To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

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

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

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

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

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

RENESAS

User's Manual

78K0/Kx2-A

8-Bit Single-Chip Microcontrollers

μPD78F0590
μPD78F0591
μPD78F0592
μPD78F0593

Document No. U19780EJ2V0UD00 (2nd edition)
Date Published February 2010 NS

© NEC Electronics Corporation 2010
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 VIL (MAX) and VIH (MIN) due to noise, etc., the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between VIL (MAX) and VIH (MIN).

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

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

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

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

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

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 February, 2010. 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.

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

(M8E0909E)

INTRODUCTION

Readers This manual is intended for user engineers who wish to understand the functions of the 78K0/Kx2-A microcontrollers and design and develop application systems and programs for these devices.

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

Organization The manual for the 78K0/Kx2-A microcontrollers is separated into two parts: this manual and the instructions edition (common to the 78K0 microcontrollers).

78K0/Kx2-A User's Manual (This Manual)	**78K/0 Series 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**.
- 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 RA78K0, and is defined as an sfr variable using the #pragma sfr directive in the CC78K0.
- To know details of the 78K0 microcontroller instructions:
  → Refer to the separate document **78K/0 Series Instructions User's Manual (U12326E)**.

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.
78K0/Kx2-A User's Manual	This manual
78K/0 Series Instructions User's Manual	U12326E

Documents Related to Development Tools (Software)

Document Name		Document No.
RA78K0 Ver.3.80 Assembler Package User's Manual [Note 1]	Operation	U17199E
	Language	U17198E
	Structured Assembly Language	U17197E
78K0 Assembler Package RA78K0 Ver.4.01 Operating Precautions (Notification Document) [Note 1]		ZUD-CD-07-0181-E
CC78K0 Ver.3.70 C Compiler User's Manual [Note 2]	Operation	U17201E
	Language	U17200E
78K0 C Compiler CC78K0 Ver. 4.00 Operating Precautions (Notification Document) [Note 2]		ZUD-CD-07-0103-E
SM+ System Simulator User's Manual	Operation	U18601E
	User Open Interface	U18212E
ID78K0-QB Ver.2.94 Integrated Debugger User's Manual	Operation	U18330E
ID78K0-QB Ver.3.00 Integrated Debugger User's Manual	Operation	U18492E
PM plus Ver.5.20[Note 3] User's Manual		U16934E
PM+ Ver.6.30[Note 4] User's Manual		U18416E

Notes 1. This document is installed into the PC together with the tool when installing RA78K0 Ver. 4.01. For descriptions not included in "78K0 Assembler Package RA78K0 Ver. 4.01 Operating Precautions", refer to the user's manual of RA78K0 Ver. 3.80.

2. This document is installed into the PC together with the tool when installing CC78K0 Ver. 4.00. For descriptions not included in "78K0 C Compiler CC78K0 Ver. 4.00 Operating Precautions", refer to the user's manual of CC78K0 Ver. 3.70.

3. PM plus Ver. 5.20 is the integrated development environment included with RA78K0 Ver. 3.80.

4. PM+ Ver. 6.30 is the integrated development environment included with RA78K0 Ver. 4.01. Software tool (assembler, C compiler, debugger, and simulator) products of different versions can be managed.

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

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

Document Name	Document No.
QB-MINI2 On-Chip Debug Emulator with Programming Function	U18371E

Documents Related to Flash Memory Programming (User's Manual)

Document Name	Document No.
PG-FP5 Flash Memory Programmer	U18865E
QB-Programmer Programming GUI Operation	U18527E

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
1.4.1 78K0/KB2-A 18
1.4.2 78K0/KC2-A 20
1.5 Block Diagram 22
1.5.1 78K0/KB2-A 22
1.5.2 78K0/KC2-A 23
1.6 Outline of Functions 24

CHAPTER 2 PIN FUNCTIONS 25

2.1 Pin Function List 25
2.1.1 78K0/KB2-A 26
2.1.2 78K0/KC2-A 29
2.2 Description of Pin Functions 34
2.2.1 P00 to P02 (port 0) 34
2.2.2 P10 to P13 (port 1) 35
2.2.3 P20 to P27 (port 2) 36
2.2.4 P31 to P35 (port 3) 37
2.2.5 P40 to P42 (port 4) 38
2.2.6 P60, P61 (Port 6) 39
2.2.7 P70 to P75 (port 7) 40
2.2.8 P80 to P83 (port 8) 40
2.2.9 P120 to P124 (port 12) 41
2.2.10 AV_{REF}, AV_{REFM}, AV_{REFP}, AV_{DD}, AV_{SS}, V_{DD}, V_{SS} 42
2.2.11 $\overline{RESET}$ 43
2.2.12 REGC 43
2.2.13 FLMD0 43
2.3 Pin I/O Circuits and Recommended Connection of Unused Pins 44

CHAPTER 3 CPU ARCHITECTURE 49

3.1 Memory Space 49
3.1.1 Internal program memory space 52
3.1.2 Internal data memory space 54
3.1.3 Special function register (SFR) area 54
3.1.4 Data memory addressing 54
3.2 Processor Registers 57
3.2.1 Control registers 57
3.2.2 General-purpose registers 60
3.2.3 Special function registers (SFRs) 62
3.3 Instruction Address Addressing 67

3.3.1 Relative addressing 67
3.3.2 Immediate addressing 68
3.3.3 Table indirect addressing 69
3.3.4 Register addressing 70
3.4 Operand Address Addressing 70
3.4.1 Implied addressing 70
3.4.2 Register addressing 71
3.4.3 Direct addressing 72
3.4.4 Short direct addressing 73
3.4.5 Special function register (SFR) addressing 74
3.4.6 Register indirect addressing 75
3.4.7 Based addressing 76
3.4.8 Based indexed addressing 77
3.4.9 Stack addressing 78

CHAPTER 4 PORT FUNCTIONS 79

4.1 Port Functions 79
4.2 Port Configuration 82
4.2.1 Port 0 83
4.2.2 Port 1 85
4.2.3 Port 2 91
4.2.4 Port 3 97
4.2.5 Port 4 100
4.2.6 Port 6 102
4.2.7 Port 7 104
4.2.8 Port 8 105
4.2.9 Port 12 109
4.3 Registers Controlling Port Function 113
4.4 Port Function Operations 119
4.4.1 Writing to I/O port 119
4.4.2 Reading from I/O port 120
4.4.3 Operations on I/O port 120
4.5 Settings of Port Mode Register and Output Latch When Using Alternate Function 120
4.6 Cautions on 1-Bit Manipulation Instruction for Port Register n (Pn) 126

CHAPTER 5 CLOCK GENERATOR 127

5.1 Functions of Clock Generator 127
5.2 Configuration of Clock Generator 128
5.3 Registers Controlling Clock Generator 131
5.4 System Clock Oscillator 140
5.4.1 X1 oscillator 140
5.4.2 XT1 oscillator 140
5.4.3 When subsystem clock is not used 143
5.4.4 Internal high-speed oscillator 143
5.4.5 Internal low-speed oscillator 143
5.4.6 Prescaler 143
5.5 Clock Generator Operation 144
5.6 Controlling Clock 148

5.6.1 Example of controlling high-speed system clock....148
5.6.2 Example of controlling internal high-speed oscillation clock....151
5.6.3 Example of controlling subsystem clock....154
5.6.4 Example of controlling internal low-speed oscillation clock....156
5.6.5 Clocks supplied to CPU and peripheral hardware....157
5.6.6 CPU clock status transition diagram....158
5.6.7 Condition before changing CPU clock and processing after changing CPU clock....164
5.6.8 Time required for switchover of CPU clock and main system clock....166
5.6.9 Conditions before clock oscillation is stopped....168
5.6.10 Peripheral hardware and source clocks....169

CHAPTER 6 16-BIT TIMER/EVENT COUNTER 00....170

6.1 Functions of 16-Bit Timer/Event Counter 00....170
6.2 Configuration of 16-Bit Timer/Event Counter 00....171
6.3 Registers Controlling 16-Bit Timer/Event Counter 00....177
6.4 Operation of 16-Bit Timer/Event Counter 00....185
6.4.1 Interval timer operation....185
6.4.2 Square wave output operation....188
6.4.3 External event counter operation....191
6.4.4 Operation in clear & start mode entered by TI000 pin valid edge input....195
6.4.5 Free-running timer operation....209
6.4.6 PPG output operation....218
6.4.7 One-shot pulse output operation....221
6.4.8 Pulse width measurement operation....226
6.5 Special Use of TM00....234
6.5.1 Rewriting CR010 during TM00 operation....234
6.5.2 Setting LVS00 and LVR00....234
6.6 Cautions for 16-Bit Timer/Event Counter 00....236

CHAPTER 7 8-BIT TIMER/EVENT COUNTERS 50 AND 51....241

7.1 Functions of 8-Bit Timer/Event Counters 50 and 51....241
7.2 Configuration of 8-Bit Timer/Event Counters 50 and 51....241
7.3 Registers Controlling 8-Bit Timer/Event Counters 50 and 51....244
7.4 Operations of 8-Bit Timer/Event Counters 50 and 51....250
7.4.1 Operation as interval timer....250
7.4.2 Operation as external event counter....252
7.4.3 Square-wave output operation....253
7.4.4 PWM output operation....254
7.5 Cautions for 8-Bit Timer/Event Counters 50 and 51....258

CHAPTER 8 8-BIT TIMERS H0 AND H1....259

8.1 Functions of 8-Bit Timers H0 and H1....259
8.2 Configuration of 8-Bit Timers H0 and H1....259
8.3 Registers Controlling 8-Bit Timers H0 and H1....263
8.4 Operation of 8-Bit Timers H0 and H1....269
8.4.1 Operation as interval timer/square-wave output....269
8.4.2 Operation as PWM output....272

8.4.3 Carrier generator operation (8-bit timer H1 only) 278

CHAPTER 9 REAL-TIME COUNTER 285

9.1 Functions of Real-Time Counter 285
9.2 Configuration of Real-Time Counter 286
9.3 Registers Controlling Real-Time Counter 288
9.4 Real-Time Counter Operation 301
9.4.1 Starting operation of real-time counter 301
9.4.2 Reading/writing real-time counter 302
9.4.3 Setting alarm of real-time counter 304
9.4.4 1 Hz output of real-time counter 305
9.4.5 32.768 kHz output of real-time counter 305
9.4.6 512 Hz, 16.384 kHz output of real-time counter 305
9.4.7 Example of watch error correction of real-time counter 306

CHAPTER 10 WATCHDOG TIMER 311

10.1 Functions of Watchdog Timer 311
10.2 Configuration of Watchdog Timer 312
10.3 Register Controlling Watchdog Timer 313
10.4 Operation of Watchdog Timer 314
10.4.1 Controlling operation of watchdog timer 314
10.4.2 Setting overflow time of watchdog timer 315
10.4.3 Setting window open period of watchdog timer 316

CHAPTER 11 CLOCK OUTPUT CONTROLLER 318

11.1 Functions of Clock Output Controller 318
11.2 Configuration of Clock Output Controller 319
11.3 Registers Controlling Clock Output Controller 319
11.4 Operations of Clock Output Controller 321
11.4.1 Operation as clock output 321

CHAPTER 12 A/D CONVERTER 322

12.1 Function of A/D Converter 322
12.2 Configuration of A/D Converter 324
12.3 Registers Used in A/D Converter 326
12.4 A/D Converter Operations 339
12.4.1 Basic operations of A/D converter 339
12.4.2 Input voltage and conversion results 341
12.4.3 A/D converter operation mode 342
12.5 How to Read A/D Converter Characteristics Table 348
12.6 Cautions for A/D Converter 350

CHAPTER 13 OPERATIONAL AMPLIFIER 354

13.1 Function of Operational Amplifier 354
13.2 Configuration of Operational Amplifier 354
13.3 Registers Used in Operational Amplifier 356

13.4 Operation of Operational Amplifier 361

CHAPTER 14 SERIAL INTERFACE UART6 362

14.1 Functions of Serial Interface UART6 362
14.2 Configuration of Serial Interface UART6 367
14.3 Registers Controlling Serial Interface UART6 370
14.4 Operation of Serial Interface UART6 379
14.4.1 Operation stop mode 379
14.4.2 Asynchronous serial interface (UART) mode 380
14.4.3 Dedicated baud rate generator 394
14.4.4 Calculation of baud rate 395

CHAPTER 15 SERIAL INTERFACE CSI10 401

15.1 Functions of Serial Interface CSI10 401
15.2 Configuration of Serial Interface CSI10 402
15.3 Registers Controlling Serial Interface CSI10 404
15.4 Operation of Serial Interface CSI10 410
15.4.1 Operation stop mode 410
15.4.2 3-wire serial I/O mode 411

CHAPTER 16 SERIAL INTERFACE IICA 423

16.1 Functions of Serial Interface IICA 423
16.2 Configuration of Serial Interface IICA 426
16.3 Registers Controlling Serial Interface IICA 429
16.4 I^2C Bus Mode Functions 442
16.4.1 Pin configuration 442
16.4.2 Setting transfer clock by using IICWL and IICWH registers 443
16.5 I^2C Bus Definitions and Control Methods 445
16.5.1 Start conditions 445
16.5.2 Addresses 446
16.5.3 Transfer direction specification 446
16.5.4 Acknowledge ($\overline{ACK}$) 447
16.5.5 Stop condition 448
16.5.6 Wait 449
16.5.7 Canceling wait 451
16.5.8 Interrupt request (INTIICA0) generation timing and wait control 452
16.5.9 Address match detection method 453
16.5.10 Error detection 453
16.5.11 Extension code 453
16.5.12 Arbitration 454
16.5.13 Wakeup function 456
16.5.14 Communication reservation 460
16.5.15 Cautions 464
16.5.16 Communication operations 465
16.5.17 Timing of I^2C interrupt request (INTIICA0) occurrence 473
16.6 Timing Charts 494

CHAPTER 17 MULTIPLIER/DIVIDER 501

17.1 Functions of Multiplier/Divider 501
17.2 Configuration of Multiplier/Divider 501
17.3 Register Controlling Multiplier/Divider 505
17.4 Operations of Multiplier/Divider 506
17.4.1 Multiplication operation 506
17.4.2 Division operation 508

CHAPTER 18 INTERRUPT FUNCTIONS 510

18.1 Interrupt Function Types 510
18.2 Interrupt Sources and Configuration 510
18.3 Registers Controlling Interrupt Functions 515
18.4 Interrupt Servicing Operations 527
18.4.1 Maskable interrupt acknowledgment 527
18.4.2 Software interrupt request acknowledgment 529
18.4.3 Multiple interrupt servicing 530
18.4.4 Interrupt request hold 533

CHAPTER 19 KEY INTERRUPT FUNCTION 534

19.1 Functions of Key Interrupt 534
19.2 Configuration of Key Interrupt 535
19.3 Register Controlling Key Interrupt 536

CHAPTER 20 STANDBY FUNCTION 537

20.1 Standby Function and Configuration 537
20.1.1 Standby function 537
20.1.2 Registers controlling standby function 538
20.2 Standby Function Operation 540
20.2.1 HALT mode 540
20.2.2 STOP mode 545

CHAPTER 21 RESET FUNCTION 552

21.1 Register for Confirming Reset Source 561

CHAPTER 22 POWER-ON-CLEAR CIRCUIT 562

22.1 Functions of Power-on-Clear Circuit 562
22.2 Configuration of Power-on-Clear Circuit 563
22.3 Operation of Power-on-Clear Circuit 563
22.4 Cautions for Power-on-Clear Circuit 566

CHAPTER 23 LOW-VOLTAGE DETECTOR 568

23.1 Functions of Low-Voltage Detector 568
23.2 Configuration of Low-Voltage Detector 569
23.3 Registers Controlling Low-Voltage Detector 569
23.4 Operation of Low-Voltage Detector 572

23.4.1 When used as reset 573
23.4.2 When used as interrupt 578
23.5 Cautions for Low-Voltage Detector 583

CHAPTER 24 OPTION BYTE 586

24.1 Functions of Option Bytes 586
24.2 Format of Option Byte 587

CHAPTER 25 FLASH MEMORY 591

25.1 Internal Memory Size Switching Register 591
25.2 Writing with Flash Memory Programmer 592
25.3 Programming Environment 592
25.4 Communication Mode 593
25.5 Connection of Pins on Board 594
25.5.1 FLMD0 pin 594
25.5.2 Serial interface pins 595
25.5.3 $\overline{\text{RESET}}$ pin 596
25.5.4 Port pins 597
25.5.5 REGC pin 597
25.5.6 Other signal pins 597
25.5.7 Power supply 597
25.6 Programming Method 598
25.6.1 Controlling flash memory 598
25.6.2 Flash memory programming mode 598
25.6.3 Selecting communication mode 599
25.6.4 Communication commands 600
25.7 Security Settings 601
25.8 Flash Memory Programming by Self-Programming 603
25.8.1 Boot swap function 610

CHAPTER 26 ON-CHIP DEBUG FUNCTION 612

26.1 Connecting QB-MINI2 to 78K0/Kx2-A microcontrollers 612
26.2 Reserved Area Used by QB-MINI2 614

CHAPTER 27 INSTRUCTION SET 615

27.1 Conventions Used in Operation List 615
27.1.1 Operand identifiers and specification methods 615
27.1.2 Description of operation column 616
27.1.3 Description of flag operation column 616
27.2 Operation List 617
27.3 Instructions Listed by Addressing Type 625

CHAPTER 28 ELECTRICAL SPECIFICATIONS 628

CHAPTER 29 PACKAGE DRAWINGS 654

29.1 78K0/KB2-A 654

29.2 78K0/KC2-A 655

CHAPTER 30 CAUTIONS FOR WAIT 656

30.1 Cautions for Wait 656
30.2 Peripheral Hardware That Generates Wait 656

CHAPTER 1 OUTLINE

1.1 Features

- Minimum instruction execution time can be changed from high speed (0.1 μs: @ 20 MHz operation with high-speed system clock) to ultra low-speed (122 μs: @ 32.768 kHz operation with subsystem clock[Note1])
- General-purpose register: 8 bits × 32 registers (8 bits × 8 registers × 4 banks)
- ROM (flash memory), RAM capacities

Program Memory (ROM)[Note2]	Data Memory (RAM)	78K0/KB2-A	78K0/KC2-A
		30 pins	48 pins
16 KB	1 KB	μPD78F0590	μPD78F0592
32 KB		μPD78F0591	μPD78F0593

- On-chip single-power-supply flash memory (with prohibition of chip erase/block erase/writing function)
- Self-programming (with boot swap 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/divider (16 bits × 16 bits, 32 bits/16 bits)
- On-chip key interrupt function: 6 channels (78K0/KC2-A only)
- On-chip clock output controller (78K0/KC2-A only)
- I/O ports
  - 78K0/KB2-A: 22 (N-ch open drain: 2)
  - 78K0/KC2-A: 40 (N-ch open drain: 2)
- On-chip 12-bit resolution A/D converter
  - 78K0/KB2-A: 10 channels
  - 78K0/KC2-A: 12 channels
- Amplifier: 3 channels
- Timer

Item / Part Number	16-bit timer/event counter	8-bit timer/event counter	8-bit timer	Watchdog timer	Real-time counter
78K0/KB2-A	1 channel	2 channels	2 channels	1 channel	–
78K0/KC2-A					1 channel

Notes 1. The 78K0/KB2-A is not provided with a subsystem clock.

2. The internal flash memory capacity can be changed using the internal memory size switching register (IMS). For IMS, see **25.1 Internal Memory Size Switching Register**.

○ Serial interface

Item / Part Number	UART supporting LIN-bus	3-wire CSI	I²C
78K0/KB2-A	1 channel	1 channel	1 channel
78K0/KC2-A		1 channel[Note]	

Note Enable control is possible when the 3-wire CSI is used as a slave.

○ Power supply voltage: V_{DD} = 1.8 to 5.5 V

○ Operating ambient temperature: T_A = –40 to +85°C

1.2 Applications

- AV equipment
- Household electrical appliances
- Industrial equipment
- Amusement machines

1.3 Ordering Information

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

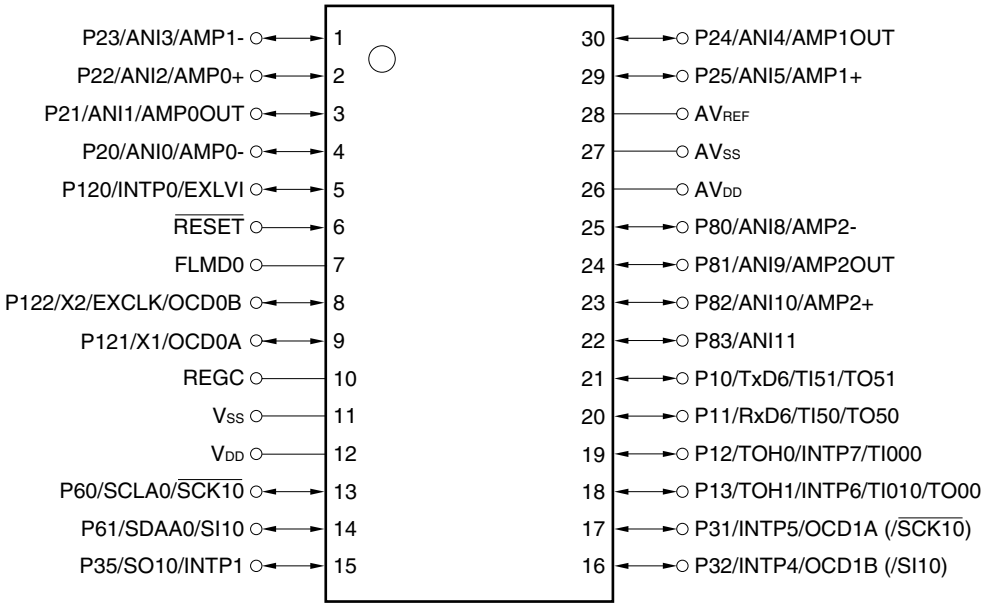
78K0/Kx2-A microcontrollers	Package	Part Number
78K0/KB2-A	30-pin plastic SSOP (7.62 mm (300))	μPD78F0590MC-CAB-AX, 78F0591MC-CAB-AX
78K0/KC2-A	48-pin plastic LQFP (fine pitch) (7x7)	μPD78F0592GA-GAM-AX, 78F0593GA-GAM-AX

Caution 78K0/Kx2-A microcontrollers have an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. NEC Electronics is not liable for problems occurring when the on-chip debug function is used.

1.4 Pin Configuration (Top View)

1.4.1 78K0/KB2-A

- 30-pin plastic SSOP (7.62 mm (300))



Cautions 1. Make AVSS the same potential as VSS.

2. Connect the REGC pin to VSS via a capacitor (0.47 to 1 μF: recommended).

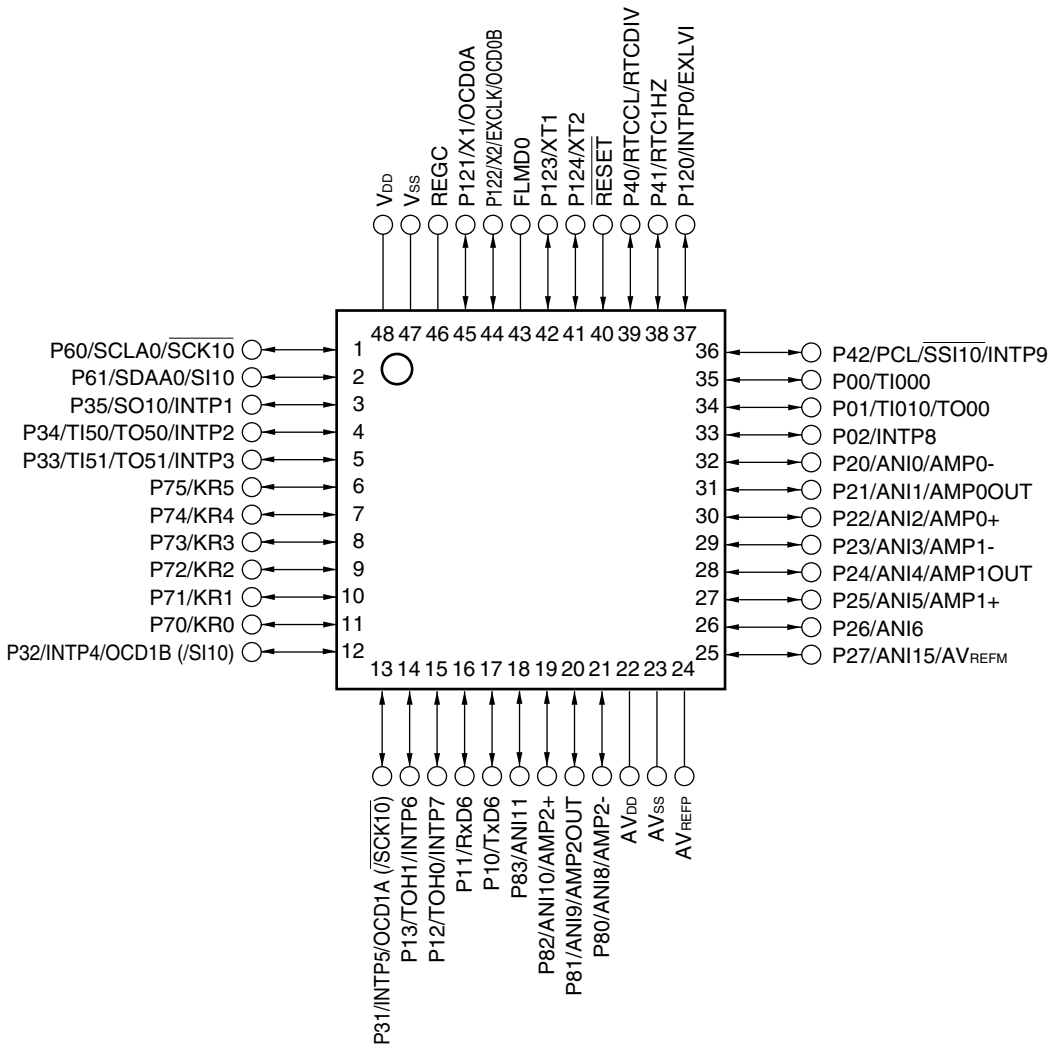
Remark The functions of pins whose names are in parentheses can be used by setting bit 2 (ISC2) of the input switch control register (ISC) to 1. If ISC2 is 0, serial interface CSI10 can only be used to send and received data as a slave.

Pin Identification

AMP0− to AMP2−:	Amplifier Input Minus	P80 to P83:	Port 8
AMP0+ to AMP2+:	Amplifier Input Plus	P120 to P122:	Port 12
AMP0OUT		REGC:	Regulator Capacitance
to AMP2OUT:	Amplifier Output	$\overline{\text{RESET}}$:	Reset
ANI0 to ANI5,		RxD6:	Receive Data
ANI8 to ANI11:	Analog Input	$\overline{\text{SCK10}}$:	Serial Clock Input/Output
AV_{DD}:	Analog Power Supply	SCLA0:	Serial Clock Input/Output
AV_{REF}:	Analog Reference Voltage	SDAA0:	Serial Data Input/Output
AV_{SS}:	Analog Ground	SI10:	Serial Data Input
EXCLK:	External Clock Input	SO10:	Serial Data Output
	(Main System Clock)	TI000, TI010	
EXLVI:	External potential Input	TI50, TI51:	Timer Input
	for Low-voltage detector	TO00, TO50, TO51,	
FLMD0:	Flash Programming Mode	TOH0, TOH1:	Timer Output
INTP0, INTP1,		TxD6:	Transmit Data
INTP4 to INTP7:	External Interrupt Input	V_{DD}:	Power Supply
OCD0A, OCD0B,		V_{SS}:	Ground
OCD1A, OCD1B:	On Chip Debug Input/Output	X1, X2:	Crystal Oscillator (Main System Clock)
P10 to P13:	Port 1		
P20 to P25:	Port 2		
P31, P32, P35:	Port 3		
P60, P61:	Port 6		

1.4.2 78K0/KC2-A

- 48-pin plastic LQFP (fine pitch) (7x7)



Cautions 1. Make AVSS the same potential as VSS.

2. Connect the REGC pin to VSS via a capacitor (0.47 to 1 μF: recommended).

Remark The functions of pins whose names are in parentheses can be used by setting bit 2 (ISC2) of the input switch control register (ISC) to 1. If ISC2 is 0, serial interface CSI10 can only be used to send and received data as a slave.

Pin Identification

AMP0− to AMP2−: Amplifier Input Minus

AMP0+ to AMP2+: Amplifier Input Plus

AMP0OUT to AMP2OUT: Amplifier Output

ANI0 to ANI6, ANI8 to ANI11, ANI15: Analog Input

AV_{DD}: Analog Power Supply

AV_{REFM}: Analog Reference Voltage Minus

AV_{REFP}: Analog Reference Voltage Plus

AV_{SS}: Analog Ground

EXCLK: External Clock Input (Main System Clock)

EXLVI: External potential Input for Low-voltage detector

FLMD0: Flash Programming Mode

INTP0 to INTP9: External Interrupt Input

KR0 to KR5: Key Return

OCD0A, OCD0B, OCD1A, OCD1B: On Chip Debug Input/Output

P00 to P02: Port 0

P10 to P13: Port 1

P20 to P27: Port 2

P31 to P35: Port 3

P40 to P42: Port 4

P60, P61: Port 6

P70 to P75: Port 7

P80 to P83: Port 8

P120 to P124: Port 12

PCL: Programmable Clock 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

RxD6: Receive Data

$\overline{\text{SCK10}}$: Serial Clock Input/Output

SCLA0: Serial Clock Input/Output

SDAA0: Serial Data Input/Output

SI10: Serial Data Input

SO10: Serial Data Output

$\overline{\text{SSI10}}$: Serial Interface Chip Select Input

TI000, TI010, TI50, TI51: Timer Input

TO00, TO50, TO51, TOH0, TOH1: Timer Output

TxD6: Transmit Data

V_{DD}: Power Supply

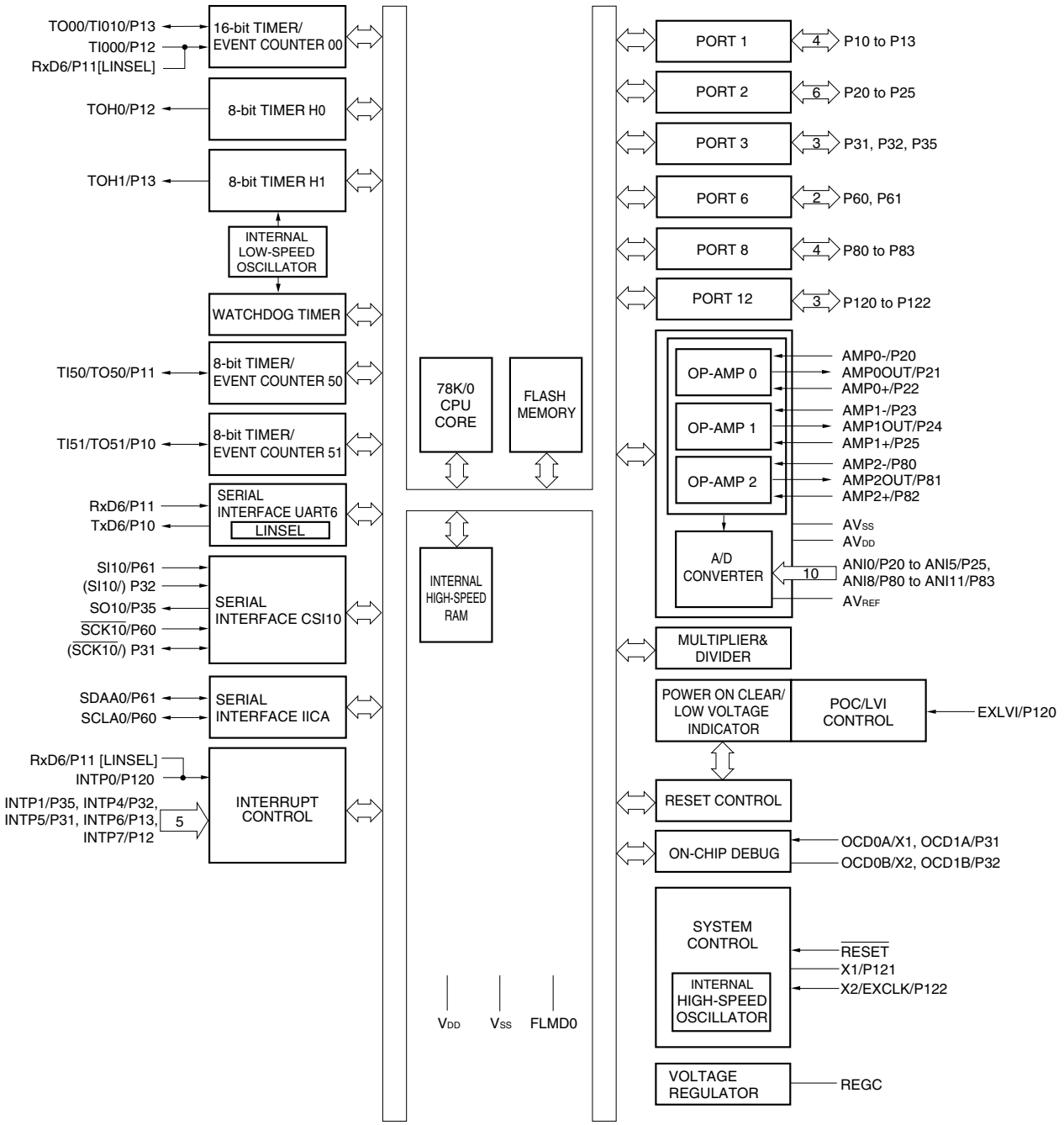
V_{SS}: Ground

X1, X2: Crystal Oscillator (Main System Clock)

XT1, XT2: Crystal Oscillator (Subsystem Clock)

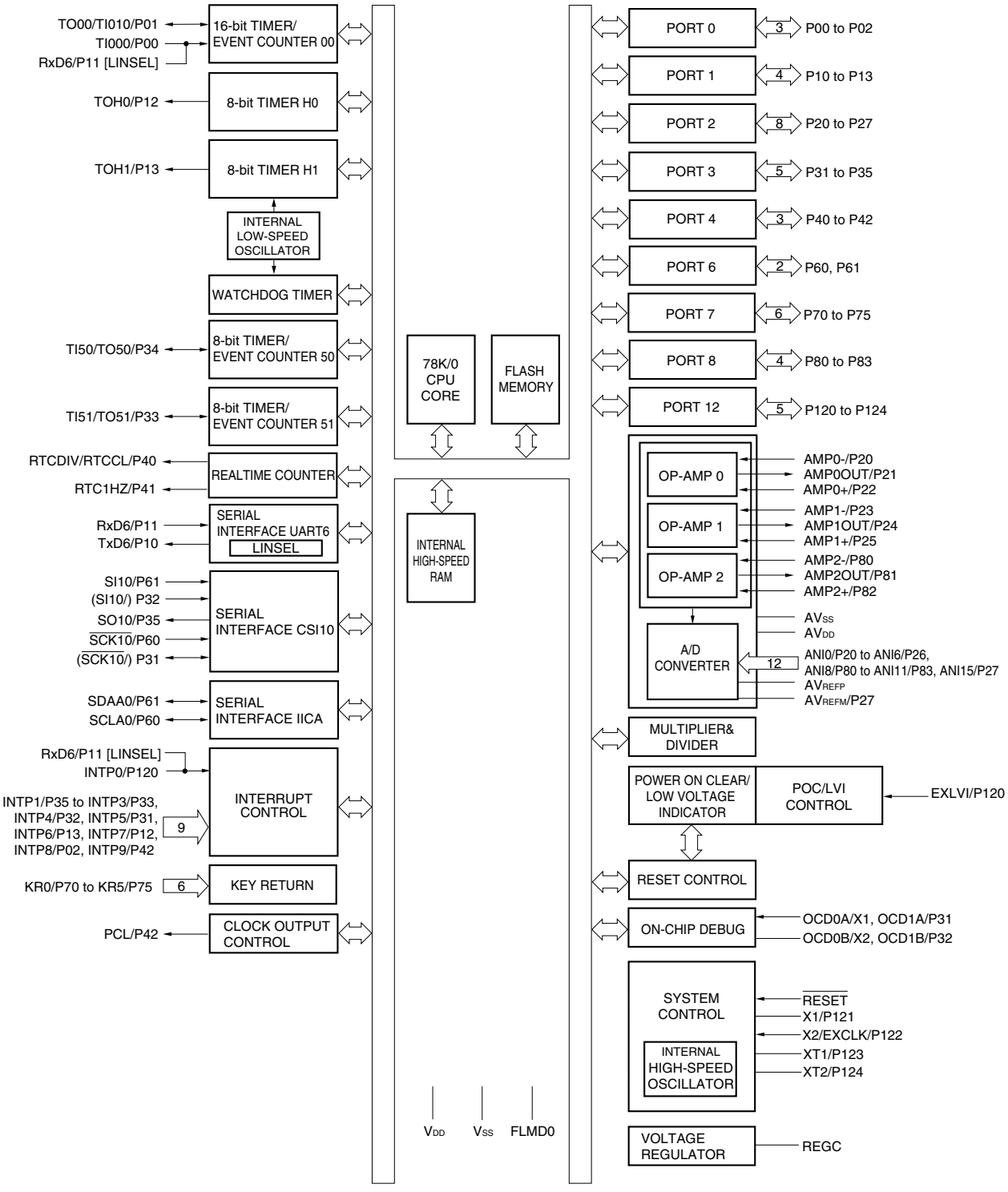
1.5 Block Diagram

1.5.1 78K0/KB2-A



Remark The functions of pins whose names are in parentheses can be used by setting bit 2 (ISC2) of the input switch control register (ISC) to 1. If ISC2 is 0, serial interface CSI10 can only be used to send and received data as a slave.

1.5.2 78K0/KC2-A



Remark The functions of pins whose names are in parentheses can be used by setting bit 2 (ISC2) of the input switch control register (ISC) to 1. If ISC2 is 0, serial interface CSI10 can only be used to send and received data as a slave.

1.6 Outline of Functions

Item \ Part Number		78K0/KB2-A		78K0/KC2-A	
		μPD78F0590	μPD78F0591	μPD78F0592	μPD78F0593
		30 Pins		48 Pins	
Internal memory	Flash memory (Self-programming supported)	16 KB	32 KB	16 KB	32 KB
	High-speed RAM	1 KB			
Power supply voltage		V_{DD} = 1.8 to 5.5 V			
Regulator		Provided			
Minimum instruction execution time		0.1 μs (20 MHz: V_{DD} = 2.7 to 5.5 V)/0.4 μs (5 MHz: V_{DD} = 1.8 to 5.5 V)			
Clock	Main: High-speed system (crystal/ceramic oscillation, external clock input)	1 to 20 MHz: V_{DD} = 2.7 to 5.5 V/1 to 5 MHz: V_{DD} = 1.8 to 5.5 V			
	Main: Internal high-speed oscillation	8 MHz(TYP.) V_{DD} = 1.8 to 5.5 V			
	Subsystem clock (crystal oscillation)	–		32.768 kHz (TYP.): V_{DD} = 1.8 to 5.5 V	
	Internal low-speed 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)			
Port	Total	22		40	
	CMOS I/O	20		38	
	N-ch I/O	2		2	
Timer	16 bits (TM0)	1 ch (PPG output: 1, Capture trigger input: 2)			
	8 bits (TM5)	2 ch (PWM output: 2)			
	8 bits (TMH)	2 ch (PWM output: 2)			
	Watchdog timer (WDT)	1 ch			
	Real-time counter	–		1 ch (RTC output: 2)	
Clock output		–		1	
Serial interface	3-wire CSI	1 ch[Note 1]		1 ch[Notes 1, 2]	
	UART supporting LIN-bus	1 ch			
	I^2C	1 ch			
12-bit A/D converter		10 ch		12 ch	
Operational amplifier		3 ch			
Vectored interrupt sources	External	6		11	
	Internal	14		16	
Key interrupt		–		6	
Reset	$\overline{RESET}$ pin	Provided			
	POC	1.59 V ±0.15 V			
	LVI	The detection level of the supply voltage is selectable.			
	WDT	Provided			
Multiplier/divider		Provided			
On-chip debug function					
Operating ambient temperature		T_A = = –40 to +85°C			
Package		30-pin plastic SSOP (7.62 mm (300))		48-pin plastic LQFP (fine pitch) (7x7)	

Notes 1. When using 3-wire CSI as the master, assign $\overline{SCK10}$ and SI10 to P31 and P32 by setting bit 2 (ISC2) of the input switch control register (ISC) to 1. For details, see **CHAPTER 15 SERIAL INTERFACE CSI10**.

2. Enable control is possible when the 3-wire CSI is used as a slave.

CHAPTER 2 PIN FUNCTIONS

2.1 Pin Function List

There are two types of pin I/O buffer power supplies: AV_{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	
	78K0/KB2-A	78K0/KC2-A
AV_{DD}	P20 to P25, P80 to P83	P20 to P27, P80 to P83
V_{DD}	Pins other than P20 to P25 and P80 to P83	Pins other than P20 to P27 and P80 to P83

2.1.1 78K0/KB2-A

(1) Port functions: 78K0/KB2-A

Function Name	I/O	Function	After Reset	Alternate Function
P10	I/O	Port 1. 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	TxD6/TI51/TO51
P11				RxD6/TI50/TO50
P12				TOH0/INTP7/TI000
P13				TOH1/INTP6/TI010/TO00
P20	I/O	Port 2. 6-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI0/AMP0−
P21				ANI1/AMP0OUT
P22				ANI2/AMP0+
P23				ANI3/AMP1−
P24				ANI4/AMP1OUT
P25				ANI5/AMP1+
P31	I/O	Port 3. 3-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP5/OCD1A (/$\overline{\text{SCK10}}$)
P32				INTP4/OCD1B (/SI10)
P35				SO10/INTP1
P60	I/O	Port 6. 2-bit I/O port. Output is N-ch open-drain output (6 V tolerance). Input/output can be specified in 1-bit units.	Input port	SCLA0/$\overline{\text{SCK10}}$
P61				SDAA0/SI10
P80	I/O	Port 8. 4-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI8/AMP2−
P81				ANI9/AMP2OUT
P82				ANI10/AMP2+
P83				ANI11
P120	I/O	Port 12. 3-bit I/O port. Input/output can be specified in 1-bit units. Only for P120, use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP0/EXLVI
P121				X1/OCD0A
P122				X2/EXCLK/OCD0B

Remark The functions of pins whose names are in parentheses can be used by setting bit 2 (ISC2) of the input switch control register (ISC) to 1.

(2) Non-port functions (1/2): 78K0/KB2-A

Function Name	I/O	Function	After Reset	Alternate Function
AMP0–	Input	Operational amplifier (–) input	Digital input port	ANI0/P20
AMP1–				ANI3/P23
AMP2–				ANI8/P80
AMP0+	Input	Operational amplifier (+) input	Digital input port	ANI2/P22
AMP1+				ANI5/P25
AMP2+				ANI10/P82
AMP0OUT	Input	Operational amplifier output	Digital input port	ANI1/P21
AMP1OUT				ANI4/P24
AMP2OUT				ANI9/P81
ANI0	Input	A/D converter analog input	Digital input port	AMP0–/P20
ANI1				AMP0OUT/P21
ANI2				AMP0+/P22
ANI3				AMP1–/P23
ANI4				AMP1OUT/P24
ANI5				AMP1+/P25
ANI8				AMP2–/P80
ANI9				AMP2OUT/P81
ANI10				AMP2+/P82
ANI11				P83
AV_{REF}	Input	A/D converter anarog reference voltage input	Analog input	–
EXLVI	Input	Potential input for external low-voltage detection	Input port	P120/INTP0
FLMD0	–	Flash memory programming mode setting	–	–
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				P35/SO10
INTP4				P32/OCD1B (/SI10)
INTP5				P31/OCD1A (/$\overline{\text{SCK10}}$)
INTP6				P13/TOH1/TI010/TO00
INTP7				P12/TOH0/TI000
REGC	–	Connecting regulator output (2.5 V) stabilization capacitance for internal operation. Connect to V_{SS} via a capacitor (0.47 to 1 μF: recommended).	–	–
$\overline{\text{RESET}}$	Input	System reset input	–	–
RxD6	Input	Serial data input to UART6	Input port	P11/TI50/TO50
$\overline{\text{SCK10}}$	I/O	Clock input/output for CSI10	Input port	P60/SCLA0
($\overline{\text{SCK10}}$)				P31/OCD1A
SCLA0	I/O	Clock input/output for I^2C	Input port	P60/$\overline{\text{SCK10}}$
SDAA0	I/O	Serial data I/O for I^2C	Input port	P61/SI10

Remark The functions of pins whose names are in parentheses can be used by setting bit 2 (ISC2) of the input switch control register (ISC) to 1.

(2) Non-port functions (2/2): 78K0/KB2-A

Function Name	I/O	Function	After Reset	Alternate Function
SI10	Input	Serial data input to CSI10	Input port	P61/SDAA0
(SI10)				P32/OCD1B
SO10	Output	Serial data output from CSI10	Input port	P35/INTP1
TI000	Input	External count clock input to 16-bit timer/event counter 00 Capture trigger input to capture registers (CR000, CR010) of 16-bit timer/event counter 00	Input port	P12/TOH0/INTP7
TI010	Input	Capture trigger input to capture register (CR000) of 16-bit timer/event counter 00	Input port	P13/TOH1/INTP6/ TO00
TI50	Input	External count clock input to 8-bit timer/event counter 50	Input port	P11/TO50/RxD6
TI51		External count clock input to 8-bit timer/event counter 51		P10/TO51/TxD6
TO00	Output	16-bit timer/event counter 00 output	Input port	P13/TOH1/TI010/ INTP6
TO50	Output	8-bit timer/event counter 50 output	Input port	P11/TI50/RxD6
TO51		8-bit timer/event counter 51 output		P10/TI51/TxD6
TOH0	Output	8-bit timer H0 output	Input port	P12/TI000/INTP7
TOH1		8-bit timer H1 output		P13/TI010/TO00/INTP6
TxD6	Output	Serial data output from UART6	Input port	P10/TI51/TO51
X1	–	Connecting resonator for main system clock	Input port	P121/OCD0A
X2	–		Input port	P122/EXCLK/OCD0B
EXCLK	Input	External clock input for main system clock	Input port	P122/X2/OCD0B
V_{DD}	–	Positive power supply for pins other than P20 to P25, P80 to P83, A/D converter, and operational amplifier.	–	–
AV_{DD}	–	Positive power supply for pins P20 to P25, P80 to P83, A/D converter, and operational amplifier.	–	–
V_{SS}	–	Ground potential for pins other than P20 to P25, P80 to P83, A/D converter, and operational amplifier.	–	–
AV_{SS}	–	Ground potential for pins P20 to P25, P80 to P83, A/D converter, and operational amplifier. Make the same potential as V_{SS}.	–	–
OCD0A	Input	Connection for on-chip debug mode setting pins	Input port	P121/X1
OCD1A				P31/INTP5 (/SCK10)
OCD0B	I/O			P122/X2/EXCLK
OCD1B				P32/INTP4 (/SI10)

Remark The functions of pins whose names are in parentheses can be used by setting bit 2 (ISC2) of the input switch control register (ISC) to 1.

2.1.2 78K0/KC2-A

(1) Port functions (1/2): 78K0/KC2-A

Function Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0. 3-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TI000
P01				TI010/TO00
P02				INTP8
P10	I/O	Port 1. 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	TxD6
P11				RxD6
P12				TOH0/INTP7
P13				TOH1/INTP6
P20	I/O	Port 2. 8-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI0/AMP0−
P21				ANI1/AMP0OUT
P22				ANI2/AMP0+
P23				ANI3/AMP1−
P24				ANI4/AMP1OUT
P25				ANI5/AMP1+
P26				ANI6
P27				ANI15/AV_{REFM}
P31	I/O	Port 3. 5-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP5/OCD1A (/$\overline{SCK10}$)
P32				INTP4/OCD1B (/SI10)
P33				TI51/TO51/INTP3
P34				TI50/TO50/INTP2
P35				SO10/INTP1
P40	I/O	Port 4. 3-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	RTCCL/RTCDIV
P41				RTC1HZ
P42				PCL/$\overline{SSI10}$/INTP9
P60	I/O	Port 6. 2-bit I/O port. Output is N-ch open-drain output (6 V tolerance). Input/output can be specified in 1-bit units.	Input port	SCLA0/$\overline{SCK10}$
P61				SDAA0/SI10
P70 to P75	I/O	Port 7. 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	KR0 to KR5

Remark The functions of pins whose names are in parentheses can be used by setting bit 2 (ISC2) of the input switch control register (ISC) to 1.

(1) Port functions (2/2): 78K0/KC2-A

Function Name	I/O	Function	After Reset	Alternate Function
P80	I/O	Port 8. 4-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI8/AMP2−
P81				ANI9/AMP2OUT
P82				ANI10/AMP2+
P83				ANI11
P120	I/O	Port 12. 5-bit I/O port. Input/output can be specified in 1-bit units. Only for P120, use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP0/EXLVI
P121				X1/OCD0A
P122				X2/EXCLK/OCD0B
P123				XT1
P124				XT2

(2) Non-port functions (1/3): 78K0/KC2-A

Function Name	I/O	Function	After Reset	Alternate Function
AMP0–	Input	Operational amplifier (–) input	Digital input port	ANI0/P20
AMP1–				ANI3/P23
AMP2–				ANI8/P80
AMP0+	Input	Operational amplifier (+) input	Digital input port	ANI2/P22
AMP1+				ANI5/P25
AMP2+				ANI10/P82
AMP0OUT	Input	Operational amplifier output	Digital input port	ANI1/P21
AMP1OUT				ANI4/P24
AMP2OUT				ANI9/P81
ANI0	Input	A/D converter analog input	Digital input port	AMP0–/P20
ANI1				AMP0OUT/P21
ANI2				AMP0+/P22
ANI3				AMP1–/P23
ANI4				AMP1OUT/P24
ANI5				AMP1+/P25
ANI6				P26
ANI8				AMP2–/P80
ANI9				AMP2OUT/P81
ANI10				AMP2+/P82
ANI11				P83
ANI15				AV_{REFM}/P27
AV_{REFM}	Input	A/D converter analog reference voltage minus	Digital input port	ANI15/P27
AV_{REFP}		A/D converter analog reference voltage plus	Analog input	–
EXLVI	Input	Potential input for external low-voltage detection	Input port	P120/INTP0
FLMD0	–	Flash memory programming mode setting	–	–
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				P35/SO10
INTP2				P34/TI50/TO50
INTP3				P33/TI51/TO51
INTP4				P32/OCD1B (/SI10)
INTP5				P31/OCD1A (/$\overline{\text{SCK10}}$)
INTP6				P13/TOH1
INTP7				P12/TOH0
INTP8				P02
INTP9				P42/PCL/$\overline{\text{SSI10}}$
KR0 to KR5	Input	Key interrupt input	Input port	P70 to P75
PCL	Output	Clock output (for trimming of high-speed system clock, subsystem clock)	Input port	P42/$\overline{\text{SSI10}}$/INTP9

Remark The functions of pins whose names are in parentheses can be used by setting bit 2 (ISC2) of the input switch control register (ISC) to 1.

(2) Non-port functions (2/3): 78K0/KC2-A

Function Name	I/O	Function	After Reset	Alternate Function
REGC	–	Connecting regulator output (2.5 V) stabilization capacitance for internal operation. Connect to V_{SS} via a capacitor (0.47 to 1 μF: recommended).	–	–
$\overline{\text{RESET}}$	Input	System reset input	–	–
RTC1HZ	Output	Real-time counter correction clock (1 Hz) output	Input port	P41
RTCCL	Output	Real-time counter clock (32 kHz original oscillation) output	Input port	P40/RTCDIV
RTCDIV	Output	Real-time counter clock(32 kHz divided frequency) output	Input port	P40/RTCCL
RxD6	Input	Serial data input to UART6	Input port	P11
$\overline{\text{SCK10}}$	I/O	Clock input/output for CSI10	Input port	P60/SCLA0
($\overline{\text{SCK10}}$)				P31/INTP5
SCLA0	I/O	Clock input/output for I^2C	Input port	P60/$\overline{\text{SCK10}}$
SDAA0	I/O	Serial data I/O for I^2C	Input port	P61/SI10
SI10	Input	Serial data input to CSI10	Input port	P61/SDAA0
(SI10)				P32/INTP4
SO10	Output	Serial data output from CSI10	Input port	P35/INTP1
$\overline{\text{SSI10}}$	Input	Chip select input to CSI10	Input port	P42/PCL/INTP9
TI000	Input	External count clock input to 16-bit timer/event counter 00 Capture trigger input to capture registers (CR000, CR010) of 16-bit timer/event counter 00	Input port	P00
TI010	Input	Capture trigger input to capture register (CR000) of 16-bit timer/event counter 00	Input port	P01/TO00
TI50	Input	External count clock input to 8-bit timer/event counter 50	Input port	P34/TO50/INTP2
TI51		External count clock input to 8-bit timer/event counter 51		P33/TO51/INTP3
TO00	Output	16-bit timer/event counter 00 output	Input port	P01/TI010
TO50	Output	8-bit timer/event counter 50 output	Input port	P34/TI50/INTP2
TO51		8-bit timer/event counter 51 output		P33/TI51/INTP3
TOH0	Output	8-bit timer H0 output	Input port	P12/INTP7
TOH1		8-bit timer H1 output		P13/INTP6
TxD6	Output	Serial data output from UART6	Input port	P10
X1	–	Connecting resonator for main system clock	Input port	P121/OCD0A
X2	–		Input port	P122/EXCLK/OCD0B
EXCLK	Input	External clock input for main system clock	Input port	P122/X2/OCD0B
XT1	–	Connecting resonator for subsystem clock	Input port	P123
XT2	–		Input port	P124

Remark The functions of pins whose names are in parentheses can be used by setting bit 2 (ISC2) of the input switch control register (ISC) to 1.

(2) Non-port functions (3/3): 78K0/KC2-A

Function Name	I/O	Function	After Reset	Alternate Function
V_{DD}	–	Positive power supply for pins other than P20 to P27, P80 to P83, A/D converter, and operational amplifier.	–	–
AV_{DD}	–	Positive power supply for pins P20 to P27, P80 to P83, A/D converter, and operational amplifier.	–	–
V_{SS}	–	Ground potential for pins other than P20 to P27, P80 to P83, A/D converter, and operational amplifier.	–	–
AV_{SS}	–	Ground potential for pins P20 to P27, P80 to P83, A/D converter, and operational amplifier. Make the same potential as V_{SS}.	–	–
OCD0A	Input	Connection for on-chip debug mode setting pins	Input port	P121/X1
OCD1A				P31/INTP5 (/$\overline{\text{SCK10}}$)
OCD0B	I/O			P122/X2/EXCLK
OCD1B				P32/INTP4 (/SI10)

Remark The functions of pins whose names are in parentheses can be used by setting bit 2 (ISC2) of the input switch control register (ISC) to 1.

2.2 Description of Pin Functions

Remark The pins mounted depend on the product. See **1.4 Ordering Information** and **2.1 Pin Function List**.

2.2.1 P00 to P02 (port 0)

P00 to P02 function as an I/O port. These pins also function as external interrupt request input and timer I/O.

78K0/KB2-A	78K0/KC2-A
–	P00/TI000
	P01/TI010/TO00
	P02/INTP8

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

(1) Port mode

P00 to P02 function as an I/O port. P00 to P02 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 P02 function as external interrupt request input and timer I/O.

(a) INTP8

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

(b) TI000

There is the pin for inputting an external count clock to 16-bit timer/event counters 00 and is also for inputting a capture trigger signal to the capture registers (CR000, CR010) of 16-bit timer/event counter 00.

(c) TI010

There is the pin for inputting a capture trigger signal to the capture register (CR000) of 16-bit timer/event counter 00.

(d) TO00

There is timer output pin of 16-bit timer/event counter 00.

2.2.2 P10 to P13 (port 1)

P10 to P13 function as an I/O port. These pins also function as pins for external interrupt request input, serial interface data I/O, and timer I/O.

78K0/KB2-A	78K0/KC2-A
P10/TxD6/TI51/TO51	P10/TxD6
P11/RxD6/TI50/TO50	P11/RxD6
P12/TOH0/INTP7/TI000	P12/TOH0/INTP7
P13/TOH1/INTP6/TI010/TO00	P13/TOH1/INTP6

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

(1) Port mode

P10 to P13 function as an I/O port. P10 to P13 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 P13 function as external interrupt request input, serial interface data I/O, and timer I/O.

(a) INTP6, INTP7

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

(b) RxD6

This is a serial data input pin of serial interface UART6.

(c) TxD6

This is a serial data output pin of serial interface UART6.

(d) TOH0, TOH1

These are the timer output pins of 8-bit timers H0 and H1.

(e) TI50, TI51

These are the pins for inputting an external count clock to 8-bit timer/event counters 50 and 51.

(f) TO50, TO51

These are the timer output pins of 8-bit timer/event counters 50 and 51.

(g) TI000

This is the pin for inputting an external count clock to 16-bit timer/event counter 00 and is also for inputting a capture trigger signal to the capture registers (CR000, CR010) of 16-bit timer/event counter 00.

(h) TI010

This is the pin for inputting a capture trigger signal to the capture register (CR000) of 16-bit timer/event counter 00.

(i) TO00

This is the timer output pin of 16-bit timer/event counter 00.

2.2.3 P20 to P27 (port 2)

P20 to P27 function as an I/O port. These pins also function as pins for A/D converter analog input, anarog reference voltage input, and operational amplifier output.

78K0/KB2-A	78K0/KC2-A
P20/ANI0/AMP0−	
P21/ANI1/AMP0OUT	
P22/ANI2/AMP0+	
P23/ANI3/AMP1−	
P24/ANI4/AMP1OUT	
P25/ANI5/AMP1+	
−	P26/ANI6
	P27/ANI15/AV_{REFM}

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

(1) Port mode

P20 to P27 function as an 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 pins for A/D converter analog input, anarog reference voltage input, and operational amplifier output.

(a) ANI0 to ANI6, ANI15

P20 to P27 function as A/D converter analog input pins. When using these pins as analog input pins, see **(5) ANI0 to ANI6, ANI8 to ANI11, and ANI15** in **12.6 Cautions for A/D Converter**.

(b) AV_{REFM}

This is an analog reference voltage input pin on the negative side of A/D converter.

(c) AMP0–, AMP1–

These are input pins on the negative side of operational amplifier 0 and 1.

(d) AMP0+, AMP1+

These are input pins on the positive side of operational amplifier 0 and 1.

(e) AMP0OUT, AMP1OUT

These are output pins of operational amplifier 0 and 1.

2.2.4 P31 to P35 (port 3)

P31 to P35 function as an I/O port. These pins also function as pins for external interrupt request input, serial interface data I/O, clock I/O, and timer I/O.

78K0/KB2-A	78K0/KC2-A
P31/INTP5/OCD1A (/$\overline{\text{SCK10}}$)	
P32/INTP4/OCD1B (/SI10)	
–	P33/TI51/TO51/INTP3
–	P34/TI50/TO50/INTP2
P35/SO10/INTP1	

Remark The functions of pins whose names are in parentheses can be used by setting bit 2 (ISC2) of the input switch control register (ISC) to 1.

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

(1) Port mode

P31 to P35 function as an I/O port. P31 to P35 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

P31 to P35 function as external interrupt request input, serial interface data I/O, clock I/O, and timer I/O.

(a) INTP1 to INTP5

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) SI10

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

(c) $\overline{\text{SCK10}}$

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

(d) SO10

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

(e) TI50, TI51

These are the pins for inputting an external count clock to 8-bit timer/event counters 50 and 51.

(f) TO50, TO51

These are the timer output pins of 8-it timer/event counters 50 and 51.

Caution Process the P31/INTP5/OCD1A (/SCK10) pin as follows, when it is not used when it is connected to a flash memory programmer or an on-chip debug emulator.

		P31/INTP5/OCD1A (/SCK10)
Flash memory programmer connection		Connect to V_{SS} via a resistor.
On-chip debug emulator connection (when it is not used as an on-chip debug mode setting pin)	During reset	
	During reset released	Input: Connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.

Remark P31 and P32 can be used as on-chip debug mode setting pins (OCD1A and OCD1B) when the on-chip debug function is used. For how to connect an on-chip debug emulator (QB-MINI2), see **CHAPTER 26 ON-CHIP DEBUG FUNCTION.**

2.2.5 P40 to P42 (port 4)

P40 to P42 function as an I/O port. These pins also function as external interrupt request input, clock output, real-time counter clock output, and chip select input for serial interface.

78K0/KB2-A	78K0/KC2-A
–	P40/RTCCL/RTCDIV
	P41/RTC1HZ
	P42/PCL/SSI10/INTP9

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

(1) Port mode

P40 to P42 function as an I/O port. P40 to P42 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).

(2) Control mode

P40 to P42 function as external interrupt request input, clock output, real-time counter clock output, and chip select input for serial interface.

(a) INTP9

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

(b) PCL

This is a clock output pin.

(c) RTCDIV

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

(d) RTCCL

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

(e) RTC1HZ

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

(f) $\overline{\text{SSI10}}$

This is a chip select input pin for serial interface CSI10.

2.2.6 P60, P61 (Port 6)

P60 and P61 function as an I/O port. These pins also function as pins for serial interface data I/O and clock I/O.

78K0/KB2-A	78K0/KC2-A
P60/SCLA0/$\overline{\text{SCK10}}$	
P61/SDAA0/SI10	

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

(1) Port mode

P60 and P61 function as an I/O port. P60 and P61 can be set to input or output port in 1-bit units using port mode register 6 (PM6).

Output of P60 and P61 is N-ch open-drain output (6 V tolerance).

(2) Control mode

P60 and P61 function as serial interface data I/O and clock I/O.

(a) SI10

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

(b) $\overline{\text{SCK10}}$

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

(c) SDAA0

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

(d) SCLA0

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

2.2.7 P70 to P75 (port 7)

P70 to P75 function as an I/O port. These pins also function as key interrupt input pins.

78K0/KB2-A	78K0/KC2-A
–	P70/KR0
	P71/KR1
	P72/KR2
	P73/KR3
	P74/KR4
	P75/KR5

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

(1) Port mode

P70 to P75 function as an I/O port. P70 to P75 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 P75 function as key interrupt input pins.

(a) KR0 to KR5

These are the key interrupt input pins.

2.2.8 P80 to P83 (port 8)

P80 to P83 function as an I/O port. These pins also function as A/D converter reference analog input and operational amplifier I/O.

78K0/KB2-A	78K0/KC2-A
P80/ANI8/AMP2–	
P81/ANI9/AMP2OUT	
P82/ANI10/AMP2+	
P83/ANI11	

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

(1) Port mode

P80 to P83 function as an I/O port. P80 to P83 can be set to input or output port in 1-bit units using port mode register 8 (PM8). 8

(2) Control mode

P80 to P83 function as A/D converter reference analog input and operational amplifier I/O.

(a) ANI8 to ANI11

P80 to P83 function as A/D converter analog input pins. When using these pins as analog input pins, see **(5) ANI0 to ANI6, ANI8 to ANI11, and ANI15** in **12.6 Cautions for A/D Converter**.

(b) AMP2–

This is an input pin on the negative side of operational amplifier 2.

(c) AMP2+

This is an input pin on the positive side of operational amplifier 2.

(d) AMP2OUT

This is an output pin of operational amplifier 2.

2.2.9 P120 to P124 (port 12)

P120 to P124 function as an I/O port. These pins also function as pins 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.

78K0/KB2-A	78K0/KC2-A
P120/INTP0/EXLVI	
P121/X1/OCD0A	
P122/X2/OCD0B	
–	P123/XT1
–	P124/XT2

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

(1) Port mode

P120 to P124 function as an I/O port. P120 to P124 can be set to input or output port using port mode register 12 (PM12). Only for P120, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 12 (PU12).

(2) Control mode

P120 to P124 function as pins 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.

(a) INTP0

This functions as an external interrupt request input (INTP0) 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.

Caution Process the P121/X1/OCD0A pin as follows, when it is not used when it is connected to a flash memory programmer or an on-chip debug emulator.

		P121/X1/OCD0A
Flash memory programmer connection		Open or connect to V_{SS} via a resistor.
On-chip debug emulator connection (when it is not used as an on-chip debug mode setting pin)	During reset	
	During reset released	Input: Connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.

Remark X1 and X2 can be used as on-chip debug mode setting pins (OCD0A and OCD0B) when the on-chip debug function is used. For how to connect an on-chip debug emulator (QB-MINI2), see **CHAPTER 26 ON-CHIP DEBUG FUNCTION.**

2.2.10 AV_{REF}, AV_{REFM}, AV_{REFP}, AV_{DD}, AV_{SS}, V_{DD}, V_{SS}

78K0/KB2-A	78K0/KC2-A
AV_{REF}	AV_{REFM}/ANI15/P27
	AV_{REFP}
AV_{DD}	
AV_{SS}	
V_{DD}	
V_{SS}	

(a) AV_{REF}

This is an analog reference voltage input pin of A/D converter.

(b) AV_{REFM}

This is an analog reference voltage input pin on the negative side of A/D converter.

(c) AV_{REFP}

This is an analog reference voltage input pin on the positive side of A/D converter.

(d) AV_{DD}

AV_{DD} is the positive power supply pin for P20 to P27, P80 to P83, A/D converter, and operational amplifier. Even when the A/D converter or operational amplifier is not used, connect this pin directly to V_{DD}[Note].

Note Make the AV_{DD} pin the same potential as the V_{DD} pin when port 2 and port 8 are used as a digital port.

(e) AV_{SS}

AV_{SS} is the ground potential pin for P20 to P27, P80 to P83, A/D converter, and operational amplifier. Even when the A/D converter or operational amplifier is not used, connect this pin directly to V_{DD}.

(f) V_{DD}

V_{DD} is the positive power supply pin for other than P20 to P27, P80 to P83, A/D converter, and operational amplifier.

(g) V_{SS}

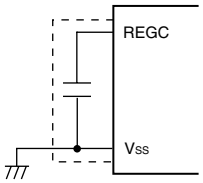
V_{SS} is the ground potential pin for other than P20 to P27, P80 to P83, A/D converter, and operational amplifier.

2.2.11 $\overline{\text{RESET}}$

This is the active-low system reset input pin.

2.2.12 REGC

This is the pin for connecting regulator output (2.5 V) stabilization capacitance for internal operation. Connect this pin to V_{SS} via a capacitor (0.47 to 1 μF: recommended).



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

2.2.13 FLMD0

This is a pin for setting flash memory programming mode.

Connect FLMD0 to V_{SS} in the normal operation mode.

In flash memory programming mode, connect this pin to the flash memory programmer.

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.
See **Figure 2-1** for the configuration of the I/O circuit of each type.

Remark The pins mounted depend on the product. See **1.5 Ordering Information (Top View)** and **2.1 Pin Function List**.

Table 2-2. Pin I/O Circuit Types (1/3)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00/TI000	5-AQ	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P01/TI010/TO00			
P02/INTP8			
P10/TxD6[Note]			
P11/RxD6[Note]			
P12/TOH0/INTP7[Note]			
P13/TOH1/INTP6[Note]			
P20/ANI0/AMP0−	11-P		<Digital input setting and analog input setting> Independently connect to AV_{DD} or AV_{SS} via a resistor. <Digital output setting> Leave open.
P21/ANI1/AMP0OUT	11-S		
P22/ANI2/AMP0+	11-N		
P23/ANI3/AMP1−	11-P		
P24/ANI4/AMP1OUT	11-S		
P25/ANI5/AMP1+	11-N		
P26/ANI6	11-G		
P27/ANI15/AV_{REFM}	11-T		

Note 78K0/KB2-A:
P10/TxD6/TI51/TO51, P11/RxD6/TI50/TO50, P12/TOH0/INTP7/TI000, P13/TOH1/INTP6/TI010/TO00
78K0/KC2-A:
P10/TxD6, P11/RxD6, P12/TOH0/INTP7, P13/TOH1/INTP6

Table 2-2. Pin I/O Circuit Types (2/3)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P31/INTP5/OCD1A (/$\overline{SCK10}$)[Note 1]	5-AQ	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P32/INTP4/OCD1B (/SI10)			
P33/TI51/TO51/INTP3			
P34/TI50/TO50/INTP2			
P35/SO10/INTP1			
P40/RTCCL/RTCDIV	5-AG		
P41/RTC1HZ			
P42/PCL/$\overline{SSI10}$/INTP9	5-AQ		
P60/SCLA0/$\overline{SCK10}$	13-AI		Input: Independently connect to V_{DD} or V_{SS} via a resistor, or connect directly to V_{SS}. Output: Leave this pin open at low-level output after clearing the output latch of the port to 0.
P61/SDAA0/SI10			
P70/KR0 to P75/KR5	5-AQ		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P80/ANI8/AMP2-	11-P		<Digital input setting and analog input setting> Independently connect to AV_{DD} or AV_{SS} via a resistor. <Digital output setting> Leave open.
P81/ANI9/AMP2OUT	11-S		
P82/ANI10/AMP2+	11-N		
P83/ANI11	11-G		
P120/INTP0/EXLVI	5-AQ		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P121/X1/OCD0A[Notes 1, 2]	37		
P122/X2/EXCLK/OCD0B[Note 2]			
P123/XT1[Note 2]			
P124/XT2[Note 2]			
AV_{REFP}	–	–	Make the same potential as the AV_{DD} or V_{DD}.[Note 3]

Notes 1. Process the P31/INTP5/OCD1A (/$\overline{SCK10}$) and P121/X1/OCD0A pins as follows, when it is not used when it is connected to a flash memory programmer or an on-chip debug emulator.

		P31/INTP5/OCD1A (/$\overline{SCK10}$)	P121/X1/OCD0A
Flash memory programmer connection		Connect to V_{SS} via a resistor.	Open or connect to V_{SS} via a resistor.
On-chip debug emulator connection (when it is not used as an on-chip debug mode setting pin)	During reset		
	During reset released	Input: Connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.	Input: Connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.

2. Use recommended connection above in I/O port mode (see **6.3 (1) Clock operation mode select register (OSCCTL), (3) Setting of operation mode for subsystem clock pin**) when these pins are not used.

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

Remark The functions of pins whose names are in parentheses can be used by setting bit 2 (ISC2) of the input switch control register (ISC) to 1.

Table 2-2. Pin I/O Circuit Types (3/3)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
AV_{DD}	–	–	<When any of P20 to P27 and P80 to P83 is specified as a digital port > Make the same potential as the V_{DD}. <When all of P20 to P27 and P80 to P83 are specified as analog ports > Specify a potential that satisfies the condition $AV_{REFP} \leq V_{DD}$.
AV_{SS}	–	–	Make the same potential as the V_{SS}.
FLMD0	38-A	–	Connect to V_{SS}.[Note]
$\overline{\text{RESET}}$	2	Input	Connect directly to V_{DD} or via a resistor.

Note FLMD0 is a pin that is used to write data to the flash memory. To rewrite the data of the flash memory on-board, connect this pin to V_{SS} via a resistor (10 kΩ: recommended). The same applies when executing on-chip debugging with a product with an on-chip debug function.

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

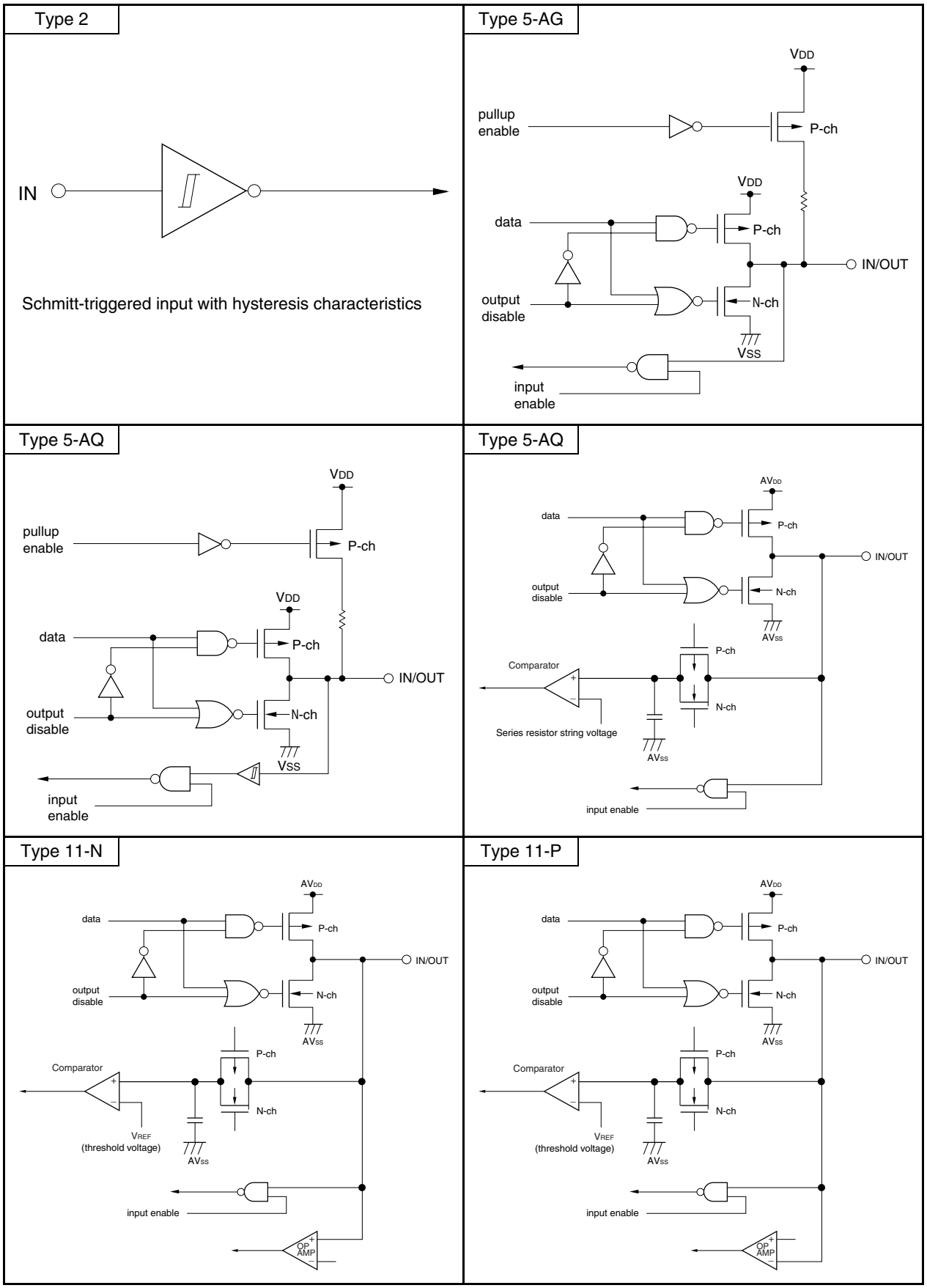
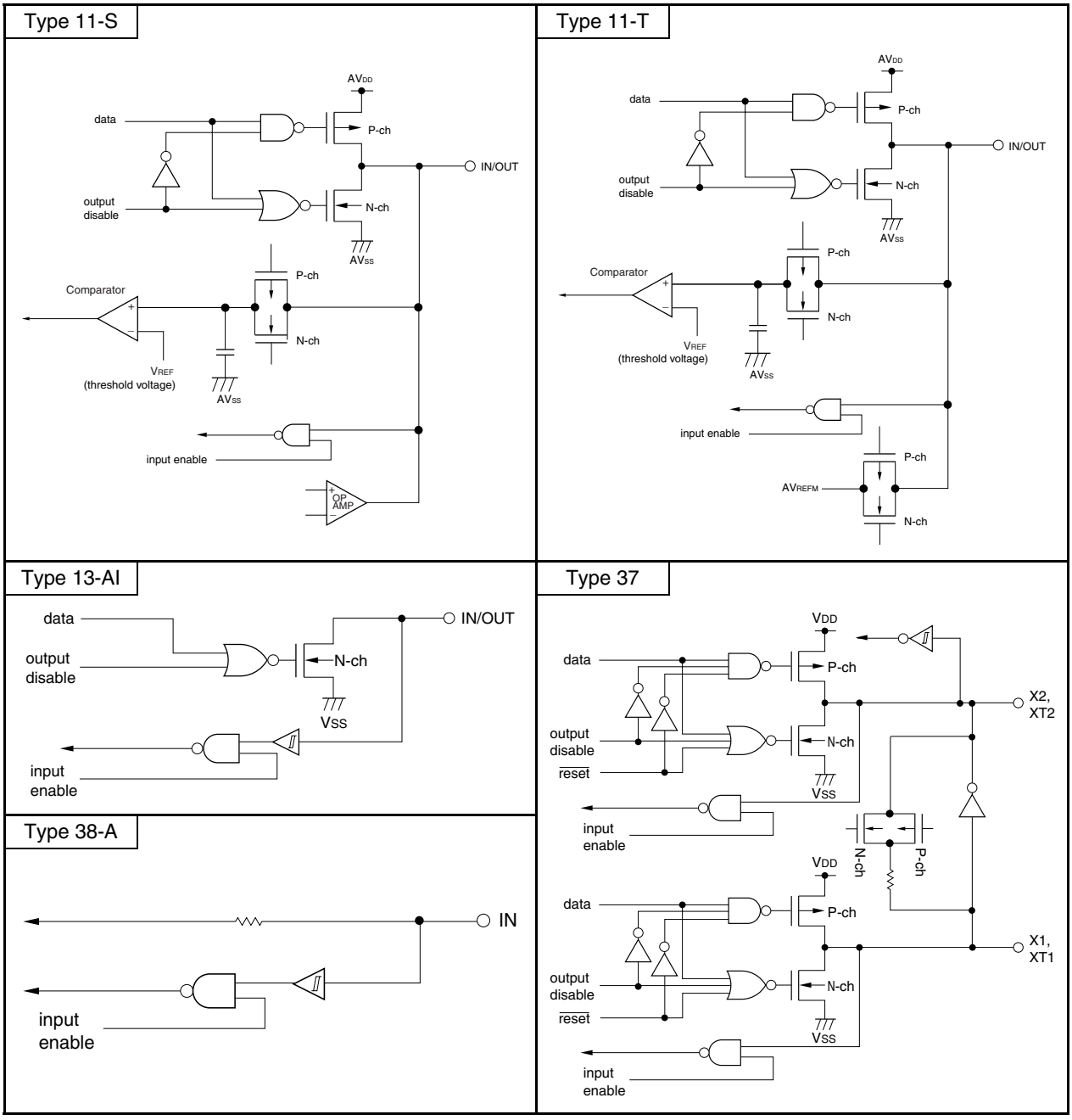


Figure 2-1. Pin I/O Circuit List (2/2)



CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

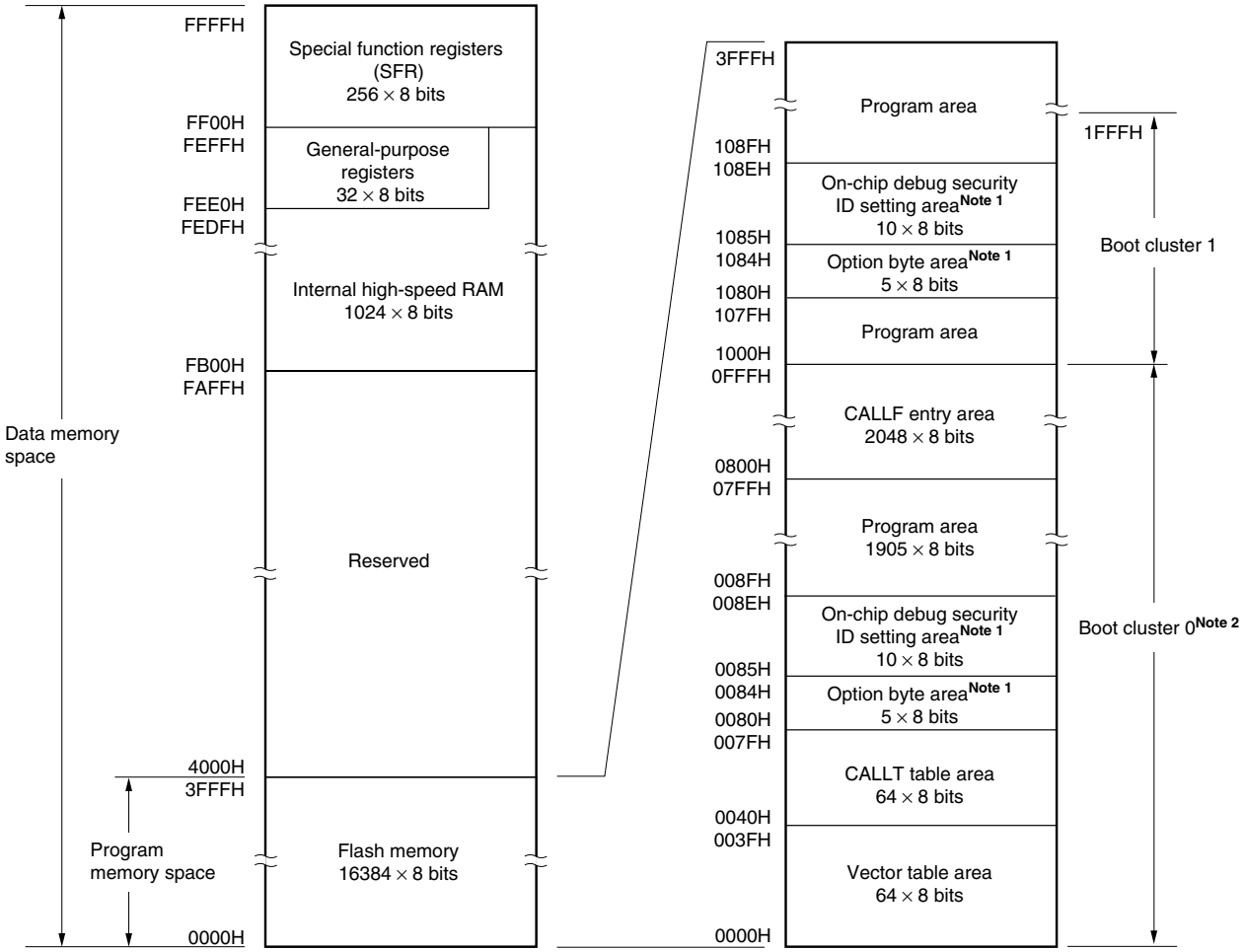
78K0/Kx2-A microcontrollers can access a 64 KB memory space. Figures 3-1 and 3-2 show the memory maps.

Caution Regardless of the internal memory capacity, the initial values of the internal memory size switching register (IMS) of 78K0/Kx2-A microcontrollers are fixed (IMS = CFH). Therefore, set the value corresponding to each product as indicated below.

Table 3-1. Set Values of Internal Memory Size Switching Register (IMS)

Part Number		IMS	ROM Capacity	Internal High-Speed RAM Capacity
78K0/KB2-A	78K0/KC2-A			
μPD78F0590	μPD78F0592	C4H	16 KB	1 KB
μPD78F0591	μPD78F0593	C8H	32 KB	

Figure 3-1. Memory Map (μPD78F0590 and 78F0592)



Notes 1. When boot swap is not used: Set the option bytes to 0080H to 0084H, and the on-chip debug security IDs to 0085H to 008EH.

When boot swap is used: Set the option bytes to 0080H to 0084H and 1080H to 1084H, and the on-chip debug security IDs to 0085H to 008EH and 1085H to 108EH.

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

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

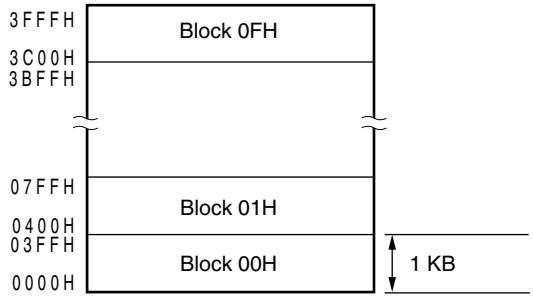
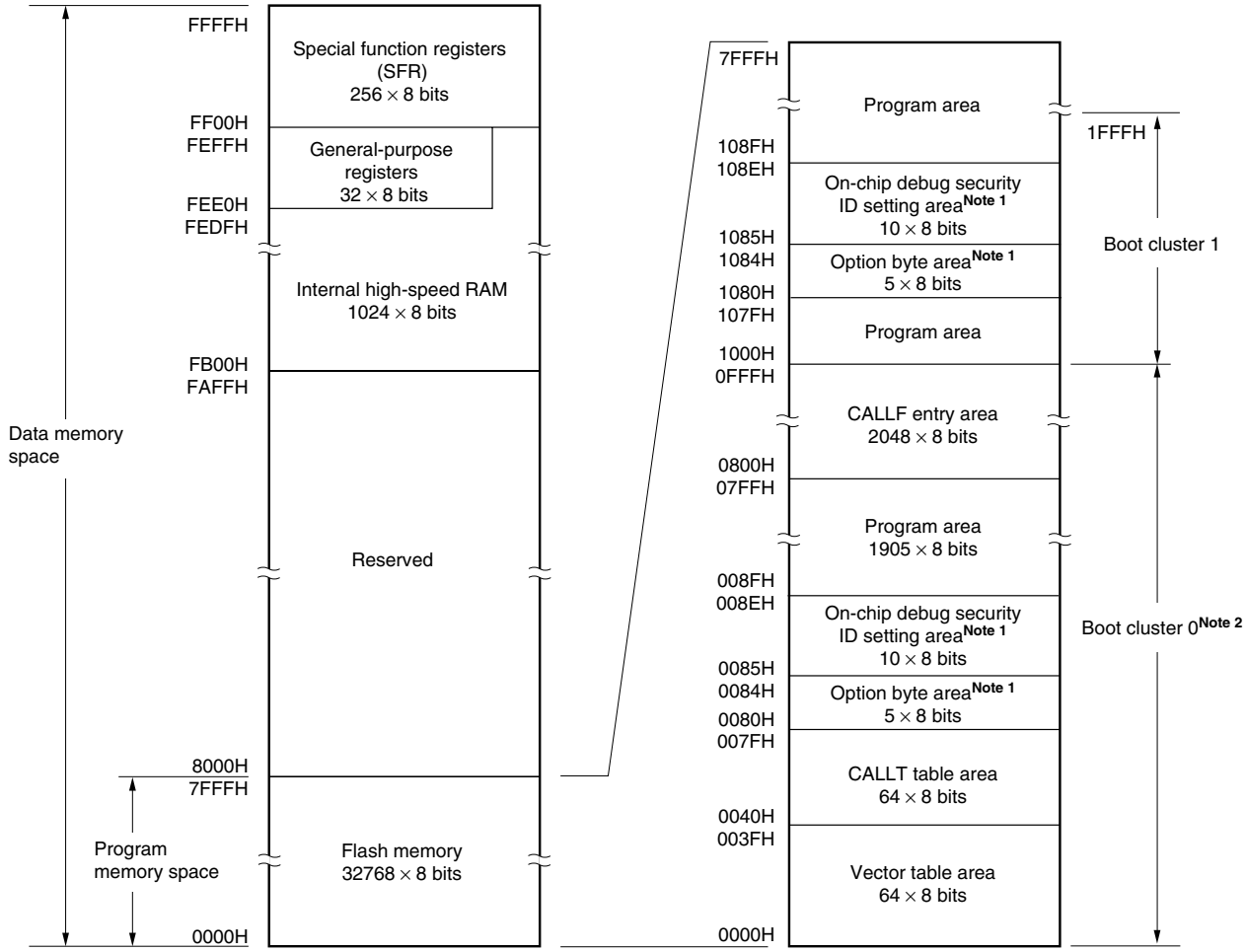


Figure 3-2. Memory Map (μPD78F0591 and 78F0593)

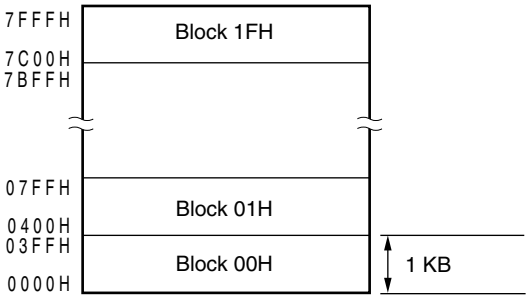


Notes 1. When boot swap is not used: Set the option bytes to 0080H to 0084H, and the on-chip debug security IDs to 0085H to 008EH.

When boot swap is used: Set the option bytes to 0080H to 0084H and 1080H to 1084H, and the on-chip debug security IDs to 0085H to 008EH and 1085H to 108EH.

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

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

Address Value	Block Number	Address Value	Block Number
0000H to 03FFH	00H	4000H to 43FFH	10H
0400H to 07FFH	01H	4400H to 47FFH	11H
0800H to 0BFFH	02H	4800H to 4BFFH	12H
0C00H to 0FFFH	03H	4C00H to 4FFFH	13H
1000H to 13FFH	04H	5000H to 53FFH	14H
1400H to 17FFH	05H	5400H to 57FFH	15H
1800H to 1BFFH	06H	5800H to 5BFFH	16H
1C00H to 1FFFH	07H	5C00H to 5FFFH	17H
2000H to 23FFH	08H	6000H to 63FFH	18H
2400H to 27FFH	09H	6400H to 67FFH	19H
2800H to 2BFFH	0AH	6800H to 6BFFH	1AH
2C00H to 2FFFH	0BH	6C00H to 6FFFH	1BH
3000H to 33FFH	0CH	7000H to 73FFH	1CH
3400H to 37FFH	0DH	7400H to 77FFH	1DH
3800H to 3BFFH	0EH	7800H to 7BFFH	1EH
3C00H to 3FFFH	0FH	7C00H to 7FFFH	1FH

Remark μPD78F0590, 78F0592: Block numbers 00H to 0FH
μPD78F0591, 78F0593: Block numbers 00H to 1FH

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).

78K0/Kx2-A microcontrollers incorporate internal ROM (flash memory), as shown below.

Table 3-3. Internal ROM Capacity

Part Number		Internal ROM (Flash memory)
78K0/KB2-A	78K0/KC2-A	
μPD78F0590	μPD78F0592	16384 × 8 bits (0000H to 3FFFH)
μPD78F0591	μPD78F0593	32768 × 8 bits (0000H to 7FFFH)

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

(1) Vector table area

The 64-byte area 0000H to 003FH 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 3-4. Vector Table

Vector Table Address	Interrupt Source	78K0/KB2-A (μPD78F0590, 78F0591)	78K0/KC2-A (μPD78F0592, 78F0593)
0000H	$\overline{\text{RESET}}$ input, POC, LVI, WDT	√	√
0004H	INTLVI	√	√
0006H	INTP0	√	√
0008H	INTP1	√	√
000AH	INTP2	–	√
000CH	INTP3	–	√
000EH	INTP4	√	√
0010H	INTP5	√	√
0012H	INTSRE6	√	√
0014H	INTSR6	√	√
0016H	INTST6	√	√
0018H	INTCSI10	√	√
001AH	INTTMH1	√	√
001CH	INTTMH0	√	√
001EH	INTTM50	√	√
0020H	INTTM000	√	√
0022H	INTTM010	√	√
0024H	INTAD	√	√
0026H	INTIICA0	√	√
0028H	INTRTCI	–	√
002AH	INTTM51	√	√
002CH	INTKR	–	√
002EH	INTRTC	–	√
0030H	INTP6	√	√
0032H	INTP7	√	√
0034H	INTDMU	√	√
0038H	INTP8	–	√
003AH	INTP9	–	√
003EH	BRK	√	√

Remark √: Mounted, –: Not mounted

(2) CALLT instruction table area

The 64-byte area 0040H to 007FH can store the subroutine entry address of a 1-byte call instruction (CALLT).

(3) Option byte area

A 5-byte area of 0080H to 0084H and 1080H to 1084H can be used as an option byte area. Set the option byte at 0080H to 0084H when the boot swap is not used, and at 0080H to 0084H and 1080H to 1084H when the boot swap is used. For details, see **CHAPTER 24 OPTION BYTE**.

(4) CALLF instruction entry area

The area 0800H to 0FFFH can perform a direct subroutine call with a 2-byte call instruction (CALLF).

(5) On-chip debug security ID setting area

A 10-byte area of 0085H to 008EH and 1085H to 108EH can be used as an on-chip debug security ID setting area. Set the on-chip debug security ID of 10 bytes at 0085H to 008EH when the boot swap is not used and at 0085H to 008EH and 1085H to 108EH when the boot swap is used. For details, see **CHAPTER 26 ON-CHIP DEBUG FUNCTION**.

3.1.2 Internal data memory space

78K0/Kx2-A microcontrollers incorporate the following RAMs.

(1) Internal high-speed RAM

The 32-byte area FEE0H to FEFFH is assigned to four general-purpose register banks consisting of eight 8-bit registers per bank.

This area cannot be used as a program area in which instructions are written and executed.

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

Table 3-5. Internal High-Speed RAM Capacity

Part Number		Internal High-Speed RAM
78K0/KB2-A	78K0/KC2-A	
μPD78F0590	μPD78F0592	1024 × 8 bits (FB00H to FEFFH)
μPD78F0591	μPD78F0593	

3.1.3 Special function register (SFR) area

On-chip peripheral hardware special function registers (SFRs) are allocated in the area FF00H to FFFFH (see **Table 3-6 Special Function Register List** in **3.2.3 Special function registers (SFRs)**).

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

3.1.4 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 78K0/Kx2-A microcontrollers, 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 3-3 and 3-4 show correspondence between data memory and addressing. For details of each addressing mode, see **3.4 Operand Address Addressing**.

Figure 3-3. Correspondence Between Data Memory and Addressing (μPD78F0590 and 78F0592)

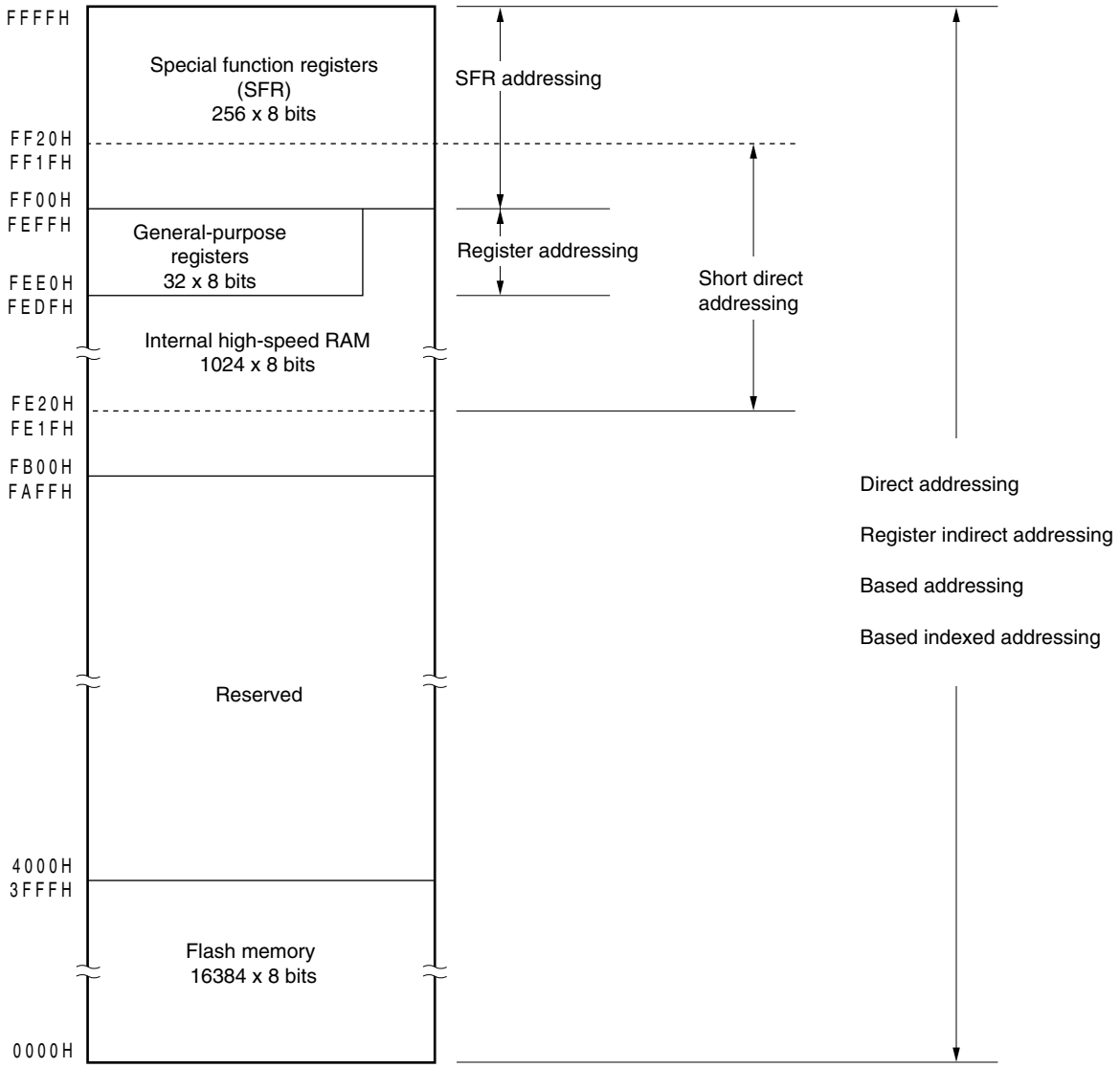
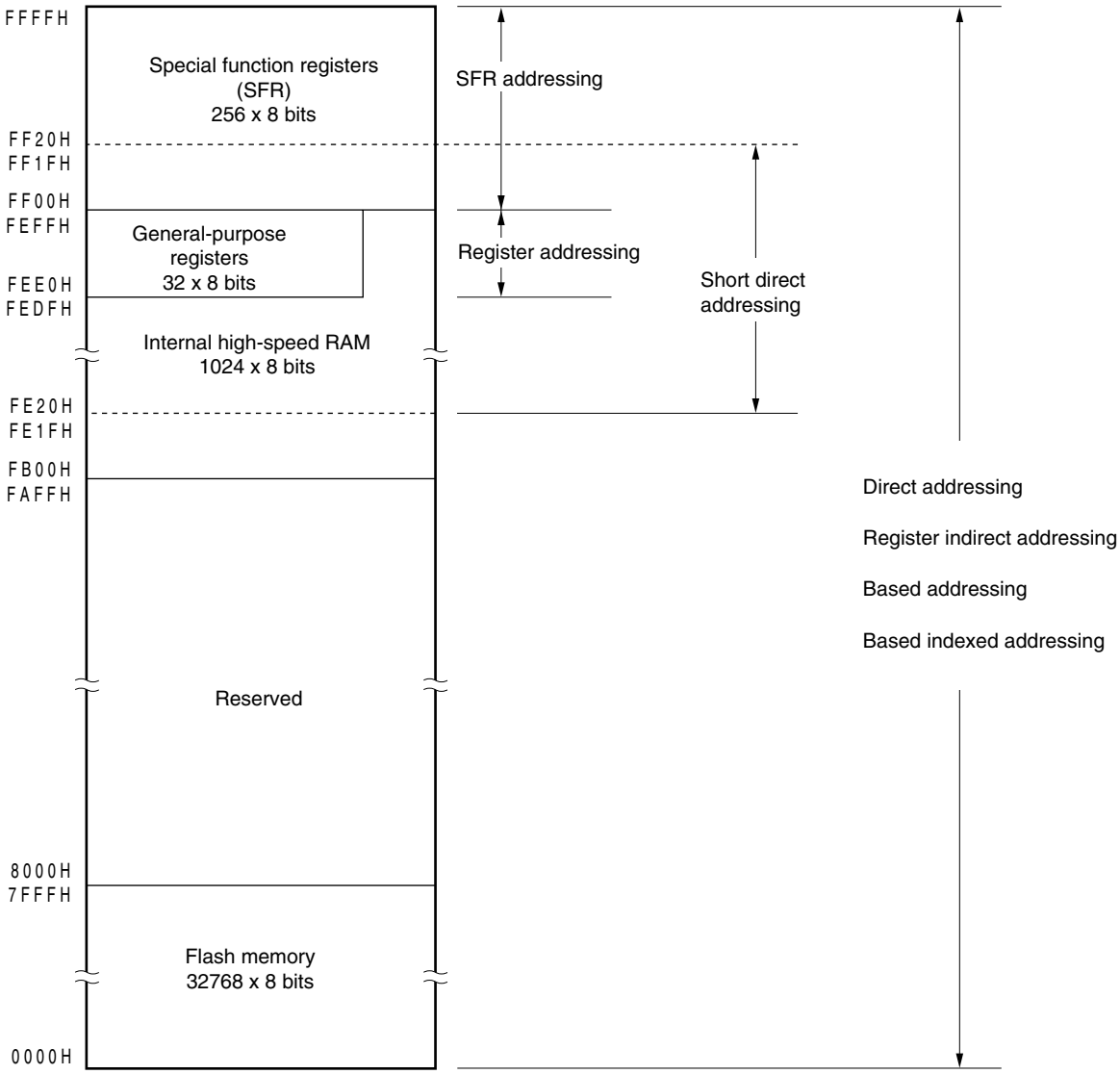


Figure 3-4. Correspondence Between Data Memory and Addressing (μPD78F0591 and 78F0593)



3.2 Processor Registers

78K0/Kx2-A microcontrollers 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 16-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 3-5. Format of Program Counter

	15															0
PC	PC15	PC14	PC13	PC12	PC11	PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0

(2) Program status word (PSW)

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

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

Reset signal generation sets PSW to 02H.

Figure 3-6. Format of Program Status Word

	7							0
PSW	IE	Z	RBS1	AC	RBS0	0	ISP	CY

(a) Interrupt enable flag (IE)

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

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

When 1, the IE flag is set to the interrupt enabled (EI) state and interrupt request acknowledgment is controlled with an in-service priority flag (ISP), 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 and 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 flag (ISP)

This flag manages the priority of acknowledgeable maskable vectored interrupts. When this flag is 0, low-level vectored interrupt requests specified by a priority specification flag register (PR0L, PR0H, PR1L, PR1H) (see **18.3 (3) Priority specification flag registers (PR0L, PR0H, PR1L, PR1H)**) can not be acknowledged. Actual request acknowledgment is controlled by the interrupt enable flag (IE).

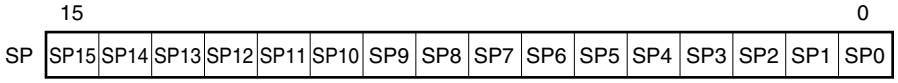
(f) Carry flag (CY)

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 high-speed RAM area can be set as the stack area.

Figure 3-7. Format of Stack Pointer



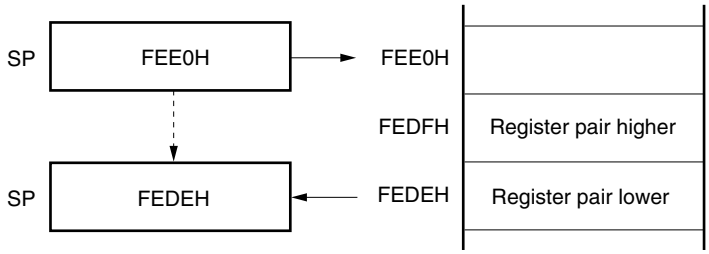
The SP is decremented ahead of write (save) to the stack memory and is incremented after read (restored) from the stack memory.

Each stack operation saves/restores data as shown in Figures 3-8 and 3-9.

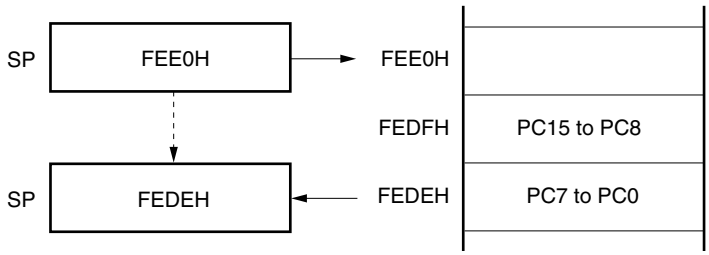
Caution Since reset signal generation makes the SP contents undefined, be sure to initialize the SP before using the stack.

Figure 3-8. Data to Be Saved to Stack Memory

(a) PUSH rp instruction (when SP = FEE0H)



(b) CALL, CALLF, CALLT instructions (when SP = FEE0H)



(c) Interrupt, BRK instructions (when SP = FEE0H)

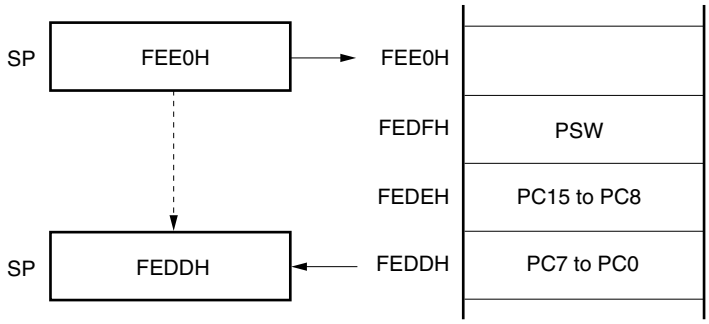
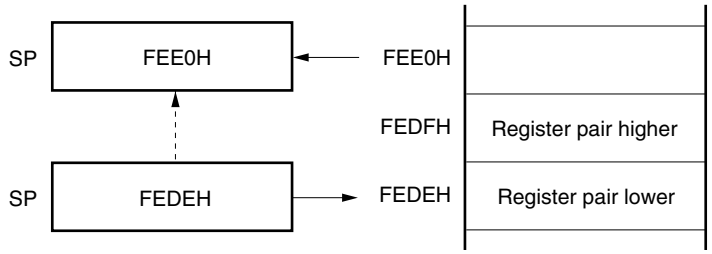
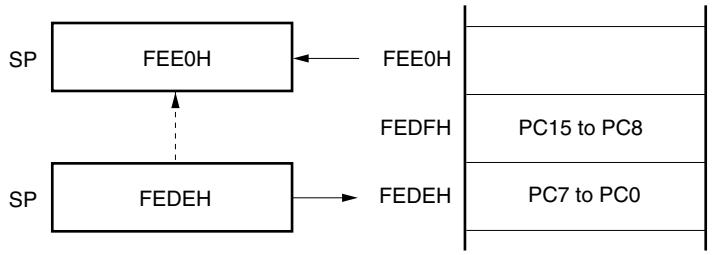


Figure 3-9. Data to Be Restored from Stack Memory

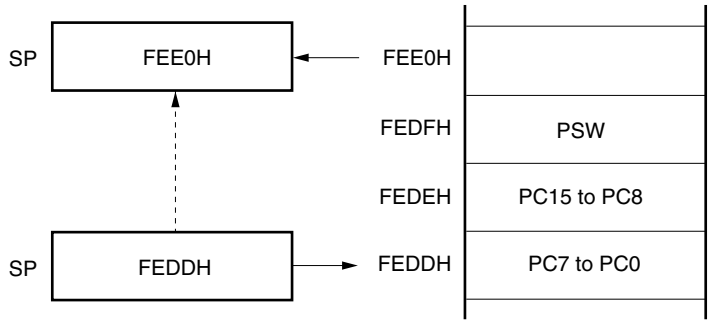
(a) POP rp instruction (when SP = FEDEH)



(b) RET instruction (when SP = FEDEH)



(c) RETI, RETB instructions (when SP = FEDDH)



3.2.2 General-purpose registers

General-purpose registers are mapped at particular addresses (FEE0H to FEFFH) 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).

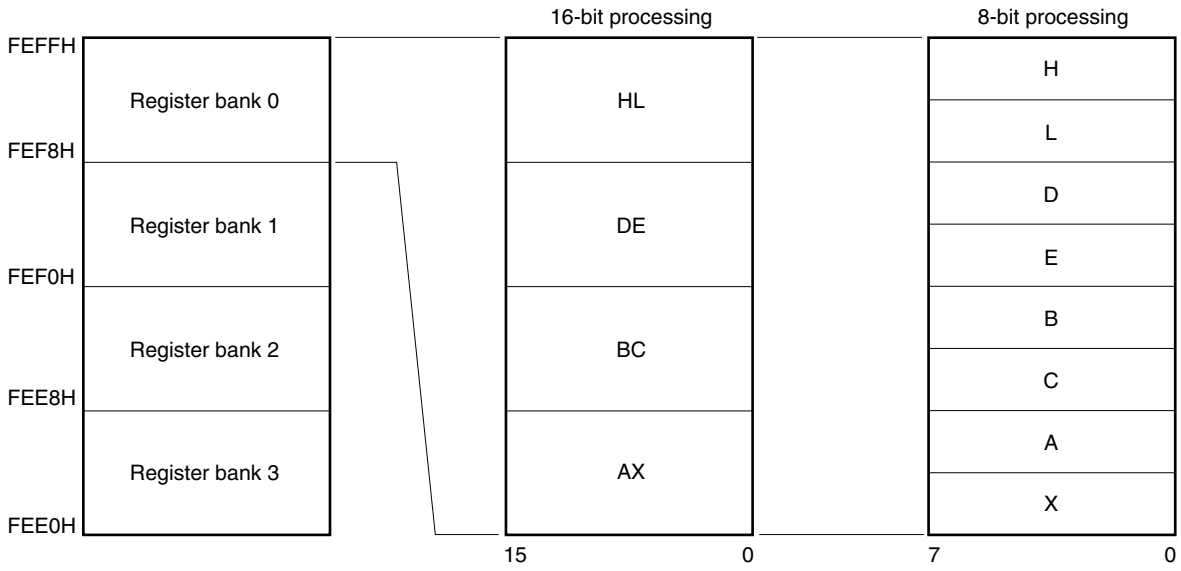
Each register can be used as an 8-bit register, and two 8-bit registers can also be used in a pair as a 16-bit register (AX, BC, DE, and HL).

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).

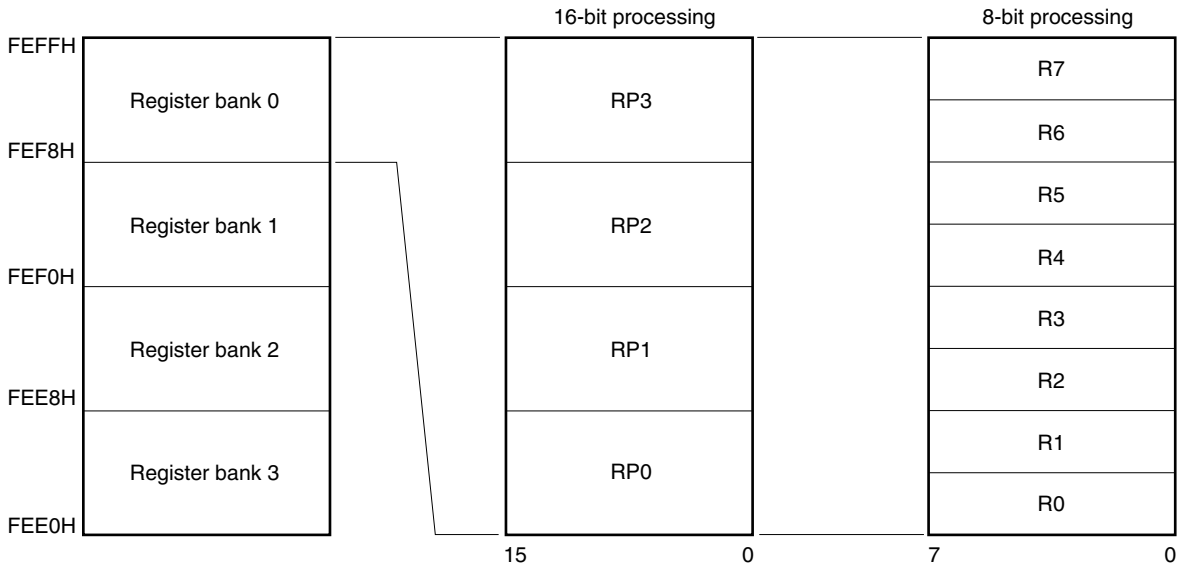
Register banks to be used for instruction execution are set by the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interrupts for each bank.

Figure 3-10. Configuration of General-Purpose Registers

(a) Function name



(b) Absolute name



3.2.3 Special function registers (SFRs)

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

SFRs are allocated to the FF00H to FFFFH area.

Special function registers can be manipulated like general-purpose registers, using operation, transfer, and bit manipulation instructions. The manipulatable bit units, 1, 8, and 16, depend on the special function register 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 3-6 gives a list of the special function registers. 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 RA78K0, and is defined as an sfr variable using the #pragma sfr directive in the CC78K0. When using the RA78K0, ID78K0-QB, SM+ for 78K0, and SM+ for 78K0/KX2, symbols can be written as an instruction operand.
- R/W
  Indicates whether the corresponding special function register can be read or written.
  R/W: Read/write enable
  R: Read only
  W: Write only
- Manipulatable bit units
  Indicates the manipulatable 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.

Table 3-6. Special Function Register List (1/4)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset	30-pin products	48-pin products
				1 Bit	8 Bits	16 Bits			
FF00H	Port register 0	P0	R/W	√	√	–	00H	–	√
FF01H	Port register 1	P1	R/W	√	√	–	00H	√	√
FF02H	Port register 2	P2	R/W	√	√	–	00H	√	√
FF03H	Port register 3	P3	R/W	√	√	–	00H	√	√
FF04H	Port register 4	P4	R/W	√	√	–	00H	–	√
FF06H	Port register 6	P6	R/W	√	√	–	00H	√	√
FF07H	Port register 7	P7	R/W	√	√	–	00H	–	√
FF08H	Port register 8	P8	R/W	√	√	–	00H	√	√
FF0AH	Receive buffer register 6	RXB6	R	–	√	–	FFH	√	√
FF0BH	Transmit buffer register 6	TXB6	R/W	–	√	–	FFH	√	√
FF0CH	Port register 12	P12	R/W	√	√	–	00H	√	√
FF0DH	8-bit timer H compare register 10	CMP10	R/W	–	√	–	00H	√	√
FF0EH	8-bit timer H compare register 00	CMP00	R/W	–	√	–	00H	√	√
FF0FH	Serial I/O shift register 10	SIO10	R	–	√	–	00H	√	√
FF10H	16-bit timer counter 00	TM00	R	–	–	√	0000H	√	√
FF11H									
FF12H	16-bit timer capture/compare register 000	CR000	R/W	–	–	√	0000H	√	√
FF13H									
FF14H	16-bit timer capture/compare register 010	CR010	R/W	–	–	√	0000H	√	√
FF15H									
FF16H	8-bit timer counter 50	TM50	R	–	√	–	00H	√	√
FF17H	8-bit timer compare register 50	CR50	R/W	–	√	–	00H	√	√
FF18H	12-bit A/D conversion result register	ADCR	R	–	–	√	0000H	√	√
FF19H	8-bit A/D conversion result register	ADCRH	R	–	√	–	00H	√	√
FF1AH	8-bit timer H compare register 01	CMP01	R/W	–	√	–	00H	√	√
FF1BH	8-bit timer H compare register 11	CMP11	R/W	–	√	–	00H	√	√
FF1FH	8-bit timer counter 51	TM51	R	–	√	–	00H	√	√
FF20H	Port mode register 0	PM0	R/W	√	√	–	FFH	–	√
FF21H	Port mode register 1	PM1	R/W	√	√	–	FFH	√	√
FF22H	Port mode register 2	PM2	R/W	√	√	–	FFH	√	√
FF23H	Port mode register 3	PM3	R/W	√	√	–	FFH	√	√
FF24H	Port mode register 4	PM4	R/W	√	√	–	FFH	–	√
FF26H	Port mode register 6	PM6	R/W	√	√	–	FFH	√	√
FF27H	Port mode register 7	PM7	R/W	√	√	–	FFH	–	√
FF28H	Port mode register 8	PM8	R/W	√	√	–	FFH	√	√
FF2CH	Port mode register 12	PM12	R/W	√	√	–	FFH	√	√
FF2EH	Analog reference voltage control register	ADVRC	R/W	√	√	–	00H	√	√
FF2FH	A/D port configuration register	ADPC	R/W	√	√	–	10H	√	√

Table 3-6. Special Function Register List (2/4)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset	30-pin products	48-pin products
					1 Bit	8 Bits	16 Bits			
FF30H	Pull-up resistor option register 0	PU0		R/W	√	√	–	00H	–	√
FF31H	Pull-up resistor option register 1	PU1		R/W	√	√	–	00H	√	√
FF33H	Pull-up resistor option register 3	PU3		R/W	√	√	–	00H	√	√
FF34H	Pull-up resistor option register 4	PU4		R/W	√	√	–	00H	–	√
FF37H	Pull-up resistor option register 7	PU7		R/W	√	√	–	00H	–	√
FF38H	A/D converter mode register	ADM		R/W	√	√	–	00H	√	√
FF39H	Analog input channel specification register	ADS		R/W	√	√	–	00H	√	√
FF3AH	A/D converter mode register 1	ADM1		R/W	√	√	–	00H	√	√
FF3BH	Operational amplifier control register	OAC		R/W	√	√	–	00H	√	√
FF3CH	Pull-up resistor option register 12	PU12		R/W	√	√	–	00H	√	√
FF40H	Clock output selection register	CKS		R/W	√	√	–	00H	–	√
FF41H	8-bit timer compare register 51	CR51		R/W	–	√	–	00H	√	√
FF43H	8-bit timer mode control register 51	TMC51		R/W	√	√	–	00H	√	√
FF48H	External interrupt rising edge enable register 0	EGP0		R/W	√	√	–	00H	√	√
FF49H	External interrupt falling edge enable register 0	EGN0		R/W	√	√	–	00H	√	√
FF4AH	External interrupt rising edge enable register 1	EGP1		R/W	√	√	–	00H	–	√
FF4BH	External interrupt falling edge enable register 1	EGN1		R/W	√	√	–	00H	–	√
FF4FH	Input switch control register	ISC		R/W	√	√	–	00H	√	√
FF50H	Asynchronous serial interface operation mode register 6	ASIM6		R/W	√	√	–	01H	√	√
FF53H	Asynchronous serial interface reception error status register 6	ASIS6		R	–	√	–	00H	√	√
FF55H	Asynchronous serial interface transmission status register 6	ASIF6		R	–	√	–	00H	√	√
FF56H	Clock selection register 6	CKSR6		R/W	–	√	–	00H	√	√
FF57H	Baud rate generator control register 6	BRGC6		R/W	–	√	–	FFH	√	√
FF58H	Asynchronous serial interface control register 6	ASICL6		R/W	√	√	–	16H	√	√
FF60H	Remainder data register 0	SD	SDR0L	R	–	√	√	00H	√	√
FF61H		R0	SDR0H		–	√		00H	√	√
FF62H	Multiplication/division data register A0	MD	MDA0LL	R/W	–	√	√	00H	√	√
FF63H		A0L	MDA0LH		–	√		00H	√	√
FF64H		MD	MDA0HL	R/W	–	√	√	00H	√	√
FF65H		A0H	MDA0HH		–	√		00H	√	√
FF66H	Multiplication/division data register B0	MD	MDB0L	R/W	–	√	√	00H	√	√
FF67H		B0	MDB0H		–	√		00H	√	√

Table 3-6. Special Function Register List (3/4)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset	30-pin products	48-pin products
				1 Bit	8 Bits	16 Bits			
FF68H	Multiplier/divider control register 0	DMUC0	R/W	√	√	–	00H	√	√
FF69H	8-bit timer H mode register 0	TMHMD0	R/W	√	√	–	00H	√	√
FF6AH	Timer clock selection register 50	TCL50	R/W	√	√	–	00H	√	√
FF6BH	8-bit timer mode control register 50	TMC50	R/W	√	√	–	00H	√	√
FF6CH	8-bit timer H mode register 1	TMHMD1	R/W	√	√	–	00H	√	√
FF6DH	8-bit timer H carrier control register 1	TMCYC1	R/W	√	√	–	00H	√	√
FF6EH	Key return mode register	KRM	R/W	√	√	–	00H	–	√
FF70H	Sub-count register	RSUBC	R	–	–	√	0000H	–	√
FF71H									
FF72H	Second count register	SEC	R/W	–	√	–	00H	–	√
FF73H	Minute count register	MIN	R/W	–	√	–	00H	–	√
FF74H	Hour count register	HOUR	R/W	–	√	–	12H[Note1]	–	√
FF75H	Week count register	WEEK	R/W	–	√	–	00H	–	√
FF76H	Day count register	DAY	R/W	–	√	–	01H	–	√
FF77H	Month count register	MONTH	R/W	–	√	–	01H	–	√
FF78H	Year count register	YEAR	R/W	–	√	–	00H	–	√
FF79H	Watch error correction register	SUBCUD	R/W	–	√	–	00H	–	√
FF7AH	Alarm minute register	ALARMWM	R/W	–	√	–	00H	–	√
FF7BH	Alarm hour register	ALARMWH	R/W	–	√	–	12H	–	√
FF7CH	Alarm week register	ALARMWW	R/W	–	√	–	00H	–	√
FF7DH	Real-time counter control register 0	RTCC0	R/W	√	√	–	00H	–	√
FF7EH	Real-time counter control register 1	RTCC1	R/W	√	√	–	00H	–	√
FF7FH	Real-time counter control register 2	RTCC2	R/W	√	√	–	00H	–	√
FF80H	Serial operation mode register 10	CSIM10	R/W	√	√	–	00H	√	√
FF81H	Serial clock selection register 10	CSIC10	R/W	√	√	–	00H	√	√
FF84H	Transmit buffer register 10	SOTB10	R/W	–	√	–	00H	√	√
FF8CH	Timer clock selection register 51	TCL51	R/W	√	√	–	00H	√	√
FF99H	Watchdog timer enable register	WDTE	R/W	–	√	–	1AH/ 9AH[Note 2]	√	√
FF9FH	Clock operation mode select register	OSCCTL	R/W	√	√	–	00H	√	√
FFA0H	Internal oscillation mode register	RCM	R/W	√	√	–	80H[Note 3]	√	√
FFA1H	Main clock mode register	MCM	R/W	√	√	–	00H	√	√
FFA2H	Main OSC control register	MOC	R/W	√	√	–	80H	√	√
FFA3H	Oscillation stabilization time counter status register	OSTC	R	√	√	–	00H	√	√
FFA4H	Oscillation stabilization time select register	OSTS	R/W	–	√	–	05H	√	√

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 WDTE is determined by setting of option byte.

3. The value of this register is 00H immediately after a reset release but automatically changes to 80H after oscillation accuracy stabilization of high-speed internal oscillator has been waited.

Table 3-6. Special Function Register List (4/4)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset	30-pin products	48-pin products
					1 Bit	8 Bits	16 Bits			
FFA6H	IICA shift register	IICA		R/W	–	√	–	00H	√	√
FFA7H	Slave address register 0	SVA0		R/W	–	√	–	00H	√	√
FFA8H	IICA control register 0	IICACTL0		R/W	√	√	–	00H	√	√
FFA9H	IICA control register 1	IICACTL1		R/W	√	√	–	00H	√	√
FFAAH	IICA flag register 0	IICAF0		R/W	√	√	–	00H	√	√
FFABH	IICA status register 0	IICAS0		R	√	√	–	00H	√	√
FFACH	Reset control flag register	RESF		R	–	√	–	00H[Note 1]	√	√
FFAEH	IICA low-level width setting register	IICWL		R/W	–	√	–	FFH	√	√
FFAFH	IICA high-level width setting register	IICWH		R/W	–	√	–	FFH	√	√
FFBAH	16-bit timer mode control register 00	TMC00		R/W	√	√	–	00H	√	√
FFBBH	Prescaler mode register 00	PRM00		R/W	√	√	–	00H	√	√
FFBCH	Capture/compare control register 00	CRC00		R/W	√	√	–	00H	√	√
FFBDH	16-bit timer output control register 00	TOC00		R/W	√	√	–	00H	√	√
FFBEH	Low-voltage detection register	LVIM		R/W	√	√	–	00H[Note 2]	√	√
FFBFH	Low-voltage detection level selection register	LVIS		R/W	√	√	–	00H[Note 2]	√	√
FFE0H	Interrupt request flag register 0L	IF0	IF0L	R/W	√	√	√	00H	√	√
FFE1H	Interrupt request flag register 0H		IF0H	R/W	√	√		00H	√	√
FFE2H	Interrupt request flag register 1L	IF1	IF1L	R/W	√	√	√	00H	√	√
FFE3H	Interrupt request flag register 1H		IF1H	R/W	√	√		00H	√	√
FFE4H	Interrupt mask flag register 0L	MK0	MK0L	R/W	√	√	√	FFH	√	√
FFE5H	Interrupt mask flag register 0H		MK0H	R/W	√	√		FFH	√	√
FFE6H	Interrupt mask flag register 1L	MK1	MK1L	R/W	√	√	√	FFH	√	√
FFE7H	Interrupt mask flag register 1H		MK1H	R/W	√	√		FFH	√	√
FFE8H	Priority specification flag register 0L	PR0	PR0L	R/W	√	√	√	FFH	√	√
FFE9H	Priority specification flag register 0H		PR0H	R/W	√	√		FFH	√	√
FFEAH	Priority specification flag register 1L	PR1	PR1L	R/W	√	√	√	FFH	√	√
FFEBH	Priority specification flag register 1H		PR1H	R/W	√	√		FFH	√	√
FFF0H	Internal memory size switching register[Note 3]	IMS		R/W	–	√	–	CFH	√	√
FFFBH	Processor clock control register	PCC		R/W	√	√	–	01H	√	√

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

2. The reset values of LVIM and LVIS vary depending on the reset source.

3. Regardless of the internal memory capacity, the initial values of the internal memory size switching register (IMS) of 78K0/Kx2-A microcontrollers are fixed (IMS = CFH). Therefore, set the value corresponding to each product as indicated in Table 3-1.

3.3 Instruction Address Addressing

An instruction address is determined by contents of the program counter (PC), and is normally incremented (+1 for each byte) automatically according to the number of bytes of an instruction to be fetched each time another instruction is executed. When a branch instruction is executed, the branch destination information is set to PC and branched by the following addressing (for details of instructions, refer to the **78K/0 Series Instructions User's Manual (U12326E)**).

3.3.1 Relative addressing

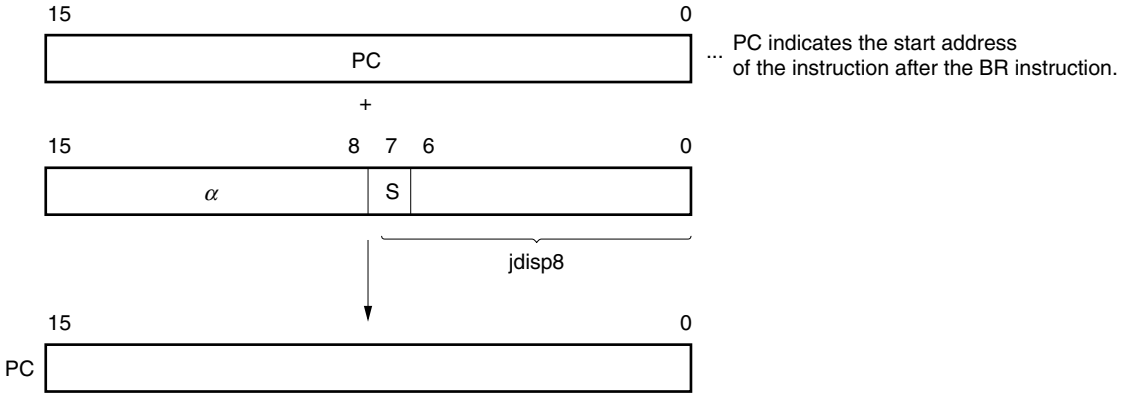
[Function]

The value obtained by adding 8-bit immediate data (displacement value: jdisp8) of an instruction code to the start address of the following instruction is transferred to the program counter (PC) and branched. The displacement value is treated as signed two's complement data (−128 to +127) and bit 7 becomes a sign bit.

In other words, relative addressing consists of relative branching from the start address of the following instruction to the −128 to +127 range.

This function is carried out when the BR $addr16 instruction or a conditional branch instruction is executed.

[Illustration]



When S = 0, all bits of α are 0.
When S = 1, all bits of α are 1.

3.3.2 Immediate addressing

[Function]

Immediate data in the instruction word is transferred to the program counter (PC) and branched.

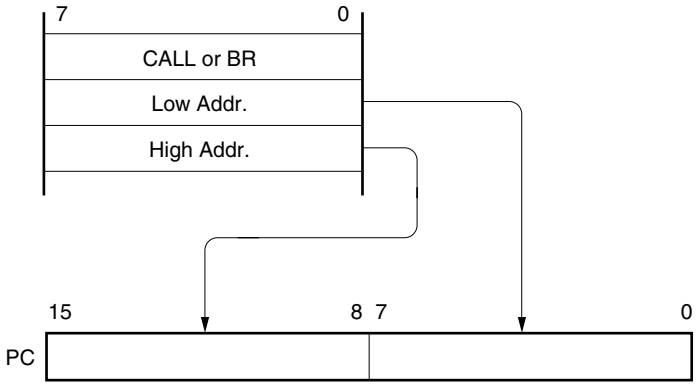
This function is carried out when the CALL !addr16 or BR !addr16 or CALLF !addr11 instruction is executed.

CALL !addr16 and BR !addr16 instructions can be branched to the entire memory space.

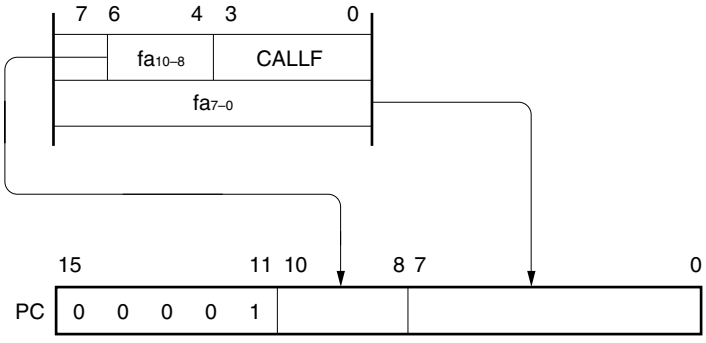
The CALLF !addr11 instruction is branched to the 0800H to 0FFFH area.

[Illustration]

In the case of CALL !addr16 and BR !addr16 instructions



In the case of CALLF !addr11 instruction



3.3.3 Table indirect addressing

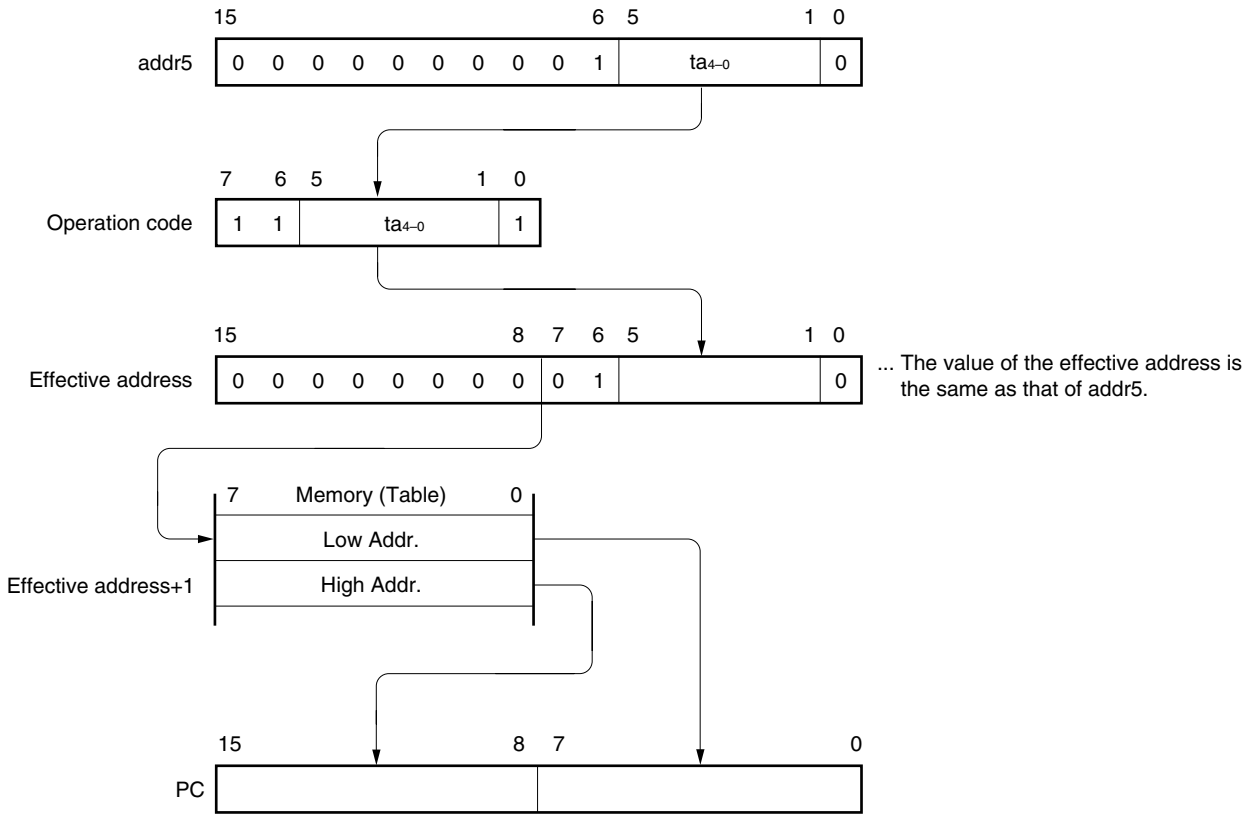
[Function]

Table contents (branch destination address) of the particular location to be addressed by bits 1 to 5 of the immediate data of an operation code are transferred to the program counter (PC) and branched.

This function is carried out when the CALLT [addr5] instruction is executed.

This instruction references the address that is indicated by addr5 and is stored in the memory table from 0040H to 007FH, and allows branching to the entire memory space.

[Illustration]



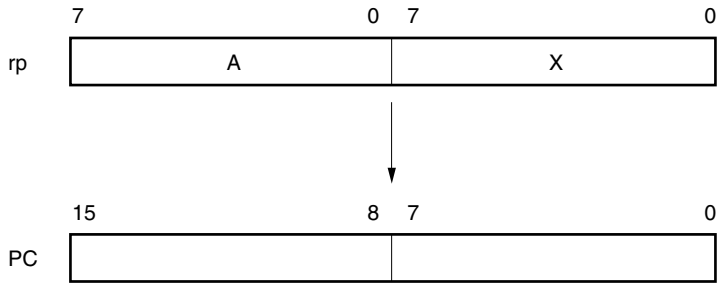
3.3.4 Register addressing

[Function]

Register pair (AX) contents to be specified with an instruction word are transferred to the program counter (PC) and branched.

This function is carried out when the BR AX instruction is executed.

[Illustration]



3.4 Operand Address Addressing

The following methods are available to specify the register and memory (addressing) to undergo manipulation during instruction execution.

3.4.1 Implied addressing

[Function]

The register that functions as an accumulator (A and AX) among the general-purpose registers is automatically (implicitly) addressed.

Of the 78K0/Kx2-A microcontrollers instruction words, the following instructions employ implied addressing.

Instruction	Register to Be Specified by Implied Addressing
MULU	A register for multiplicand and AX register for product storage
DIVUW	AX register for dividend and quotient storage
ADJBA/ADJBS	A register for storage of numeric values that become decimal correction targets
ROR4/ROL4	A register for storage of digit data that undergoes digit rotation

[Operand format]

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

[Description example]

In the case of MULU X

With an 8-bit × 8-bit multiply instruction, the product of the A register and X register is stored in AX. In this example, the A and AX registers are specified by implied addressing.

3.4.2 Register addressing

[Function]

The general-purpose register to be specified is accessed as an operand with the register bank select flags (RBS0 to RBS1) and the register specify codes of an operation code.

Register addressing is carried out when an instruction with the following operand format is executed. When an 8-bit register is specified, one of the eight registers is specified with 3 bits in the operation code.

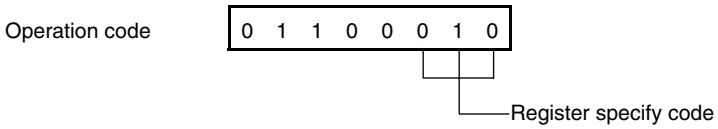
[Operand format]

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

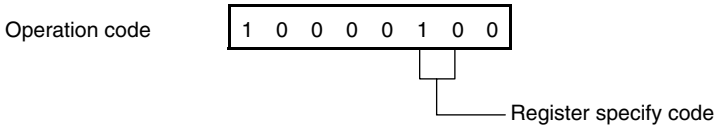
'r' and 'rp' can be described by absolute names (R0 to R7 and RP0 to RP3) as well as function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL).

[Description example]

MOV A, C; when selecting C register as r



INCW DE; when selecting DE register pair as rp



3.4.3 Direct addressing

[Function]

The memory to be manipulated is directly addressed with immediate data in an instruction word becoming an operand address.

This addressing can be carried out for all of the memory spaces.

[Operand format]

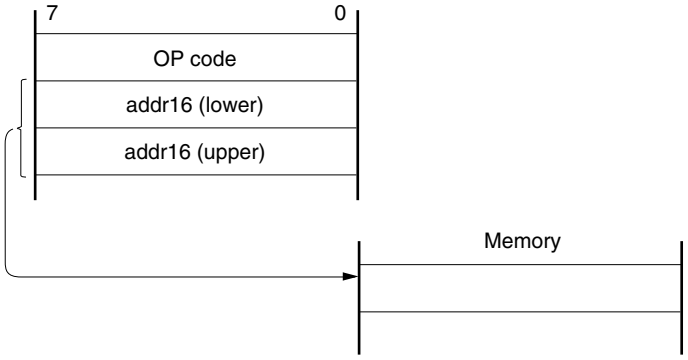
Identifier	Description
addr16	Label or 16-bit immediate data

[Description example]

MOV A, !0FE00H; when setting !addr16 to FE00H

Operation code	1 0 0 0 1 1 1 0	OP code
	0 0 0 0 0 0 0 0	00H
	1 1 1 1 1 1 1 0	FEH

[Illustration]



3.4.4 Short direct addressing

[Function]

The memory to be manipulated in the fixed space is directly addressed with 8-bit data in an instruction word.

This addressing is applied to the 256-byte space FE20H to FF1FH. Internal high-speed RAM and special function registers (SFRs) are mapped at FE20H to FEFFH and FF00H to FF1FH, respectively.

The SFR area (FF00H to FF1FH) where short direct addressing is applied is a part of the overall SFR area. Ports that are frequently accessed in a program and compare and capture registers of the timer/event counter are mapped in this area, allowing SFRs to be manipulated with a small number of bytes and clocks.

When 8-bit immediate data is at 20H to FFH, bit 8 of an effective address is set to 0. When it is at 00H to 1FH, bit 8 is set to 1. See the **[Illustration]** shown below.

[Operand format]

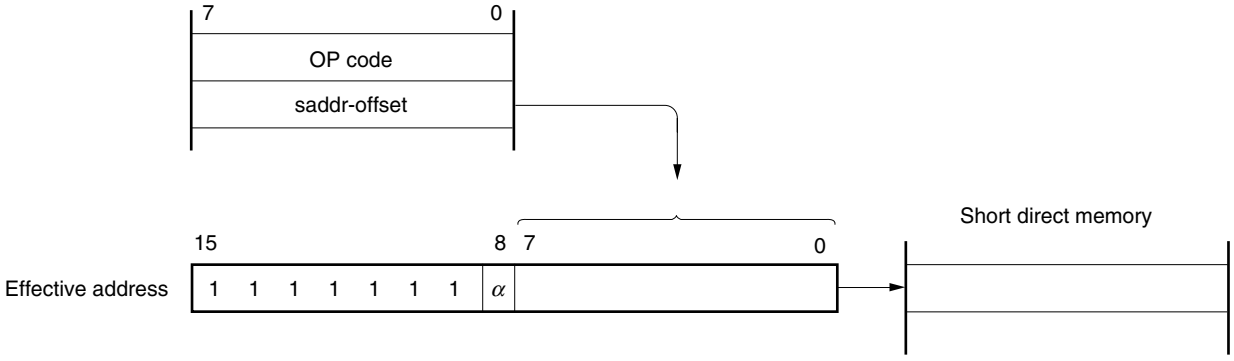
Identifier	Description
saddr	Immediate data that indicate label or FE20H to FF1FH
saddrp	Immediate data that indicate label or FE20H to FF1FH (even address only)

[Description example]

```
LB1 EQU 0FE30H   ; Defines FE30H by LB1.
      :
MOV LB1, A       ; When LB1 indicates FE30H of the saddr area and the value of register A is transferred to
                   that address
```

Operation code	1 1 1 1 0 0 1 0	OP code
	0 0 1 1 0 0 0 0	30H (saddr-offset)

[Illustration]



When 8-bit immediate data is 20H to FFH, $\alpha = 0$

When 8-bit immediate data is 00H to 1FH, $\alpha = 1$

3.4.5 Special function register (SFR) addressing

[Function]

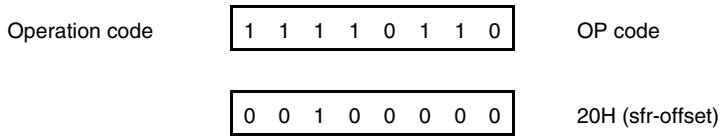
A memory-mapped special function register (SFR) is addressed with 8-bit immediate data in an instruction word. This addressing is applied to the 240-byte spaces FF00H to FFCFH and FFE0H to FFFFH. However, the SFRs mapped at FF00H to FF1FH can be accessed with short direct addressing.

[Operand format]

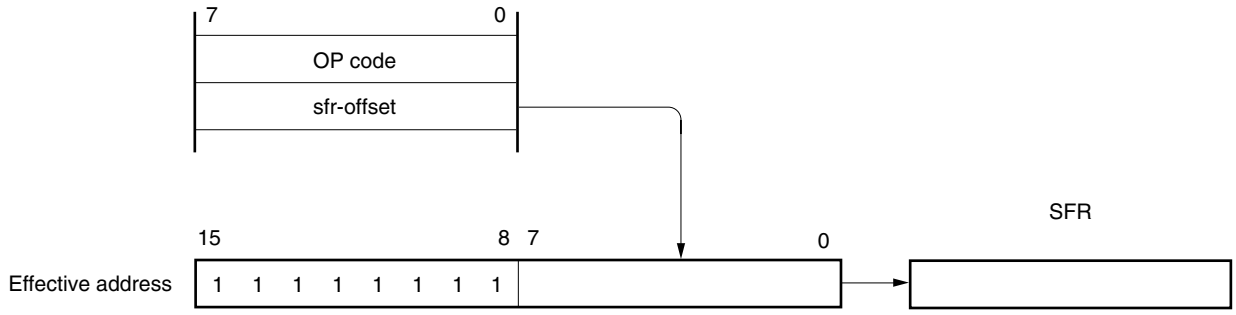
Identifier	Description
sfr	Special function register name
sfrp	16-bit manipulatable special function register name (even address only)

[Description example]

MOV PM0, A; when selecting PM0 (FF20H) as sfr



[Illustration]



3.4.6 Register indirect addressing

[Function]

Register pair contents specified by a register pair specify code in an instruction word and by a register bank select flag (RBS0 and RBS1) serve as an operand address for addressing the memory.

This addressing can be carried out for all of the memory spaces.

[Operand format]

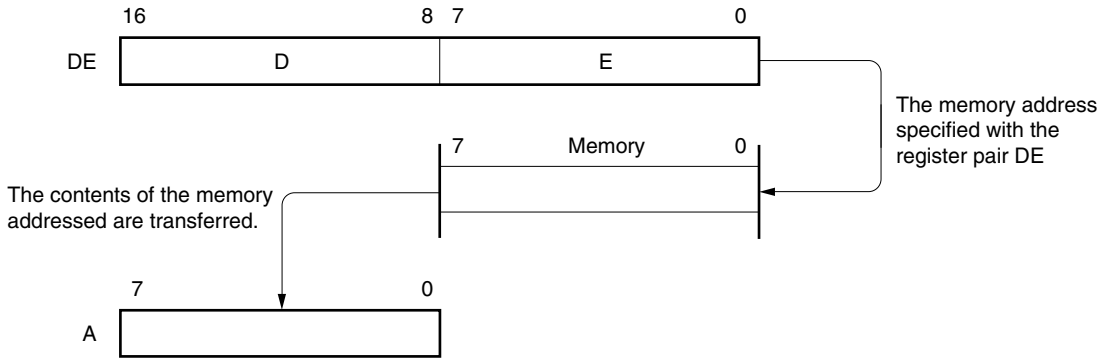
Identifier	Description
–	[DE], [HL]

[Description example]

MOV A, [DE]; when selecting [DE] as register pair

Operation code	1	0	0	0	0	1	0	1

[Illustration]



3.4.7 Based addressing

[Function]

8-bit immediate data is added as offset data to the contents of the base register, that is, the HL register pair in the register bank specified by the register bank select flag (RBS0 and RBS1), and the sum is used to address the memory. Addition is performed by expanding the offset data as a positive number to 16 bits. A carry from the 16th bit is ignored.

This addressing can be carried out for all of the memory spaces.

[Operand format]

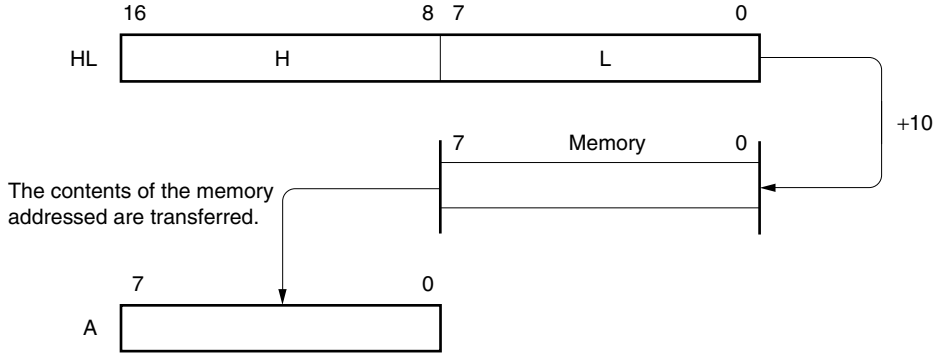
Identifier	Description
–	[HL + byte]

[Description example]

MOV A, [HL + 10H]; when setting byte to 10H

Operation code	1	0	1	0	1	1	1	0
	0	0	0	1	0	0	0	0

[Illustration]



3.4.8 Based indexed addressing

[Function]

The B or C register contents specified in an instruction word are added to the contents of the base register, that is, the HL register pair in the register bank specified by the register bank select flag (RBS0 and RBS1), and the sum is used to address the memory. Addition is performed by expanding the B or C register contents as a positive number to 16 bits. A carry from the 16th bit is ignored.

This addressing can be carried out for all of the memory spaces.

[Operand format]

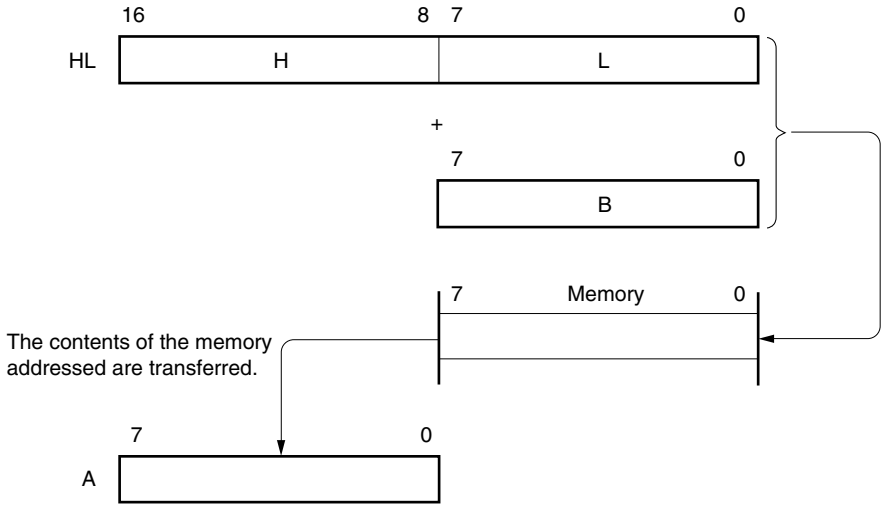
Identifier	Description
–	[HL + B], [HL + C]

[Description example]

MOV A, [HL +B]; when selecting B register

Operation code	1	0	1	0	1	0	1	1

[Illustration]



3.4.9 Stack addressing

[Function]

The stack area is indirectly addressed with the stack pointer (SP) contents.

This addressing method is automatically employed when the PUSH, POP, subroutine call and return instructions are executed or the register is saved/reset upon generation of an interrupt request.

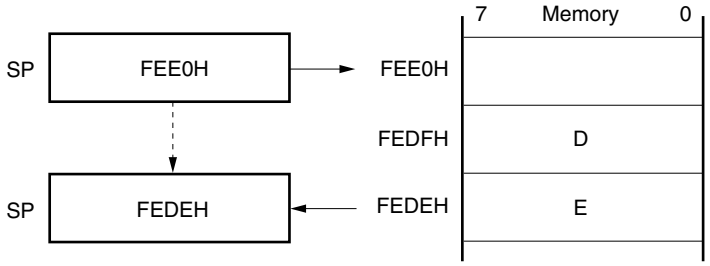
With stack addressing, only the internal high-speed RAM area can be accessed.

[Description example]

PUSH DE; when saving DE register

Operation code	1	0	1	1	0	1	0	1

[Illustration]



CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

There are two types of pin I/O buffer power supplies: AV_{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	
	78K0/KB2-A	78K0/KC2-A
AV_{DD}	P20 to P25, P80 to P83	P20 to P27, P80 to 83
V_{DD}	Pins other than P20 to P25, P80 to P83	Pins other than P20 to P27, P80 to 83

78K0/Kx2-A microcontrollers are provided with digital I/O ports, which enable variety of control operations. The functions of each port are shown in Tables 4-2 and 4-3.

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

Table 4-2. Port Functions (78K0/KB2-A)

Function Name	I/O	Function	After Reset	Alternate Function
P10	I/O	Port 1. 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	TxD6/TI51/TO51
P11				RxD6/TI50/TO50
P12				TOH0/INTP7/TI000
P13				TOH1/INTP6/TI010 /TO00
P20	I/O	Port 2. 6-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI0/AMP0-
P21				ANI1/AMP0OUT
P22				ANI2/AMP0+
P23				ANI3/AMP1−
P24				ANI4/AMP1OUT
P25				ANI5/AMP1+
P31	I/O	Port 3. 3-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP5/OCD1A (/$\overline{\text{SCK10}}$)
P32				INTP4/OCD1B/ (/SI10)
P35				SO10/INTP1
P60	I/O	Port 6. 2-bit I/O port. Output is N-ch open-drain output (6 V tolerance).Input/output can be specified in 1-bit units.	Input port	SCLA0/$\overline{\text{SCK10}}$
P61				SDAA0/SI10
P80	I/O	Port 8. 4-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI8/AMP2−
P81				ANI9/AMP2OUT
P82				ANI10/AMP2+
P83				ANI11
P120	I/O	Port 12. 3-bit I/O port. Input/output can be specified in 1-bit units. Only for P120, use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP0/EXLVI
P121				X1/OCD0A
P122				X2/EXCLK/OCD0B

Remark The functions of pins whose names are in parentheses can be used by setting bit 2 (ISC2) of the input switch control register (ISC) to 1.

Table 4-3. Port Functions (78K0/KC2-A)

Function Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0. 3-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TI000
P01				TI010/TO00
P02				INTP8
P10	I/O	Port 1. 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	TxD6
P11				RxD6
P12				TOH0/INTP7
P13				TOH1/INTP6
P20	I/O	Port 2. 8-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI0/AMP0−
P21				ANI1/AMP0OUT
P22				ANI2/AMP0+
P23				ANI3/AMP1−
P24				ANI4/AMP1OUT
P25				ANI5/AMP1+
P26				ANI6
P27				ANI15/AV_{REFM}
P31	I/O	Port 3. 5-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP5/OCD1A (/$\overline{SCK10}$)
P32				INTP4/OCD1B (/SI10)
P33				TI51/TO51/INTP3
P34				TI50/TO50/INTP2
P35				SO10/INTP1
P40	I/O	Port 4. 3-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	RTCCL/RTCDIV
P41				RTC1HZ
P42				PCL/$\overline{SSI10}$/INTP9
P60	I/O	Port 6. 2-bit I/O port. Output is N-ch open-drain output (6 V tolerance). Input/output can be specified in 1-bit units.	Input port	SCLA0/$\overline{SCK10}$
P61				SDAA0/SI10
P70 to P75	I/O	Port 7. 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	KR0 to KR5
P80	I/O	Port 8. 4-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI8/AMP2−
P81				ANI9/AMP2OUT
P82				ANI10/AMP2+
P83				ANI11
P120	I/O	Port 12. 5-bit I/O port. Input/output can be specified in 1-bit units. Only for P120, use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP0/EXLVI
P121				X1/OCD0A
P122				X2/EXCLK/OCD0B
P123				XT1
P124				XT2

Remark The functions of pins whose names are in parentheses can be used by setting bit 2 (ISC2) of the input switch control register (ISC) to 1.

4.2 Port Configuration

Ports include the following hardware.

Table 4-4. Port Configuration

Item	Configuration
Control registers	• 78K0/KB2-A Port mode register (PMxx): PM1 to PM3, PM6, PM8, PM12 Port register (Pxx): P1 to P3, P6, P8, P12 Pull-up resistor option register (PUxx): PU1, PU3, PU12 A/D port configuration register (ADPC) • 78K0/KC2-A Port mode register (PMxx): PM0 to PM4, PM6 to PM8, PM12 Port register (Pxx): P0 to P4, P6 to P8, P12 Pull-up resistor option register (PUxx): PU0, PU1, PU3, PU4, PU7, PU12 A/D port configuration register (ADPC)
Port	• 78K0/KB2-A: Total: 22 (CMOS I/O: 20, N-ch open drain I/O: 2) • 78K0/KC2-A: Total: 40 (CMOS I/O: 38, N-ch open drain I/O: 2)
Pull-up resistor	• 78K0/KB2-A: Total: 8 • 78K0/KC2-A: Total: 22

4.2.1 Port 0

78K0/KB2-A	78K0/KC2-A
–	P00/TI000
	P01/TI010/TO00
	P02/INTP8

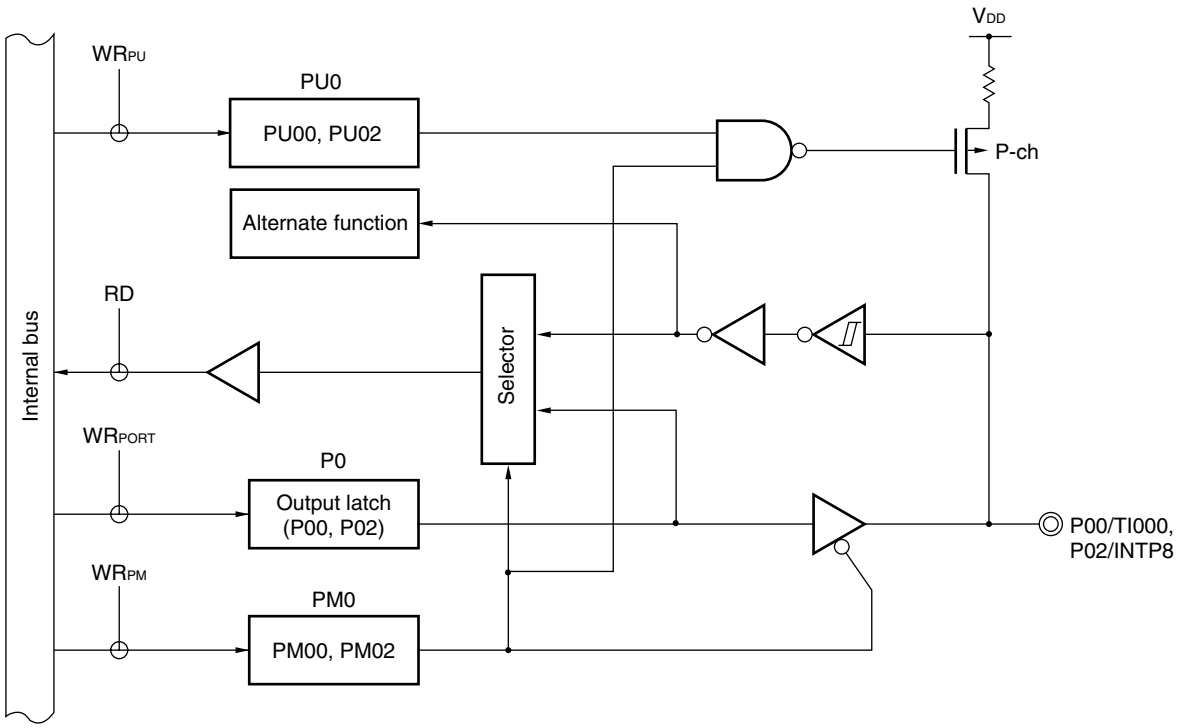
Port 0 is an I/O port with an output latch. Port 0 can be set to the input mode or output mode in 1-bit units using port mode register 0 (PM0). When the P00 to P02 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 0 (PU0).

This port can also be used for external interrupt request input t and timer I/O.

Reset signal generation sets port 0 to input mode.

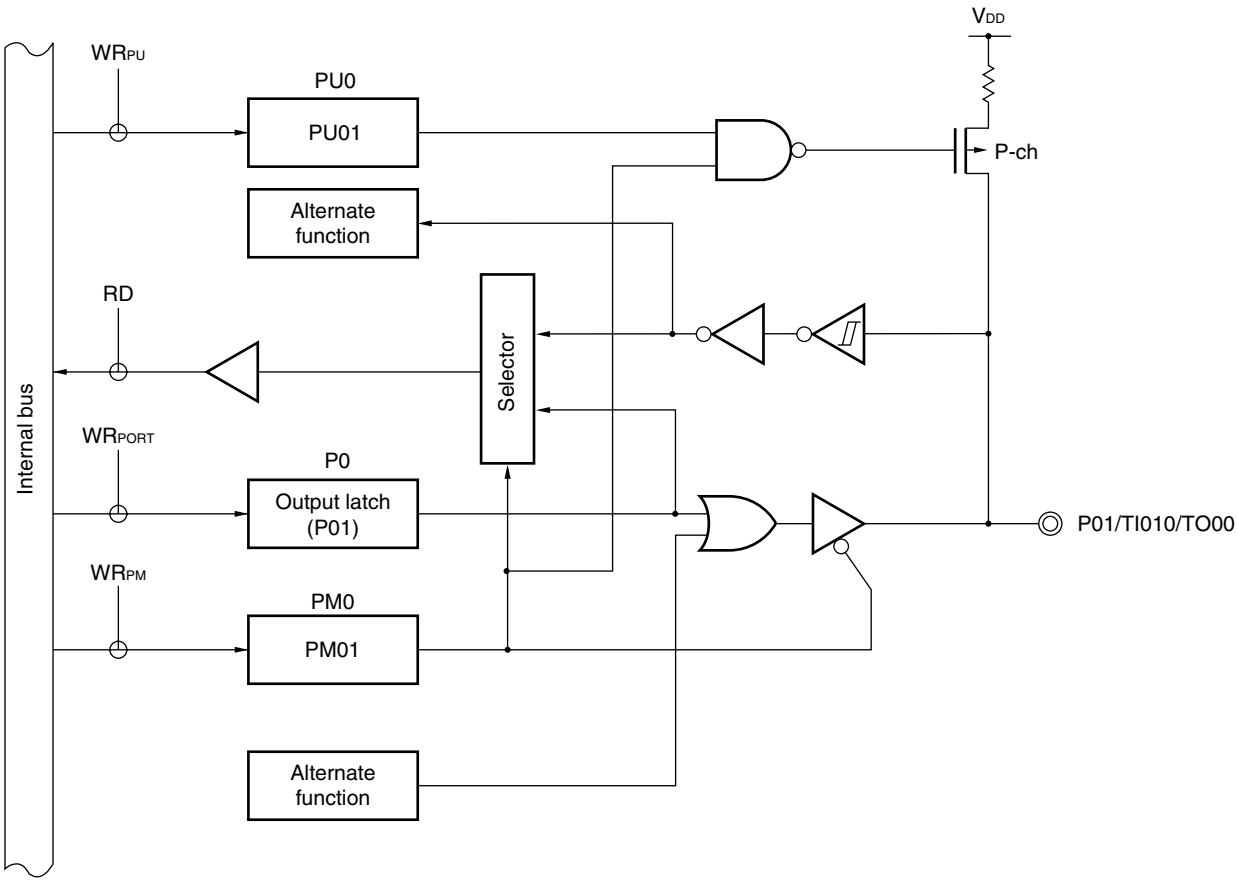
Figures 4-1 and 4-2 show block diagrams of port 0.

Figure 4-1. Block Diagram of P00, P02



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

Figure 4-2. 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

4.2.2 Port 1

78K0/KB2-A	78K0/KC2-A
P10/TxD6/TI51/TO51	P10/TxD6
P11/RxD6/TI50/TO50	P11/RxD6
P12/TOH0/INTP7/TI000	P12/TOH0/INTP7
P13/TOH1/INTP6/TI010/TO00	P13/TOH1/INTP6

Port 1 is an I/O port with an output latch. Port 1 can be set to the input mode or output mode in 1-bit units using port mode register 1 (PM1). When the P10 to P13 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, and timer I/O.

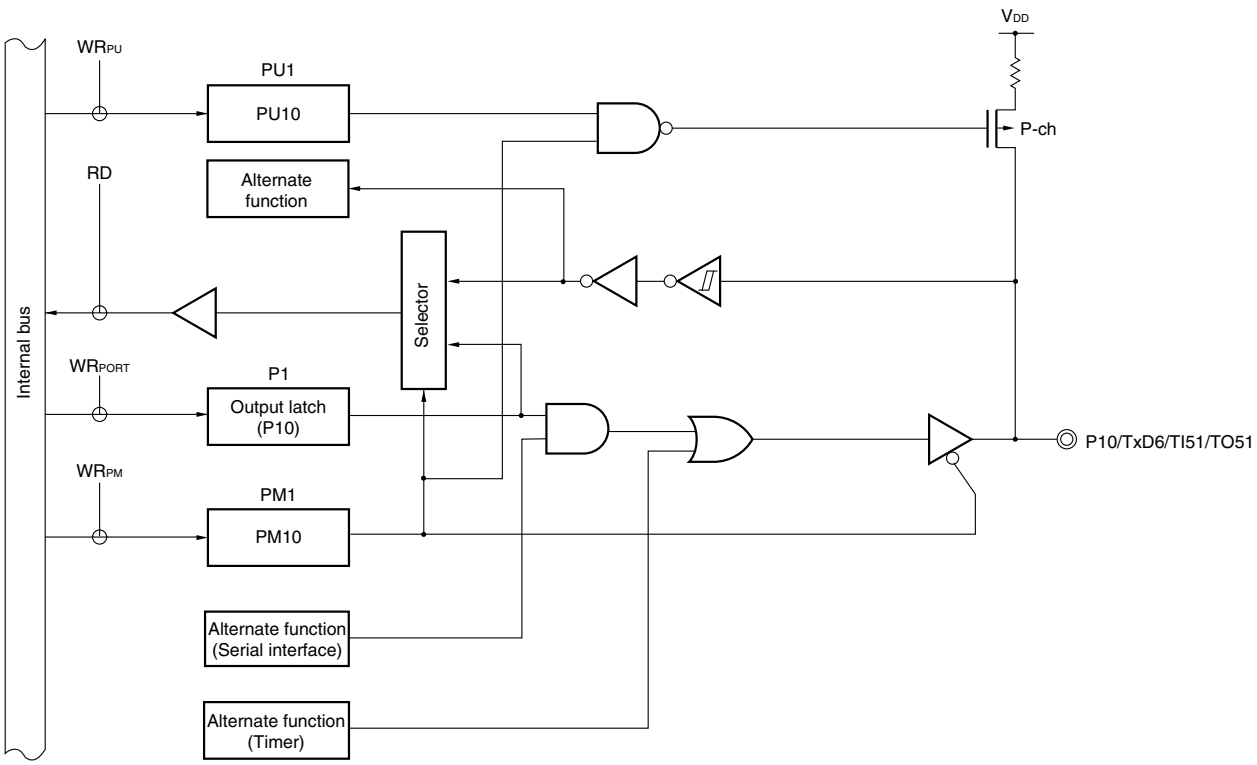
Reset signal generation sets port 1 to input mode.

Figures 4-3 to 4-5 show block diagrams of port 1.

Cautions To use P10/TxD6 as general-purpose port, clear bit 0 (TXDLV6) of asynchronous serial interface control register 6 (ASICL6) to 0 (normal output of TxD6).

Figure 4-3. Block Diagram of P10 (1/2)

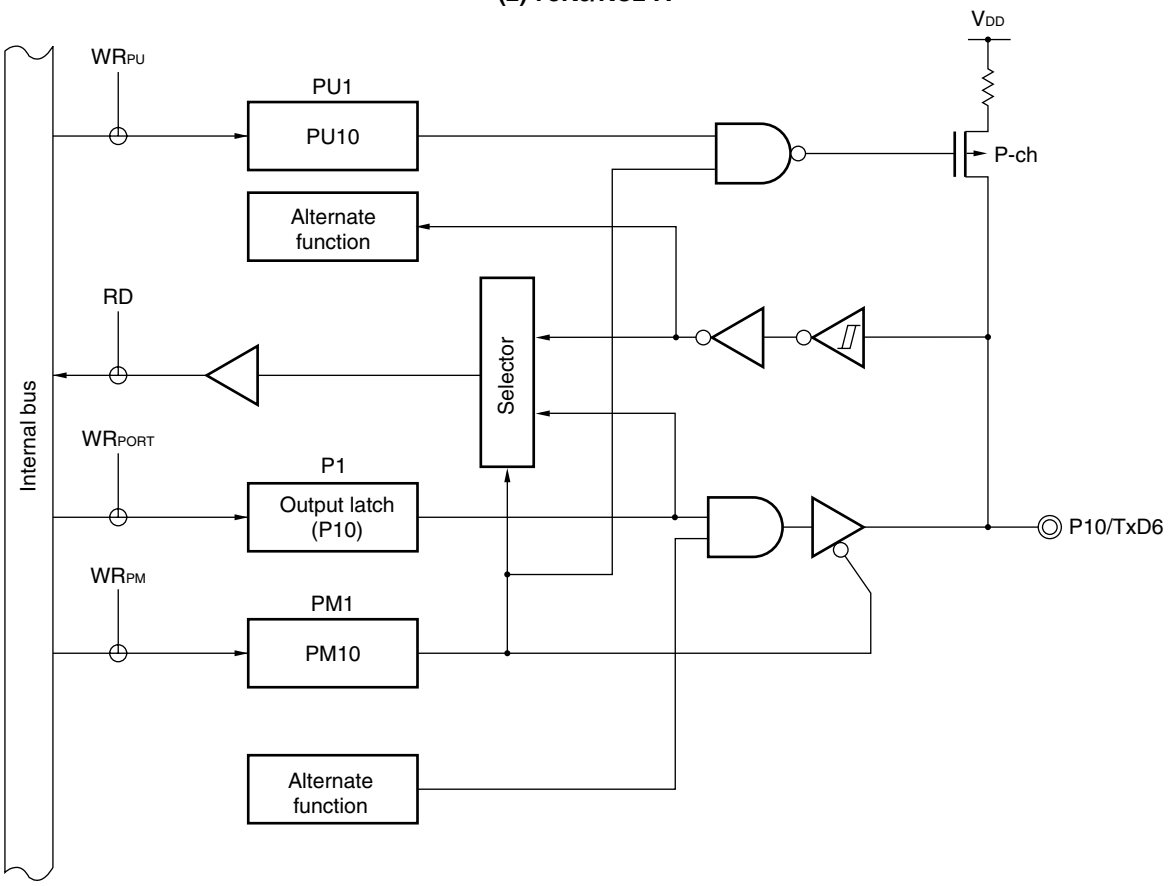
(1) 78K0/KB2-A



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

Figure 4-3. Block Diagram of P10 (2/2)

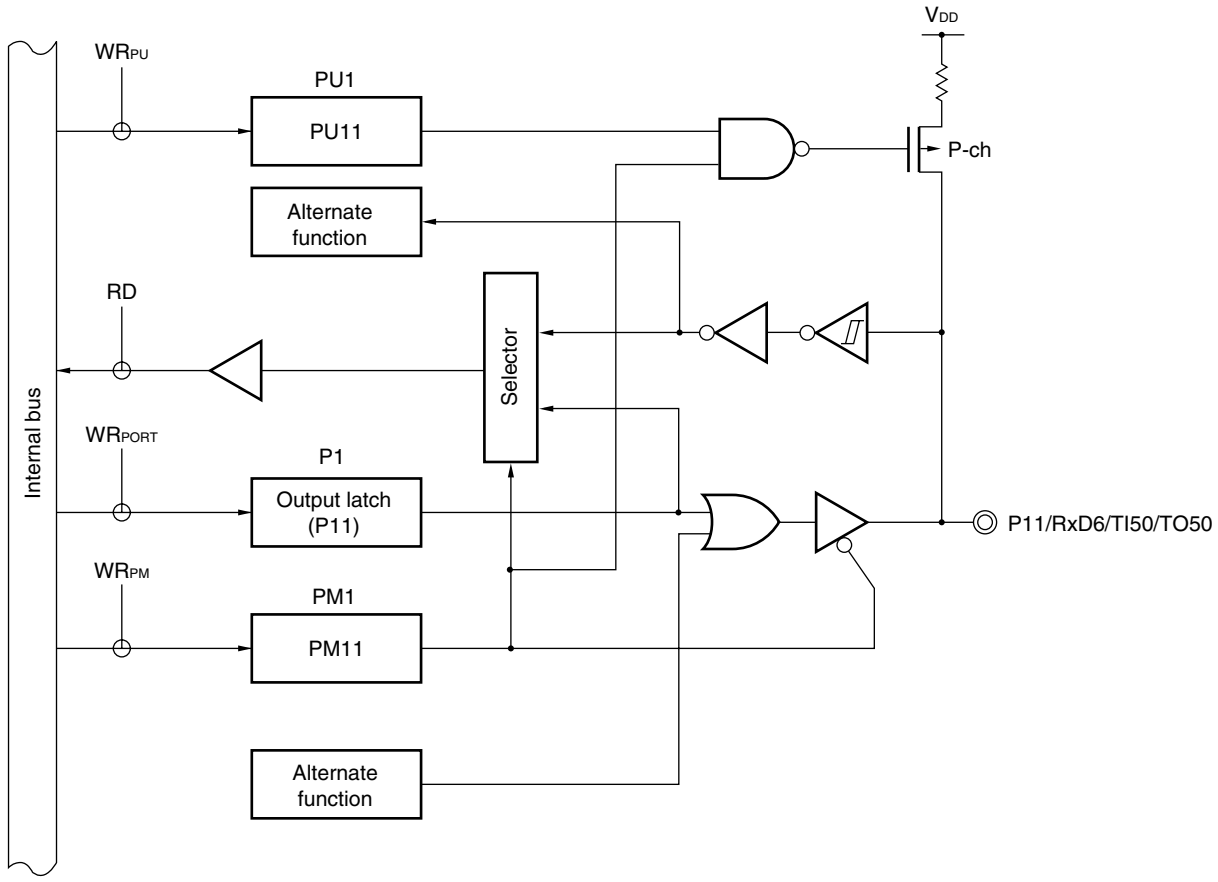
(2) 78K0/KC2-A



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

Figure 4-4. Block Diagram of P11 (1/2)

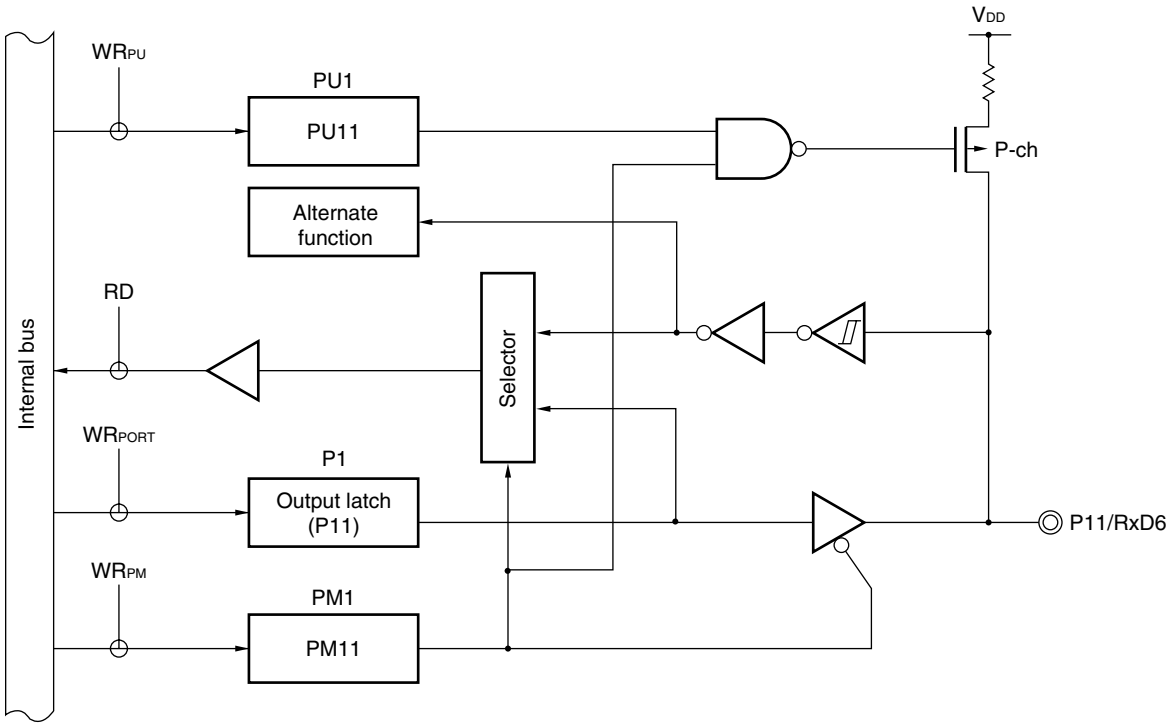
(1) 78K0/KB2-A



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

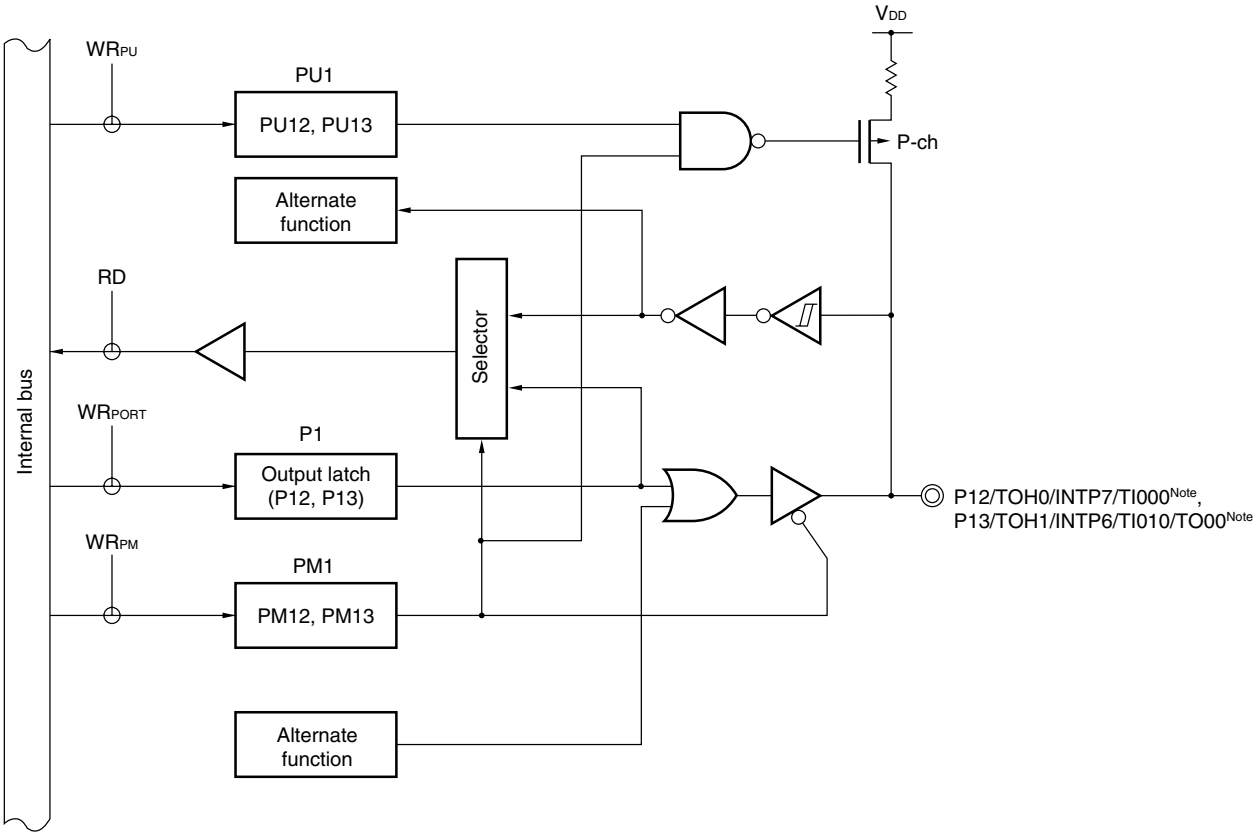
Figure 4-4. Block Diagram of P11 (2/2)

(2) 78K0/KC2-A



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

Figure 4-5. 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

Note 78K0/KB2-A: P12/TOH0/INTP7/TI000, P13/TOH1/INTP6/TI010/TO00
78K0/KC2-A: P12/TOH0/INTP7, P13/TOH1/INTP6

4.2.3 Port 2

78K0/KB2-A	78K0/KC2-A
P20/ANI0/AMP0−	
P21/ANI1/AMP0OUT	
P22/ANI2/AMP0+	
P23/ANI3/AMP1−	
P24/ANI4/AMP1OUT	
P25/ANI5/AMP1+	
−	P26/ANI6
	P27/ANI15/AV_{REFM}

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

This port can also be used for A/D converter analog input, analog reference voltage input, and operational amplifier I/O.

When using P20/ANI0/AMP0− to P27/ANI15/AV_{REFM}, set the registers according to the pin function to be used (refer to Tables 4-5 to 4-8).

To use P20/ANI0/AMP0− to P27/ANI15/AV_{REFM} and P80/ANI8/AMP2− to P83/ANI11 as a digital input or a digital output, it is recommended to select a pin to use starting with the furthest P20/ANI0/AMP0 pin from AV_{DD}.

All P20/ANI0/AMP0− to P27/ANI15/AV_{REFM} are set in the digital input mode when the reset signal is generated.

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

P20/ANI0/AMP0− to P27/ANI15/AV_{REFM} pins are as shown below depending on the settings of ADPC, ADS, PM2, OAENn, and ADREF.

Table 4-5. Setting Functions of P20/ANI0/AMP0–, P22/ANI2/AMP0+, P23/ANI3/AMP1–, P25/ANI5/AMP1+ Pins

ADPC	PM2	OAENn	ADS	P20/ANI0/AMP0−, P22/ANI2/AMP0+, P23/ANI3/AMP1−, P25/ANI5/AMP1+ Pin
Digital I/O selection	Input mode	0	–	Digital input
		1	–	Setting prohibited
	Output mode	0	–	Digital output
		1	–	Setting prohibited
Analog input selection	Input mode	0	Selects ANI.	Analog input (to be converted)
			Does not select ANI.	Analog input (not to be converted)
		1	Selects ANI.	Setting prohibited
			Does not select ANI.	Operational amplifier input
	Output mode	–	–	Setting prohibited

Table 4-6. Setting Functions of P21/ANI1/AMP0OUT, P24/ANI4/AMP1OUT Pins

ADPC	PM2	OAENn	ADS	P21/ANI1/AMP0OUT, P24/ANI4/AMP1OUT Pin
Digital I/O selection	Input mode	0	–	Digital input
		1	–	Setting prohibited
	Output mode	0	–	Digital output
		1	–	Setting prohibited
Analog input selection	Input mode	0	Selects ANI.	Analog input (to be converted)
			Does not select ANI.	Analog input (not to be converted)
		1	Selects ANI.	Operational amplifier output (to be converted)
			Does not select ANI.	Operational amplifier output (not to be converted)
	Output mode	–	–	Setting prohibited

Table 4-7. Setting Functions of P26/ANI6 Pin

ADPC	PM2	ADS	P26/ANI6 Pin
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	–	Setting prohibited

Caution When using an operational amplifier, the AMPn+, AMPn-, and AMPnOUT pins are used as I/O pins for the operational amplifier, and therefore cannot be used as analog input pins. However, the operational amplifier's output signal can be used as an analog input.

Remark n = 0, 1

Table 4-8. Setting Functions of P27/ANI15/AV$_{REFM}$ Pin

ADPC	PM2	ADREF	ADS	P27/ANI15/AV$_{REFM}$ Pin
Digital I/O selection	Input mode	0	–	Digital input
		1	–	Setting prohibited
	Output mode	0	–	Digital output
		1	–	Setting prohibited
Analog input selection	Input mode	0	Selects ANI.	Analog input (to be converted)
			Does not select ANI.	Analog input (not to be converted)
		1	–	Reference voltage input on the negative side of A/D converter
	Output mode	–	–	Setting prohibited

Figures 4-6 to 4-10 show block diagrams of port 2.

Figure 4-6. Block Diagram of P20 and P23

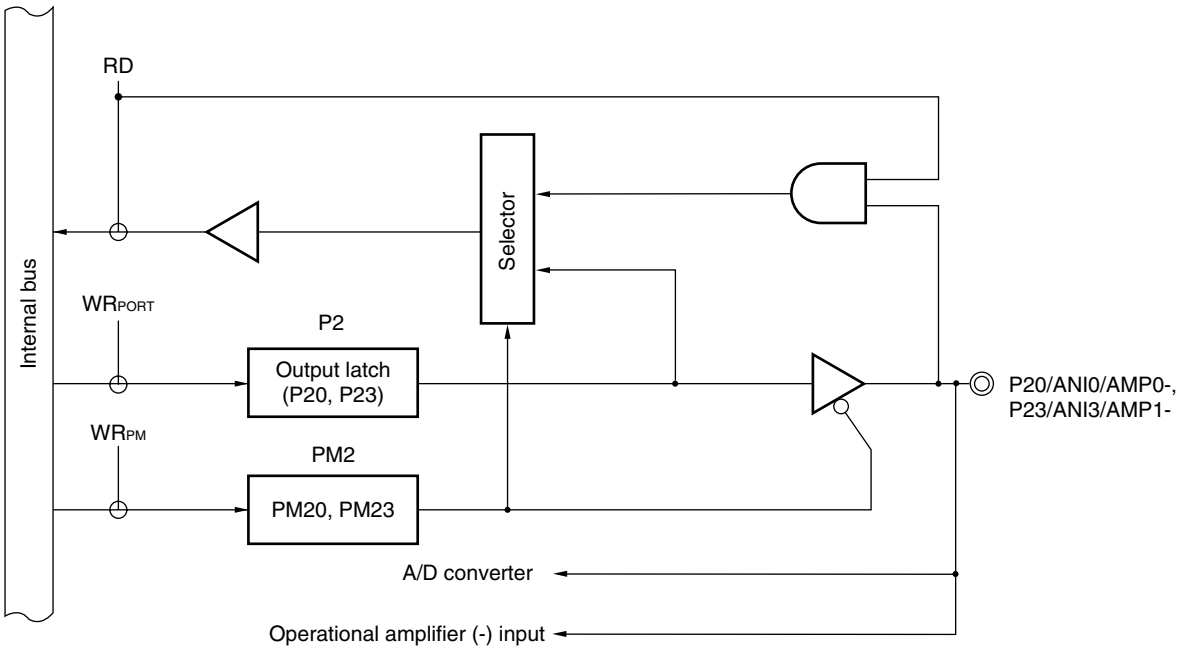
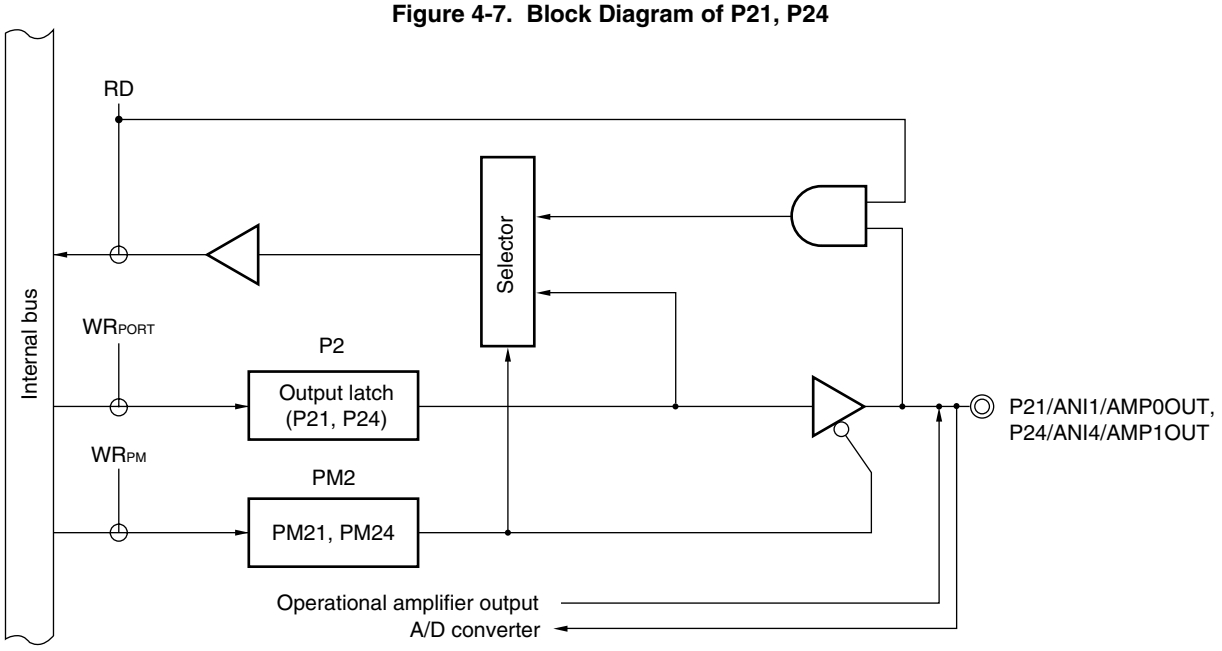


Figure 4-7. Block Diagram of P21, P24



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

Figure 4-8. Block Diagram of P21, P24

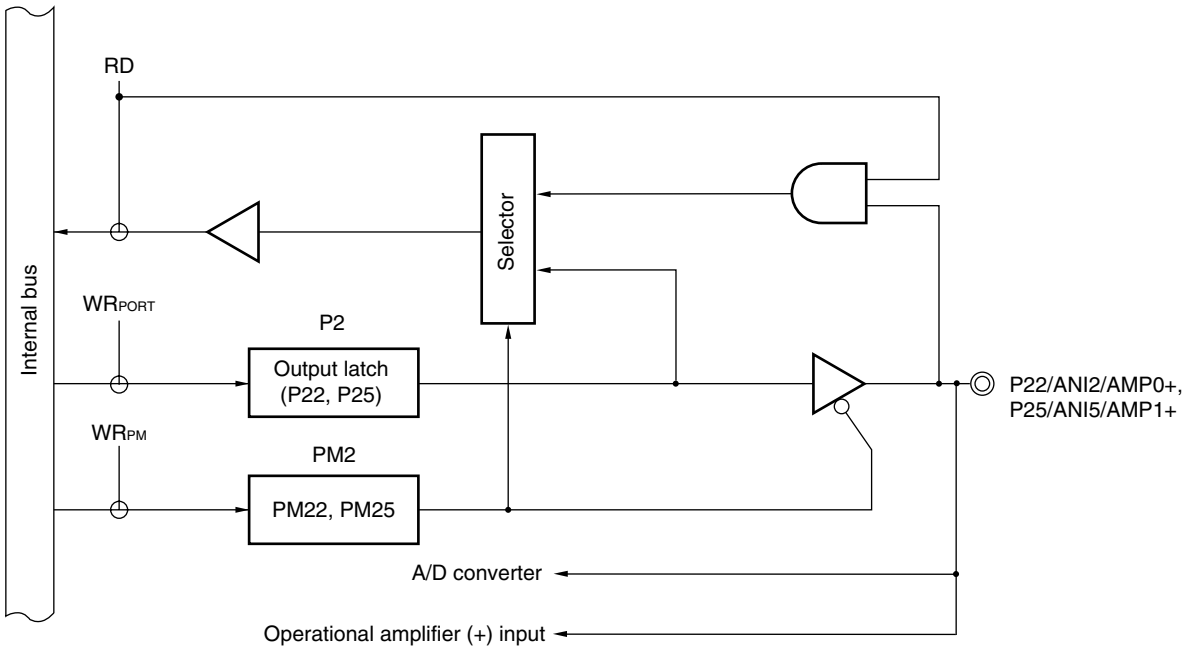
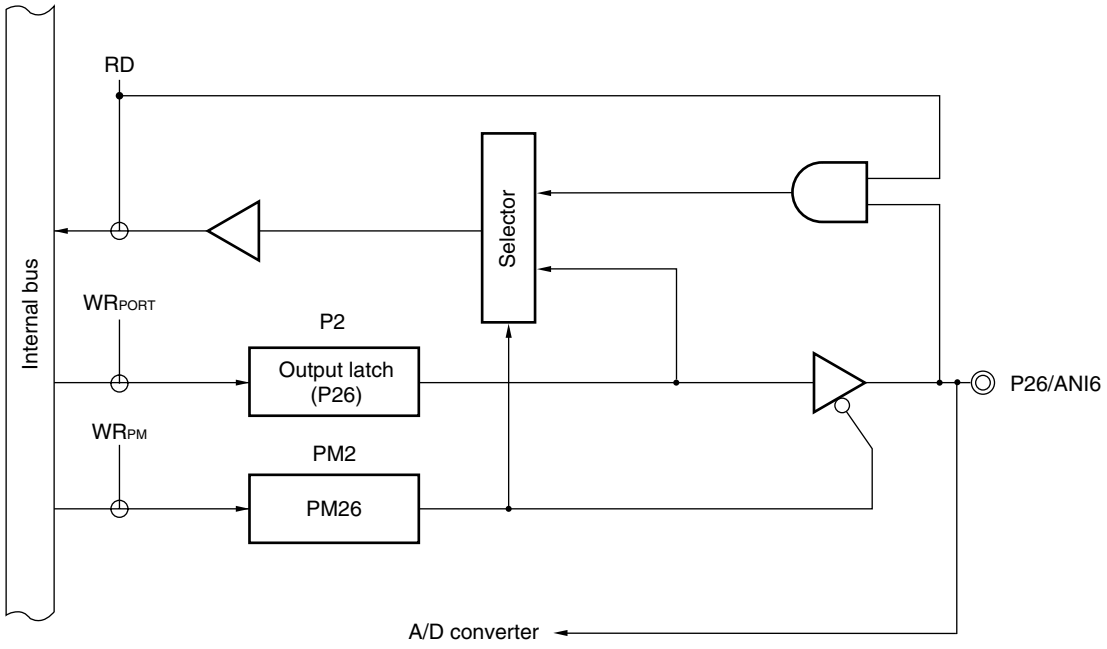
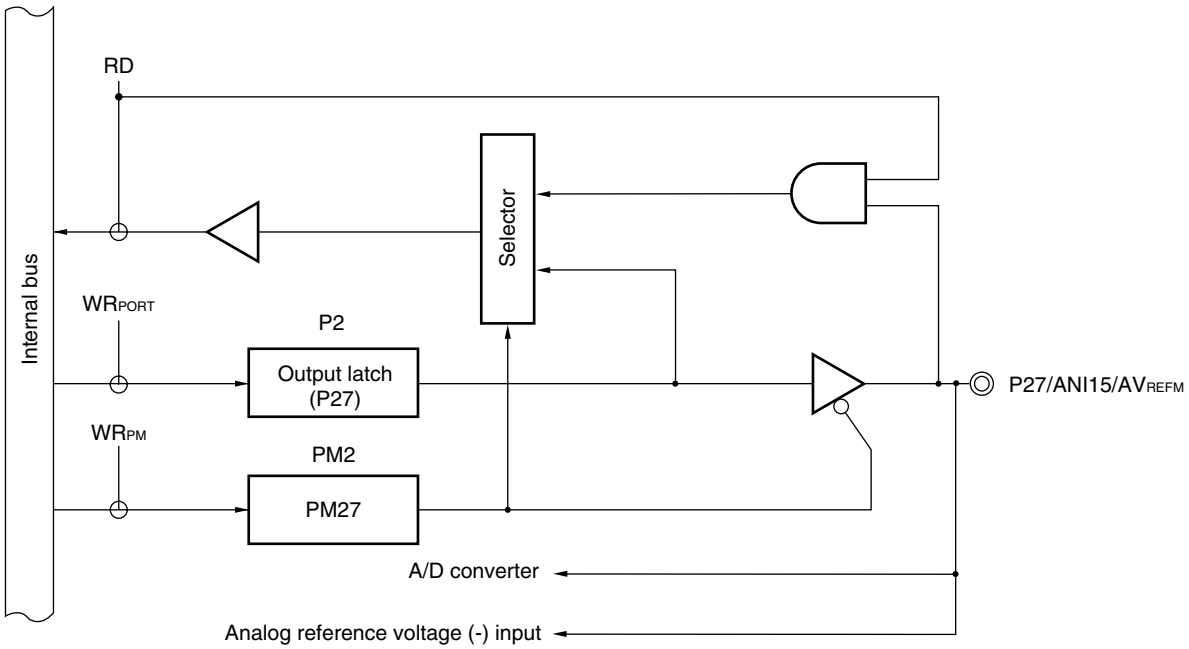


Figure 4-9. Block Diagram of P26



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

Figure 4-10. Block Diagram of P27



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

4.2.4 Port 3

78K0/KB2-A	78K0/KC2-A
P31/INTP5/OCD1A (/$\overline{\text{SCK10}}$)	
P32/INTP4/OCD1B (/SI10)	
–	P33/TI51/TO51/INTP
–	P34/TI50/TO50/INTP2
P35/SO10/INTP1	

Remark The functions of pins whose names are in parentheses can be used by setting bit 2 (ISC2) of the input switch control register (ISC) to 1.

Port 3 is an I/O port with an output latch. Port 3 can be set to the input mode or output mode in 1-bit units using port mode register 3 (PM3). When the P31 to P35 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, serial interface data I/O, and clock I/O.

Reset signal generation sets port 3 to input mode.

Figures 4-11 and 4-13 show block diagrams of port 3.

Cautions 1. Process the P31/INTP5/OCD1A (/$\overline{\text{SCK10}}$) pin as follows, when it is not used when it is connected to a flash memory programmer or an on-chip debug emulator.

		P31/INTP5/OCD1A (/$\overline{\text{SCK10}}$)
Flash memory programmer connection		Connect to V_{SS} via a resistor.
On-chip debug emulator connection (when it is not used as an on-chip debug mode setting pin)	During reset	
	During reset released	Input: Connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.

2. To use P31/INTP5/OCD1A (/$\overline{\text{SCK10}}$) and P35/SO10/INTP1 as general-purpose ports, set serial operation mode register 10 (CSIM10) and serial clock selection register 10 (CSIC10) to the default status (00H).

Remark P31 and P32 can be used as on-chip debug mode setting pins (OCD1A, OCD1B) when the on-chip debug function is used. For how to connect an on-chip debug emulator (QB-MINI2), see **CHAPTER 26 ON-CHIP DEBUG FUNCTION.**

Figure 4-11. Block Diagram of P31

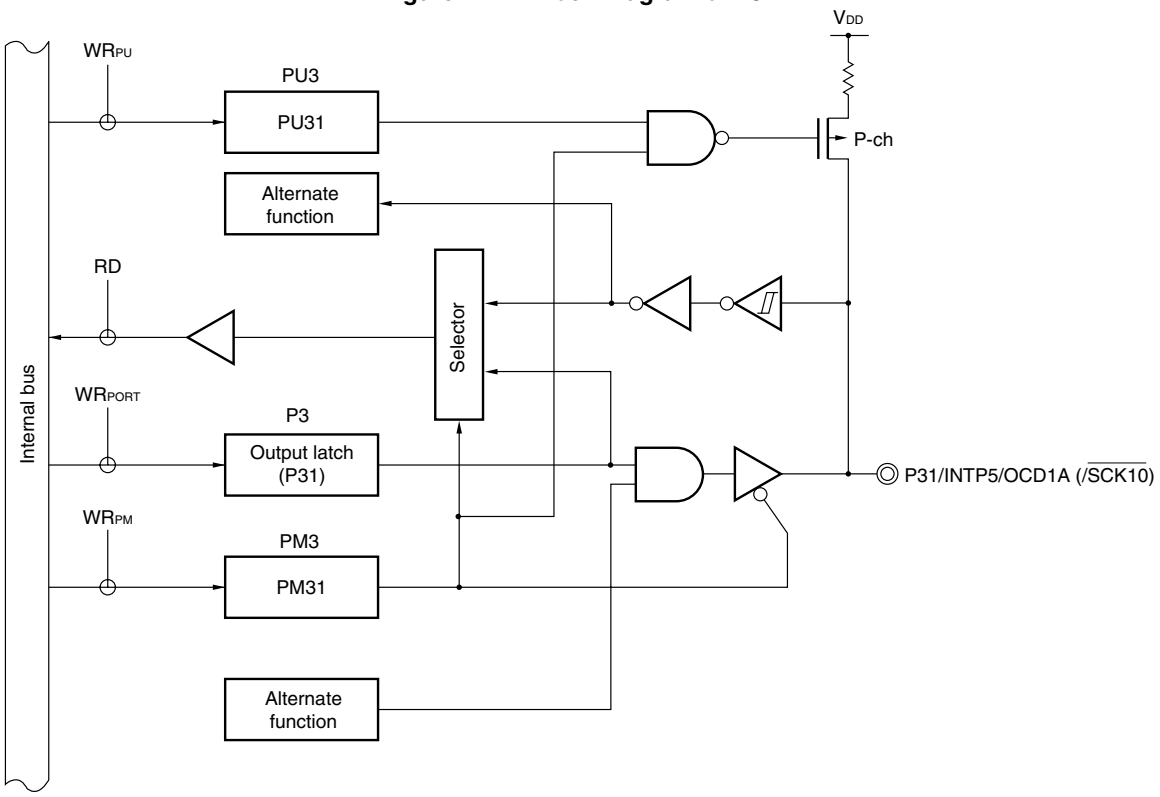
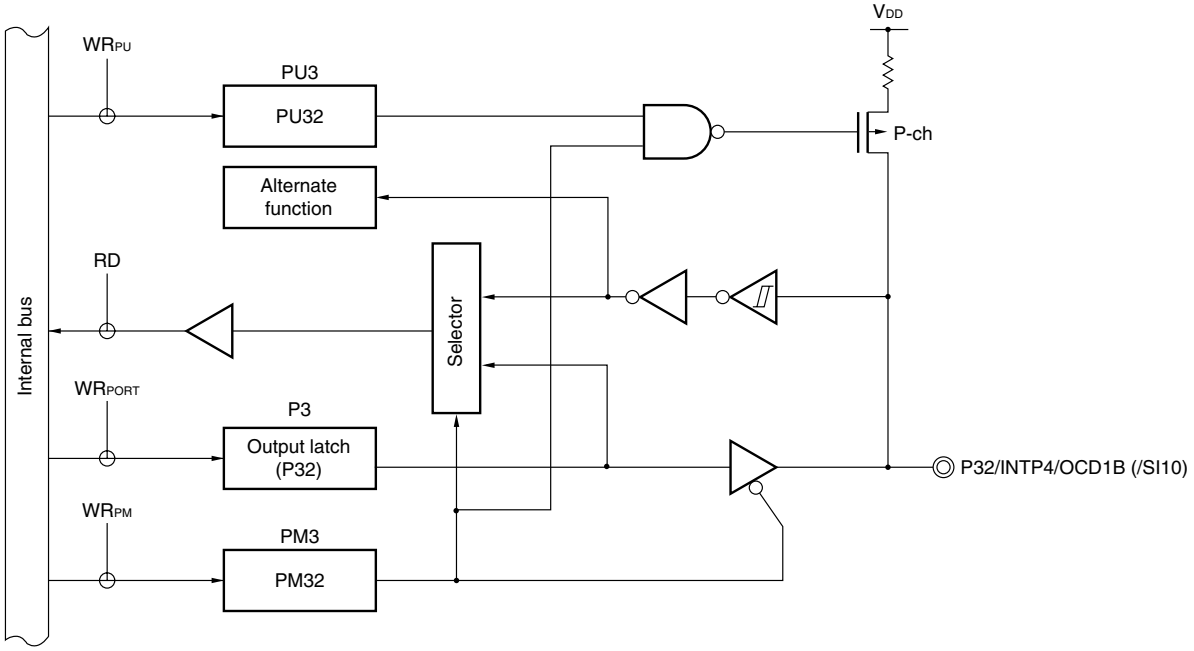


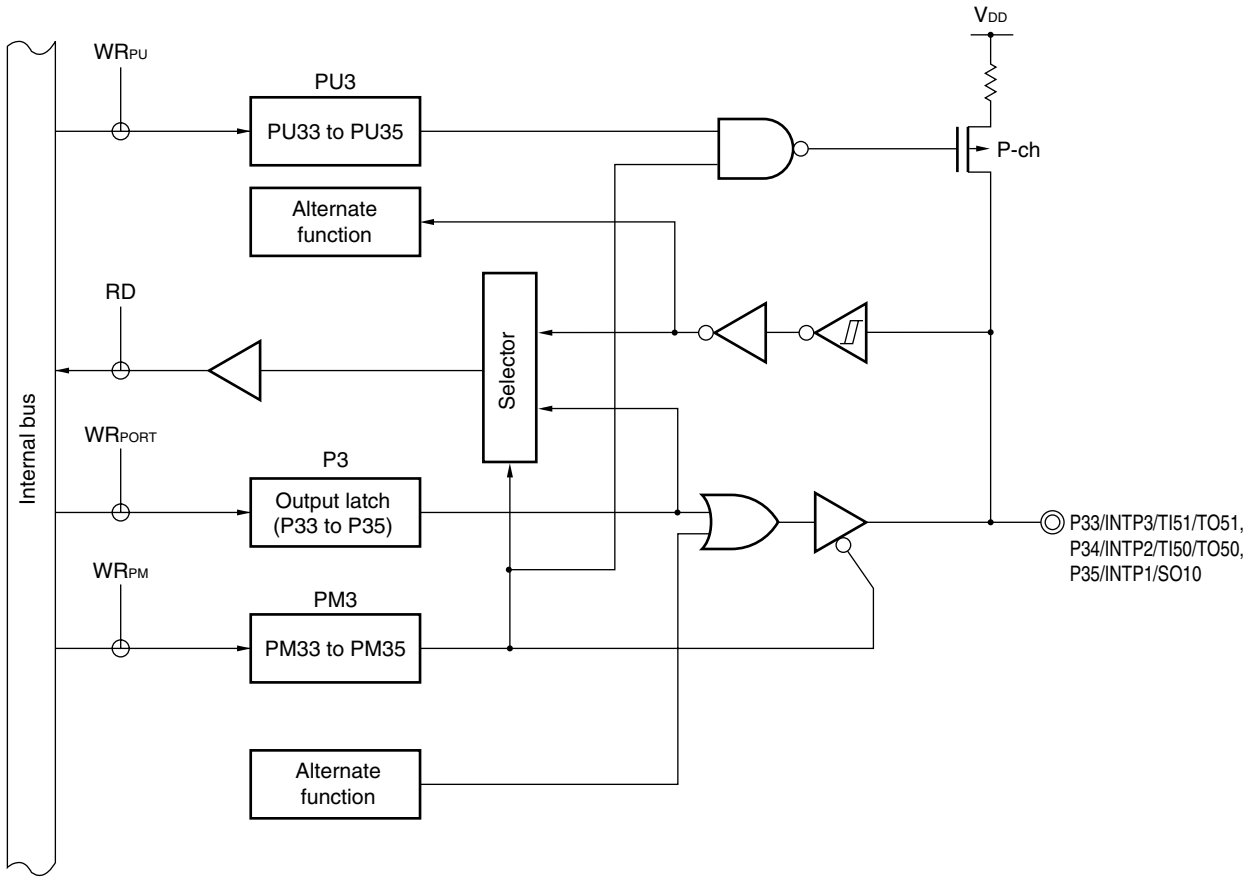
Figure 4-12. Block Diagram of P32



P3: Port register 3
PU3: Pull-up resistor option register 3
PM3: Port mode register 3
RD: Read signal
WR××: Write signal

Remark The functions of pins whose names are in parentheses can be used by setting bit 2 (ISC2) of the input switch control register (ISC) to 1.

Figure 4-13. Block Diagram of P33 to P35



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

78K0/KB2-A	78K0/KC2-A
–	P40/RTCCL/RTCDIV
	P41/RTC1HZ
	P42/PCL/$\overline{\text{SSI10}}$/INTP9

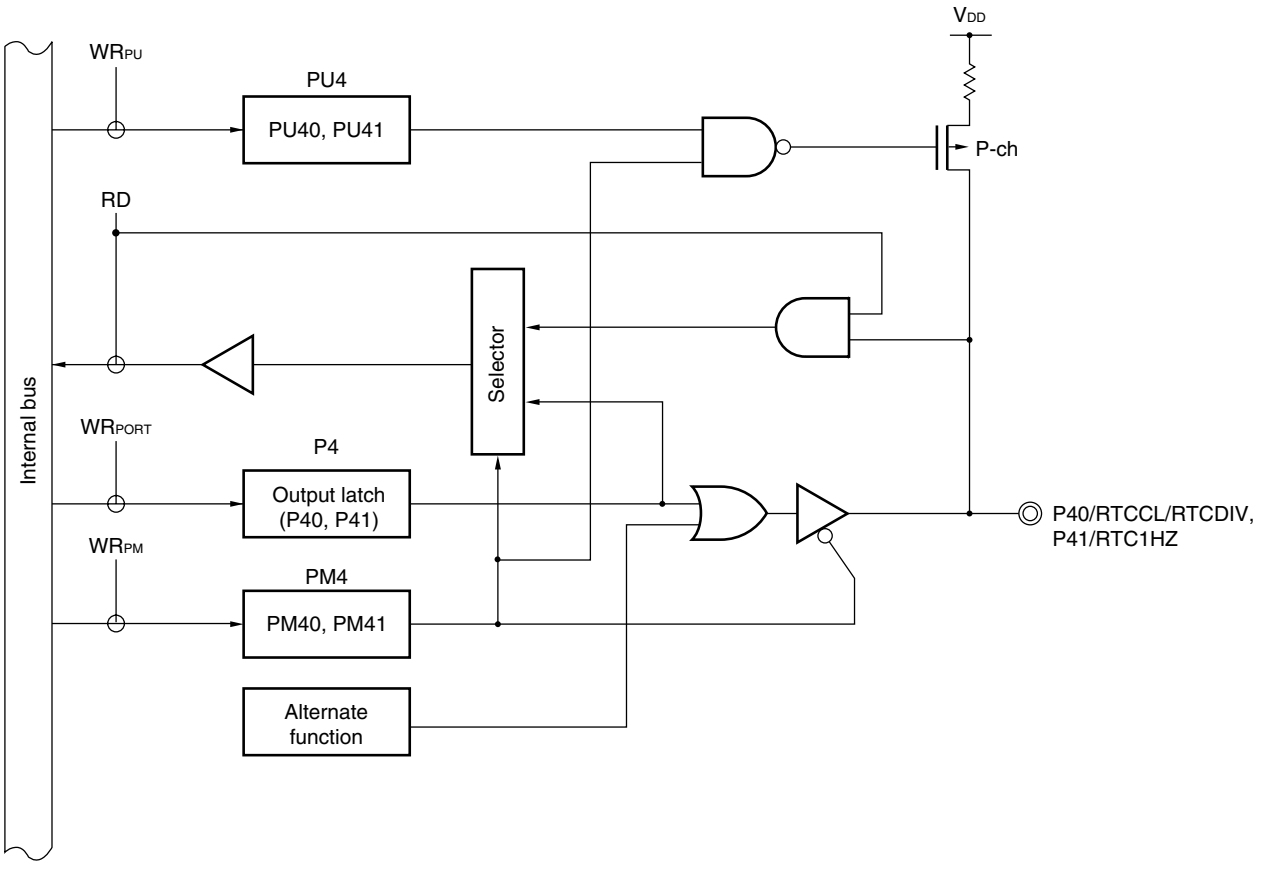
Port 4 is an I/O port with an output latch. Port 4 can be set to the input mode or output mode in 1-bit units using port mode register 4 (PM4). When the P40 to P42 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 4 (PU4).

This port can also be used for external interrupt request input, clock output, real-time counter clock output, and chip select input for serial interface.

Reset signal generation sets port 4 to input mode.

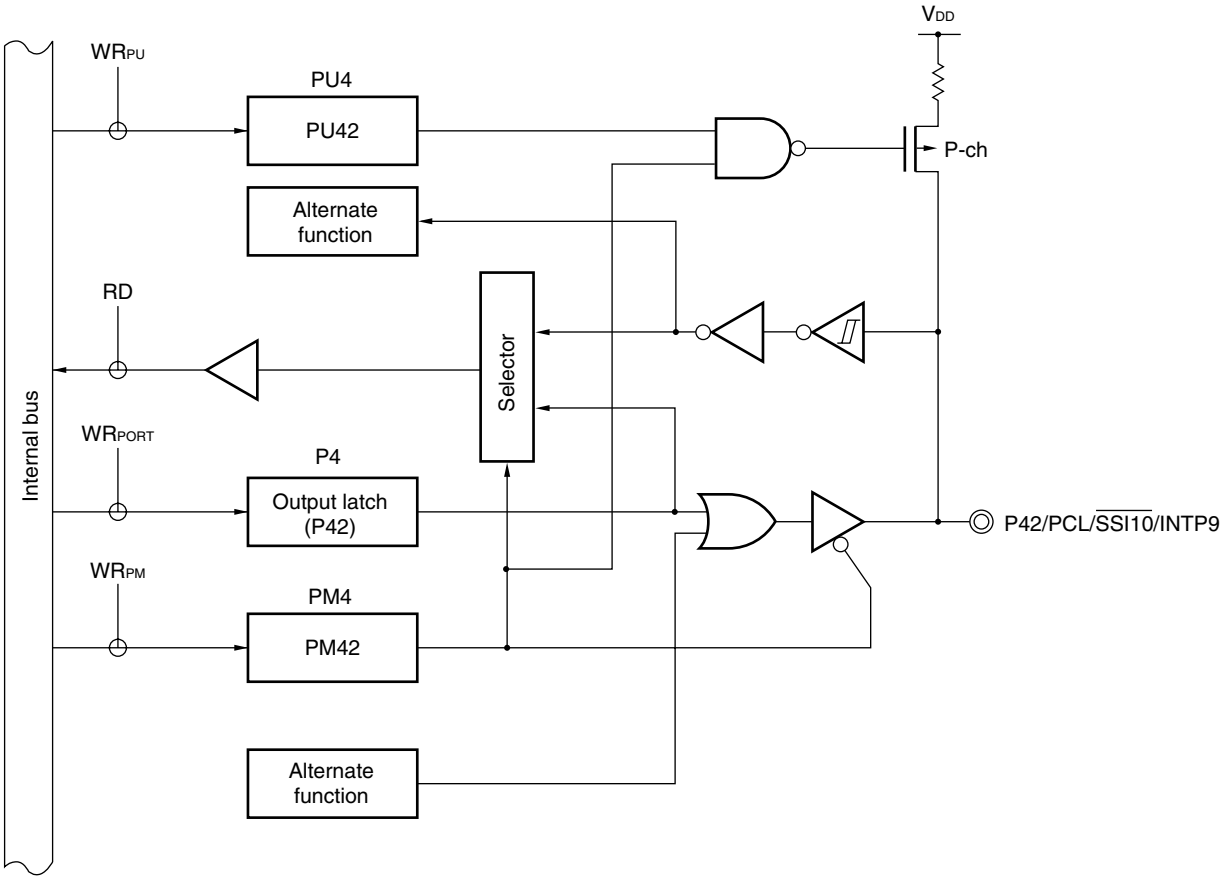
Figures 4-14 and 4-15 show block diagrams of port 4.

Figure 4-14. Block Diagram of P40 and P41



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

Figure 4-15. 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

4.2.6 Port 6

78K0/KB2-A	78K0/KC2-A
P60/SCLA0/$\overline{\text{SCK10}}$	
P61/SDAA0/SI10	

Port 6 is an I/O port with an output latch. Port 6 can be set to the input mode or output mode in 1-bit units using port mode register 6 (PM6).

The output of the P60 and P61 pins is N-ch open-drain output (6 V tolerance).

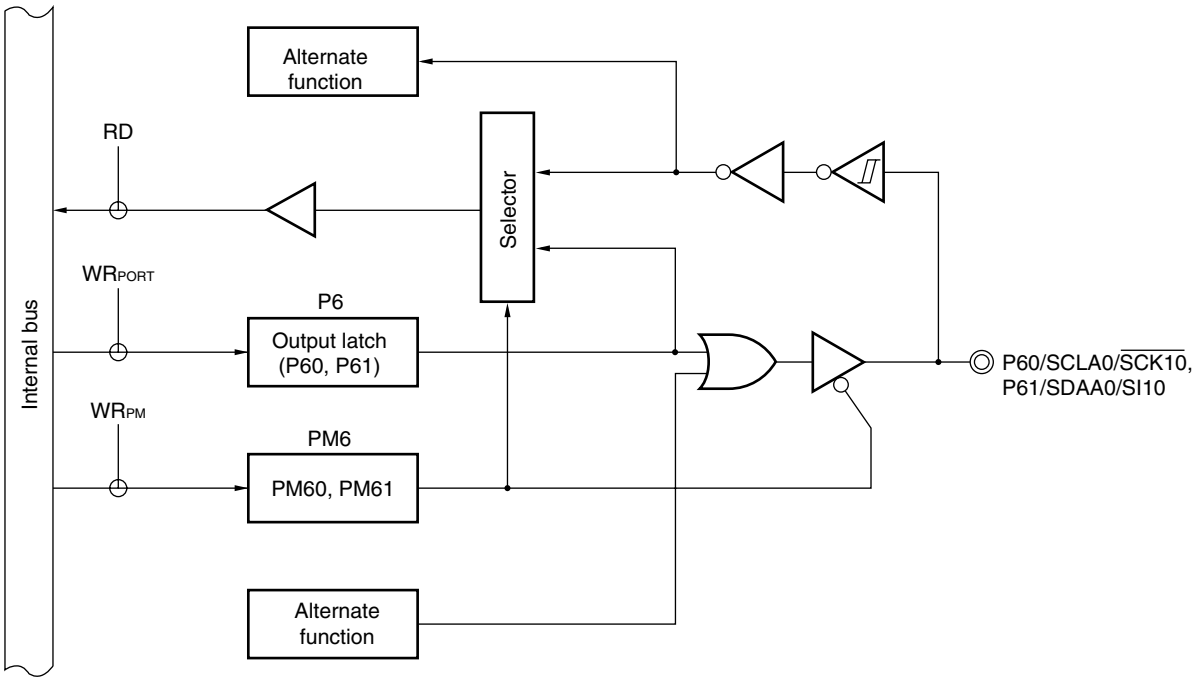
This port can also be used for serial interface data I/O and clock I/O.

Reset signal generation sets port 6 to input mode.

Figures 4-16 shows block diagram of port 6.

Caution To use P60/$\overline{\text{SCK10}}$ as general-purpose ports, set serial operation mode register 10 (CSIM10) and serial clock selection register 10 (CSIC10) to the default status (00H).

Figure 4-16. Block Diagram of P60 and P61



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

Caution A through current flows through P60 and P61 if an intermediate potential is input to these pins, because the input buffer is also turned on when P60 and P61 are in output mode. Consequently, do not input an intermediate potential when P60 and P61 are in output mode.

4.2.7 Port 7

78K0/KB2-A	78K0/KC2-A
–	P70/KR0
	P71/KR1
	P72/KR2
	P73/KR3
	P74/KR4
	P75/KR5

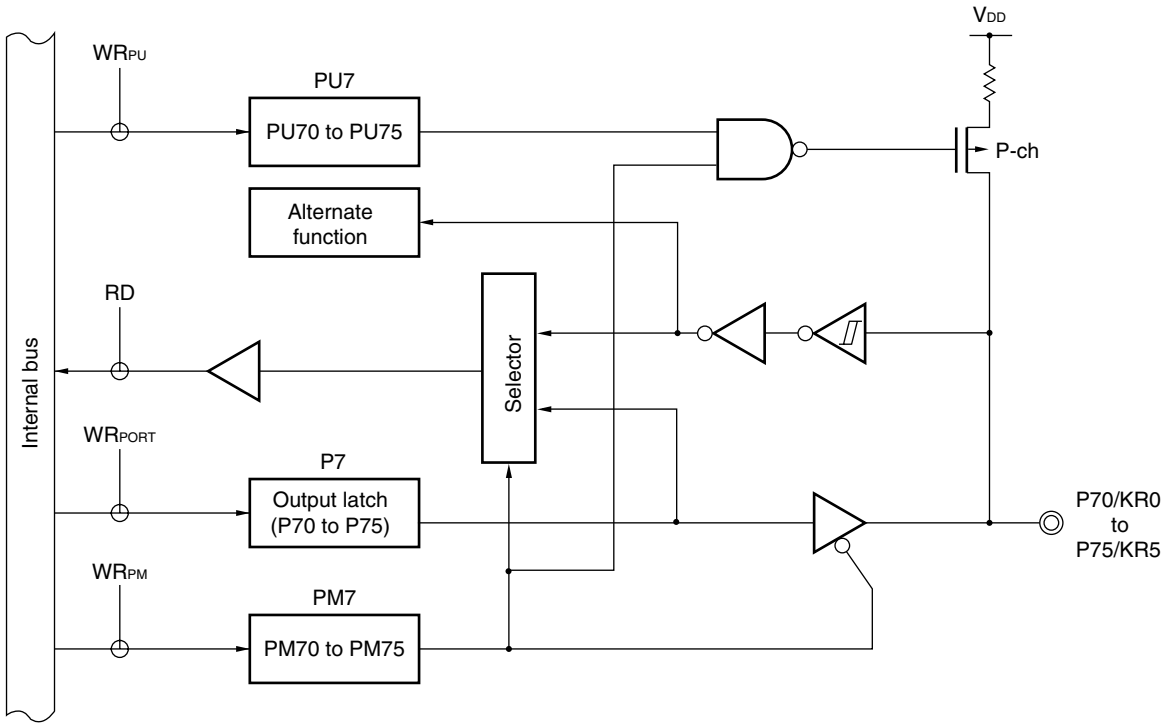
Port 7 is an I/O port with an output latch. Port 7 can be set to the input mode or output mode in 1-bit units using port mode register 7 (PM7). When the P70 to P75 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 7 (PU7).

This port can also be used for key return input.

Reset signal generation sets port 7 to input mode.

Figure 4-17 shows a block diagram of port 7.

Figure 4-17. Block Diagram of P70 to P75



P7: Port register 7
PU7: Pull-up resistor option register 7
PM7: Port mode register 7
RD: Read signal
WR××: Write signal

4.2.8 Port 8

78K0/KB2-A	78K0/KC2-A
P80/ANI8/AMP2−	
P81/ANI9/AMP2OUT	
P82/ANI10/AMP2+	
P83/ANI11	

Port 8 is an I/O port with an output latch. Port 8 can be set to the input mode or output mode in 1-bit units using port mode register 8 (PM8).

This port can also be used for A/D converter analog input and Operational amplifier I/O.

When using P80/ANI8/AMP2− to P83/ANI11, set the registers according to the pin function to be used (refer to Tables 4-9 to 4-11).

To use P20/ANI0/AMP0− to P27/ANI15/AV_{REFM} and P80/ANI8/AMP2− to P83/ANI11 as a digital input or a digital output, it is recommended to select a pin to use starting with the furthest P20/ANI0/AMP0 pin from AV_{DD}.

Reset signal generation sets P80/ANI8/AMP2− to P83/ANI11 to digital input mode.

Caution Make the AV_{DD} pin the same potential as the V_{DD} pin when port 8 is used as a digital port.

P80/ANI8/AMP2− to P83/ANI11 pins are as shown below depending on the settings of ADPC, ADS, PM8, and OAEN2.

Table 4-9. Setting Functions of P80/ANI8/AMP2− and P82/ANI10/AMP2+ Pins

ADPC	PM8	OAEN2	ADS	P80/ANI8/AMP2−, P82/ANI10/AMP2+
Digital I/O selection	Input mode	0	−	Digital input
		1	−	Setting prohibited
	Output mode	0	−	Digital output
		1	−	Setting prohibited
Analog input selection	Input mode	0	Selects ANI.	Analog input (to be converted)
			Does not select ANI.	Analog input (not to be converted)
		1	Selects ANI.	Setting prohibited
			Does not select ANI.	Operational amplifier input
	Output mode	−	−	Setting prohibited

Table 4-10. Setting Functions of P81/ANI9/AMP2OUT Pin

ADPC	PM8	OAEN2	ADS	P81/ANI9/AMP2OUT Pin
Digital I/O selection	Input mode	0	−	Digital input
		1	−	Setting prohibited
	Output mode	0	−	Digital output
		1	−	Setting prohibited
Analog input selection	Input mode	0	Selects ANI.	Analog input (to be converted)
			Does not select ANI.	Analog input (not to be converted)
		1	Selects ANI.	Operational amplifier output (to be converted)
			Does not select ANI.	Operational amplifier output (not to be converted)
	Output mode	−	−	Setting prohibited

Table 4-11. Setting Functions of P83/ANI11 Pin

ADPC	PM8	ADS	P83/ANI11 Pin
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	−	Setting prohibited

Caution When using an operational amplifier, the AMPn+, AMPn-, and AMPnOUT pins are used as I/O pins for the operational amplifier, and therefore cannot be used as analog input pins. However, the operational amplifier's output signal can be used as an analog input.

Figure 4-18 to 4-21 show block diagrams of port 8.

Figure 4-18. Block Diagram of P80

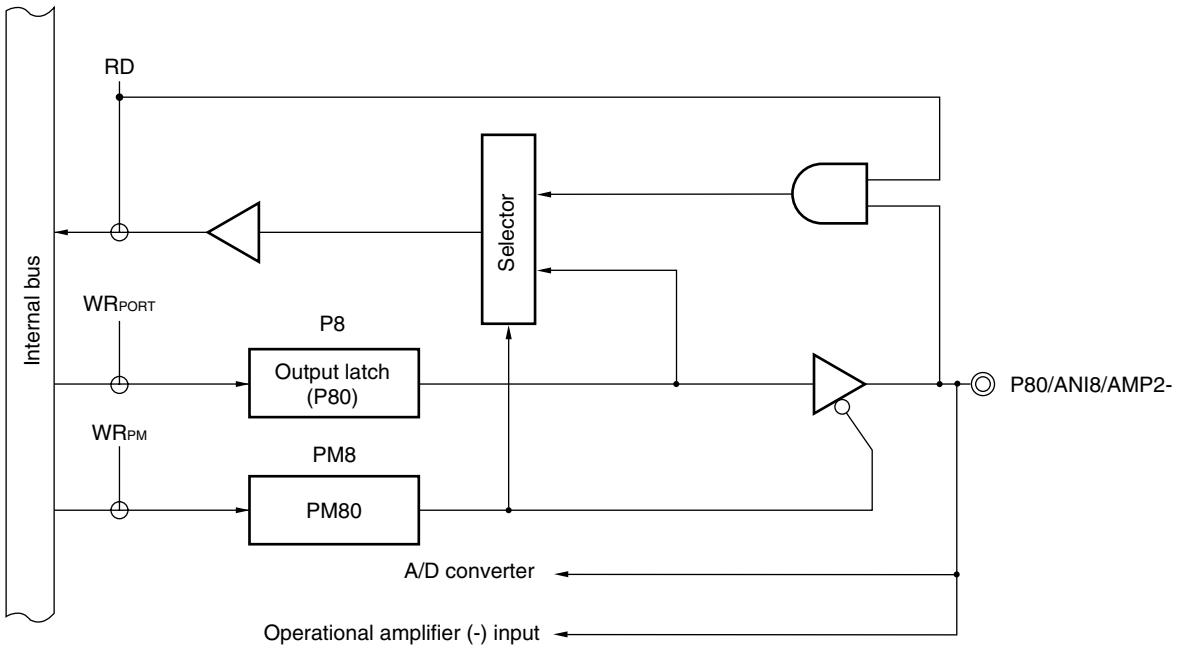
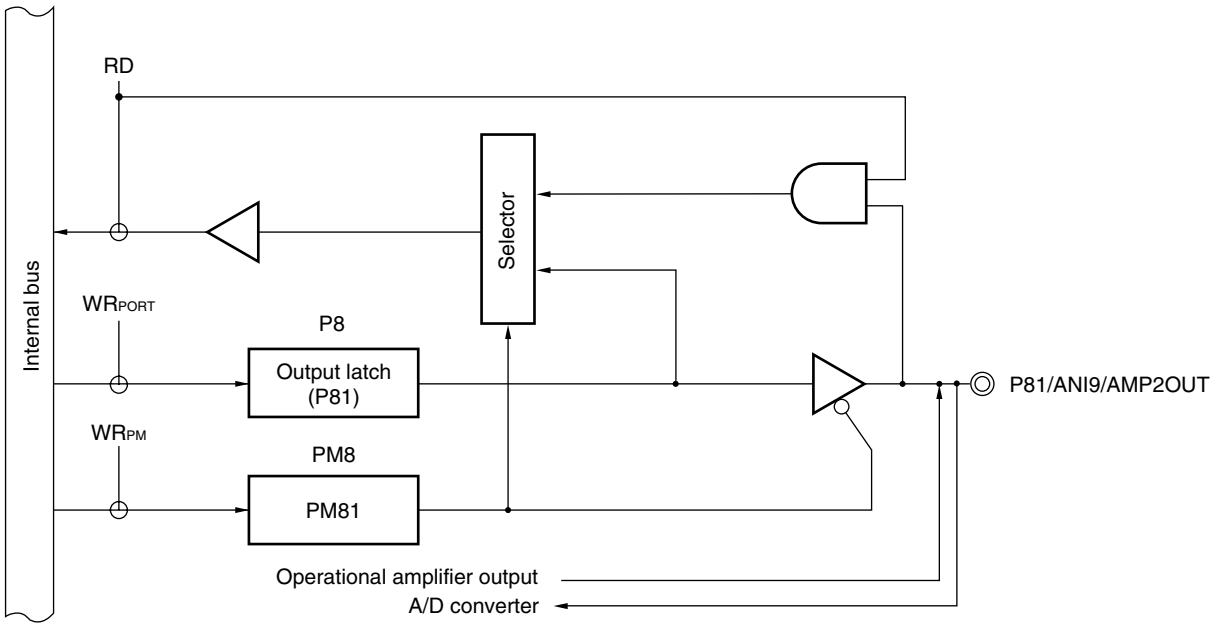


Figure 4-19. Block Diagram of P81



P8: Port register 8
PM8: Port mode register 8
RD: Read signal
WR××: Write signal

Figure 4-20. Block Diagram of P82

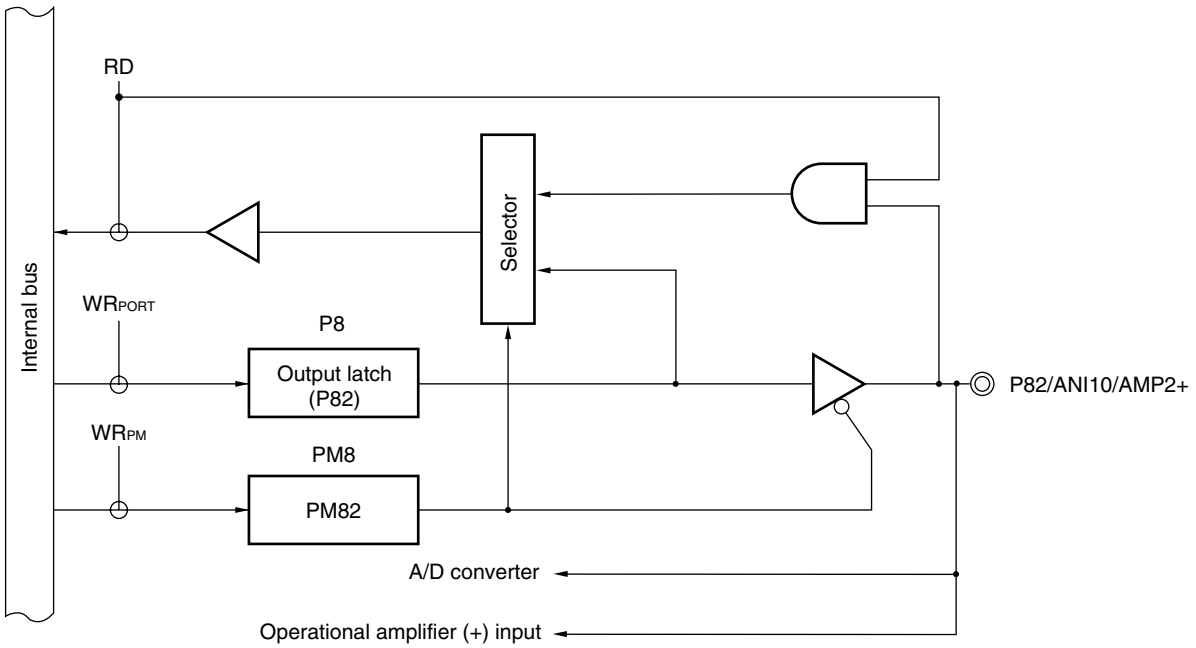
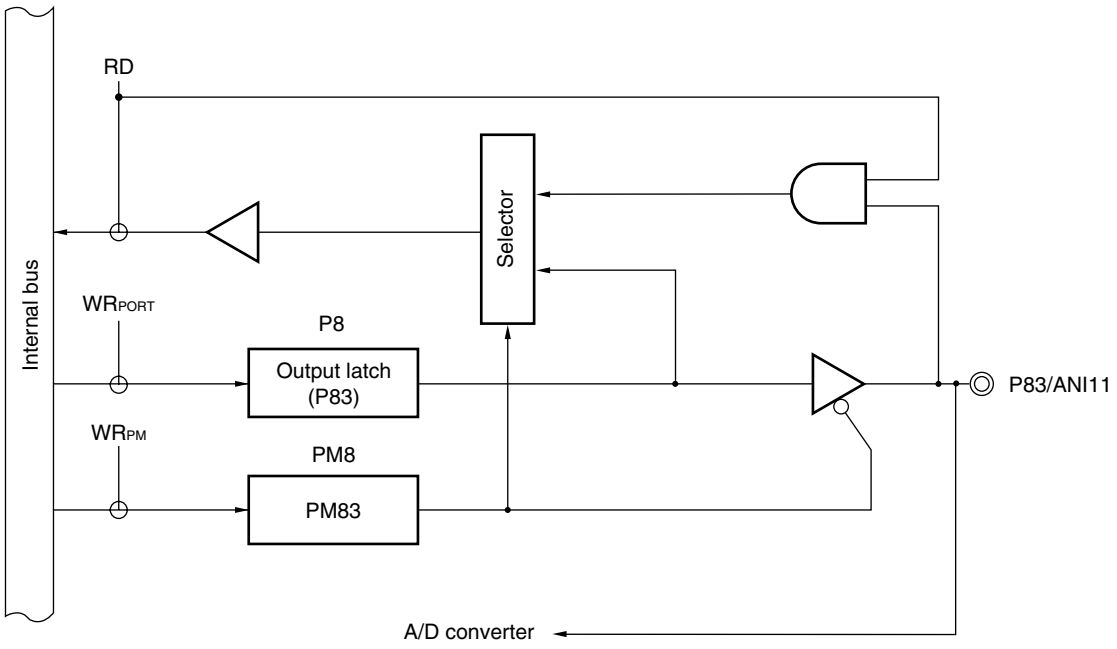


Figure 4-21. Block Diagram of P83



P8: Port register 8
PM8: Port mode register 8
RD: Read signal
WR××: Write signal

4.2.9 Port 12

78K0/KB2-A	78K0/KC2-A
P120/INTP0/EXLVI	
P121/X1/OCD0A	
P122/X2/OCD0B	
–	P123/XT1
–	P124/XT2

Port 12 is an 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 only for P120, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 12 (PU12).

This port can also be used as pins 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 4-22 to 4-24 show block diagrams of port 12.

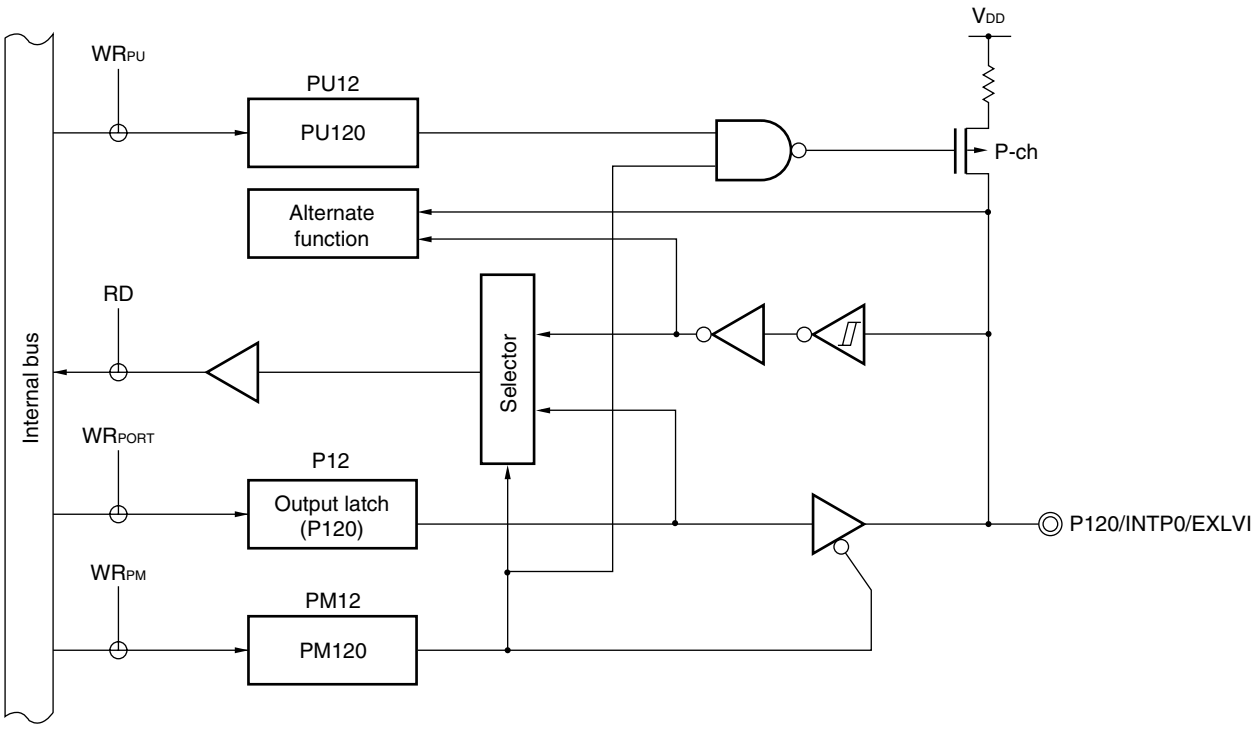
Cautions 1. When using the P121 to P124 pins to connect a resonator for the main system clock (X1, X2) or subsystem clock (XT1, XT2), or to input an external clock for the main system clock (EXCLK), the X1 oscillation mode, XT1 oscillation mode, or external clock input mode must be set by using the clock operation mode select register (OSCCTL) (for details, see 5.3 (1) Clock operation mode select register (OSCCTL)). The reset value of OSCCTL is 00H (all of the P121 to P124 pins are I/O port pins). At this time, setting of the PM121 to PM124 and P121 to P124 pins is not necessary.

2. Process the P121/X1/OCD0A pin as follows, when it is not used when it is connected to a flash memory programmer or an on-chip debug emulator.

		P121/X1/OCD0A
Flash memory programmer connection		Connect to V_{SS} via a open or resistor.
On-chip debug emulator connection (when it is not used as an on-chip debug mode setting pin)	During reset	
	During reset released	Input: Connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.

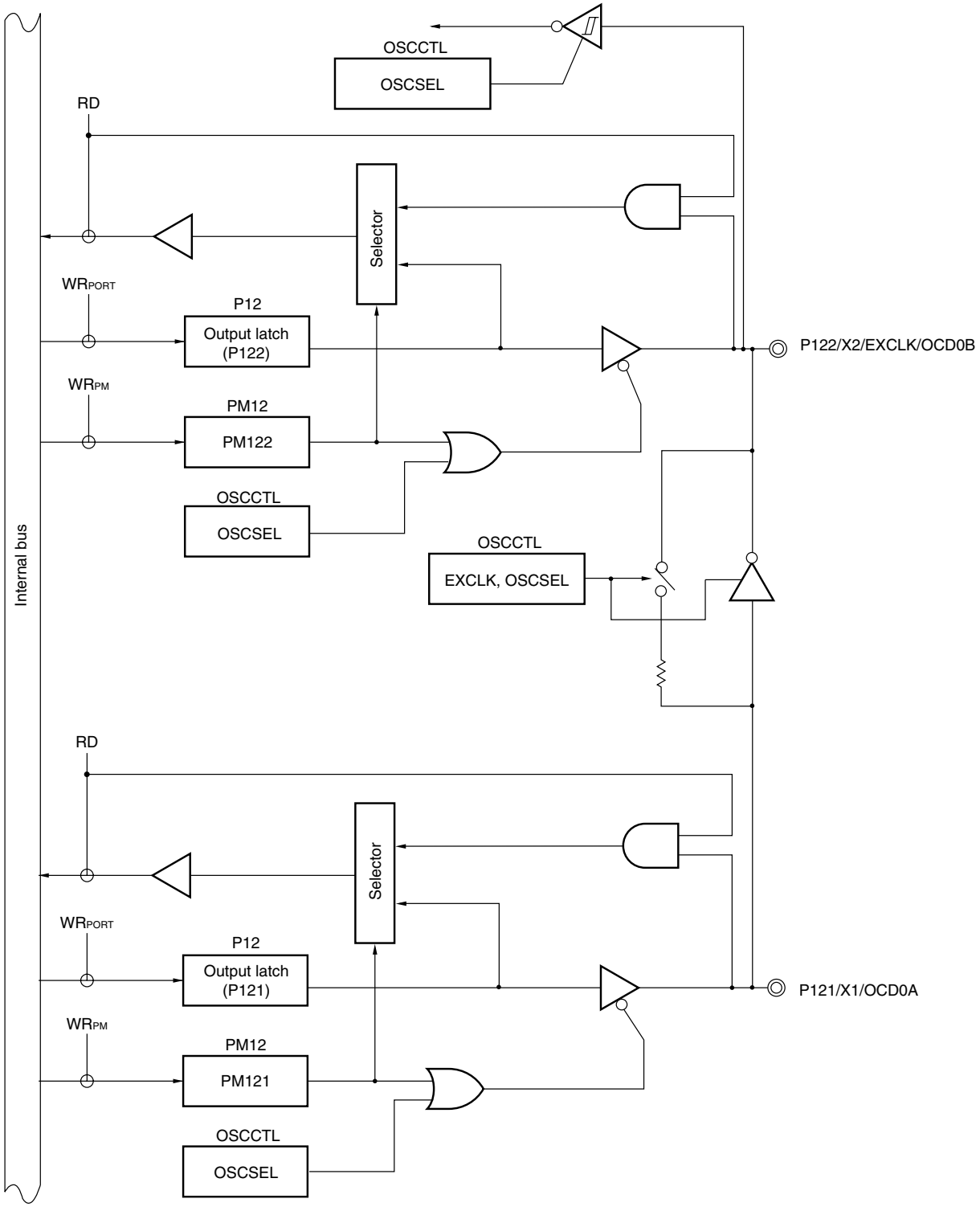
Remark X1 and X2 can be used as on-chip debug mode setting pins (OCD0A, OCD0B) when the on-chip debug function is used. For how to connect an on-chip debug emulator (QB-MINI2), see **CHAPTER 26 ON-CHIP DEBUG FUNCTION.**

Figure 4-22. Block Diagram of P120



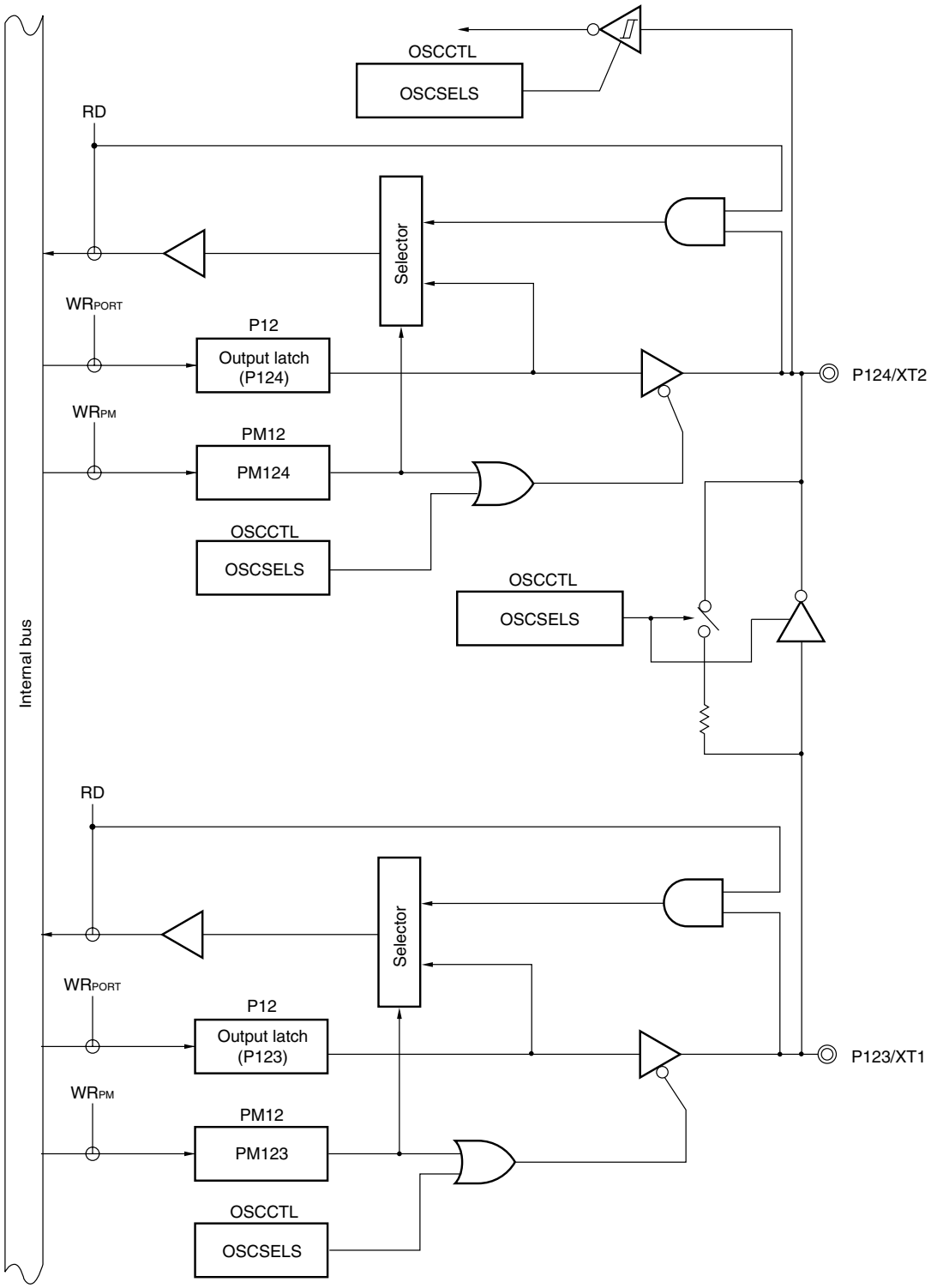
P12: Port register 12
PU12: Pull-up resistor option register 12
PM12: Port mode register 12
RD: Read signal
WR××: Write signal

Figure 4-23. Block Diagram of P121 and P122



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

Figure 4-24. Block Diagram of P123 and P124



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

4.3 Registers Controlling Port Function

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

- Port mode registers (PMxx)
- Port registers (Pxx)
- Pull-up resistor option registers (PUxx)
- A/D port configuration register (ADPC)

(1) Port mode registers (PMxx)

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

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

Reset signal generation sets these registers to FFH.

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

Figure 4-25. Format of Port Mode Register (1/2)

(1) 78K0/KB2-A

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM1	1	1	1	1	PM13	PM12	PM11	PM10	FF21H	FFH	R/W
PM2	1	1	PM25	PM24	PM23	PM22	PM21	PM20	FF22H	FFH	R/W
PM3	1	1	PM35	1	1	PM32	PM31	1	FF23H	FFH	R/W
PM6	1	1	1	1	1	1	PM61	PM60	FF26H	FFH	R/W
PM8	1	1	1	1	PM83	PM82	PM81	PM80	FF28H	FFH	R/W
PM12	1	1	1	1	1	PM122	PM121	PM120	FF2CH	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 1 to 3, 6, 8, 12; n = 0 to 5)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Caution Be sure to set bits 4 to 7 of PM1, bits 6 and 7 of PM2, bits 0, 3, 4, 6, and 7 of PM3, bits 2 to 7 of PM6, bits 4 to 7 of PM8, and bits 3 to 7 of PM12 to 1.

(2) 78K0/KC2-A

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	1	1	1	1	PM02	PM01	PM00	FF20H	FFH	R/W
PM1	1	1	1	1	PM13	PM12	PM11	PM10	FF21H	FFH	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FF22H	FFH	R/W
PM3	1	1	PM35	PM34	PM33	PM32	PM31	1	FF23H	FFH	R/W
PM4	1	1	1	1	1	PM42	PM41	PM40	FF24H	FFH	R/W
PM6	1	1	1	1	1	1	PM61	PM60	FF26H	FFH	R/W
PM7	1	1	PM75	PM74	PM73	PM72	PM71	PM70	FF27H	FFH	R/W
PM8	1	1	1	1	PM83	PM82	PM81	PM80	FF28H	FFH	R/W
PM12	1	1	1	PM124	PM123	PM122	PM121	PM120	FF2CH	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0 to 4, 6 to 8, 12 ; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Caution Be sure to set bits 3 to 7 of PM0, bits 4 to 7 of PM1, bits 0, 6, and 7 of PM3, bits 3 to 7 of PM4, bits 2 to 7 of PM6, bits 6 and 7 of PM7, bits 4 to 7 of PM8, and bits 5 to 7 of PM12 to 1.

(2) Port registers (Pxx)

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.

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

Reset signal generation clears these registers to 00H.

Figure 4-26. Format of Port Register (1/2)

(1) 78K0/KB2-A

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P1	0	0	0	0	P13	P12	P11	P10	FF01H	00H (output latch)	R/W
P2	0	0	P25	P24	P23	P22	P21	P20	FF02H	00H (output latch)	R/W
P3	0	0	P35	0	0	P32	P31	0	FF03H	00H (output latch)	R/W
P6	0	0	0	0	0	0	P61	P60	FF06H	00H (output latch)	R/W
P8	0	0	0	0	P83	P82	P81	P80	FF08H	00H (output latch)	R/W
P12	0	0	0	0	0	P122[Note]	P121[Note]	P120	FF0CH	00H (output latch)	R/W

Pmn	m = 1 to 3, 6, 8, 12; n = 0 to 5	
	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 “0” is always read from the output latch of P121 and P122 if the pin is in the external clock input mode.

Figure 4-26. Format of Port Register (2/2)

(2) 78K0/KC2-A

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	0	0	0	0	0	P02	P01	P00	FF00H	00H (output latch)	R/W
P1	0	0	0	0	P13	P12	P11	P10	FF01H	00H (output latch)	R/W
P2	P27	P26	P25	P24	P23	P22	P21	P20	FF02H	00H (output latch)	R/W
P3	0	0	P35	P34	P33	P32	P31	0	FF03H	00H (output latch)	R/W
P4	0	0	0	0	0	P42	P41	P40	FF04H	00H (output latch)	R/W
P6	0	0	0	0	0	0	P61	P60	FF06H	00H (output latch)	R/W
P7	0	0	P75	P74	P73	P72	P71	P70	FF07H	00H (output latch)	R/W
P8	0	0	0	0	P83	P82	P81	P80	FF08H	00H (output latch)	R/W
P12	0	0	0	P124	P123	P122[Note]	P121[Note]	P120	FF0CH	00H (output latch)	R/W

Pmn	m = 0 to 4, 6 to 8, 12; 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 “0” is always read from the output latch of P121 and P122 if the pin is in the external clock input mode.

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

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

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

Reset signal generation clears these registers to 00H.

Figure 4-27. Format of Pull-up Resistor Option Register (1/2)

(1) 78K0/KB2-A

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU1	0	0	0	0	PU13	PU12	PU11	PU10	FF31H	00H	R/W
PU3	0	0	PU35	0	0	PU32	PU31	0	FF33H	00H	R/W
PU12	0	0	0	0	0	0	0	PU120	FF3CH	00H	R/W

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

Figure 4-27. Format of Pull-up Resistor Option Register (2/2)

(2) 78K0/KC2-A

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	0	0	0	0	PU02	PU01	PU00	FF30H	00H	R/W
PU1	0	0	0	0	PU13	PU12	PU11	PU10	FF31H	00H	R/W
PU3	0	0	PU35	PU34	PU33	PU32	PU31	0	FF33H	00H	R/W
PU4	0	0	0	0	0	PU42	PU41	PU40	FF34H	00H	R/W
PU7	0	0	PU75	PU74	PU73	PU72	PU71	PU70	FF37H	00H	R/W
PU12	0	0	0	0	0	0	0	PU120	FF3CH	00H	R/W

PUmn	Pmn pin on-chip pull-up resistor selection (m = 0, 1, 3, 4, 7, 12 ; n = 0 to 5)
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

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

This register switches the P20/ANI0/AMP0− to P26/ANI6, P80/ANI8/AMP2− to P83/ANI11, and P27/ANI15/AV_{REFM} 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 clears this register to 10H.

Note 78K0/KB2-A: P20/ANI0/AMP0− to P25/ANI5/AMP1+, P80/ANI8/AMP2− to P83/ANI11

78K0/KC2-A: P20/ANI0/AMP0− to P26/ANI6, P80/ANI8/AMP2− to P83/ANI11, P27/ANI15/AV_{REFM}

Figure 4-28. Format of A/D Port Configuration Register (ADPC)

Address: FF2FH 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											
						ANI15 /AV_{REFM} /P27	ANI11 /P83	ANI10 /AMP2+ /P82	ANI9 /$AMP2_{OUT}$ /P81	ANI8 /AMP2– /P80	ANI6 /P26	ANI5 /AMP1+ /P25	ANI4 /$AMP1_{OUT}$ /P24	ANI3 /AMP1– /P23	ANI2 /AMP0+ /P22	ANI1 /$AMP0_{OUT}$ /P21	ANI0 /AMP0– /P20
	0	0	0	0	0	A	A	A	A	A	A	A	A	A	A	A	A
	0	0	0	0	1	A	A	A	A	A	A	A	A	A	A	A	D
	0	0	0	1	0	A	A	A	A	A	A	A	A	A	A	D	D
	0	0	0	1	1	A	A	A	A	A	A	A	A	A	D	D	D
	0	0	1	0	0	A	A	A	A	A	A	A	A	D	D	D	D
	0	0	1	0	1	A	A	A	A	A	A	A	D	D	D	D	D
Note –>	0	0	1	1	0	A	A	A	A	A	A	D	D	D	D	D	D
	0	0	1	1	1	Setting prohibited											
	0	1	0	0	0	A	A	A	A	A	D	D	D	D	D	D	D
	0	1	0	0	1	A	A	A	A	D	D	D	D	D	D	D	D
	0	1	0	1	0	A	A	A	D	D	D	D	D	D	D	D	D
	0	1	0	1	1	A	A	D	D	D	D	D	D	D	D	D	D
Note –>	0	1	1	1	1	A	D	D	D	D	D	D	D	D	D	D	D
	1	0	0	0	0	D	D	D	D	D	D	D	D	D	D	D	D
	Other than above					Setting prohibited											

Note In case of 78K0/KB2-A, setting prohibited

Cautions 1. Set the channel used for A/D conversion to the input mode by using port mode register 2 and 8 (PM2 and PM8).

2. Do not set the pin that is set by the A/D port configuration register (ADPC) by the analog input channel specification register (ADS).

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.

4.5 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 Tables 4-12 and 4-13.

Remark The port pins mounted depend on the product. See Tables 4-2 and 4-3.

Table 4-12. Settings of Port Mode Register and Output Latch When Using Alternate Function (78K0/KB2-A) (1/2)

Pin Name	Alternate Function		PM××	P××
	Function Name	I/O		
P10	TxD6	Output	0	1
	TI51	Input	1	×
	TO51	Output	0	0
P11	RxD6	Input	1	×
	TI50	Input	1	×
	TO50	Output	0	0
P12	TOH0	Output	0	0
	INTP7	Input	1	×
	TI000	Input	1	×
P13	TOH1	Output	0	0
	TI010	Input	1	×
	TO00	Output	0	0
	INTP6	Input	1	×
P20	ANI0[Note 1]	Input	1	×
	AMP0− [Note 1]	Input	1	×
P21	ANI1[Note 1]	Input	1	×
	AMP0OUT[Note 1]	Output	1	×
P22	ANI2[Note 1]	Input	1	×
	AMP0+ [Note 1]	Input	1	×
P23	ANI3[Note 1]	Input	1	×
	AMP1− [Note 1]	Input	1	×
P24	ANI4[Note 1]	Input	1	×
	AMP1OUT[Note 1]	Output	1	×
P25	ANI5[Note 1]	Input	1	×
	AMP1+ [Note 1]	Input	1	×
P31	INTP5	Input	1	×
	($\overline{SCK10}$)[Note 2]	Input	1	×
		Output	0	1
P32	INTP4	Input	1	×
	(SI10)[Note 2]	Input	1	×
P35	SO10	Output	0	0
	INTP1	Input	1	×

(Notes and remarks are on the next page.)

Notes 1. The function can be selected by using the ADPC, ADS, PM2, OAENn (n = 0, 1). Refer to Tables 4-5 to 4-8.

2. When using the P121 and P122 pins to connect a resonator for the main system clock (X1, X2) or to input an external clock for the main system clock (EXCLK), the X1 oscillation mode or external clock input mode must be set by using the clock operation mode select register (OSCCTL) (for details, see **5.3 (1) Clock operation mode select register (OSCCTL)**). The reset value of OSCCTL is 00H (P121 and P122 pins are I/O port pins). At this time, setting of the PM121, PM122, P121, and P122 pins are not necessary.

Remarks 1. ×: Don't care

PM××: Port mode register

P××: Port output latch

2. X1 and X2 can be used as on-chip debug mode setting pins (OCD0A, OCD0B, OCD1A, and OCD1B) when the on-chip debug function is used. For how to connect an on-chip debug emulator (QB-MINI2), see **CHAPTER 26 ON-CHIP DEBUG FUNCTION**.

Table 4-12. Settings of Port Mode Register and Output Latch When Using Alternate Function (78K0/KB2-A) (2/2)

Pin Name	Alternate Function		PM××	P××
	Function Name	I/O		
P60	SCLA0	I/O	0	0
	$\overline{SCK10}$	Input	1	×
P61	SDAA0	I/O	0	0
	SI10	Input	1	×
P80	ANI8[Note 1]	Input	1	×
	AMP2− [Note 1]	Input	1	×
P81	ANI9[Note 1]	Input	1	×
	AMP2OUT[Note 1]	Output	1	×
P82	ANI10[Note 1]	Input	1	×
	AMP2+ [Note 1]	Input	1	×
P83	ANI11[Note 1]	Input	1	×
P120	INTP0	Input	1	×
	EXLVI	Input	1	×
P121	X1[Note 2]	–	×	×
P122	X2[Note 2]	–	×	×
	EXCLK[Note 2]	Input	×	×

Notes 1. The function can be selected by using the ADPC, ADS, PM8, OAEN2. Refer to Tables 4-9 to 4-11.

2. The functions of pins whose names are in parentheses can be used by setting bit 2 (ISC2) of the input switch control register (ISC) to 1.

Remarks 1. ×: Don't care

PM××: Port mode register

P××: Port output latch

2. X1 and X2 can be used as on-chip debug mode setting pins (OCD0A, OCD0B, OCD1A, and OCD1B) when the on-chip debug function is used. For how to connect an on-chip debug emulator (QB-MINI2), see **CHAPTER 26 ON-CHIP DEBUG FUNCTION**..

Table 4-13. Settings of Port Mode Register and Output Latch When Using Alternate Function (78K0/KC2-A) (1/3)

Pin Name	Alternate Function		PM××	P××
	Function Name	I/O		
P00	TI000	Input	1	×
P01	TI010	Input	1	×
	TO00	Output	0	0
P02	INTP8	Input	1	×
P10	TxD6	Output	0	1
P11	RxD6	Input	1	×
P12	TOH0	Output	0	0
	INTP7	Input	1	×
P13	TOH1	Output	0	0
	INTP6	Input	1	×
P20	ANI0[Note 1]	Input	1	×
	AMP0−[Note 1]	Input	1	×
P21	ANI1[Note 1]	Input	1	×
	AMP0OUT[Note 1]	Output	1	×
P22	ANI2[Note 1]	Input	1	×
	AMP0+[Note 1]	Input	1	×
P23	ANI3[Note 1]	Input	1	×
	AMP1−[Note 1]	Input	1	×
P24	ANI4[Note 1]	Input	1	×
	AMP1OUT[Note 1]	Output	1	×
P25	ANI5[Note 1]	Input	1	×
	AMP1+[Note 1]	Input	1	×
P26	ANI6[Note 1]	Input	1	×
P27	ANI15[Note 1]	Input	1	×
	AV_{REFM}[Note 1]	Input	1	×
P31	INTP5	Input	1	×
	($\overline{SCK10}$)[Note 2]	Input	1	×
		Output	0	1
P32	INTP4	Input	1	×
	(SI10)[Note 2]	Input	1	×
P33	TI51	Input	1	×
	TO51	Output	0	0
	INTP3	Input	1	×

Notes 1. The function can be selected by using the ADPC, ADS, PM2, OAENn (n = 0, 1). Refer to Tables 4-5 to 4-8.

2. The functions of pins whose names are in parentheses can be used by setting bit 2 (ISC2) of the input switch control register (ISC) to 1.

Remarks 1. ×: Don't care

PM××: Port mode register

P××: Port output latch

2. P31 and P32 can be used as on-chip debug mode setting pins (OCD0A, OCD0B, OCD1A, and OCD1B) when the on-chip debug function is used. For how to connect an on-chip debug emulator (QB-MINI2), see **CHAPTER 26 ON-CHIP DEBUG FUNCTION**.

Table 4-13. Settings of Port Mode Register and Output Latch When Using Alternate Function (78K0/KC2-A) (2/3)

Pin Name	Alternate Function		PM××	P××
	Function Name	I/O		
P34	TI50	Input	1	×
	TO50	Output	0	0
	INTP2	Input	1	×
P35	SO10	Output	0	0
	INTP1	Input	1	×
P40	RTCCL	Output	0	0
	RTCDIV	Output	0	0
P41	RTC1HZ	Output	0	0
P42	PCL	Output	0	0
	$\overline{SSI10}$	Input	1	×
	INTP9	Input	1	×
P60	SCLA0	I/O	0	0
	$\overline{SCK10}$	Input	1	×
P61	SDAA0	I/O	0	0
	SI10	Input	1	×
P70-P75	KR0-KR5	Input	1	×
P80	ANI8[Note]	Input	1	×
	AMP2−[Note]	Input	1	×
P81	ANI9[Note]	Input	1	×
	AMP2OUT[Note]	Output	1	×
P82	ANI10[Note]	Input	1	×
	AMP2+[Note]	Input	1	×
P83	ANI11[Note]	Input	1	×

Note The function can be selected by using the ADPC, ADS, PM8, OAEN2. Refer to Tables 4-9 to 4-11.

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

Table 4-13. Settings of Port Mode Register and Output Latch When Using Alternate Function (78K0/KC2-A) (3/3)

Pin Name	Alternate Function		PM××	P××
	Function Name	I/O		
P120	INTP0	Input	1	×
	EXLVI	Input	1	×
P121	X1[Note]	–	×	×
P122	X2[Note]	–	×	×
	EXCLK[Note]	Input	×	×
P123	XT1[Note]	–	×	×
P124	XT2[Note]	–	×	×

Note When using the P121 to P124 pins to connect a resonator for the main system clock (X1, X2) or subsystem clock (XT1, XT2), or to input an external clock for the main system clock (EXCLK), the X1 oscillation mode, XT1 oscillation mode, or external clock input mode must be set by using the clock operation mode select register (OSCCTL) (for details, see **5.3 (1) Clock operation mode select register (OSCCTL)**). The reset value of OSCCTL is 00H (all of the P121 to P124 are I/O port pins). At this time, setting of PM121 to PM124 and P121 to P124 is not necessary.

Remarks 1. ×: Don't care
PM××: Port mode register
P××: Port output latch

2. X1 and X2 can be used as on-chip debug mode setting pins (OCD0A, OCD0B, OCD1A, and OCD1B) when the on-chip debug function is used. For how to connect an on-chip debug emulator (QB-MINI2), see **CHAPTER 26 ON-CHIP DEBUG FUNCTION**.

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 P13 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 0FH.

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 78K0/Kx2-A microcontrollers.

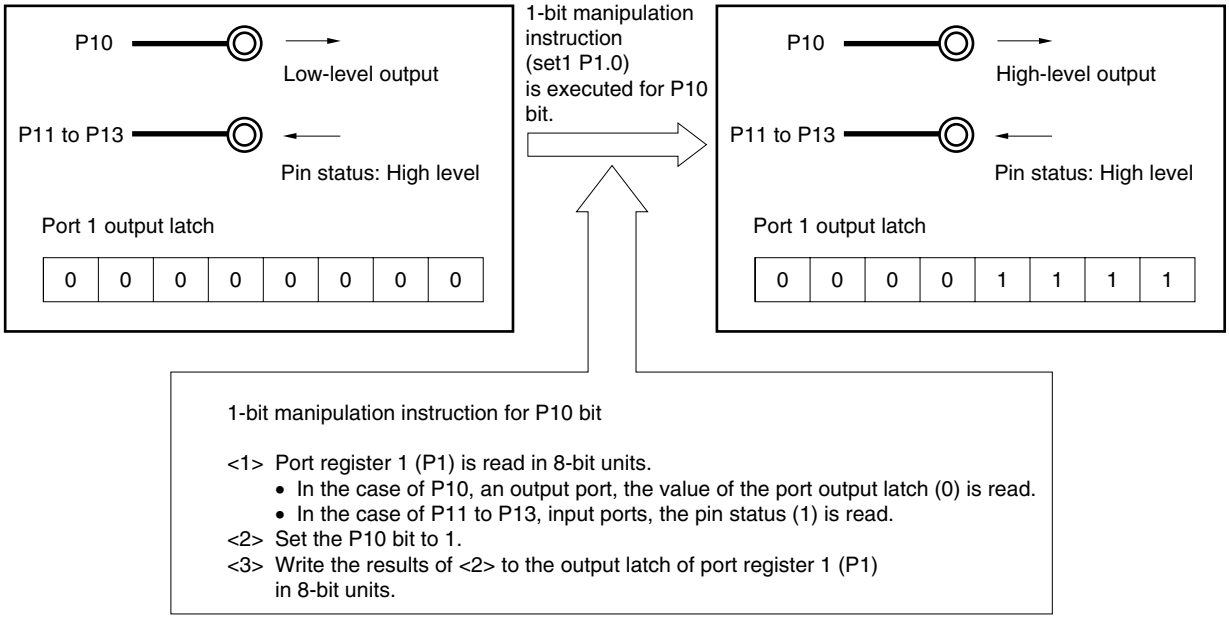
<1> The Pn register is read in 8-bit units.
<2> The targeted one bit is manipulated.
<3> The Pn register is written in 8-bit units.

In step <1>, the output latch value (0) of P10, which is an output port, is read, while the pin statuses of P11 to P13, which are input ports, are read. If the pin statuses of P11 to P13 are high level at this time, the read value is 0EH.

The value is changed to 0FH by the manipulation in <2>.

0FH is written to the output latch by the manipulation in <3>.

Figure 4-29. 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 = 1 to 20 MHz by connecting a resonator to X1 and X2.
Oscillation can be stopped by executing the STOP instruction or using the main OSC control register (MOC).

<2> Internal high-speed oscillator

This circuit oscillates a clock of f_{RH} = 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 using the internal oscillation mode register (RCM).

An external main system clock (f_{EXCLK} = 1 to 20 MHz) can also be supplied from the EXCLK/X2/P122 pin. An external main system clock input can be disabled by executing the STOP instruction or using RCM.
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 using the main clock mode register (MCM).

(2) Subsystem clock[Note]

- **Subsystem clock oscillator**

  This circuit oscillates at a frequency of f_{XT} = 32.768 kHz by connecting a 32.768 kHz resonator across XT1 and XT2. Oscillation can be stopped by using the clock operation mode select register (OSCCTL).

Note The 78K0/KB2-A is not provided with a subsystem clock.

Remark f_X: X1 clock oscillation frequency
f_{RH}: Internal high-speed oscillation clock frequency
f_{EXCLK}: External main system clock frequency
f_{XT}: XT1 clock oscillation frequency

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

- **Internal low-speed oscillator**

  This circuit oscillates a clock of f_{RL} = 240 kHz (TYP.). After a reset release, the internal low-speed oscillation clock always starts operating.

  Oscillation can be stopped by using the internal oscillation mode register (RCM) when "internal low-speed oscillator can be stopped by software" is set by option byte.

  The internal low-speed oscillation clock cannot be used as the CPU clock. The following hardware operates with the internal low-speed oscillation clock.

  - Watchdog timer
  - 8-bit timer H1 (when f_{RL}, $f_{RL}/2^7$, or $f_{RL}/2^9$ is selected)

Remark f_{RL}: Internal low-speed oscillation clock frequency

5.2 Configuration of Clock Generator

The clock generator includes the following hardware.

Table 5-1. Configuration of Clock Generator

Item	Configuration
Control registers	Clock operation mode select register (OSCCTL) Processor clock control register (PCC) Internal oscillation mode register (RCM) Main OSC control register (MOC) Main clock mode register (MCM) Oscillation stabilization time counter status register (OSTC) Oscillation stabilization time select register (OSTS)
Oscillators	X1 oscillator XT1 oscillator[Note] Internal high-speed oscillator Internal low-speed oscillator

Note The 78K0/KB2-A is not provided with an XT1 oscillator (subsystem clock).

Figure 5-1. Block Diagram of Clock Generator (78K0/KB2-A)

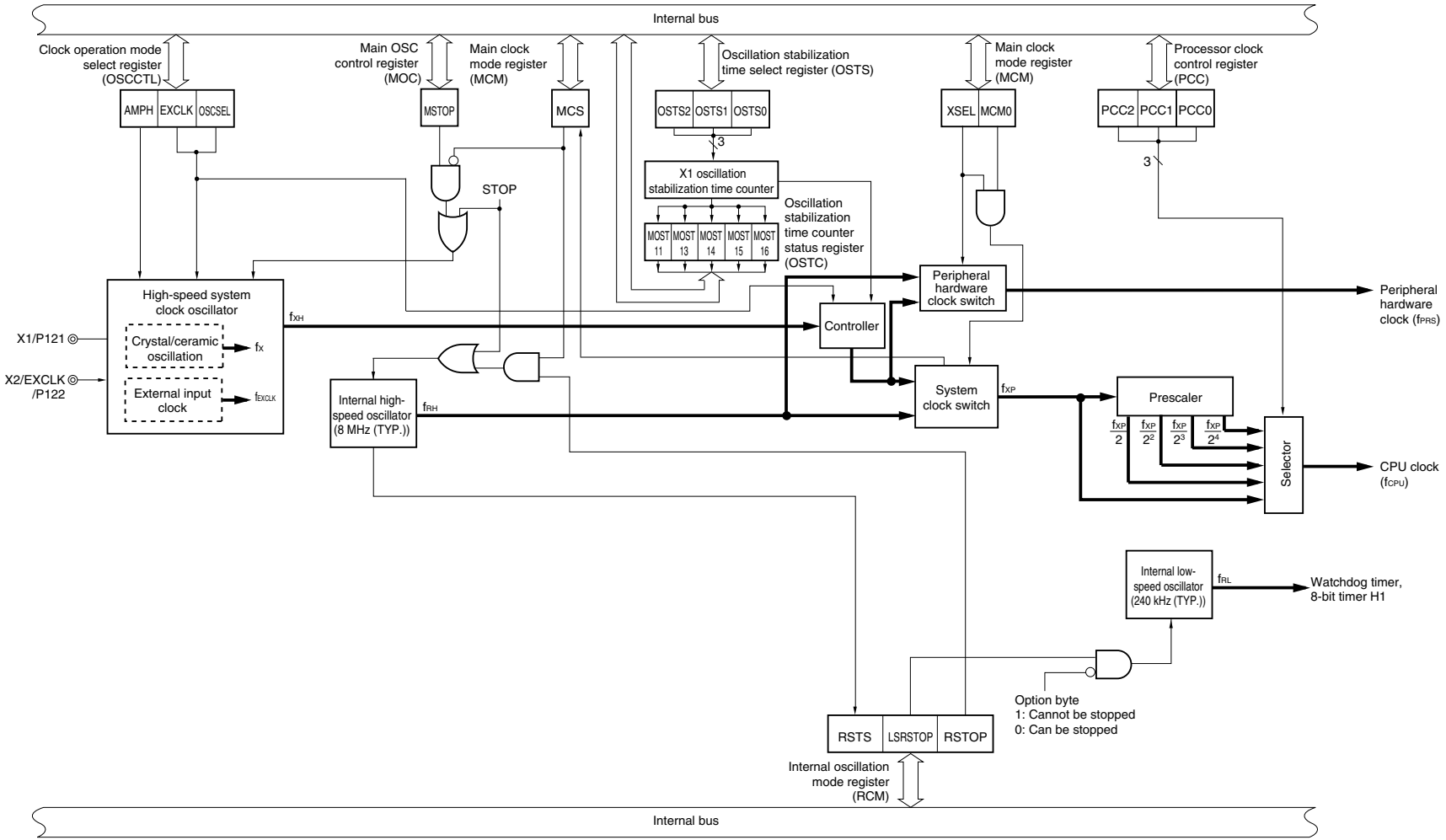
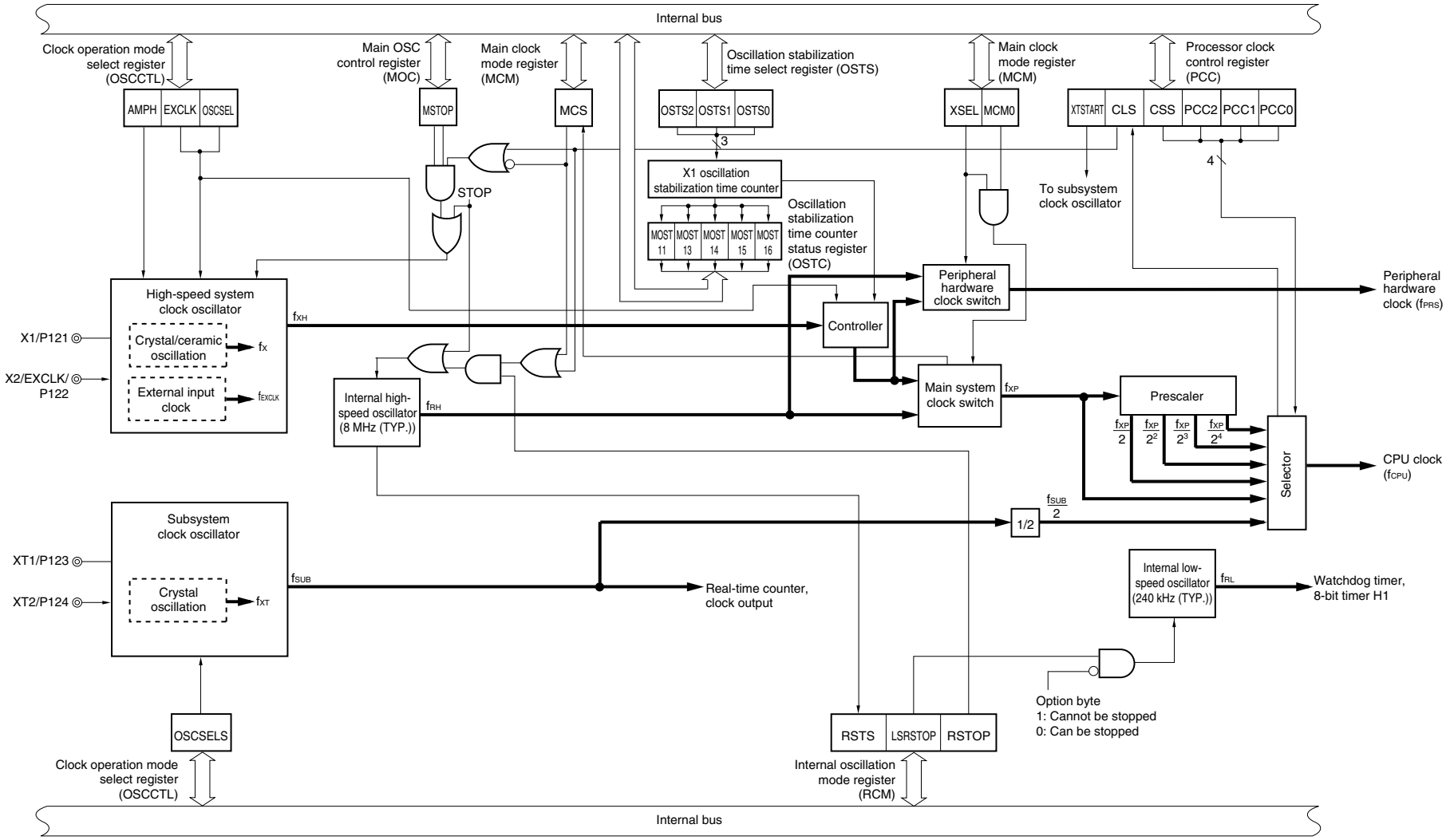


Figure 5-2. Block Diagram of Clock Generator (78K0/KC2-A)



Remark f_X: X1 clock oscillation frequency
f_{RH}: Internal high-speed oscillation clock frequency
f_{EXCLK}: External main system clock frequency
f_{XH}: High-speed system clock frequency
f_{XP}: Main system clock frequency
f_{PRS}: Peripheral hardware clock frequency
f_{CPU}: CPU clock frequency
f_{XT}: XT1 clock oscillation frequency
f_{SUB}: Subsystem clock frequency
f_{RL}: Internal low-speed oscillation clock frequency

5.3 Registers Controlling Clock Generator

The following seven registers are used to control the clock generator.

- Clock operation mode select register (OSCCTL)
- Processor clock control register (PCC)
- Internal oscillation mode register (RCM)
- Main OSC control register (MOC)
- Main clock mode register (MCM)
- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)

(1) Clock operation mode select register (OSCCTL)

This register selects the operation modes of the high-speed system and subsystem clocks, and the gain of the on-chip oscillator.

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

Reset signal generation clears this register to 00H.

Figure 5-3. Format of Clock Operation Mode Select Register (OSCCTL)

<1> 78K0/KB2-A

Address: FF9FH After reset: 00H R/W

Symbol	<7>	<6>	5	4	3	2	1	<0>
OSCCTL	EXCLK	OSCSEL	0	0	0	0	0	AMPH

<2> 78K0/KC2-A

Address: FF9FH After reset: 00H R/W

Symbol	<7>	<6>	5	<4>	3	2	1	<0>
OSCCTL	EXCLK	OSCSEL	0	OSCSELS	0	0	0	AMPH

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

OSCSELS	Subsystem clock pin operation mode	P123/XT1 pin	P124/XT2 pin
0	I/O port mode	I/O port	
1	XT1 operating mode	Crystal resonator connection	

AMPH	High-speed system clock operating frequency control
0	1 MHz ≤ f_{XH} ≤ 10 MHz
1	10 MHz < f_{XH} ≤ 20 MHz

Cautions
1. **Be sure to set AMPH to 1 if the high-speed system clock oscillation frequency exceeds 10 MHz.**
2. **Set AMPH before setting the main system mode register (MCM).**
3. **Set AMPH before setting the peripheral functions after a reset release. The value of AMPH can be changed only once after a reset release. When the high-speed system clock (X1 oscillation) is selected as the CPU clock, supply of the CPU clock is stopped for 4.06 to 16.12 μs after AMPH is set to 1. When the high-speed system clock (external clock input) is selected as the CPU clock, supply of the CPU clock is stopped for the duration of 160 external clocks after AMPH is set to 1.**
4. **If the STOP instruction is executed when AMPH = 1, supply of the CPU clock is stopped for 4.06 to 16.12 μs after the STOP mode is released when the internal high-speed oscillation clock is selected as the CPU clock, or for the duration of 160 external clocks when the high-speed system clock (external clock input) is selected as the CPU clock. When the high-speed system clock (X1 oscillation) is selected as the CPU clock, the oscillation stabilization time is counted after the STOP mode is released.**
5. **To change the value of EXCLK and OSCSEL, be sure to confirm that bit 7 (MSTOP) of the main OSC control register (MOC) is 1 (the X1 oscillator stops or the external clock from the EXCLK pin is disabled).**
6. **To change the value of OSCSELS, be sure to confirm that bit 5 (CLS) of the processor clock control register (PCC) is 1 (CPU is operating with main system clock).**
7. **For the 78K0/KB2-A, be sure to set bits 1 to 5 to "0". For the 78K0/KC2-A, be sure to set bits 1 to 3 and 5 to "0".**

Remark f_{XH}: High-speed system clock oscillation frequency

(2) Processor clock control register (PCC)

This register is used to select the CPU clock, the division ratio, and operation mode for subsystem clock.

PCC is set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets PCC to 01H.

Figure 5-4. Format of Processor Clock Control Register (PCC)

<1> 78K0/KB2-A

Address: FFFBH After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
PCC	0	0	0	0	0	PCC2	PCC1	PCC0

<2> 78K0/KC2-A

Address: FFFBH After reset: 01H R/W[Note]

Symbol	7	6	<5>	<4>	3	2	1	0
PCC	0	0	CLS	CSS	0	PCC2	PCC1	PCC0

CLS	CPU clock status
0	Main system clock
1	Subsystem clock

CSS	PCC2	PCC1	PCC0	CPU clock (f_{CPU}) selection
0	0	0	0	f_{XP}
	0	0	1	$f_{XP}/2$ (default)
	0	1	0	$f_{XP}/2^2$
	0	1	1	$f_{XP}/2^3$
	1	0	0	$f_{XP}/2^4$
1	0	0	0	$f_{SUB}/2$
	0	0	1	
	0	1	0	
	0	1	1	
	1	0	0	
Other than above				Setting prohibited

Note Bit 5 is read-only.

Cautions 1. For the 78K0/KB2-A, be sure to set bits 3 to 7 to "0". For the 78K0/KC2-A, be sure to set bits 3, 6, and 7 to "0".

2. The peripheral hardware clock (f_{PRS}) is not divided when the division ratio of the PCC is set.

3. The CPU clock (f_{CPU}) is supplied to the CPU and the A/D converter. If f_{CPU} is changed, therefore, the A/D converter's conversion clock (f_{AD}) is also changed. When changing f_{CPU} by using the PCC register, therefore, be sure to stop the A/D converter first (by setting ADCS to 0).

Remarks 1. f_{XP}: Main system clock oscillation frequency

2. f_{SUB}: Subsystem clock oscillation frequency

The fastest instruction can be executed in 2 clocks of the CPU clock in the 78K0/Kx2-A microcontrollers. Therefore, the relationship between the CPU clock (f_{CPU}) and the minimum instruction execution time is as shown in Table 5-2.

Table 5-2. Relationship Between CPU Clock and Minimum Instruction Execution Time

CPU Clock (f_{CPU})	Minimum Instruction Execution Time: $2/f_{CPU}$			
	Main System Clock			Subsystem Clock[Note 2]
	High-Speed System Clock[Note 1]		Internal High-Speed Oscillation Clock[Note 1]	
	At 10 MHz Operation	At 20 MHz Operation	At 8 MHz (TYP.) Operation	At 32.768 kHz Operation
f_{XP}	0.2 μs	0.1 μs	0.25 μs (TYP.)	–
$f_{XP}/2$	0.4 μs	0.2 μs	0.5 μs (TYP.)	–
$f_{XP}/2^2$	0.8 μs	0.4 μs	1.0 μs (TYP.)	–
$f_{XP}/2^3$	1.6 μs	0.8 μs	2.0 μs (TYP.)	–
$f_{XP}/2^4$	3.2 μs	1.6 μs	4.0 μs (TYP.)	–
$f_{SUB}/2$[Note 2]	–		–	122.1 μs

Notes 1. The main clock mode register (MCM) is used to set the main system clock supplied to CPU clock (high-speed system clock/internal high-speed oscillation clock) (see **Figure 5-7**).

2. The 78K0/KB2-A is not provided with a subsystem clock.

(3) Internal oscillation mode register (RCM)

This register sets the operation mode of internal oscillator.

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

Reset signal generation sets this register to 80H[Note 1].

Figure 5-5. Format of Internal Oscillation Mode Register (RCM)

Address: FFA0H After reset: 80H[Note 1] R/W[Note 2]

Symbol	<7>	6	5	4	3	2	<1>	<0>
RCM	RSTS	0	0	0	0	0	LSRSTOP	RSTOP

RSTS	Status of internal high-speed oscillator
0	Waiting for accuracy stabilization of internal high-speed oscillator
1	Stability operating of internal high-speed oscillator

LSRSTOP	Internal low-speed oscillator oscillating/stopped
0	Internal low-speed oscillator oscillating
1	Internal low-speed oscillator stopped

RSTOP	Internal high-speed oscillator oscillating/stopped
0	Internal high-speed oscillator oscillating
1	Internal high-speed oscillator stopped

Notes 1. The value of this register is 00H immediately after a reset release but automatically changes to 80H after internal high-speed oscillator has been stabilized.

2. Bit 7 is read-only.

Caution When setting RSTOP to 1, be sure to confirm that the CPU operates with a clock other than the internal high-speed oscillation clock. Specifically, set under either of the following conditions.

<1> 78K0/KB2-A

- **When MCS = 1 (when CPU operates with the high-speed system clock)**

<2> 78K0/KC2-A

- **When MCS = 1 (when CPU operates with the high-speed system clock)**
- **When CLS = 1 (when CPU operates with the subsystem clock)**

In addition, stop peripheral hardware that is operating on the internal high-speed oscillation clock before setting RSTOP to 1.

(4) Main OSC control register (MOC)

This register selects the operation mode of the high-speed system clock.

This register is used to stop the X1 oscillator or to disable an external clock input from the EXCLK pin when the CPU operates with a clock other than the high-speed system clock.

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

Reset signal generation sets this register to 80H.

Figure 5-6. Format of Main OSC Control Register (MOC)

Address: FFA2H After reset: 80H R/W

Symbol	<7>	6	5	4	3	2	1	0
MOC	MSTOP	0	0	0	0	0	0	0

MSTOP	Control of high-speed system clock operation	
	X1 oscillation mode	External clock input mode
0	X1 oscillator operating	External clock from EXCLK pin is enabled
1	X1 oscillator stopped	External clock from EXCLK pin is disabled

Cautions 1. When setting MSTOP to 1, be sure to confirm that the CPU operates with a clock other than the high-speed system clock. Specifically, set under either of the following conditions.

<1> 78K0/KB2-A

- **When MCS = 0 (when CPU operates with the internal high-speed oscillation clock)**

<2> 78K0/KC2-A

- **When MCS = 0 (when CPU operates with the internal high-speed oscillation clock)**
- **When CLS = 1 (when CPU operates with the subsystem clock)**

In addition, stop peripheral hardware that is operating on the high-speed system clock before setting MSTOP to 1.

2. Do not clear MSTOP to 0 while bit 6 (OSCSEL) of the clock operation mode select register (OSCCTL) is 0 (I/O port mode).

3. The peripheral hardware cannot operate when the peripheral hardware clock is stopped. To resume the operation of the peripheral hardware after the peripheral hardware clock has been stopped, initialize the peripheral hardware.

(5) Main clock mode register (MCM)

This register selects the main system clock supplied to CPU clock and clock supplied to peripheral hardware clock.

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

Reset signal generation clears this register to 00H.

Figure 5-7. Format of Main Clock Mode Register (MCM)

Address: FFA1H After reset: 00H R/W[Note]

Symbol	7	6	5	4	3	<2>	<1>	<0>
MCM	0	0	0	0	0	XSEL	MCS	MCM0

XSEL	MCM0	Selection of clock supplied to main system clock and peripheral hardware	
		Main system clock (f_{XP})	Peripheral hardware clock (f_{PRS})
0	0	Internal high-speed oscillation clock (f_{RH})	Internal high-speed oscillation clock (f_{RH})
0	1		
1	0		High-speed system clock (f_{XH})
1	1	High-speed system clock (f_{XH})	

MCS	Main system clock status
0	Operates with internal high-speed oscillation clock
1	Operates with high-speed system clock

Note Bit 1 is read-only.

Cautions 1. XSEL can be changed only once after a reset release.

2. Do not rewrite MCM0 when the CPU clock operates with the subsystem clock.

3. A clock other than f_{PRS} is supplied to the following peripheral functions regardless of the setting of XSEL and MCM0.

- **Watchdog timer (operates with internal low-speed oscillation clock)**
- **When "f_{RL}", "$f_{RL}/2^7$", or "$f_{RL}/2^9$" is selected as the count clock for 8-bit timer H1 (operates with internal low-speed oscillation clock)**
- **Peripheral hardware selects the external clock as the clock source (Except when the external count clock of TM00 is selected (TI000 pin valid edge))**

4. The CPU clock (f_{CPU}) is supplied to the CPU and the A/D converter. If f_{CPU} is changed, therefore, the A/D converter's conversion clock (f_{AD}) is also changed. When changing f_{CPU} by using the PCC register, therefore, be sure to stop the A/D converter first (by setting ADCS to 0).

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

This is the register that indicates the count status of the X1 clock oscillation stabilization time counter. When X1 clock oscillation starts with the internal high-speed oscillation clock or subsystem clock used as the CPU clock, the X1 clock oscillation stabilization time can be checked.

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, and WDT), the STOP instruction and MSTOP (bit 7 of MOC register) = 1 clear OSTC to 00H.

Figure 5-8. Format of Oscillation Stabilization Time Counter Status Register (OSTC)

Address: FFA3H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
OSTC	0	0	0	MOST11	MOST13	MOST14	MOST15	MOST16

MOST11	MOST13	MOST14	MOST15	MOST16	Oscillation stabilization time status		
						f_X = 10 MHz	f_X = 20 MHz
1	0	0	0	0	$2^{11}/f_X$ min.	204.8 μs min.	102.4 μs min.
1	1	0	0	0	$2^{13}/f_X$ min.	819.2 μs min.	409.6 μs min.
1	1	1	0	0	$2^{14}/f_X$ min.	1.64 ms min.	819.2 μs min.
1	1	1	1	0	$2^{15}/f_X$ min.	3.27 ms min.	1.64 ms min.
1	1	1	1	1	$2^{16}/f_X$ min.	6.55 ms min.	3.27 ms min.

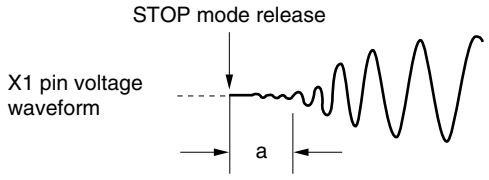
Cautions 1. After the above time has elapsed, the bits are set to 1 in order from MOST11 and remain 1.

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

(7) 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 OSTS to 05H.

Figure 5-9. Format of Oscillation Stabilization Time Select Register (OSTS)

Address: FFA4H After reset: 05H 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	1	$2^{11}/f_X$	204.8 μs	102.4 μs
0	1	0	$2^{13}/f_X$	819.2 μs	409.6 μs
0	1	1	$2^{14}/f_X$	1.64 ms	819.2 μs
1	0	0	$2^{15}/f_X$	3.27 ms	1.64 ms
1	0	1	$2^{16}/f_X$	6.55 ms	3.27 ms
Other than above			Setting prohibited		

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

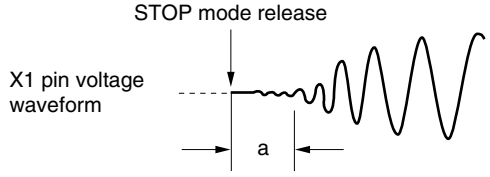
2. Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.

3. 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.

4. 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

5.4 System Clock Oscillator

5.4.1 X1 oscillator

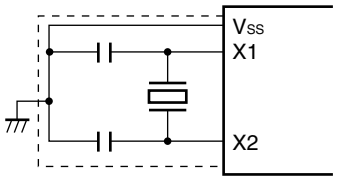
The X1 oscillator oscillates with a crystal resonator or ceramic resonator (1 to 20 MHz) connected to the X1 and X2 pins.

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

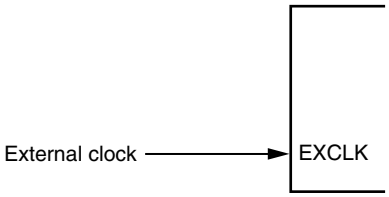
Figure 5-10 shows an example of the external circuit of the X1 oscillator.

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.

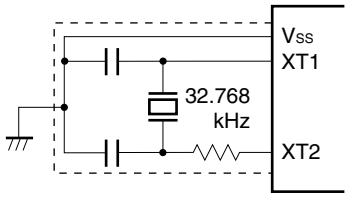
5.4.2 XT1 oscillator

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

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

Note The 78K0/KB2-A is not provided with an XT1 oscillator.

Figure 5-11. Example of External Circuit of XT1 Oscillator



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.

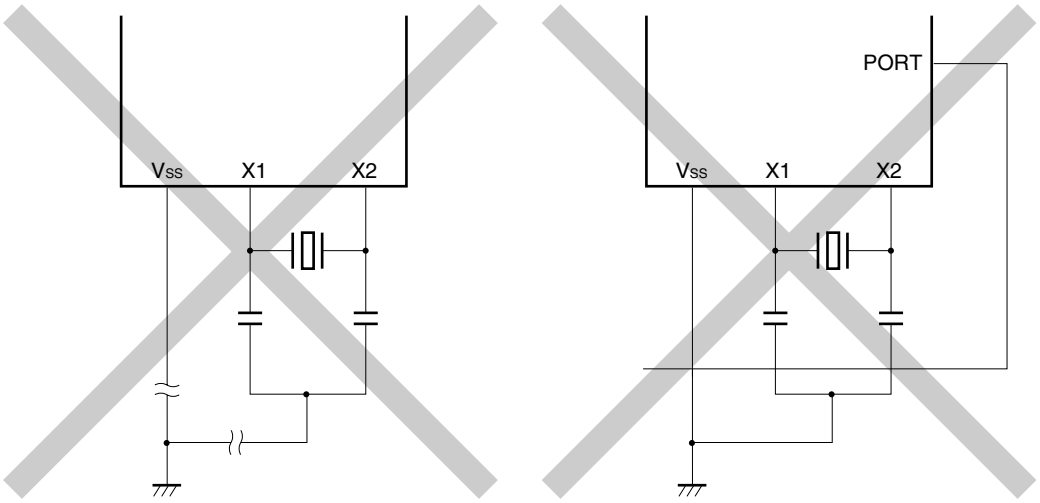
- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as V_{SS}. Do not ground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

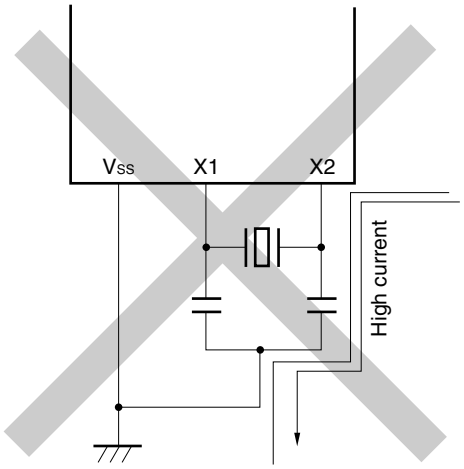
(a) Too long wiring **(b) Crossed signal line**



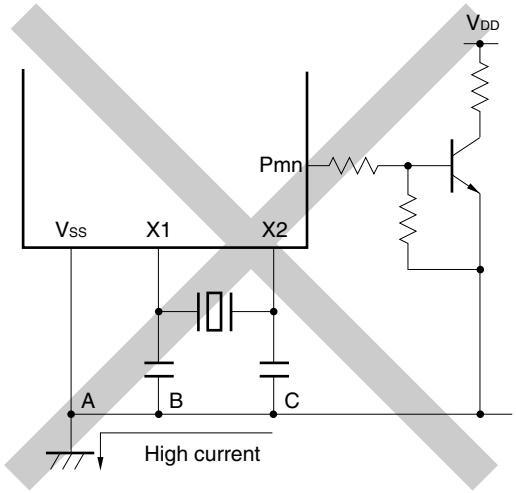
Remark When using the subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Also, insert resistors in series on the XT2 side.

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

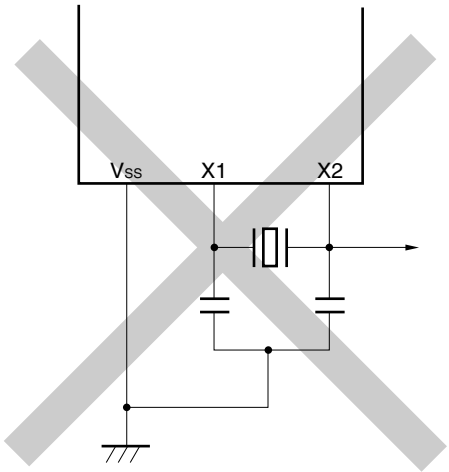
(c) Wiring near high alternating current



(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 When subsystem clock is not used

If it is not necessary to use the subsystem clock[Note] for low power consumption operations and real-time counter, or if not using the subsystem clock as an I/O port, set the XT1 and XT2 pins to I/O mode (OSCSELS = 0) and connect them as follows.

Note The 78K0/KB2-A is not provided with a subsystem clock.

Input (PM123/PM124 = 1): Independently connect to V_{DD} or V_{SS} via a resistor.
Output (PM123/PM124 = 0): Leave open.

Remark OSCSELS: Bit 4 of clock operation mode select register (OSCCTL)
PM123, PM124: Bits 3 and 4 of port mode register 12 (PM12)

5.4.4 Internal high-speed oscillator

The internal high-speed oscillator is incorporated in the 78K0/Kx2-A microcontrollers. Oscillation can be controlled by the internal oscillation mode register (RCM).

After a reset release, the internal high-speed oscillator automatically starts oscillation (8 MHz (TYP.)).

5.4.5 Internal low-speed oscillator

The internal low-speed oscillator is incorporated in the 78K0/Kx2-A microcontrollers.

The internal low-speed oscillation clock is only used as the watchdog timer and the clock of 8-bit timer H1. The internal low-speed oscillation clock cannot be used as the CPU clock.

“Can be stopped by software” or “Cannot be stopped” can be selected by the option byte. When “Can be stopped by software” is set, oscillation can be controlled by the internal oscillation mode register (RCM).

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 using the option byte.

5.4.6 Prescaler

The prescaler generates the CPU clock by dividing the main system clock when the main system clock is selected as the clock to be supplied to the CPU.

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** and **5-2**).

- Main system clock f_{XP}
  - High-speed system clock f_{XH}
    X1 clock f_X
    External main system clock f_{EXCLK}
  - Internal high-speed oscillation clock f_{RH}
- Subsystem clock f_{SUB} Note
  - XT1 clock f_{XT}
- Internal low-speed oscillation clock f_{RL}
- CPU clock f_{CPU}
- Peripheral hardware clock f_{PRS}

Note The 78K0/KB2-A is not provided with a subsystem clock.

The CPU starts operation when the internal high-speed oscillator starts outputting after a reset release in the 78K0/Kx2-A microcontrollers, thus enabling the following.

(1) Enhancement of security function

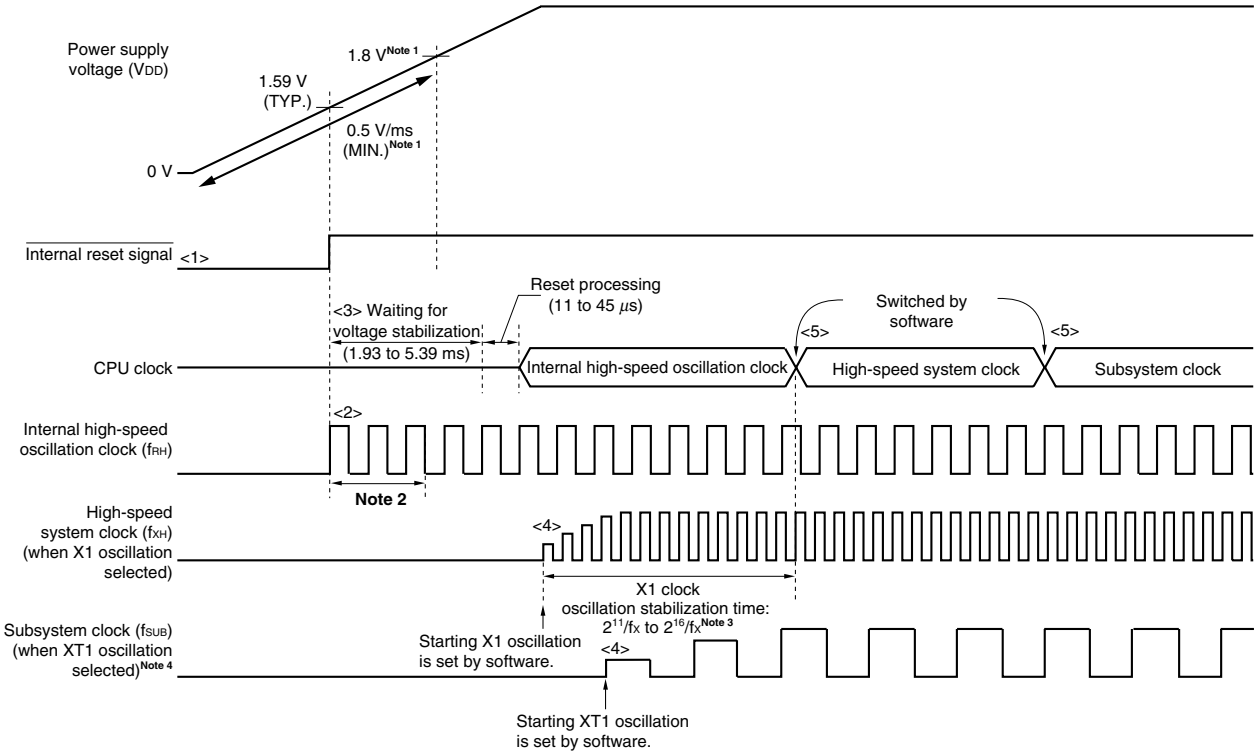
When the X1 clock is set as the CPU clock by the default setting, the device cannot operate if the X1 clock is damaged or badly connected and therefore does not operate after reset is released. However, the start clock of the CPU is the internal high-speed oscillation clock, so the device can be started by the internal high-speed oscillation clock after a reset release. Consequently, the system can be safely shut down by performing a minimum operation, such as acknowledging a reset source by software or performing safety processing when there is a malfunction.

(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 Figures 5-13 and 5-14.

Figure 5-13. Clock Generator Operation When Power Supply Voltage Is Turned On (When 1.59 V POC Mode Is Set (Option Byte: POCMODE = 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 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 **(2)** in **5.6.3 Example of controlling subsystem clock**).

Notes 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 2.7 V/1.59 V POC mode by using the option byte (POCMODE = 1) (see **Figure 5-14**). When a low level has been input to the $\overline{\text{RESET}}$ pin until the voltage reaches 1.8 V, the CPU operates with the same timing as <2> and thereafter in Figure 5-13, after the reset has been released by the $\overline{\text{RESET}}$ pin.

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

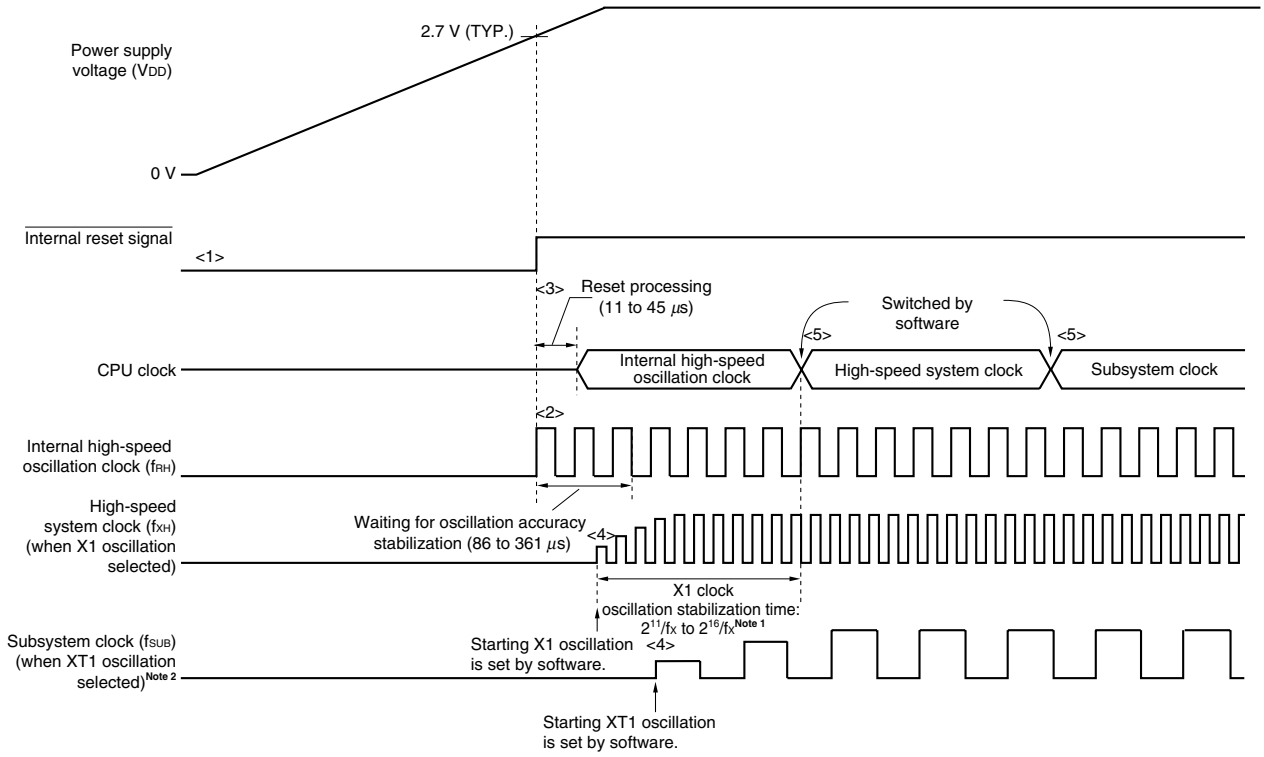
3. When releasing a reset (above figure) or releasing STOP mode while the CPU is operating on the internal high-speed oscillation clock, confirm the oscillation stabilization time for the X1 clock using the oscillation stabilization time counter status register (OSTC). If the CPU operates on the high-speed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

4. The 78K0/KB2-A is not provided with a subsystem clock.

Caution 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 **(3)** in **5.6.3 Example of controlling subsystem clock**).

Figure 5-14. Clock Generator Operation When Power Supply Voltage Is Turned On (When 2.7 V/1.59 V POC Mode Is Set (Option Byte: POCMODE = 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 2.7 V (TYP.), the reset is released and the internal high-speed oscillator automatically starts oscillation.

<3> After the reset is released and reset processing is performed, the CPU starts operation on the internal high-speed oscillation clock.

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

<5> When switching the CPU clock to the X1 or XT1 clock, wait for the clock oscillation to stabilize, and then set switching via software (see **(3)** in **5.6.1 Example of controlling high-speed system clock** and **(2)** in **5.6.3 Example of controlling subsystem clock**).

Notes 1. 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).

2. The 78K0/KB2-A is not provided with a subsystem clock.

Cautions 1. A voltage oscillation stabilization time of 1.93 to 5.39 ms is required after the supply voltage reaches 1.59 V (TYP.). If the supply voltage rises from 1.59 V (TYP.) to 2.7 V (TYP.) within 1.93 ms, the power supply oscillation stabilization time of 0 to 5.39 ms 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 **(3)** 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 across 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 I/O port pins.

Caution The X1/P121 and X2/EXCLK/P122 pins are in the I/O 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 clock and peripheral hardware clock
(4) When stopping high-speed system clock

(1) Example of setting procedure when oscillating the X1 clock

<1> Setting frequency (OSCCTL register)

Using AMPH, set the gain of the on-chip oscillator according to the frequency to be used.

AMPH[Note]	Operating Frequency Control
0	1 MHz ≤ f_{XH} ≤ 10 MHz
1	10 MHz < f_{XH} ≤ 20 MHz

Note Set AMPH before setting the peripheral functions after a reset release. The value of AMPH can be changed only once after a reset release. When AMPH is set to 1, the clock supply to the CPU is stopped for 4.06 to 16.12 μs.

Remark f_{XH}: High-speed system clock oscillation frequency

<2> Setting P121/X1 and P122/X2/EXCLK pins and selecting X1 clock or external clock (OSCCTL register)

When EXCLK is cleared to 0 and OSCSEL is set to 1, the mode is switched from port mode to X1 oscillation mode.

EXCLK	OSCSEL	Operation Mode of High-Speed System Clock Pin	P121/X1 Pin	P122/X2/EXCLK Pin
0	1	X1 oscillation mode	Crystal/ceramic resonator connection	

<3> Controlling oscillation of X1 clock (MOC register)

If MSTOP is cleared to 0, the X1 oscillator starts oscillating.

<4> 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. Do not change the value of EXCLK and OSCSEL while the X1 clock is operating.
2. Set the X1 clock after the supply voltage has reached the operable voltage of the clock to be used (see CHAPTER 28 ELECTRICAL SPECIFICATIONS).

(2) Example of setting procedure when using the external main system clock

<1> Setting frequency (OSCCTL register)
Using AMPH, set the frequency to be used.

AMPH[Note]	Operating Frequency Control
0	1 MHz ≤ f_{XH} ≤ 10 MHz
1	10 MHz < f_{XH} ≤ 20 MHz

Note Set AMPH before setting the peripheral functions after a reset release. The value of AMPH can be changed only once after a reset release. The clock supply to the CPU is stopped for the duration of 160 external clocks after AMPH is set to 1.

Remark f_{XH}: High-speed system clock oscillation frequency

<2> Setting P121/X1 and P122/X2/EXCLK pins and selecting operation mode (OSCCTL register)
When EXCLK and OSCSEL are set to 1, the mode is switched from port mode to external clock input mode.

EXCLK	OSCSEL	Operation Mode of High-Speed System Clock Pin	P121/X1 Pin	P122/X2/EXCLK Pin
1	1	External clock input mode	I/O port	External clock input

<3> Controlling external main system clock input (MOC register)
When MSTOP is cleared to 0, the input of the external main system clock is enabled.

Cautions 1. Do not change the value of EXCLK and OSCSEL while the external main system clock is operating.
2. Set the X1 clock after the supply voltage has reached the operable voltage of the clock to be used (see CHAPTER 28 ELECTRICAL SPECIFICATIONS).

(3) Example of setting procedure when using high-speed system clock as CPU clock and 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 main system clock (MCM register)

When XSEL and MCM0 are set to 1, the high-speed system clock is supplied as the main system clock and peripheral hardware clock.

XSEL	MCM0	Selection of Main System Clock and Clock Supplied to Peripheral Hardware	
		Main System Clock (f_{XP})	Peripheral Hardware Clock (f_{PRS})
1	1	High-speed system clock (f_{XH})	High-speed system clock (f_{XH})

Caution If the high-speed system clock is selected as the main system clock, a clock other than the high-speed system clock cannot be set as the peripheral hardware clock.

<3> Setting the main system clock as the CPU clock and selecting the division ratio (PCC register)

When CSS is cleared to 0, the main system clock is supplied to the CPU. To select the CPU clock division ratio, use PCC0, PCC1, and PCC2.

CSS	PCC2	PCC1	PCC0	CPU Clock (f_{CPU}) Selection
0	0	0	0	f_{XP}
	0	0	1	$f_{XP}/2$ (default)
	0	1	0	$f_{XP}/2^2$
	0	1	1	$f_{XP}/2^3$
	1	0	0	$f_{XP}/2^4$
	Other than above			Setting prohibited

(4) Example of setting procedure when stopping the high-speed system clock

The high-speed system clock can be stopped in the following two ways.

- Executing the STOP instruction and stopping the X1 oscillation (disabling clock input if the external clock is used)
- Setting MSTOP to 1 and stopping the X1 oscillation (disabling clock input if the external clock is used)

(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 20 STANDBY FUNCTION**).

<2> Setting the X1 clock oscillation stabilization time after standby release

When the CPU is operating on the X1 clock, set the value of the OSTS register before the STOP instruction is executed.

<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 (PCC and MCM registers)

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 a clock other than the high-speed system clock.

• 78K0/KB2-A

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

• 78K0/KC2-A

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

<2> Stopping the high-speed system clock (MOC register)

When MSTOP is set to 1, X1 oscillation is stopped (the input of the external clock is disabled).

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 clock, and internal high-speed oscillation clock or high-speed system clock as 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]

<1> Setting restart of oscillation of the internal high-speed oscillation clock (RCM register)
When RSTOP is cleared to 0, the internal high-speed oscillation clock starts operating.

<2> Waiting for the oscillation accuracy stabilization time of internal high-speed oscillation clock (RCM register)
Wait until RSTS is set to 1[Note 2].

Notes 1. After a reset release, the internal high-speed oscillator automatically starts oscillating and the internal high-speed oscillation clock is selected as the CPU clock.

2. This wait time is not necessary if high accuracy is not necessary for the CPU clock and peripheral hardware clock.

(2) Example of setting procedure when using internal high-speed oscillation clock as CPU clock, and internal high-speed oscillation clock or high-speed system clock as 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 oscillation of the internal high-speed oscillation clock**).
- Oscillating the high-speed system clock[Note]
(This setting is required when using the high-speed system clock as the peripheral hardware clock. 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 the internal high-speed oscillation clock or high-speed system clock is already operating.

<2> Selecting the clock supplied as the main system clock and peripheral hardware clock (MCM register)
Set the main system clock and peripheral hardware clock using XSEL and MCM0.

XSEL	MCM0	Selection of Main System Clock and Clock Supplied to Peripheral Hardware	
		Main System Clock (f_{XP})	Peripheral Hardware Clock (f_{PRS})
0	0	Internal high-speed oscillation clock (f_{RH})	Internal high-speed oscillation clock (f_{RH})
0	1		
1	0		High-speed system clock (f_{XH})

<3> Selecting the CPU clock division ratio (PCC register)
When CSS is cleared to 0, the main system clock is supplied to the CPU. To select the CPU clock division ratio, use PCC0, PCC1, and PCC2.

CSS	PCC2	PCC1	PCC0	CPU Clock (f_{CPU}) Selection
0	0	0	0	f_{XP}
	0	0	1	$f_{XP}/2$ (default)
	0	1	0	$f_{XP}/2^2$
	0	1	1	$f_{XP}/2^3$
	1	0	0	$f_{XP}/2^4$
	Other than above			Setting prohibited

(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 to set the STOP mode
- Setting RSTOP to 1 and stopping the internal high-speed oscillation clock

(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 20 STANDBY FUNCTION**).

<2> Setting the X1 clock oscillation stabilization time after standby release

When the CPU is operating on the X1 clock, set the value of the OSTS register before the STOP instruction is executed. To operate the CPU immediately after the STOP mode has been released, set MCM0 to 0, switch the CPU clock to the internal high-speed oscillation clock, and check that RSTS is 1.

<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 RSTOP to 1

<1> Confirming the CPU clock status (PCC and MCM registers)

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 a clock other than the internal high-speed oscillation clock.

- 78K0/KB2-A

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

- 78K0/KC2-A

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 (RCM register)

When RSTOP is set to 1, internal high-speed oscillation clock is stopped.

Caution Be sure to confirm that MCS = 1 or CLS = 1 when setting RSTOP 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 following type of subsystem clock[Note] is available.

- XT1 clock: Crystal/ceramic resonator is connected across the XT1 and XT2 pins.

When the subsystem clock is not used, the XT1/P123 and XT2/P124 pins can be used as I/O port pins.

Note The 78K0/KB2-A is not provided with a subsystem clock.

Cautions 1. The XT1/P123 and XT2/P124 pins are in the I/O port mode after a reset release.

2. Do not start the peripheral hardware operation with the external clock from peripheral hardware pins when the internal high-speed oscillation clock and high-speed system clock are stopped while the CPU operates with the subsystem clock, or when in the STOP mode.

The following describes examples of setting procedures for the following cases.

(1) When oscillating XT1 clock

(2) When using subsystem clock as CPU clock

(3) When stopping subsystem clock

(1) Example of setting procedure when oscillating the XT1 clock

<1> Setting XT1 and XT2 pins and selecting operation mode (OSCCTL register)
When OSCSELS is set to 1, the mode is switched from port mode to XT1 oscillation mode.

<2> 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.

(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 XT1 clock**.)

Note The setting of <1> is not necessary when while the subsystem clock is operating.

<2> Switching the CPU clock (PCC register)
When CSS is set to 1, the subsystem clock is supplied to the CPU.

CSS	PCC2	PCC1	PCC0	CPU Clock (f_{CPU}) Selection
1	0	0	0	$f_{SUB}/2$
	0	0	1	
	0	1	0	
	0	1	1	
	1	0	0	
	Other than above			Setting prohibited

(3) Example of setting procedure when stopping the subsystem clock

<1> Confirming the CPU clock status (PCC and MCM registers)

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 a clock other than the 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 subsystem clock (OSCCTL register)

When OSCSELS is cleared to 0, XT1 oscillation is stopped.

Cautions 1. Be sure to confirm that CLS = 0 when clearing OSCSELS to 0. In addition, stop the real-time counter and clock output controller 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.

Only the following peripheral hardware can operate with this clock.

- Watchdog timer
- 8-bit timer H1 (if f_{RL} is selected as the count clock)

In addition, the following operation modes can be selected by the option byte.

- Internal low-speed oscillator cannot be stopped
- Internal low-speed oscillator can be stopped by software

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 has been enabled by the option byte.

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

<1> Setting LSRSTOP to 1 (RCM register)

When LSRSTOP is set to 1, the internal low-speed oscillation clock is stopped.

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

<1> Clearing LSRSTOP to 0 (RCM register)

When LSRSTOP is cleared to 0, the internal low-speed oscillation clock is restarted.

Caution If "Internal low-speed oscillator cannot be stopped" is selected by the option byte, oscillation of the internal low-speed oscillation clock cannot be controlled.

5.6.5 Clocks supplied to CPU and peripheral hardware

The following table shows the relation among the clocks supplied to the CPU and peripheral hardware, and setting of registers.

Table 5-3. Clocks Supplied to CPU and Peripheral Hardware, and Register Setting (78K0/KB2-A)

Supplied Clock		XSEL	MCM0	EXCLK
Clock Supplied to CPU	Clock Supplied to Peripheral Hardware			
Internal high-speed oscillation clock		0	×	×
Internal high-speed oscillation clock	X1 clock	1	0	0
	External main system clock	1	0	1
X1 clock		1	1	0
External main system clock		1	1	1

Remarks XSEL: Bit 2 of the main clock mode register (MCM)
CSS: Bit 4 of processor clock control register (PCC)
MCM0: Bit 0 of MCM
EXCLK: Bit 7 of the clock operation mode select register (OSCCTL)
×: don't care

Table 5-4. Clocks Supplied to CPU and Peripheral Hardware, and Register Setting (78K0/KC2-A)

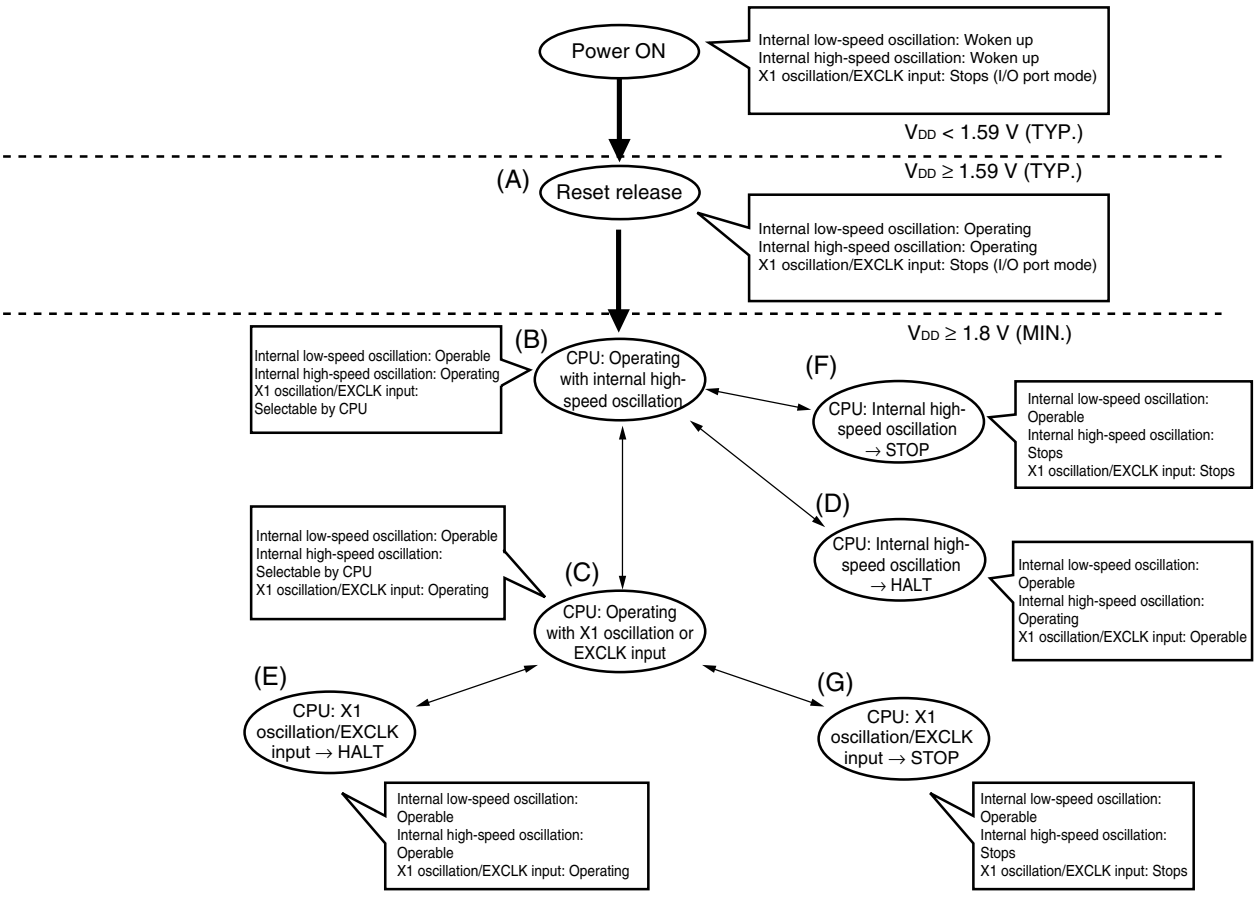
Supplied Clock		XSEL	CSS	MCM0	EXCLK
Clock Supplied to CPU	Clock Supplied to Peripheral Hardware				
Internal high-speed oscillation clock		0	0	×	×
Internal high-speed oscillation clock	X1 clock	1	0	0	0
	External main system clock	1	0	0	1
X1 clock		1	0	1	0
External main system clock		1	0	1	1
Subsystem clock	Internal high-speed oscillation clock	0	1	×	×
	X1 clock	1	1	0	0
		1	1	1	0
	External main system clock	1	1	0	1
		1	1	1	1

Remark XSEL: Bit 2 of the main clock mode register (MCM)
CSS: Bit 4 of the processor clock control register (PCC)
MCM0: Bit 0 of MCM
EXCLK: Bit 7 of the clock operation mode select register (OSCCTL)
×: don't care

5.6.6 CPU clock status transition diagram

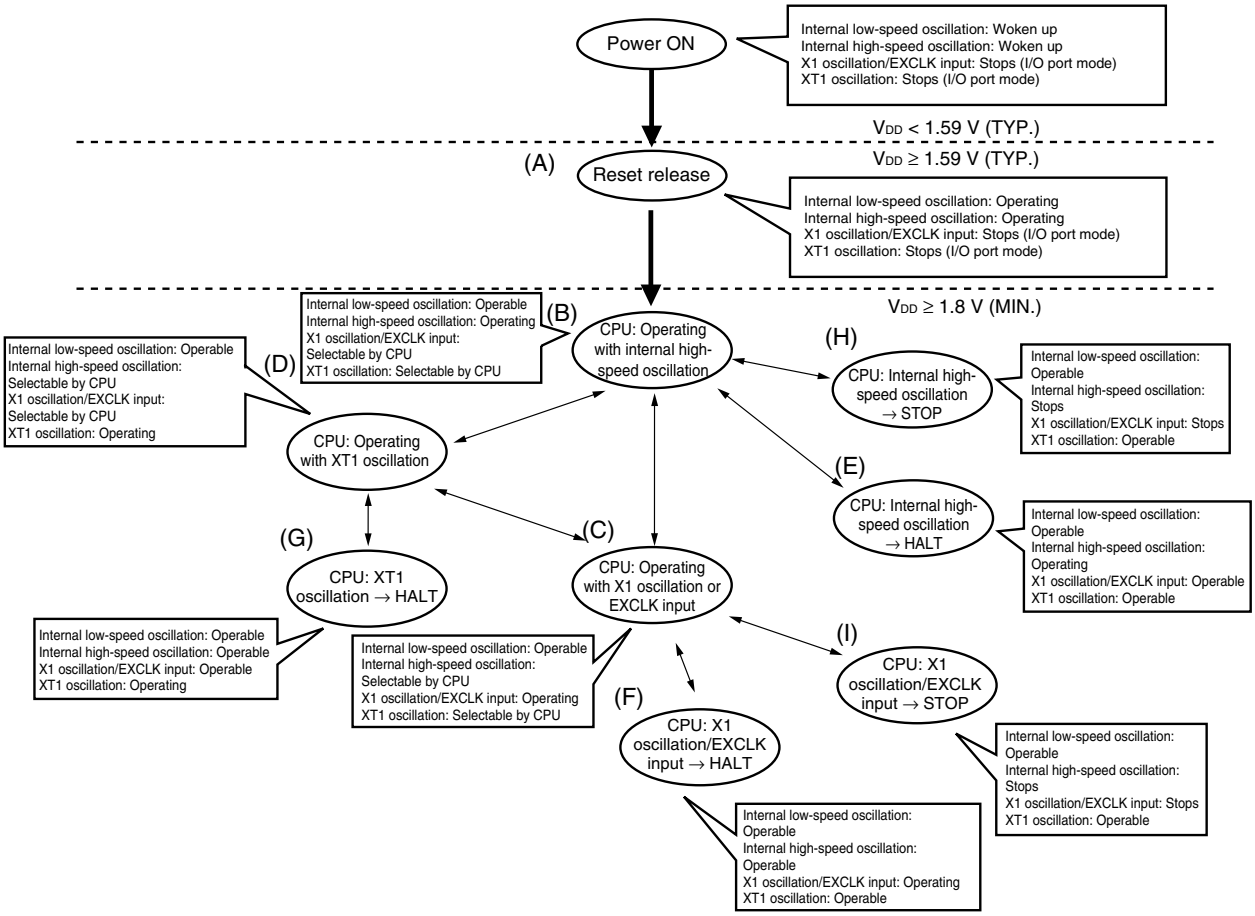
Figures 5-15 and 5-16 show the CPU clock status transition diagrams of this product.

Figure 5-15. CPU Clock Status Transition Diagram
(When 1.59 V POC Mode Is Set (Option Byte: POCMODE = 0), 78K0/KB2-A)



Remark In the 2.7 V/1.59 V POC mode (option byte: POCMODE = 1), the CPU clock status changes to (A) in the above figure when the supply voltage exceeds 2.7 V (TYP.), and to (B) after reset processing (11 to 45 μs).

Figure 5-16. CPU Clock Status Transition Diagram
(When 1.59 V POC Mode Is Set (Option Byte: POCMODE = 0), 78K0/KC2-A)



Remark In the 2.7 V/1.59 V POC mode (option byte: POCMODE = 1), the CPU clock status changes to (A) in the above figure when the supply voltage exceeds 2.7 V (TYP.), and to (B) after reset processing (11 to 45 μs).

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

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

(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	AMPH	EXCLK	OSCSEL	MSTOP	OSTC Register	XSEL	MCM0
(A) → (B) → (C) (X1 clock: 1 MHz ≤ f_{XH} ≤ 10 MHz)	0	0	1	0	Must be checked	1	1
(A) → (B) → (C) (external main clock: 1 MHz ≤ f_{XH} ≤ 10 MHz)	0	1	1	0	Must not be checked	1	1
(A) → (B) → (C) (X1 clock: 10 MHz < f_{XH} ≤ 20 MHz)	1	0	1	0	Must be checked	1	1
(A) → (B) → (C) (external main clock: 10 MHz < f_{XH} ≤ 20 MHz)	1	1	1	0	Must not be checked	1	1

Caution Set the X1 clock after the supply voltage has reached the operable voltage of the clock to be used (see CHAPTER 28 ELECTRICAL SPECIFICATIONS).

(3) CPU operating with subsystem clock (D) after reset release (A)[Note]

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

Note The 78K0/KB2-A is not provided with a subsystem clock.

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	OSCSELS	Waiting for Oscillation Stabilization	CSS
(A) → (B) → (D)	1	Necessary	1

Remarks 1. (A) to (I) in Table 5-5 correspond to (A) to (I) in Figures 5-15 and 5-16.

2. EXCLK, OSCSEL, OSCSELS, AMPH: Bits 7, 6, 4 and 0 of the clock operation mode select register (OSCCTL)
MSTOP: Bit 7 of the main OSC control register (MOC)
XSEL, MCM0: Bits 2 and 0 of the main clock mode register (MCM)
CSS: Bit 4 of the processor clock control register (PCC)

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

(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	AMPH[Note]	EXCLK	OSCSEL	MSTOP	OSTC Register	XSEL[Note]	MCM0
(B) → (C) (X1 clock: 1 MHz ≤ f_{XH} ≤ 10 MHz)	0	0	1	0	Must be checked	1	1
(B) → (C) (external main clock: 1 MHz ≤ f_{XH} ≤ 10 MHz)	0	1	1	0	Must not be checked	1	1
(B) → (C) (X1 clock: 10 MHz < f_{XH} ≤ 20 MHz)	1	0	1	0	Must be checked	1	1
(B) → (C) (external main clock: 10 MHz < f_{XH} ≤ 20 MHz)	1	1	1	0	Must not be checked	1	1

Unnecessary if these registers are already set

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

Note The value of this flag can be changed only once after a reset release. This setting is not necessary if it has already been set.

Caution Set the X1 clock after the supply voltage has reached the operable voltage of the clock to be used (see CHAPTER 28 ELECTRICAL SPECIFICATIONS).

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

Note The 78K0/KB2-A is not provided with a subsystem clock.

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	OSCSELS	Waiting for Oscillation Stabilization	CSS
(B) → (D)	1	Necessary	1

Unnecessary if the CPU is operating with the subsystem clock

Remarks 1. (A) to (I) in Table 5-5 correspond to (A) to (I) in Figures 5-15 and 5-16.

2. EXCLK, OSCSEL, OSCSELS, AMPH: Bits 7, 6, 4 and 0 of the clock operation mode select register (OSCCTL)
MSTOP: Bit 7 of the main OSC control register (MOC)
XSEL, MCM0: Bits 2 and 0 of the main clock mode register (MCM)
CSS: Bit 4 of the processor clock control register (PCC)

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

(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	RSTOP	RSTS	MCM0
(C) → (B)	0	Confirm this flag is 1.	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)[Note]

Note The 78K0/KB2-A is not provided with a subsystem clock.

(Setting sequence of SFR registers) →

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

Unnecessary if the CPU is operating with the subsystem clock

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

Note The 78K0/KB2-A is not provided with a subsystem clock.

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	RSTOP	RSTS	MCM0	CSS
(D) → (B)	0	Confirm this flag is 1.	0	0

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

↑ Unnecessary if XSEL is 0

Remarks 1. (A) to (I) in Table 5-5 correspond to (A) to (I) in Figures 5-15 and 5-16.

2. MCM0: Bit 0 of the main clock mode register (MCM)
OSCSELS: Bit 4 of the clock operation mode select register (OSCCTL)
RSTS, RSTOP: Bits 7 and 0 of the internal oscillation mode register (RCM)
CSS: Bit 4 of the processor clock control register (PCC)

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

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

Note The 78K0/KB2-A is not provided with a subsystem clock.

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	AMPH[Note]	EXCLK	OSCSEL	MSTOP	OSTC Register	XSEL[Note]	MCM0	CSS
(D) → (C) (X1 clock: 1 MHz ≤ f_{XH} ≤ 10 MHz)	0	0	1	0	Must be checked	1	1	0
(D) → (C) (external main clock: 1 MHz ≤ f_{XH} ≤ 10 MHz	0	1	1	0	Must not be checked	1	1	0
(D) → (C) (X1 clock: 10 MHz < f_{XH} ≤ 20 MHz)	1	0	1	0	Must be checked	1	1	0
(D) → (C) (external main clock: 10 MHz < f_{XH} ≤ 20 MHz)	1	1	1	0	Must not be checked	1	1	0

Unnecessary if these registers are already set

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

Unnecessary if this register is already set

Note The value of this flag can be changed only once after a reset release. This setting is not necessary if it has already been set.

Caution Set the X1 clock after the supply voltage has reached the operable voltage of the clock to be used (see CHAPTER 28 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)[Note]

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

Note The 78K0/KB2-A is not provided with a subsystem clock.

Remarks 1. (A) to (I) in Table 5-5 correspond to (A) to (I) in Figures 5-15 and 5-16.

2. EXCLK, OSCSEL, AMPH: Bits 7, 6, and 0 of the clock operation mode select register (OSCCTL)
MSTOP: Bit 7 of the main OSC control register (MOC)
XSEL, MCM0: Bits 2 and 0 of the main clock mode register (MCM)
CSS: Bit 4 of the processor clock control register (PCC)

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

(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) (C) → (I)	Stopping peripheral functions that cannot operate in STOP mode	Executing STOP instruction

Remarks 1. (A) to (I) in Table 5-5 correspond to (A) to (I) in Figures 5-15 and 5-16.

2. EXCLK, OSCSEL, AMPH: Bits 7, 6 and 0 of the clock operation mode select register (OSCCTL)
MSTOP: Bit 7 of the main OSC control register (MOC)
XSEL, MCM0: Bits 2 and 0 of the main clock mode register (MCM)
CSS: Bit 4 of the processor clock control register (PCC)

5.6.7 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 5-6. Changing CPU Clock (1/2)

(1) 78K0/KB2-A

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
Internal high-speed oscillation clock	X1 clock	Stabilization of X1 oscillation • MSTOP = 0, OSCSEL = 1, EXCLK = 0 • After elapse of oscillation stabilization time	• Internal high-speed oscillator can be stopped (RSTOP = 1). • Clock supply to CPU is stopped for 4.06 to 16.12 μs after AMPH has been set to 1.
	External main system clock	Enabling input of external clock from EXCLK pin • MSTOP = 0, OSCSEL = 1, EXCLK = 1	• Internal high-speed oscillator can be stopped (RSTOP = 1). • Clock supply to CPU is stopped for the duration of 160 external clocks from the EXCLK pin after AMPH has been set to 1.
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			External main system clock input can be disabled (MSTOP = 1).

Table 5-6. Changing CPU Clock (2/2)

(2) 78K0/KC2-A

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
Internal high-speed oscillation clock	X1 clock	Stabilization of X1 oscillation • MSTOP = 0, OSCSEL = 1, EXCLK = 0 • After elapse of oscillation stabilization time	• Internal high-speed oscillator can be stopped (RSTOP = 1). • Clock supply to CPU is stopped for 4.06 to 16.12 μs after AMPH has been set to 1.
	External main system clock	Enabling input of external clock from EXCLK pin • MSTOP = 0, OSCSEL = 1, EXCLK = 1	• Internal high-speed oscillator can be stopped (RSTOP = 1). • Clock supply to CPU is stopped for the duration of 160 external clocks from the EXCLK pin after AMPH has been set to 1.
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			External main system clock input can be disabled (MSTOP = 1).
Internal high-speed oscillation clock	XT1 clock	Stabilization of XT1 oscillation • OSCSELS = 1 • After elapse of oscillation stabilization time	Operating current can be reduced by stopping internal high-speed oscillator (RSTOP = 1).
X1 clock			X1 oscillation can be stopped (MSTOP = 1).
External main system clock			External main system clock input can be disabled (MSTOP = 1).
XT1 clock, external 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 • RSTOP = 0, MCS = 0	XT1 oscillation can be stopped (OSCSELS = 0).
	X1 clock	Stabilization of X1 oscillation and selection of high-speed system clock as main system clock • MSTOP = 0, OSCSEL = 1, EXCLK = 0 • After elapse of oscillation stabilization time • MCS = 1	• XT1 oscillation can be stopped (OSCSELS = 0). • Clock supply to CPU is stopped for 4.06 to 16.12 μs after AMPH has been set to 1.
	External main system clock	Enabling input of external clock from EXCLK pin and selection of high-speed system clock as main system clock • MSTOP = 0, OSCSEL = 1, EXCLK = 1 • MCS = 1	• XT1 oscillation can be stopped (OSCSELS = 0). • Clock supply to CPU is stopped for the duration of 160 external clocks from the EXCLK pin after AMPH has been set to 1.

5.6.8 Time required for switchover of CPU clock and main system clock

By setting bits 0 to 2 (PCC0 to PCC2) and bit 4 (CSS) of the processor clock control register (PCC), the CPU clock can be switched (between the main system clock and the subsystem clock) and the division ratio of the main system clock can be changed.

The actual switchover operation is not performed immediately after rewriting to PCC; operation continues on the pre-switchover clock for several clocks (see **Tables 5-7** and **5-8**).

Whether the CPU is operating on the main system clock or the subsystem clock[Note] can be ascertained using bit 5 (CLS) of the PCC register.

Note The 78K0/KB2-A is not provided with a subsystem clock.

Table 5-7. Time Required for Switchover of CPU Clock and Main System Clock Cycle Division Factor (78K0/KB2-A)

Set Value Before Switchover			Set Value After Switchover															
PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0
			0	0	0	0	0	1	0	1	0	0	1	1	1	0	0
0	0	0				16 clocks			16 clocks			16 clocks			16 clocks		
0	0	1	8 clocks						8 clocks			8 clocks			8 clocks		
0	1	0	4 clocks			4 clocks						4 clocks			4 clocks		
0	1	1	2 clocks			2 clocks			2 clocks						2 clocks		
1	0	0	1 clock			1 clock			1 clock			1 clock					

Remark The number of clocks listed in Table 5-7 is the number of CPU clocks before switchover.

Table 5-8. Time Required for Switchover of CPU Clock and Main System Clock Cycle Division Factor (78K0/KC2-A)

Set Value Before Switchover				Set Value After Switchover																									
CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0
				0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	1	×	×	×
0	0	0	0					16 clocks				16 clocks				16 clocks				16 clocks				$2f_{XP}/f_{SUB}$ clocks			
	0	0	1	8 clocks								8 clocks				8 clocks				8 clocks				f_{XP}/f_{SUB} clocks			
	0	1	0	4 clocks				4 clocks								4 clocks				4 clocks				$f_{XP}/2f_{SUB}$ clocks			
	0	1	1	2 clocks				2 clocks				2 clocks								2 clocks				$f_{XP}/4f_{SUB}$ clocks			
	1	0	0	1 clock				1 clock				1 clock				1 clock								$f_{XP}/8f_{SUB}$ clocks			
1	×	×	×	2 clocks				2 clocks				2 clocks				2 clocks				2 clocks							

Caution Selection of the main system clock cycle division factor (PCC0 to PCC2) and switchover from the main system clock to the subsystem clock (changing CSS from 0 to 1) should not be set simultaneously.

Simultaneous setting is possible, however, for selection of the main system clock cycle division factor (PCC0 to PCC2) and switchover from the subsystem clock to the main system clock (changing CSS from 1 to 0).

Remark 1. The number of clocks listed in Table 5-8 is the number of CPU clocks before switchover.

Remark 2. When switching the CPU clock from the main system clock to the subsystem clock, calculate the number of clocks by rounding up to the next clock and discarding the decimal portion, as shown below.

Example When switching CPU clock from $f_{XP}/2$ to $f_{SUB}/2$ (@ oscillation with f_{XP} = 10 MHz, f_{SUB} = 32.768 kHz)

$f_{XP}/f_{SUB} = 10000/32.768 \cong 305.1 \rightarrow 306$ clocks

By setting bit 0 (MCM0) of the main clock mode register (MCM), the main system clock can be switched (between the internal high-speed oscillation clock and the high-speed system clock).

The actual switchover operation is not performed immediately after rewriting to MCM0; operation continues on the pre-switchover clock for several clocks (see **Table 5-9**).

Whether the CPU is operating on the internal high-speed oscillation clock or the high-speed system clock can be ascertained using bit 1 (MCS) of MCM.

Table 5-9. Maximum Time Required for Main System Clock Switchover

Set Value Before Switchover	Set Value After Switchover	
MCM0	MCM0	
	0	1
0		1 + $2f_{RH}/f_{XH}$ clock
1	1 + $2f_{XH}/f_{RH}$ clock	

Cautions 1. When switching the internal high-speed oscillation clock to the high-speed system clock, bit 2 (XSEL) of MCM must be set to 1 in advance. The value of XSEL can be changed only once after a reset release.

2. Do not rewrite MCM0 when the CPU clock operates with the subsystem clock.

Remarks 1. The number of clocks listed in Table 5-9 is the number of main system clocks before switchover.

2. Calculate the number of clocks in 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_{RH} = 8 MHz, f_{XH} = 10 MHz)

$1 + 2f_{RH}/f_{XH} = 1 + 2 \times 8/10 = 1 + 2 \times 0.8 = 1 + 1.6 = 2.6 \rightarrow 2$ clocks

5.6.9 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 (78K0/KB2-A)

Clock[Note]	Conditions Before Clock Oscillation Is Stopped (External Clock Input Disabled)	Flag Settings of SFR Register
Internal high-speed oscillation clock	MCS = 1 (The CPU is operating on the high-speed system clock)	RSTOP = 1
X1 clock	MCS = 0 (The CPU is operating on the internal high-speed oscillation clock)	MSTOP = 1
External main system clock		

Table 5-11. Conditions Before the Clock Oscillation Is Stopped and Flag Settings (78K0/KC2-A)

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)	RSTOP = 1
X1 clock	MCS = 0 or CLS = 1 (The CPU is operating on a clock other than the high-speed system clock)	MSTOP = 1
External main system clock		
XT1 clock	CLS = 0 (The CPU is operating on a clock other than the subsystem clock)	OSCSELS = 0

5.6.10 Peripheral hardware and source clocks

The following lists peripheral hardware and source clocks incorporated in the 78K0/Kx2-A microcontrollers.

Remark The peripheral hardware depends on the product. See **1.5 Block Diagram** and **1.6 Outline of Functions**.

Table 5-12. Peripheral Hardware and Source Clocks

Peripheral Hardware \ Source Clock		CPU Clock (f_{CPU})	Peripheral Hardware Clock (f_{PRS})	Subsystem Clock (f_{SUB})[Note 1]	Internal Low-Speed Oscillation Clock (f_{RL})	TM50 Output	External Clock from Peripheral Hardware Pins
16-bit timer/ event counter 00		N	Y	N	N	N	Y (TI000 pin)[Note 2]
8-bit timer/ event counter	50	N	Y	N	N	N	Y (TI50 pin)[Note 2]
	51	N	Y	N	N	N	Y (TI51 pin)[Note 2]
8-bit timer	H0	N	Y	N	N	Y	N
	H1	N	Y	N	Y	N	N
Real-time counter		N	N	Y	N	N	N
Watchdog timer		N	N	N	Y	N	N
Clock output		N	Y	Y	N	N	N
A/D converter		Y	N	N	N	N	N
Operational amplifier		N	Y	N	N	N	N
Serial interface	UART6	N	Y	N	N	Y	N
	CSI10	N	Y	N	N	N	Y ($\overline{SCK10}$ pin)[Note 2]
	IICA	N	Y	N	N	N	Y (SCLA0 pin)[Note 2]

Notes 1. The 78K0/KB2-A is not provided with a subsystem clock.

2. Do not start the peripheral hardware operation with the external clock from peripheral hardware pins when the internal high-speed oscillation clock and high-speed system clock are stopped while the CPU operates with the subsystem clock, or when in the STOP mode.

Remark Y: Can be selected, N: Cannot be selected

CHAPTER 6 16-BIT TIMER/EVENT COUNTER 00

6.1 Functions of 16-Bit Timer/Event Counter 00

16-bit timer/event counters 00 is mounted onto all 78K0/Kx2-A microcontrollers.

16-bit timer/event counter 00 has the following functions.

(1) Interval timer

16-bit timer/event counter 00 generates an interrupt request at the preset time interval.

(2) Square-wave output

16-bit timer/event counter 00 can output a square wave with any selected frequency.

(3) External event counter

16-bit timer/event counter 00 can measure the number of pulses of an externally input signal.

(4) One-shot pulse output

16-bit timer event counter 00 can output a one-shot pulse whose output pulse width can be set freely.

(5) PPG output

16-bit timer/event counter 00 can output a rectangular wave whose frequency and output pulse width can be set freely.

(6) Pulse width measurement

16-bit timer/event counter 00 can measure the pulse width of an externally input signal.

6.2 Configuration of 16-Bit Timer/Event Counter 00

16-bit timer/event counter 00 includes the following hardware.

Table 6-1. Configuration of 16-Bit Timer/Event Counter 00

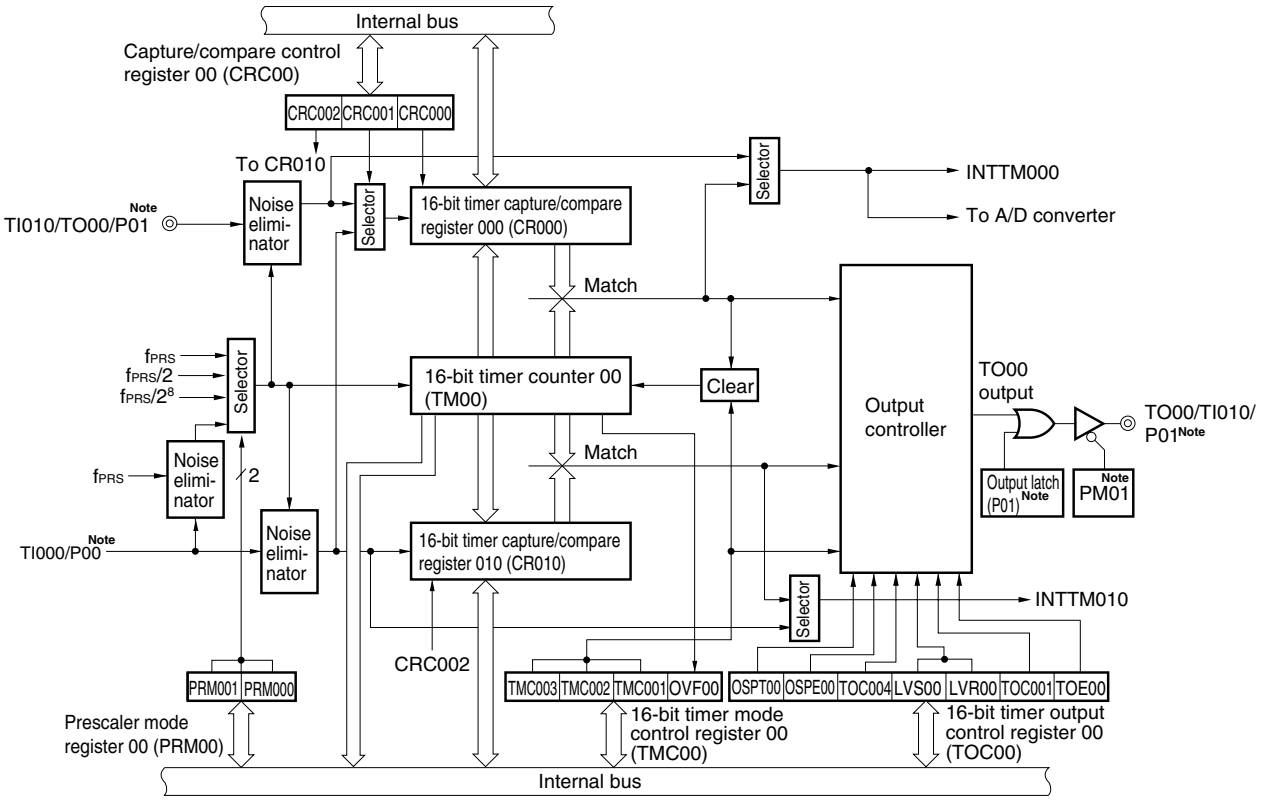
Item	Configuration
Time/counter	16-bit timer counter 00 (TM00)
Register	16-bit timer capture/compare registers 000, 010 (CR000, CR010)
Timer input	TI000, TI010 pins[Note]
Timer output	TO00 pin[Note], output controller
Control registers	16-bit timer mode control register 00 (TMC00) Capture/compare control register 00 (CRC00) 16-bit timer output control register 00 (TOC00) Prescaler mode register 00 (PRM00) Input switch control register (ISC) Port mode register 0, 1 (PM0, PM1)[Note] Port register 0, 1 (P0, P1)[Note]

Note The port pins with which the I/O pins for 16-bit timer/event counter 00 are shared differ depending on the product as follows:

- 78K0/KB2-A: Shared with the pins of port 1
- 78K0/KC2-A: Shared with the pins of port 0

Figures 6-1 shows the block diagrams.

Figure 6-1. Block Diagram of 16-Bit Timer/Event Counter 00



Note 78K0/KB2-A: TI000/TOH0/INTP7/P12, TI010/TO00/TOH1/INTP6/P13, output latch (P13), PM13
78K0/KB2-A: TI000/P00, TI010/TO00/P01, output latch (P01), PM01

Cautions 1. In case of 78K0/KB2-A, The valid edge of TI010 and timer output (TO00) cannot be used for the P13 pin at the same time. Select either of the functions.

2. In case of 78K0/KC2-A, The valid edge of TI010 and timer output (TO00) cannot be used for the P01 pin at the same time. Select either of the functions.

3. If clearing of bits 3 and 2 (TMC003 and TMC002) of 16-bit timer mode control register 00 (TMC00) to 00 and input of the capture trigger conflict, then the captured data is undefined.

4. To change the mode from the capture mode to the comparison mode, first clear the TMC003 and TMC002 bits to 00, and then change the setting.
A value that has been once captured remains stored in CR000 unless the device is reset. If the mode has been changed to the comparison mode, be sure to set a comparison value.

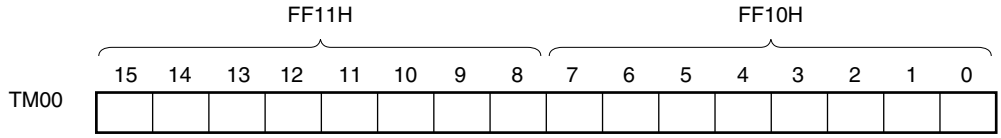
(1) 16-bit timer counter 00 (TM00)

TM00 is a 16-bit read-only register that counts count pulses.

The counter is incremented in synchronization with the rising edge of the count clock.

Figure 6-2. Format of 16-Bit Timer Counter 00 (TM00)

Address: FF10H, FF11H After reset: 0000H R



The count value of TM00 can be read by reading TM00 when the value of bits 3 and 2 (TMC003 and TMC002) of 16-bit timer mode control register 00 (TMC00) is other than 00. The value of TM00 is 0000H if it is read when TMC003 and TMC002 = 00.

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

- At reset signal generation
- If TMC003 and TMC002 are cleared to 00
- If the valid edge of the TI000 pin is input in the mode in which the clear & start occurs when inputting the valid edge to the TI000 pin
- If TM00 and CR000 match in the mode in which the clear & start occurs when TM00 and CR000 match
- OSPT00 is set to 1 in one-shot pulse output mode or the valid edge is input to the TI000 pin

Caution Even if TM00 is read, the value is not captured by CR010.

(2) 16-bit timer capture/compare register 000 (CR000), 16-bit timer capture/compare register 010 (CR010)

CR000 and CR010 are 16-bit registers that are used with a capture function or comparison function selected by using CRC00.

Change the value of CR000 while the timer is stopped (TMC003 and TMC002 = 00).

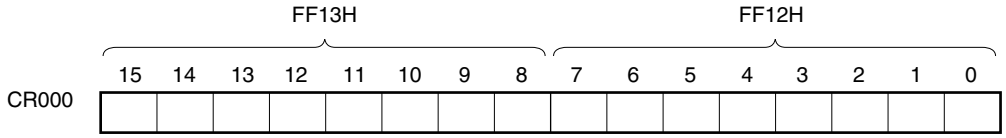
The value of CR010 can be changed during operation if the value has been set in a specific way. For details, see **6.5.1 Rewriting CR010 during TM00 operation**.

These registers can be read or written in 16-bit units.

Reset signal generation clears these registers to 0000H.

Figure 6-3. Format of 16-Bit Timer Capture/Compare Register 000 (CR000)

Address: FF12H, FF13H After reset: 0000H R/W



(i) When CR000 is used as a compare register

The value set in CR000 is constantly compared with the TM00 count value, and an interrupt request signal (INTTM000) is generated if they match. The value is held until CR000 is rewritten.

Caution CR000 does not perform the capture operation when it is set in the comparison mode, even if a capture trigger is input to it.

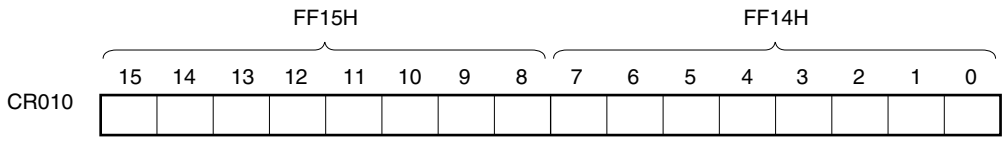
(ii) When CR000 is used as a capture register

The count value of TM00 is captured to CR000 when a capture trigger is input.

As the capture trigger, an edge of a phase reverse to that of the TI000 pin or the valid edge of the TI010 pin can be selected by using CRC00 or PRM00.

Figure 6-4. Format of 16-Bit Timer Capture/Compare Register 010 (CR010)

Address: FF14H, FF15H After reset: 0000H R/W



(i) When CR010 is used as a compare register

The value set in CR010 is constantly compared with the TM00 count value, and an interrupt request signal (INTTM010) is generated if they match.

Caution CR010 does not perform the capture operation when it is set in the comparison mode, even if a capture trigger is input to it.

(ii) When CR010 is used as a capture register

The count value of TM00 is captured to CR010 when a capture trigger is input.

It is possible to select the valid edge of the TI000 pin as the capture trigger. The TI000 pin valid edge is set by PRM00.

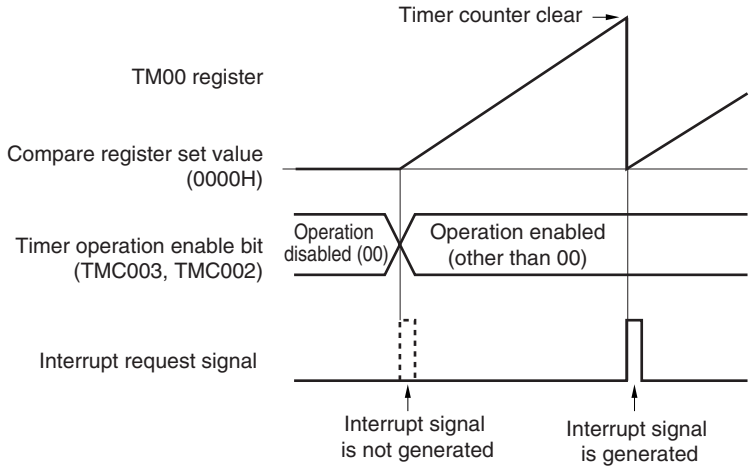
(iii) Setting range when CR000 or CR010 is used as a compare register

When CR000 or CR010 is used as a compare register, set it as shown below.

Operation	CR000 Register Setting Range	CR010 Register Setting Range
Operation as interval timer	0000H < N ≤ FFFFH	0000H[Note] ≤ M ≤ FFFFH Normally, this setting is not used. Mask the match interrupt signal (INTTM010).
Operation as square-wave output		
Operation as external event counter		
Operation in the clear & start mode entered by TI000 pin valid edge input	0000H[Note] ≤ N ≤ FFFFH	0000H[Note] ≤ M ≤ FFFFH
Operation as free-running timer		
Operation as PPG output	M < N ≤ FFFFH	0000H[Note] ≤ M < N
Operation as one-shot pulse output	0000H[Note] ≤ N ≤ FFFFH (N ≠ M)	0000H[Note] ≤ M ≤ FFFFH (M ≠ N)

Note When 0000H is set, a match interrupt immediately after the timer operation does not occur and timer output is not changed, and the first match timing is as follows. A match interrupt occurs at the timing when the timer counter (TM00 register) is changed from 0000H to 0001H.

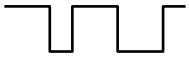
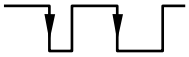
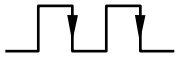
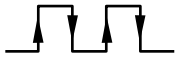
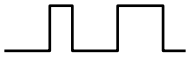
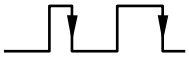
- When the timer counter is cleared due to overflow
- When the timer counter is cleared due to TI000 pin valid edge (when clear & start mode is entered by TI000 pin valid edge input)
- When the timer counter is cleared due to compare match (when clear & start mode is entered by match between TM00 and CR000 (CR000 = other than 0000H, CR010 = 0000H))



Remarks 1. N: CR000 register set value, M: CR010 register set value

2. For details of TMC003 and TMC002, see **6.3 (1) 16-bit timer mode control register 00 (TMC00)**.

Table 6-2. Capture Operation of CR000 and CR010

External Input Signal / Capture Operation	TI000 Pin Input		TI010 Pin Input	
Capture operation of CR000	CRC001 = 1 TI000 pin input (reverse phase) 	Set values of ES001 and ES000 Position of edge to be captured	CRC001 bit = 0 TI010 pin input 	Set values of ES101 and ES100 Position of edge to be captured
		01: Rising		01: Rising
		00: Falling 		00: Falling 
		11: Both edges (cannot be captured)		11: Both edges 
	Interrupt signal	INTTM000 signal is not generated even if value is captured.	Interrupt signal	INTTM000 signal is generated each time value is captured.
Capture operation of CR010	TI000 pin input[Note] 	Set values of ES001 and ES000 Position of edge to be captured		
		01: Rising		
		00: Falling 		
		11: Both edges		
	Interrupt signal	INTTM010 signal is generated each time value is captured.		

Note The capture operation of CR010 is not affected by the setting of the CRC001 bit.

Caution To capture the count value of the TM00 register to the CR000 register by using the phase reverse to that input to the TI000 pin, the interrupt request signal (INTTM000) is not generated after the value has been captured. If the valid edge is detected on the TI010 pin during this operation, the capture operation is not performed but the INTTM000 signal is generated as an external interrupt signal. To not use the external interrupt, mask the INTTM000 signal.

Remark CRC001: See **6.3 (2) Capture/compare control register 00 (CRC00)**.
ES101, ES100, ES001, ES000: See **6.3 (4) Prescaler mode register 00 (PRM00)**.

6.3 Registers Controlling 16-Bit Timer/Event Counter 00

Registers used to control 16-bit timer/event counter 00 are shown below.

- 16-bit timer mode control register 00 (TMC00)
- Capture/compare control register 00 (CRC00)
- 16-bit timer output control register 00 (TOC00)
- Prescaler mode register 00 (PRM00)
- Input switch control register (ISC)
- Port mode register 0, 1 (PM0, PM1)[Note]
- Port register 0, 1 (P0, P1)[Note]

Note The port pins with which the I/O pins for 16-bit timer/event counter 00 are shared differ depending on the product as follows:
- 78K0/KB2-A: Shared with the pins of port 1
- 78K0/KC2-A: Shared with the pins of port 0

(1) 16-bit timer mode control register 00 (TMC00)

TMC00 is an 8-bit register that sets the 16-bit timer/event counter 00 operation mode, TM00 clear mode, and output timing, and detects an overflow.

Rewriting TMC00 is prohibited during operation (when TMC003 and TMC002 = other than 00). However, it can be changed when TMC003 and TMC002 are cleared to 00 (stopping operation) and when OVF00 is cleared to 0.

TMC00 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears TMC00 to 00H.

Caution 16-bit timer/event counter 00 starts operation at the moment TMC002 and TMC003 are set to values other than 00 (operation stop mode), respectively. Set TMC002 and TMC003 to 00 to stop the operation.

Figure 6-5. Format of 16-Bit Timer Mode Control Register 00 (TMC00)

Address: FFBAH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
TMC00	0	0	0	0	TMC003	TMC002	TMC001	OVF00

TMC003	TMC002	Operation enable of 16-bit timer/event counter 00
0	0	Disables 16-bit timer/event counter 00 operation. Stops supplying operating clock. Clears 16-bit timer counter 00 (TM00).
0	1	Free-running timer mode
1	0	Clear & start mode entered by TI000 pin valid edge input[Note]
1	1	Clear & start mode entered upon a match between TM00 and CR000

TMC001	Condition to reverse timer output (TO00)
0	• Match between TM00 and CR000 or match between TM00 and CR010
1	• Match between TM00 and CR000 or match between TM00 and CR010 • Trigger input of TI000 pin valid edge

OVF00	TM00 overflow flag
Clear (0)	Clears OVF00 to 0 or TMC003 and TMC002 = 00
Set (1)	Overflow occurs.
OVF00 is set to 1 when the value of TM00 changes from FFFFH to 0000H in all the operation modes (free-running timer mode, clear & start mode entered by TI000 pin valid edge input, and clear & start mode entered upon a match between TM00 and CR000). It can also be set to 1 by writing 1 to OVF00.	

Note The TI000 pin valid edge is set by bits 5 and 4 (ES001, ES000) of prescaler mode register 00 (PRM00).

(2) Capture/compare control register 00 (CRC00)

CRC00 is the register that controls the operation of CR000 and CR010.

Changing the value of CRC00 is prohibited during operation (when TMC003 and TMC002 = other than 00).

CRC00 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears CRC00 to 00H.

Figure 6-6. Format of Capture/Compare Control Register 00 (CRC00)

Address: FFBCH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CRC00	0	0	0	0	0	CRC002	CRC001	CRC000

CRC002	CR010 operating mode selection
0	Operates as compare register
1	Operates as capture register

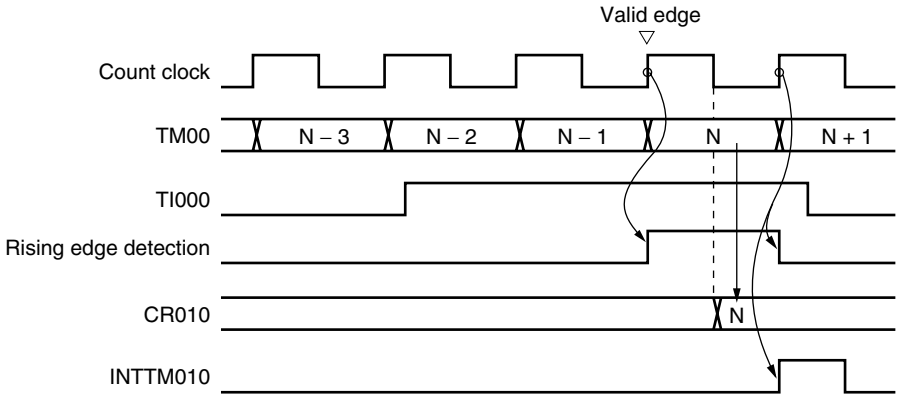
CRC001	CR000 capture trigger selection
0	Captures on valid edge of TI010 pin
1	Captures on valid edge of TI000 pin by reverse phase[Note]
The valid edge of the TI010 and TI000 pin is set by PRM00. If ES001 and ES000 are set to 11 (both edges) when CRC001 is 1, the valid edge of the TI000 pin cannot be detected.	

CRC000	CR000 operating mode selection
0	Operates as compare register
1	Operates as capture register
If TMC003 and TMC002 are set to 11 (clear & start mode entered upon a match between TM00 and CR000), be sure to set CRC000 to 0.	

Note When the valid edge is detected from the TI010 pin, the capture operation is not performed but the INTTM000 signal is generated as an external interrupt signal.

Caution To ensure that the capture operation is performed properly, the capture trigger requires a pulse two cycles longer than the count clock selected by prescaler mode register 00 (PRM00).

Figure 6-7. Example of CR010 Capture Operation (When Rising Edge Is Specified)



(3) 16-bit timer output control register 00 (TOC00)

TOC00 is an 8-bit register that controls TO00 output.

TOC00 can be rewritten while only OSPT00 is operating (when TMC003 and TMC002 = other than 00). Rewriting the other bits is prohibited during operation.

However, TOC004 can be rewritten during timer operation as a means to rewrite CR010 (see **6.5.1 Rewriting CR010 during TM00 operation**).

TOC00 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears TOC00 to 00H.

Caution Be sure to set TOC00 using the following procedure.

<1> Set TOC004 and TOC001 to 1.

<2> Set only TOE00 to 1.

<3> Set either of LVS00 or LVR00 to 1.

Figure 6-8. Format of 16-Bit Timer Output Control Register 00 (TOC00)

Address: FFBDH After reset: 00H R/W

Symbol	7	<6>	<5>	4	<3>	<2>	1	<0>
TOC00	0	OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00

OSPT00	One-shot pulse output trigger via software
0	–
1	One-shot pulse output
The value of this bit is always “0” when it is read. Do not set this bit to 1 in a mode other than the one-shot pulse output mode. If it is set to 1, TM00 is cleared and started.	

OSPE00	One-shot pulse output operation control
0	Successive pulse output
1	One-shot pulse output
One-shot pulse output operates correctly in the free-running timer mode or clear & start mode entered by TI000 pin valid edge input. The one-shot pulse cannot be output in the clear & start mode entered upon a match between TM00 and CR000.	

TOC004	TO00 output control on match between CR010 and TM00
0	Disables inversion operation
1	Enables inversion operation
The interrupt signal (INTTM010) is generated even when TOC004 = 0.	

LVS00	LVR00	Setting of TO00 output status
0	0	No change
0	1	Initial value of TO00 output is low level (TO00 output is cleared to 0).
1	0	Initial value of TO00 output is high level (TO00 output is set to 1).
1	1	Setting prohibited

- LVS00 and LVR00 can be used to set the initial value of the TO00 output level. If the initial value does not have to be set, leave LVS00 and LVR00 as 00.
- Be sure to set LVS00 and LVR00 when TOE00 = 1.
  LVS00, LVR00, and TOE00 being simultaneously set to 1 is prohibited.
- LVS00 and LVR00 are trigger bits. By setting these bits to 1, the initial value of the TO00 output level can be set. Even if these bits are cleared to 0, TO00 output is not affected.
- The values of LVS00 and LVR00 are always 0 when they are read.
- For how to set LVS00 and LVR00, see **6.5.2 Setting LVS00 and LVR00**.
- In case of 78K0/KB2-A, The actual pin output is determined depending on PM13 and P13, In case of 78K0/KC2-A, The actual pin output is determined depending on PM01 and P01, besides TO00 output.

TOC001	TO00 output control on match between CR000 and TM00
0	Disables inversion operation
1	Enables inversion operation
The interrupt signal (INTTM000) is generated even when TOC001 = 0.	

TOE00	TO00 output control
0	Disables output (TO00 output fixed to low level)
1	Enables output

(4) Prescaler mode register 00 (PRM00)

PRM00 is the register that sets the TM00 count clock and TI000 and TI010 pin input valid edges.

Rewriting PRM00 is prohibited during operation (when TMC003 and TMC002 = other than 00).

PRM00 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears PRM00 to 00H.

Cautions 1. Do not apply the following setting when setting the PRM001 and PRM000 bits to 11 (to specify the valid edge of the TI000 pin as a count clock).

- **Clear & start mode entered by the TI000 pin valid edge**
- **Setting the TI000 pin as a capture trigger**

2. If the operation of the 16-bit timer/event counter 00 is enabled when the TI000 or TI010 pin is at high level and when the valid edge of the TI000 or TI010 pin is specified to be the rising edge or both edges, the high level of the TI000 or TI010 pin is detected as a rising edge. Note this when the TI000 or TI010 pin is pulled up. However, the rising edge is not detected when the timer operation has been once stopped and then is enabled again.

3. In case of 78K0/KB2-A, the valid edge of TI010 and timer output (TO00) cannot be used for the P13 pin at the same time. Select either of the functions.

4. In case of 78K0/KC2-A, the valid edge of TI010 and timer output (TO00) cannot be used for the P01 pin at the same time. Select either of the functions.

Figure 6-9. Format of Prescaler Mode Register 00 (PRM00)

Address: FFBBH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PRM00	ES101	ES100	ES001	ES000	0	0	PRM001	PRM000

ES101	ES100	TI010 pin valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES001	ES000	TI000 pin valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

PRM001	PRM000	Count clock selection[Note 1]				
			f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	f_{PRS}[Note 2]	2 MHz	5 MHz	10 MHz	20 MHz[Note 3]
0	1	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz
1	0	$f_{PRS}/2^8$	7.81 kHz	19.53 kHz	39.06 kHz	78.12 kHz
1	1	TI000 valid edge[Notes 4, 5]				

Notes 1. The frequency that can be used for the peripheral hardware clock (f_{PRS}) differs depending on the power supply voltage.

Power supply voltage	Use frequency range of peripheral hardware clock (f_{PRS})
2.7 V ≤ V_{DD} ≤ 5.5 V	f_{PRS} ≤ 20 MHz
1.8 V ≤ V_{DD} ≤ 2.7 V	f_{PRS} ≤ 5 MHz

(The values shown in the table above are those when $f_{PRS} = f_{XH}$ (XSEL = 1).)

2. If the peripheral hardware clock (f_{PRS}) operates on the internal high-speed oscillation clock (f_{RH}) (XSEL = 0), when 1.8 V ≤ V_{DD} < 2.7 V, the setting of PRM001 = PRM000 = 0 (count clock: f_{PRS}) is prohibited.

3. This is settable only if 4.0 V ≤ V_{DD} ≤ 5.5 V

4. The external clock from the TI000 pin requires a pulse longer than twice the cycle of the peripheral hardware clock (f_{PRS}).

5. Do not start timer operation with the external clock from the TI000 pin when the internal high-speed oscillation clock and high-speed system clock are stopped while the CPU operates with the subsystem clock, or when in the STOP mode.

Remark f_{PRS} **:** Peripheral hardware clock frequency**.**

(5) Port input mode registers 0, 1 (PM0, PM1)

These registers specify input or output mode for the port 0 and 1 in 1-bit units.

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

Reset signal generation clears this register to FFH.

- 78K0/KB2-A

When using the P13/TO00/TI010/TOH1/INTP6 pin for timer output, set PM13 and the output latches of P13 to 0.

When using the P12/TI000/TOH0/INTP7 and P13/TO00/TI010/TOH1/INTP6 pins for timer input, set PM12 and PM13 to 1. At this time, the output latches of P12 and P13 may be 0 or 1.

Figure 6-10. Format of Port Mode Register 1(PM1)

Address: FF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PIM1	1	1	1	1	PM13	PM12	PM11	PM10

PIM1n	P1n pin I/O mode selection (n = 0 to 3)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

- 78K0/KC2-A

When using the P01/TO00/TI010 pin for timer output, set PM01 and the output latches of P01 to 0.

When using the P00/TI000 and P01/TO00/TI010 pin for timer input, set PM00 and PM01 to 1. At this time, the output latches of P00 and P01 may be 0 or 1.

Figure 6-11. Format of Port Mode Register 0(PM0)

Address: FF20H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	1	1	1	1	1	PM02	PM01	PM00

PM0n	P0n pin I/O mode selection (n = 0 to 2)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

6.4 Operation of 16-Bit Timer/Event Counter 00

6.4.1 Interval timer operation

If bits 3 and 2 (TMC003 and TMC002) of the 16-bit timer mode control register (TMC00) are set to 11 (clear & start mode entered upon a match between TM00 and CR000), the count operation is started in synchronization with the count clock.

When the value of TM00 later matches the value of CR000, TM00 is cleared to 0000H and a match interrupt signal (INTTM000) is generated. This INTTM000 signal enables TM00 to operate as an interval timer.

Remarks 1. For the setting of I/O pins, see **6.3 (5) Port mode registers 0, 1 (PM0, PM1)**.

2. For how to enable the INTTM000 interrupt, see **CHAPTER 18 INTERRUPT FUNCTIONS**.

Figure 6-12. Block Diagram of Interval Timer Operation

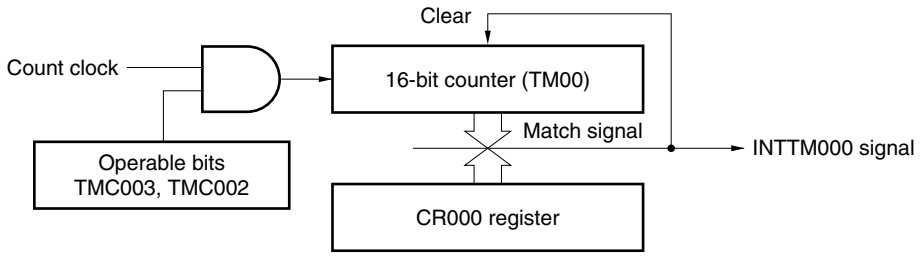


Figure 6-13. Basic Timing Example of Interval Timer Operation

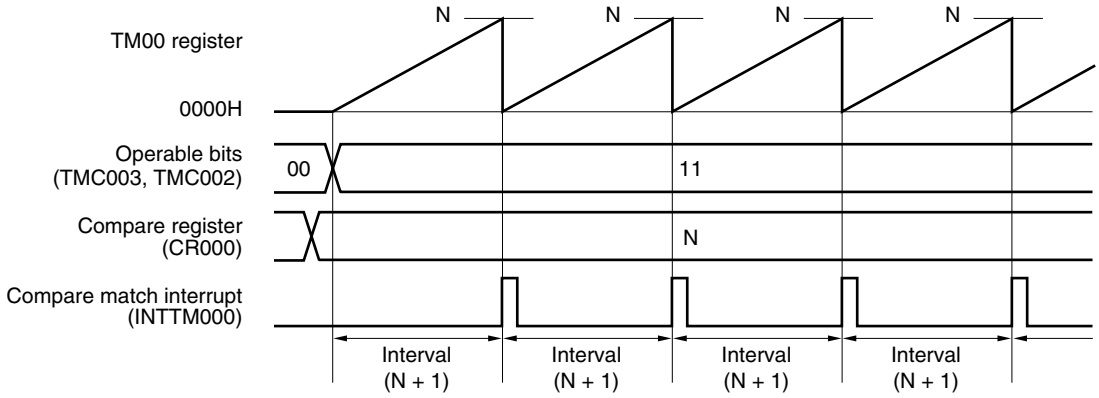


Figure 6-14. Example of Register Settings for Interval Timer Operation

(a) 16-bit timer mode control register 00 (TMC00)

				TMC003	TMC002	TMC001	OVF00
0	0	0	0	1	1	0	0

Clears and starts on match between TM00 and CR000.

(b) Capture/compare control register 00 (CRC00)

					CRC002	CRC001	CRC000
0	0	0	0	0	0	0	0

CR000 used as compare register

(c) 16-bit timer output control register 00 (TOC00)

	OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00
0	0	0	0	0	0	0	0

(d) Prescaler mode register 00 (PRM00)

ES101	ES100	ES001	ES000	3	2	PRM001	PRM000
0	0	0	0	0	0	0/1	0/1

Selects count clock

(e) 16-bit timer counter 00 (TM00)

By reading TM00, the count value can be read.

(f) 16-bit capture/compare register 000 (CR000)

If M is set to CR000, the interval time is as follows.

- Interval time = (M + 1) × Count clock cycle

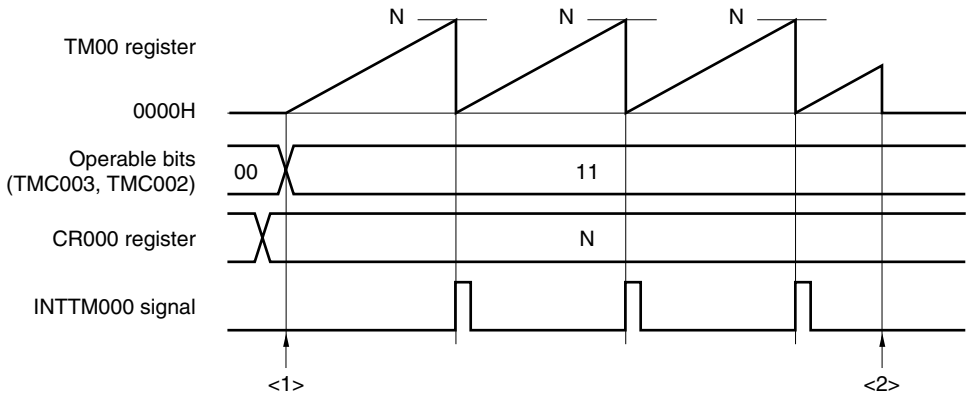
Setting CR000 to 0000H is prohibited.

(g) 16-bit capture/compare register 010 (CR010)

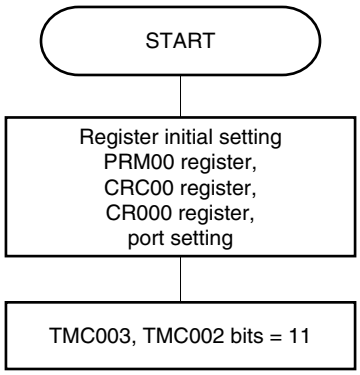
Usually, CR010 is not used for the interval timer function. However, a compare match interrupt (INTTM010) is generated when the set value of CR010 matches the value of TM00.

Therefore, mask the interrupt request by using the interrupt mask flag (TMMK010).

Figure 6-15. Example of Software Processing for Interval Timer Function



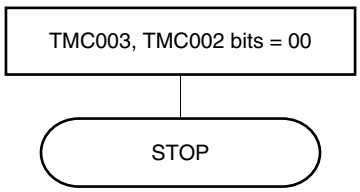
<1> Count operation start flow



Initial setting of these registers is performed before setting the TMC003 and TMC002 bits to 11.

Starts count operation

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TMC003 and TMC002 bits to 00.

6.4.2 Square wave output operation

When 16-bit timer/event counter 00 operates as an interval timer (see **6.4.1**), a square wave can be output from the TO00 pin by setting the 16-bit timer output control register 00 (TOC00) to 03H.

When TMC003 and TMC002 are set to 11 (count clear & start mode entered upon a match between TM00 and CR000), the counting operation is started in synchronization with the count clock.

When the value of TM00 later matches the value of CR000, TM00 is cleared to 0000H, an interrupt signal (INTTM000) is generated, and TO00 output is inverted. This TO00 output that is inverted at fixed intervals enables TO00 to output a square wave.

Remarks 1. For the setting of I/O pins, see **6.3 (5) Port mode registers 0 and 1 (PM0, PM1)**.

2. For how to enable the INTTM000 signal interrupt, see **CHAPTER 18 INTERRUPT FUNCTIONS**.

Figure 6-16. Block Diagram of Square Wave Output Operation

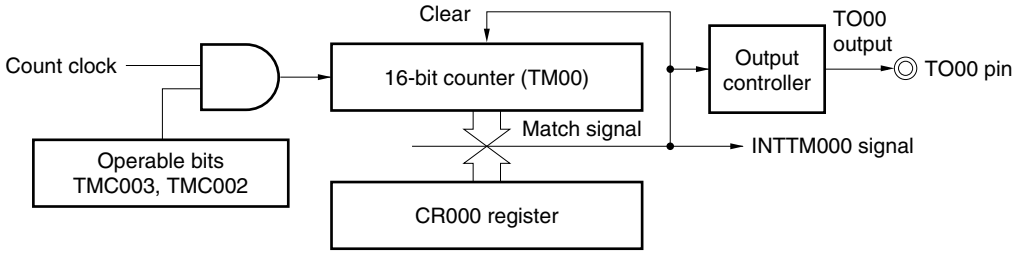


Figure 6-17. Basic Timing Example of Square Wave Output Operation

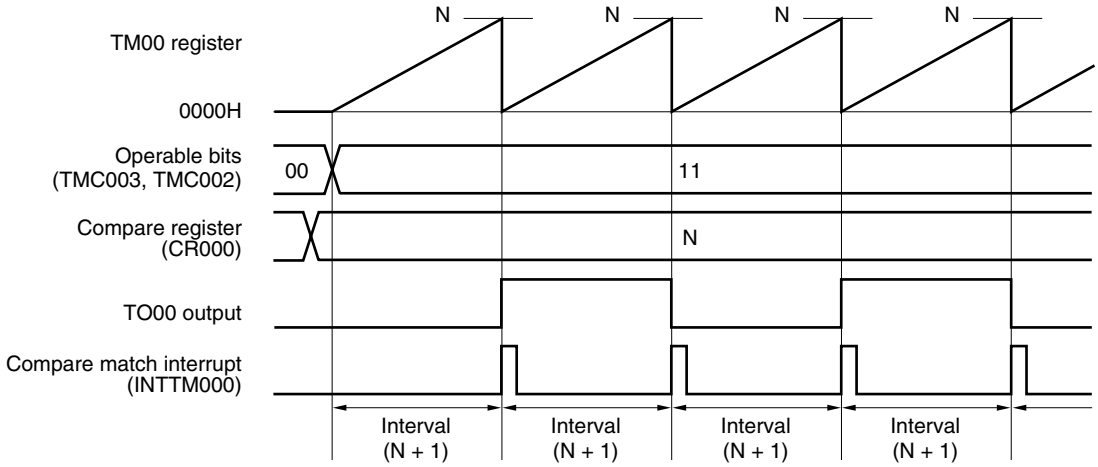
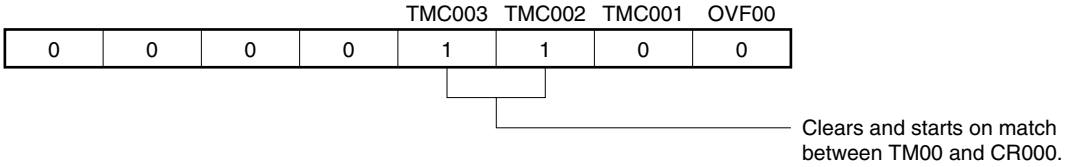
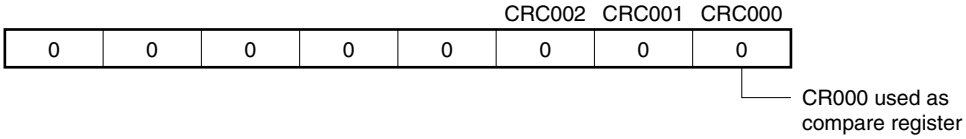


Figure 6-18. Example of Register Settings for Square Wave Output Operation

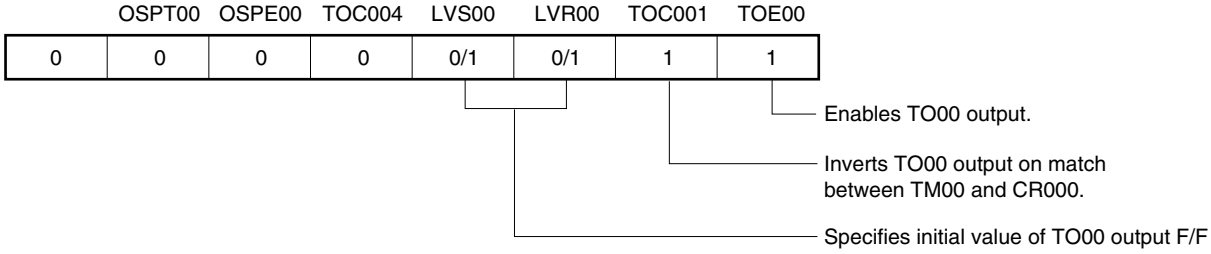
(a) 16-bit timer mode control register 00 (TMC00)



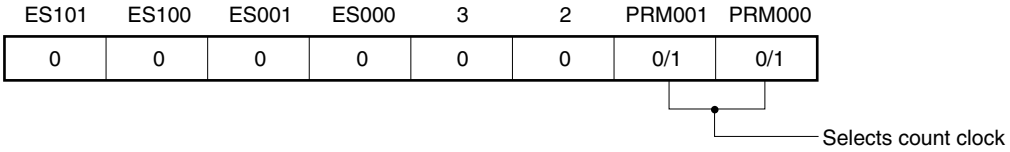
(b) Capture/compare control register 00 (CRC00)



(c) 16-bit timer output control register 00 (TOC00)



(d) Prescaler mode register 00 (PRM00)



(e) 16-bit timer counter 00 (TM00)

By reading TM00, the count value can be read.

(f) 16-bit capture/compare register 000 (CR000)

If M is set to CR000, the interval time is as follows.

- Square wave frequency = 1 / [2 × (M + 1) × Count clock cycle]

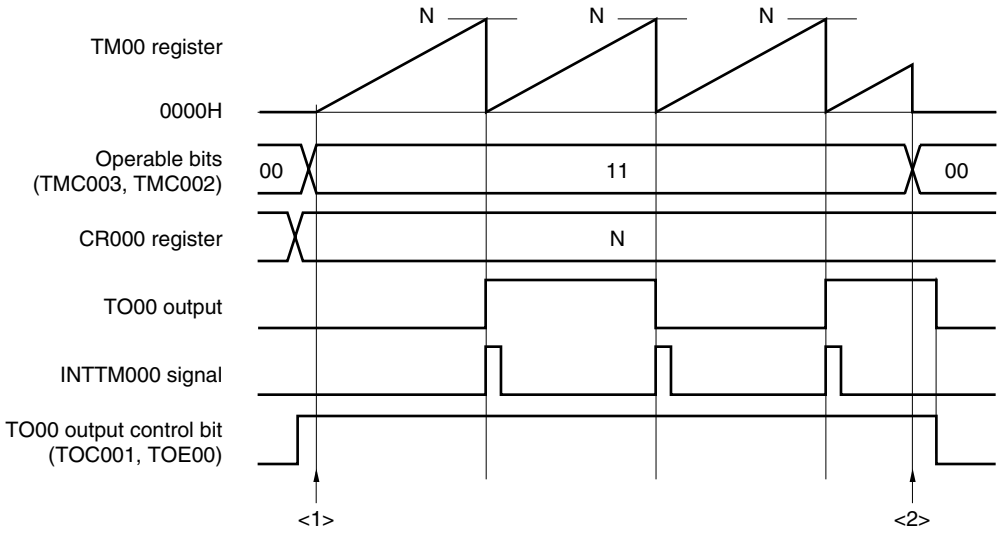
Setting CR000 to 0000H is prohibited.

(g) 16-bit capture/compare register 010 (CR010)

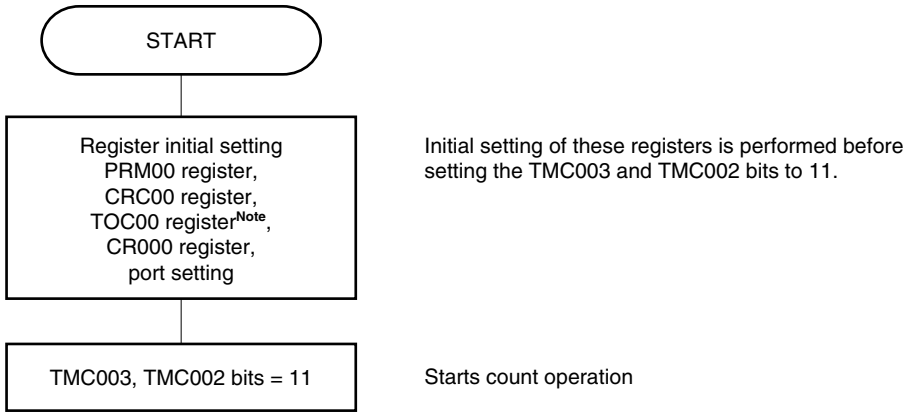
Usually, CR010 is not used for the square wave output function. However, a compare match interrupt (INTTM010) is generated when the set value of CR010 matches the value of TM00.

Therefore, mask the interrupt request by using the interrupt mask flag (TMMK010).

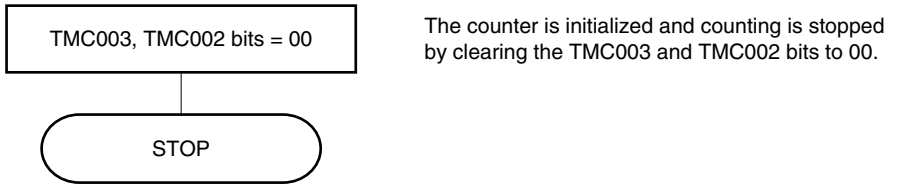
Figure 6-19. Example of Software Processing for Square Wave Output Function



<1> Count operation start flow



<2> Count operation stop flow



Note Care must be exercised when setting TOC00. For details, see **6.3 (3) 16-bit timer output control register 00 (TOC00)**.

6.4.3 External event counter operation

When bits 1 and 0 (PRM001 and PRM000) of the prescaler mode register 00 (PRM00) are set to 11 (for counting up with the valid edge of the TI000 pin) and bits 3 and 2 (TMC003 and TMC002) of 16-bit timer mode control register 00 (TMC00) are set to 11, the valid edge of an external event input is counted, and a match interrupt signal indicating matching between TM00 and CR000 (INTTM000) is generated.

To input the external event, the TI000 pin is used. Therefore, the timer/event counter cannot be used as an external event counter in the clear & start mode entered by the TI000 pin valid edge input (when TMC003 and TMC002 = 10).

The INTTM000 signal is generated with the following timing.

- Timing of generation of INTTM000 signal (second time or later)
  = Number of times of detection of valid edge of external event × (Set value of CR000 + 1)

However, the first match interrupt immediately after the timer/event counter has started operating is generated with the following timing.

- Timing of generation of INTTM000 signal (first time only)
  = Number of times of detection of valid edge of external event input × (Set value of CR000 + 2)

To detect the valid edge, the signal input to the TI000 pin is sampled during the clock cycle of f_{PRS}. The valid edge is not detected until it is detected two times in a row. Therefore, a noise with a short pulse width can be eliminated.

Remarks 1. For the setting of I/O pins, see **6.3 (5) Port mode registers 0 and 1 (PM0, PM1)**.

2. For how to enable the INTTM000 signal interrupt, see **CHAPTER 18 INTERRUPT FUNCTIONS**.

Figure 6-20. Block Diagram of External Event Counter Operation

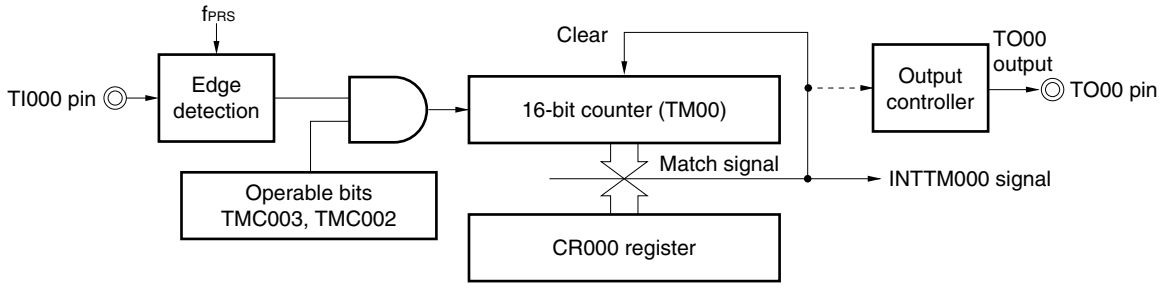
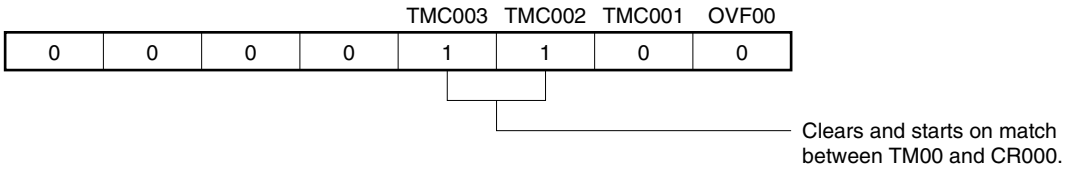
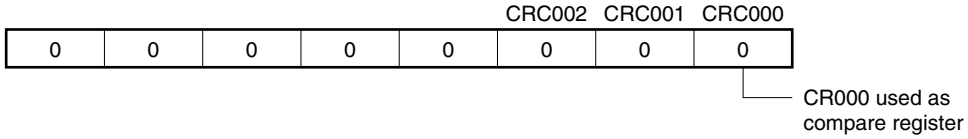


Figure 6-21. Example of Register Settings in External Event Counter Mode (1/2)

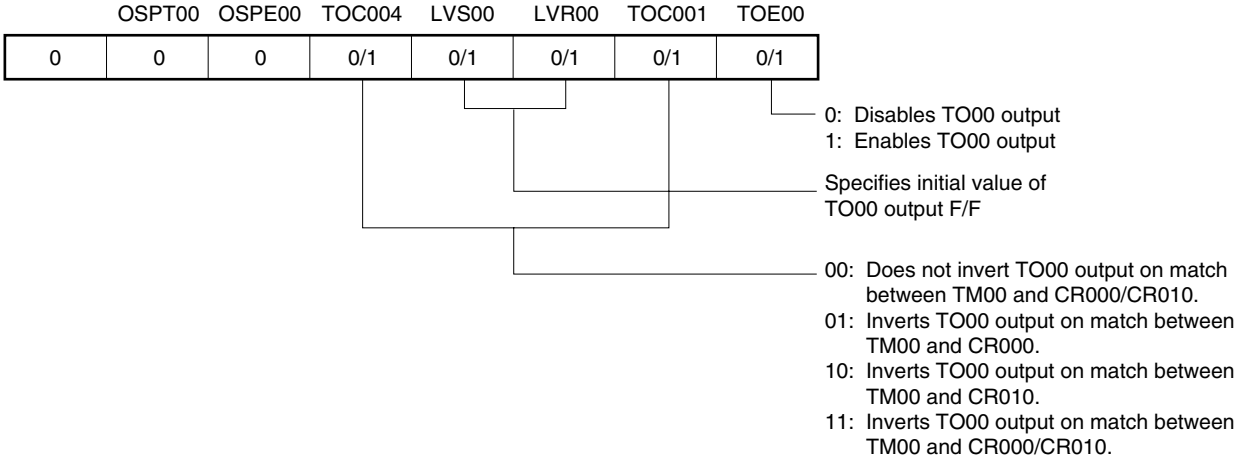
(a) 16-bit timer mode control register 00 (TMC00)



(b) Capture/compare control register 00 (CRC00)



(c) 16-bit timer output control register 00 (TOC00)



(d) Prescaler mode register 00 (PRM00)

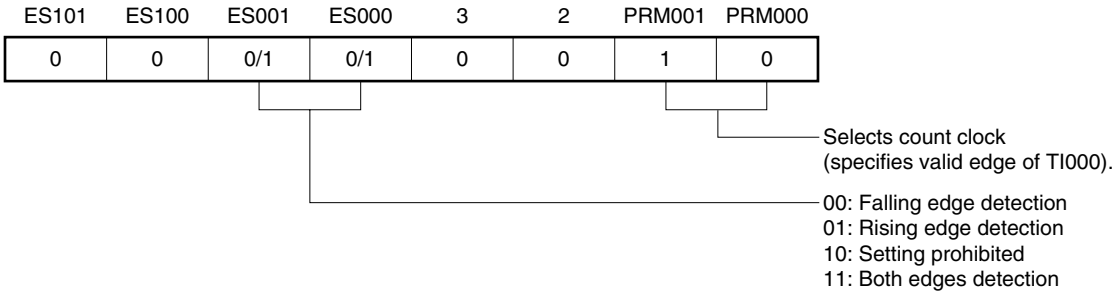


Figure 6-21. Example of Register Settings in External Event Counter Mode (2/2)

(e) 16-bit timer counter 00 (TM00)

By reading TM00, the count value can be read.

(f) 16-bit capture/compare register 000 (CR000)

If M is set to CR000, the interrupt signal (INTTM000) is generated when the number of external events reaches (M + 1).

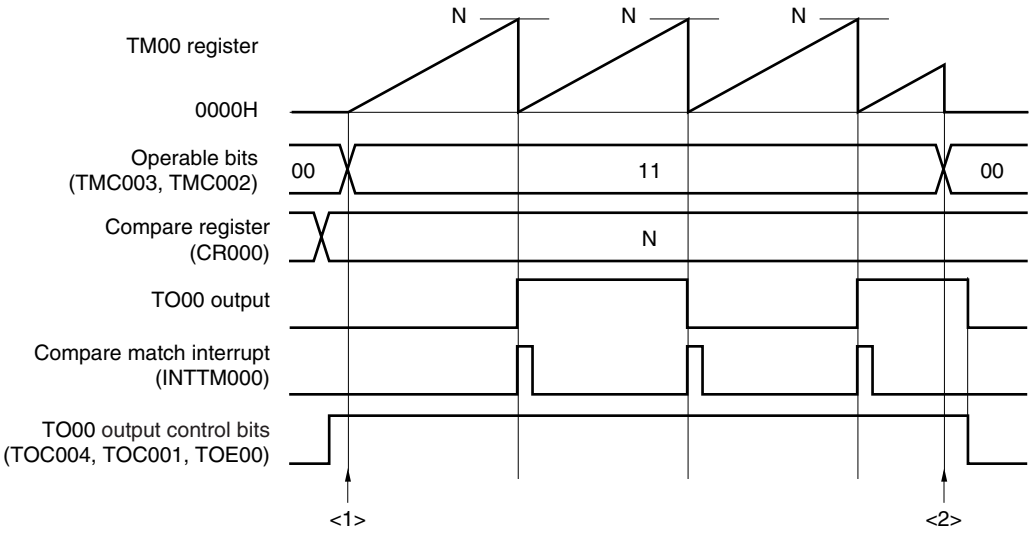
Setting CR000 to 0000H is prohibited.

(g) 16-bit capture/compare register 010 (CR010)

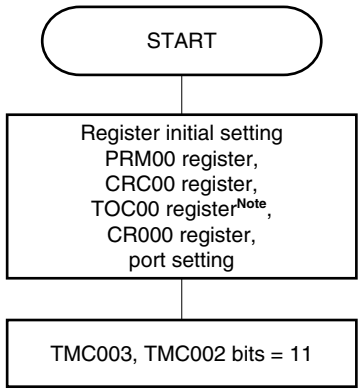
Usually, CR010 is not used in the external event counter mode. However, a compare match interrupt (INTTM010) is generated when the set value of CR010 matches the value of TM00.

Therefore, mask the interrupt request by using the interrupt mask flag (TMMK010).

Figure 6-22. Example of Software Processing in External Event Counter Mode



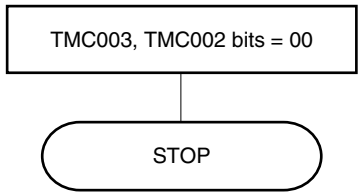
<1> Count operation start flow



Initial setting of these registers is performed before setting the TMC003 and TMC002 bits to 11.

Starts count operation

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TMC003 and TMC002 bits to 00.

Note Care must be exercised when setting TOC00. For details, see **6.3 (3) 16-bit timer output control register 00 (TOC00)**.

6.4.4 Operation in clear & start mode entered by TI000 pin valid edge input

When bits 3 and 2 (TMC003 and TMC002) of 16-bit timer mode control register 00 (TMC00) are set to 10 (clear & start mode entered by the TI000 pin valid edge input) and the count clock (set by PRM00) is supplied to the timer/event counter, TM00 starts counting up. When the valid edge of the TI000 pin is detected during the counting operation, TM00 is cleared to 0000H and starts counting up again. If the valid edge of the TI000 pin is not detected, TM00 overflows and continues counting.

The valid edge of the TI000 pin is a cause to clear TM00. Starting the counter is not controlled immediately after the start of the operation.

CR000 and CR010 are used as compare registers and capture registers.

(a) When CR000 and CR010 are used as compare registers

Signals INTTM000 and INTTM010 are generated when the value of TM00 matches the value of CR000 and CR010.

(b) When CR000 and CR010 are used as capture registers

The count value of TM00 is captured to CR000 and the INTTM000 signal is generated when the valid edge is input to the TI010 pin (or when the phase reverse to that of the valid edge is input to the TI000 pin).

When the valid edge is input to the TI000 pin, the count value of TM00 is captured to CR010 and the INTTM010 signal is generated. As soon as the count value has been captured, the counter is cleared to 0000H.

Caution Do not set the count clock as the valid edge of the TI000 pin (PRM001, and PRM000 = 11). When PRM001, and PRM000 = 11, TM00 is cleared.

Remarks 1. For the setting of the I/O pins, see **6.3 (5) Port mode registers 0 and 1 (PM0, PM1)**.

2. For how to enable the INTTM000 signal interrupt, see **CHAPTER 18 INTERRUPT FUNCTIONS**.

(1) Operation in clear & start mode entered by TI000 pin valid edge input (CR000: compare register, CR010: compare register)

Figure 6-23. Block Diagram of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Compare Register, CR010: Compare Register)

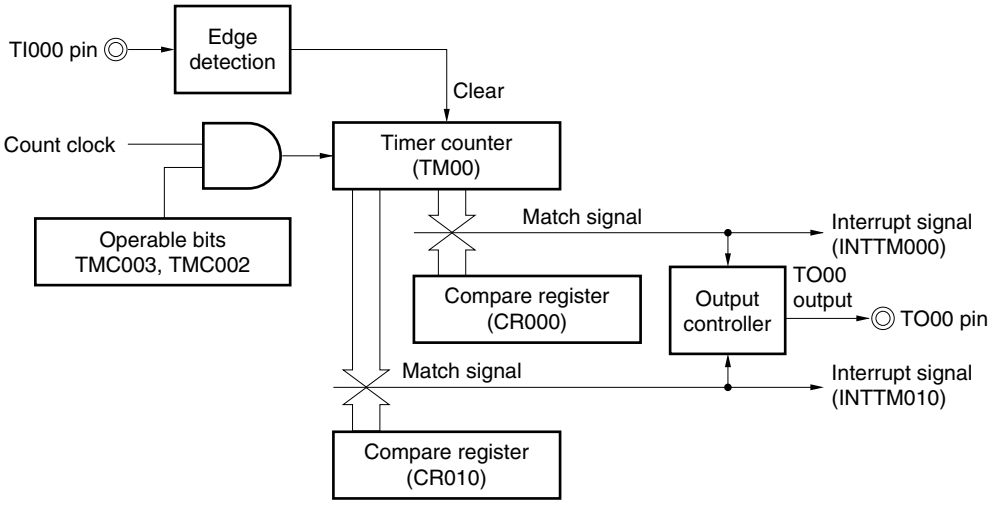
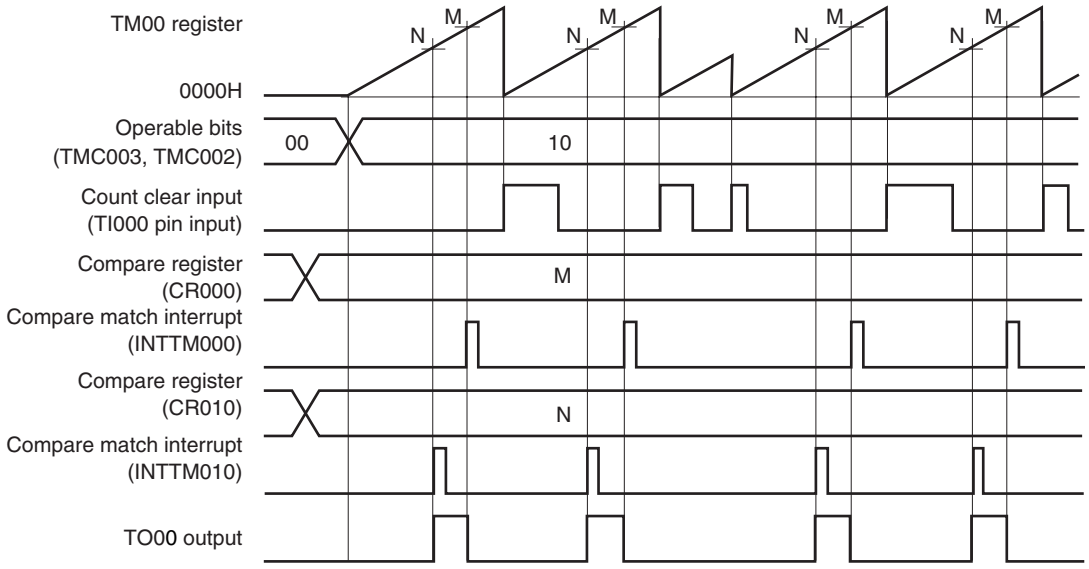
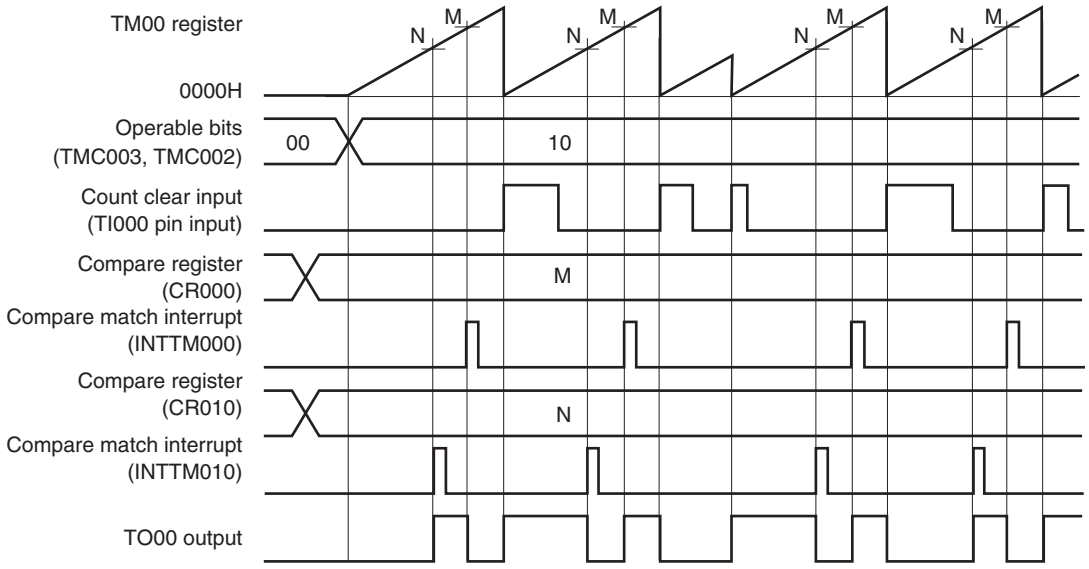


Figure 6-24. Timing Example of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Compare Register, CR010: Compare Register)

(a) TOC00 = 13H, PRM00 = 10H, CRC00, = 00H, TMC00 = 08H



(b) TOC00 = 13H, PRM00 = 10H, CRC00, = 00H, TMC00 = 0AH



(a) and (b) differ as follows depending on the setting of bit 1 (TMC001) of the 16-bit timer mode control register 00 (TMC00).

(a) The TO00 output level is inverted when TM00 matches a compare register.

(b) The TO00 output level is inverted when TM00 matches a compare register or when the valid edge of the TI000 pin is detected.

(2) Operation in clear & start mode entered by TI000 pin valid edge input (CR000: compare register, CR010: capture register)

Figure 6-25. Block Diagram of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Compare Register, CR010: Capture Register)

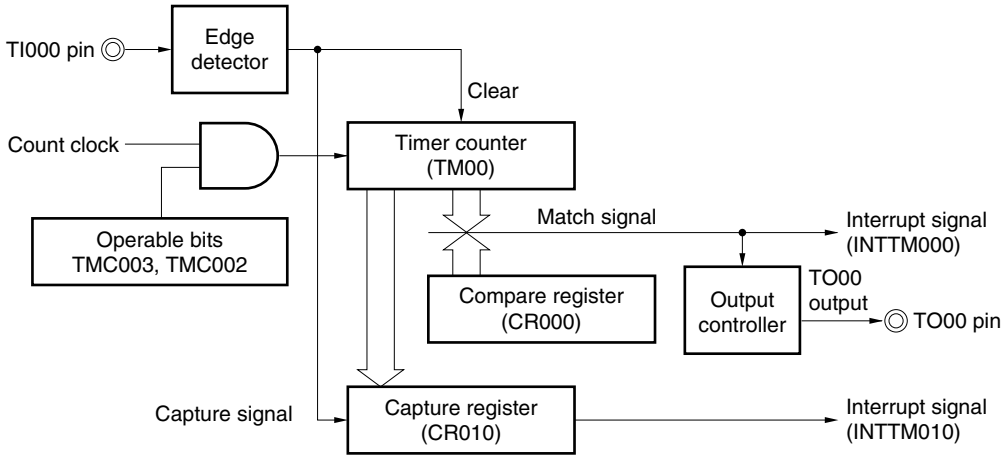
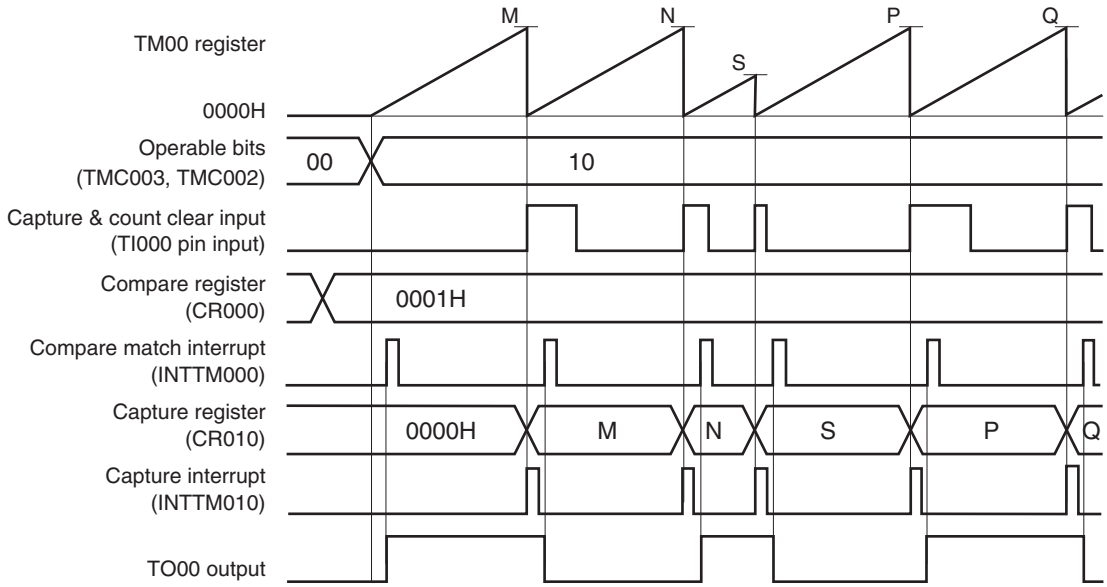


Figure 6-26. Timing Example of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Compare Register, CR010: Capture Register) (1/2)

(a) TOC00 = 13H, PRM00 = 10H, CRC00, = 04H, TMC00 = 08H, CR000 = 0001H

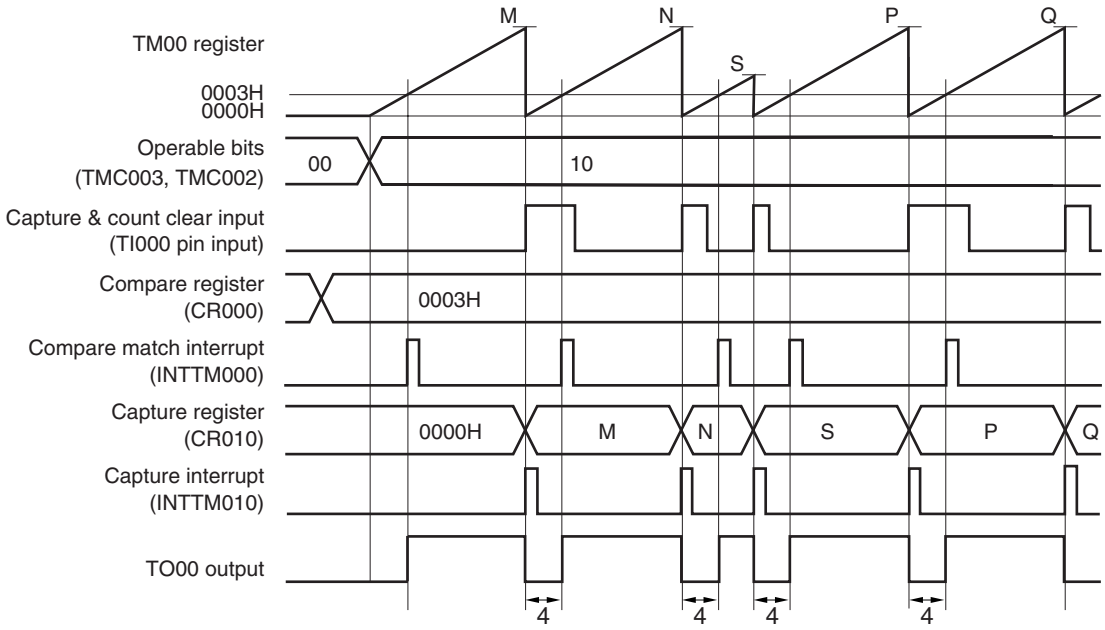


This is an application example where the TO00 output level is inverted when the count value has been captured & cleared.

The count value is captured to CR010 and TM00 is cleared (to 0000H) when the valid edge of the TI000 pin is detected. When the count value of TM00 is 0001H, a compare match interrupt signal (INTTM000) is generated, and the TO00 output level is inverted.

Figure 6-26. Timing Example of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Compare Register, CR010: Capture Register) (2/2)

(b) TOC00 = 13H, PRM00 = 10H, CRC00, = 04H, TMC00 = 0AH, CR000 = 0003H



This is an application example where the width set to CR000 (4 clocks in this example) is to be output from the TO00 pin when the count value has been captured & cleared.

The count value is captured to CR010, a capture interrupt signal (INTTM010) is generated, TM00 is cleared (to 0000H), and the TO00 output is inverted when the valid edge of the TI000 pin is detected. When the count value of TM00 is 0003H (four clocks have been counted), a compare match interrupt signal (INTTM000) is generated and the TO00 output level is inverted.

(3) Operation in clear & start mode by entered TI000 pin valid edge input (CR000: capture register, CR010: compare register)

Figure 6-27. Block Diagram of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Capture Register, CR010: Compare Register)

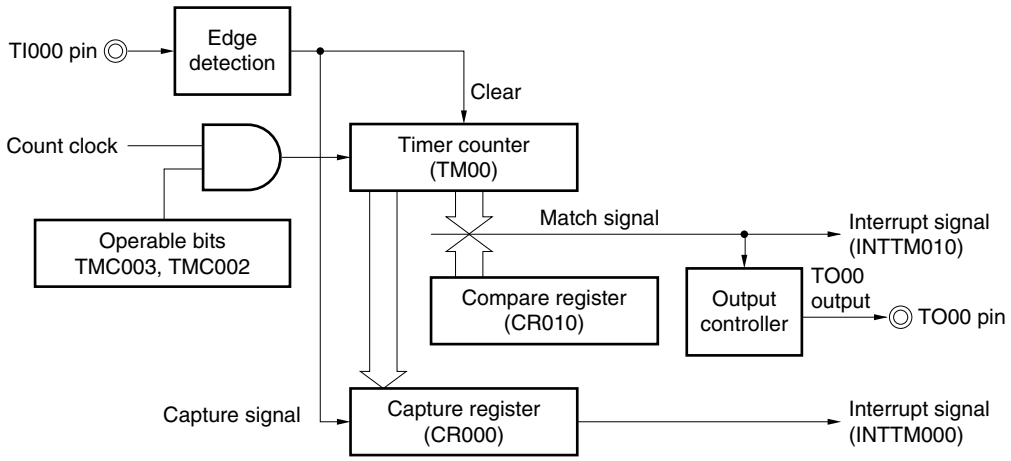
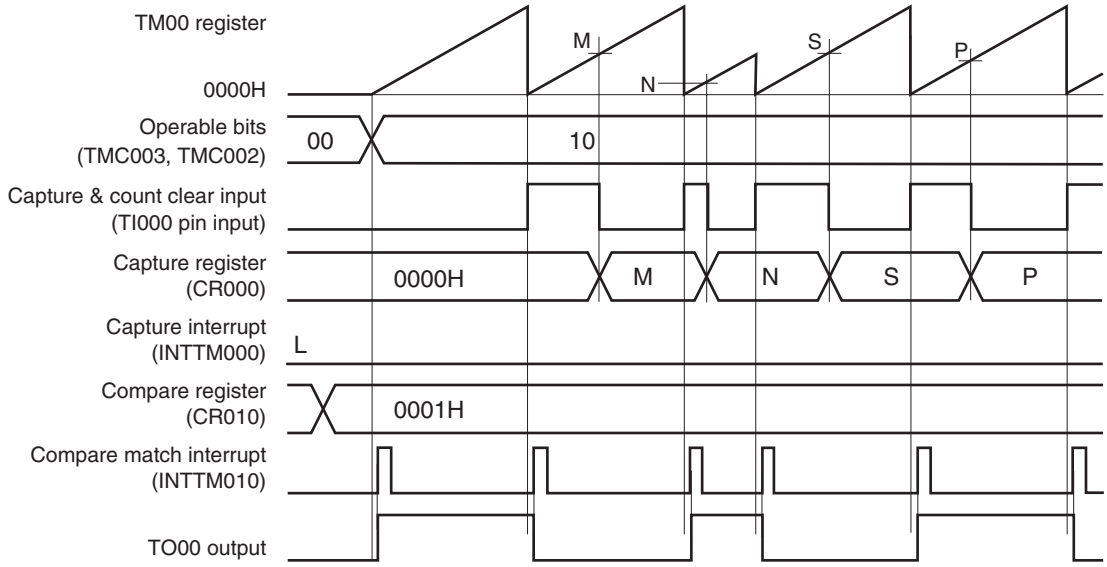


Figure 6-28. Timing Example of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Capture Register, CR010: Compare Register) (1/2)

(a) TOC00 = 13H, PRM00 = 10H, CRC00, = 03H, TMC00 = 08H, CR010 = 0001H



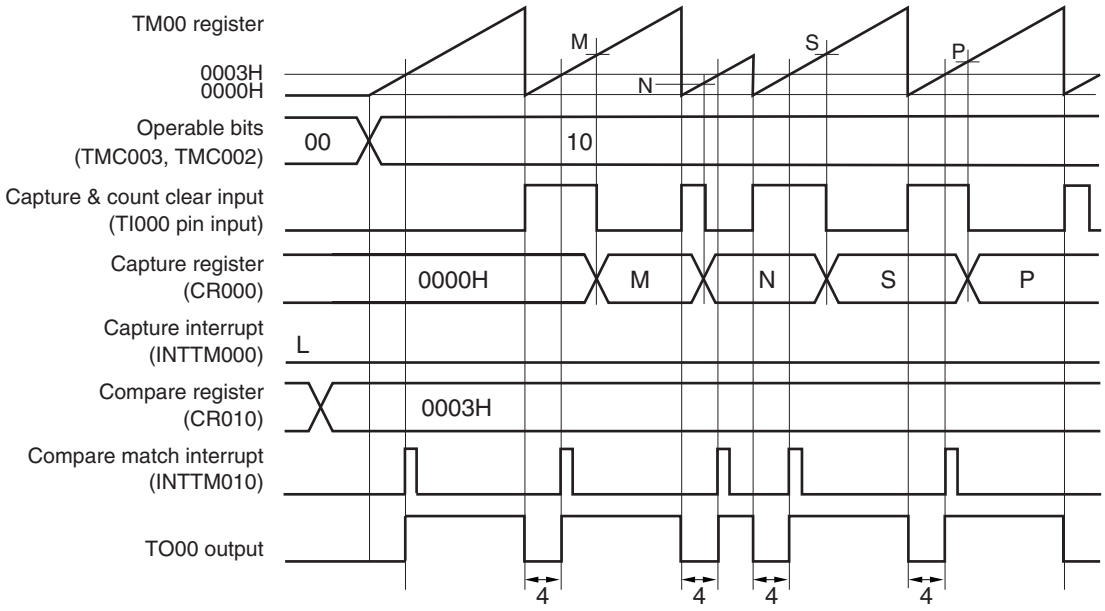
This is an application example where the TO00 output level is to be inverted when the count value has been captured & cleared.

TM00 is cleared at the rising edge detection of the TI000 pin and it is captured to CR000 at the falling edge detection of the TI000 pin.

When bit 1 (CRC001) of capture/compare control register 00 (CRC00) is set to 1, the count value of TM00 is captured to CR000 in the phase reverse to that of the signal input to the TI000 pin, but the capture interrupt signal (INTTM000) is not generated. However, the INTTM000 signal is generated when the valid edge of the TI010 pin is detected. Mask the INTTM000 signal when it is not used.

Figure 6-28. Timing Example of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Capture Register, CR010: Compare Register) (2/2)

(b) TOC00 = 13H, PRM00 = 10H, CRC00, = 03H, TMC00 = 0AH, CR010 = 0003H



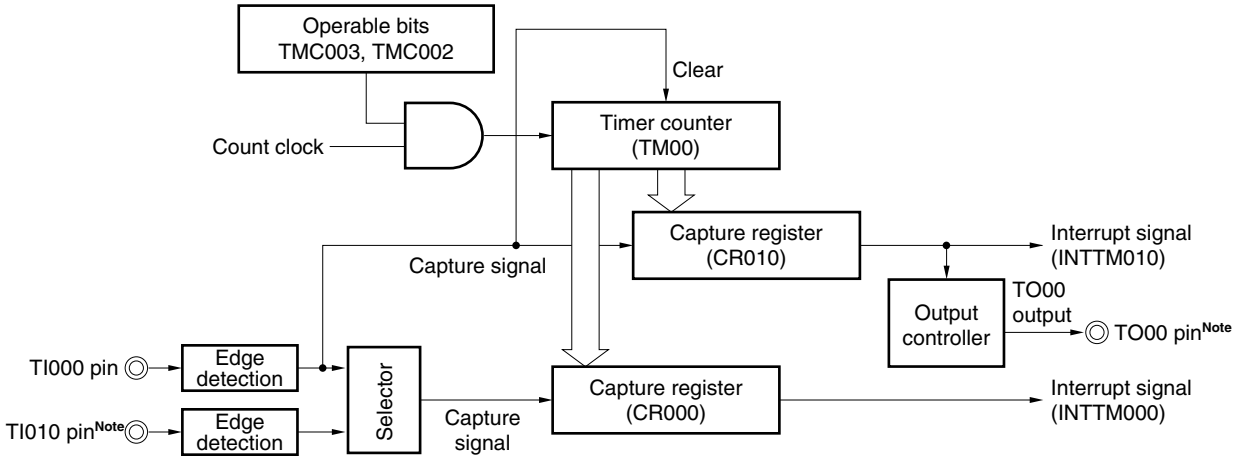
This is an application example where the width set to CR010 (4 clocks in this example) is to be output from the TO00 pin when the count value has been captured & cleared.

TM00 is cleared (to 0000H) at the rising edge detection of the TI000 pin and captured to CR000 at the falling edge detection of the TI000 pin. The TO00 output is inverted when TM00 is cleared (to 0000H) because the rising edge of the TI000 pin has been detected or when the value of TM00 matches that of a compare register (CR010).

When bit 1 (CRC001) of capture/compare control register 00 (CRC00) is 1, the count value of TM00 is captured to CR000 in the phase reverse to that of the input signal of the TI000 pin, but the capture interrupt signal (INTTM000) is not generated. However, the INTTM000 interrupt is generated when the valid edge of the TI010 pin is detected. Mask the INTTM000 signal when it is not used.

(4) Operation in clear & start mode entered by TI000 pin valid edge input (CR000: capture register, CR010: capture register)

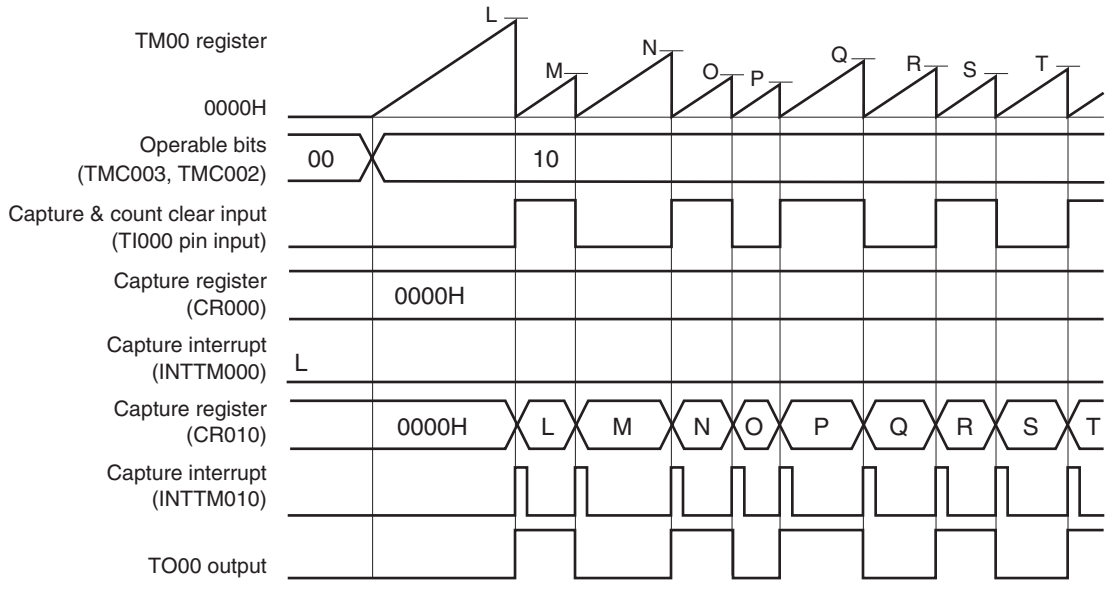
Figure 6-29. Block Diagram of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Capture Register, CR010: Capture Register)



Note The timer output (TO00) cannot be used when detecting the valid edge of the TI010 pin is used.

Figure 6-30. Timing Example of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Capture Register, CR010: Capture Register) (1/3)

(a) TOC00 = 13H, PRM00 = 30H, CRC00 = 05H, TMC00 = 0AH

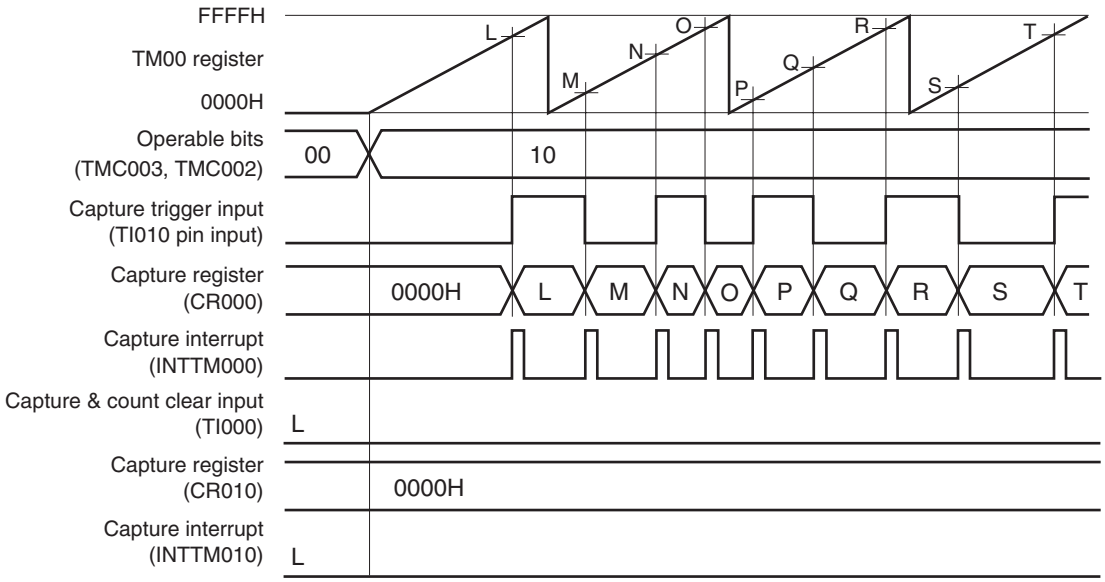


This is an application example where the count value is captured to CR010, TM00 is cleared, and the TO00 output is inverted when the rising or falling edge of the TI000 pin is detected.

When the edge of the TI010 pin is detected, an interrupt signal (INTTM000) is generated. Mask the INTTM000 signal when it is not used.

Figure 6-30. Timing Example of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Capture Register, CR010: Capture Register) (2/3)

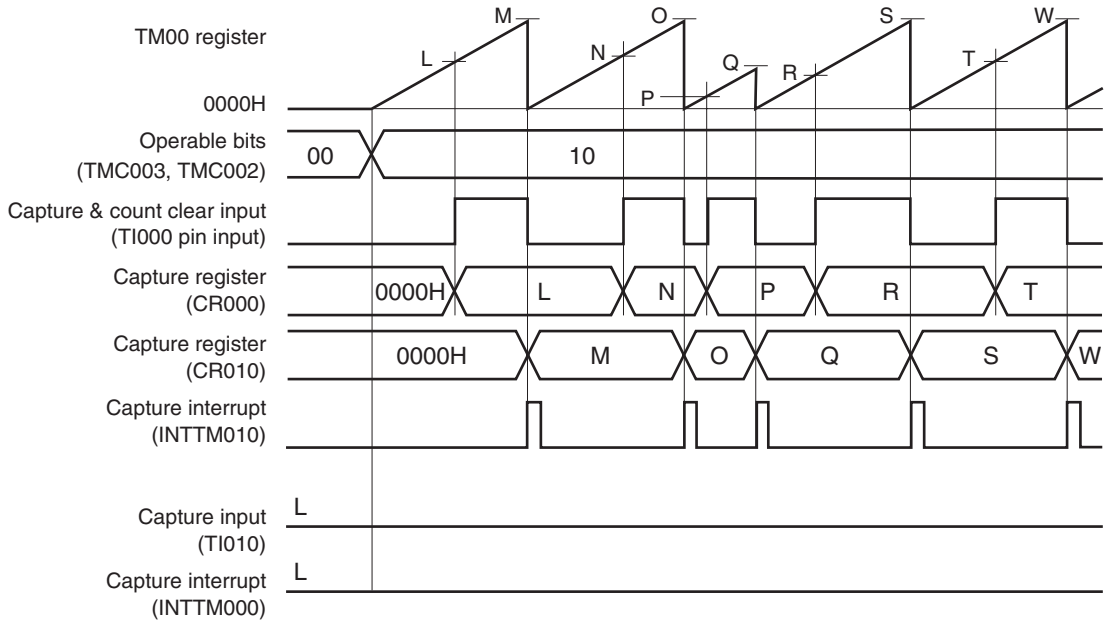
(b) TOC00 = 13H, PRM00 = C0H, CRC00 = 05H, TMC00 = 0AH



This is a timing example where an edge is not input to the TI000 pin, in an application where the count value is captured to CR000 when the rising or falling edge of the TI010 pin is detected.

Figure 6-30. Timing Example of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Capture Register, CR010: Capture Register) (3/3)

(c) TOC00 = 13H, PRM00 = 00H, CRC00 = 07H, TMC00 = 0AH



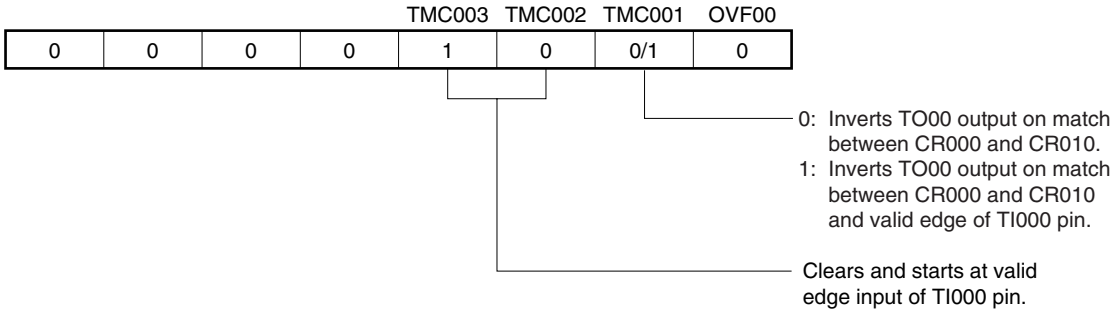
This is an application example where the pulse width of the signal input to the TI000 pin is measured.
By setting CRC00, the count value can be captured to CR000 in the phase reverse to the falling edge of the TI000 pin (i.e., rising edge) and to CR010 at the falling edge of the TI000 pin.
The high- and low-level widths of the input pulse can be calculated by the following expressions.

- High-level width = [CR010 value] – [CR000 value] × [Count clock cycle]
- Low-level width = [CR000 value] × [Count clock cycle]

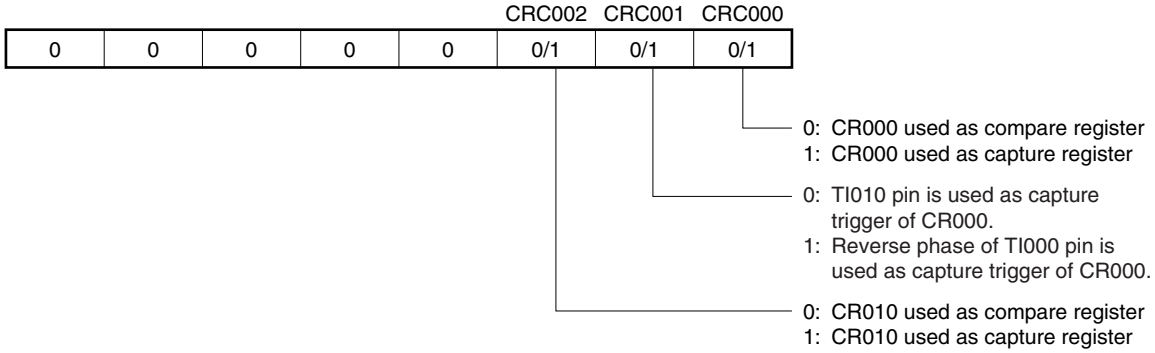
If the reverse phase of the TI000 pin is selected as a trigger to capture the count value to CR000, the INTTM000 signal is not generated. Read the values of CR000 and CR010 to measure the pulse width immediately after the INTTM010 signal is generated.
However, if the valid edge specified by bits 6 and 5 (ES101 and ES100) of prescaler mode register 00 (PRM00) is input to the TI010 pin, the count value is not captured but the INTTM000 signal is generated. To measure the pulse width of the TI000 pin, mask the INTTM000 signal when it is not used.

Figure 6-31. Example of Register Settings in Clear & Start Mode Entered by TI000 Pin Valid Edge Input (1/2)

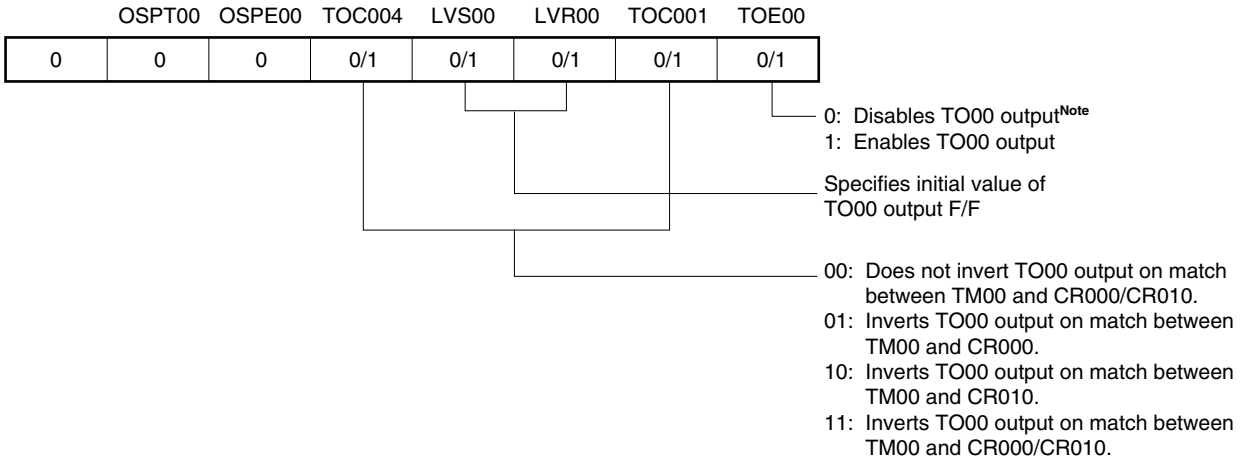
(a) 16-bit timer mode control register 00 (TMC00)



(b) Capture/compare control register 00 (CRC00)



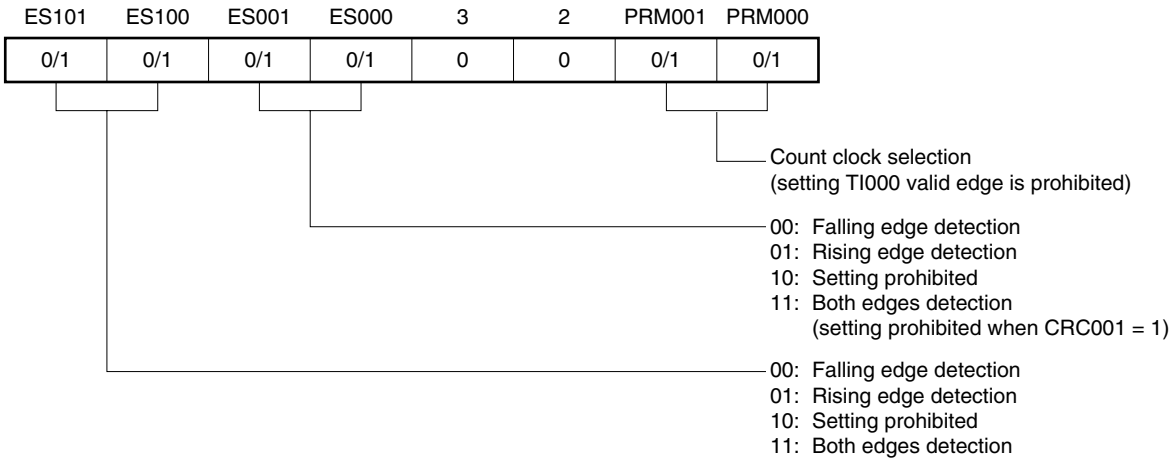
(c) 16-bit timer output control register 00 (TOC00)



Note The timer output (TO00) cannot be used when detecting the valid edge of the TI010 pin is used.

Figure 6-31. Example of Register Settings in Clear & Start Mode Entered by TI000 Pin Valid Edge Input (2/2)

(d) Prescaler mode register 00 (PRM00)



(e) 16-bit timer counter 00 (TM00)

By reading TM00, the count value can be read.

(f) 16-bit capture/compare register 000 (CR000)

When this register is used as a compare register and when its value matches the count value of TM00, an interrupt signal (INTTM000) is generated. The count value of TM00 is not cleared.

To use this register as a capture register, select either the TI000 or TI010 pin[Note] input as a capture trigger. When the valid edge of the capture trigger is detected, the count value of TM00 is stored in CR000.

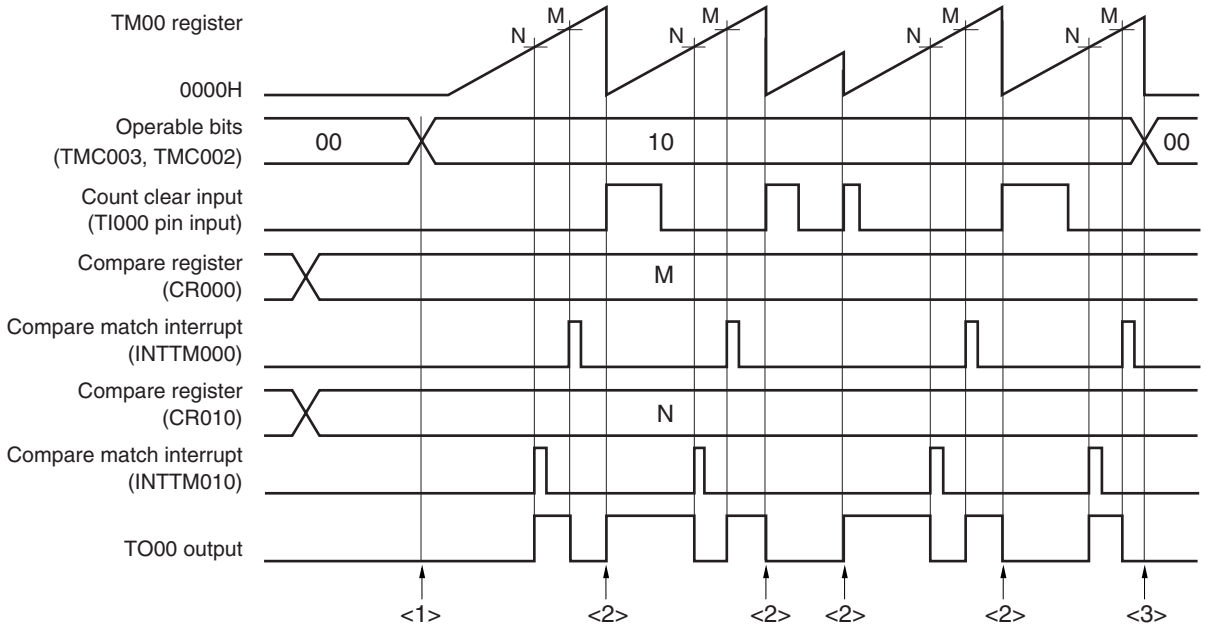
Note The timer output (TO00) cannot be used when detection of the valid edge of the TI010 pin is used.

(g) 16-bit capture/compare register 010 (CR010)

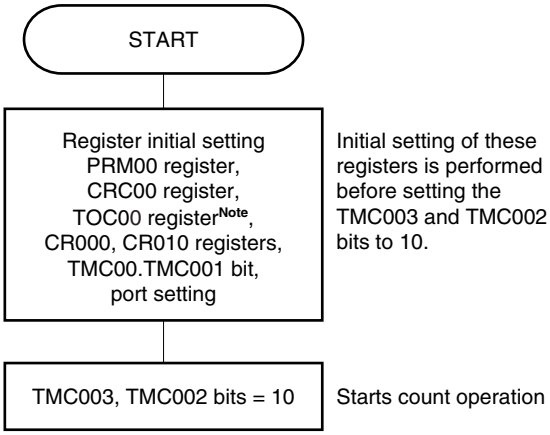
When this register is used as a compare register and when its value matches the count value of TM00, an interrupt signal (INTTM010) is generated. The count value of TM00 is not cleared.

When this register is used as a capture register, the TI000 pin input is used as a capture trigger. When the valid edge of the capture trigger is detected, the count value of TM00 is stored in CR010.

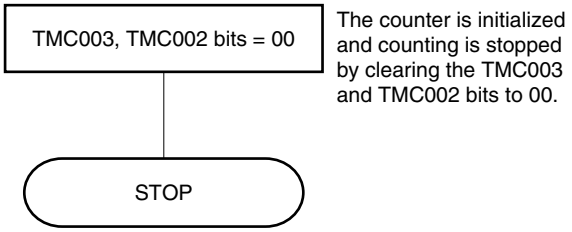
Figure 6-32. Example of Software Processing in Clear & Start Mode Entered by TI000 Pin Valid Edge Input



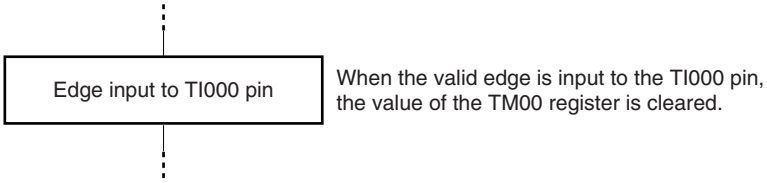
<1> Count operation start flow



<3> Count operation stop flow



<2> TM00 register clear & start flow



Note Care must be exercised when setting TOC00. For details, see **6.3 (3) 16-bit timer output control register 00 (TOC00)**.

6.4.5 Free-running timer operation

When bits 3 and 2 (TMC003 and TMC002) of 16-bit timer mode control register 00 (TMC00) are set to 01 (free-running timer mode), 16-bit timer/event counter 00 continues counting up in synchronization with the count clock. When it has counted up to FFFFH, the overflow flag (OVF00) is set to 1 at the next clock, and TM00 is cleared (to 0000H) and continues counting. Clear OVF00 to 0 by executing the CLR instruction via software.

The following three types of free-running timer operations are available.

- Both CR000 and CR010 are used as compare registers.
- One of CR000 or CR010 is used as a compare register and the other is used as a capture register.
- Both CR000 and CR010 are used as capture registers.

Remarks 1. For the setting of the I/O pins, see **6.3 (5) Port mode registers 0 and 1 (PM0, PM1)**.

2. For how to enable the INTTM000 signal interrupt, see **CHAPTER 18 INTERRUPT FUNCTIONS**.

(1) Free-running timer mode operation (CR000: compare register, CR010: compare register)

Figure 6-33. Block Diagram of Free-Running Timer Mode (CR000: Compare Register, CR010: Compare Register)

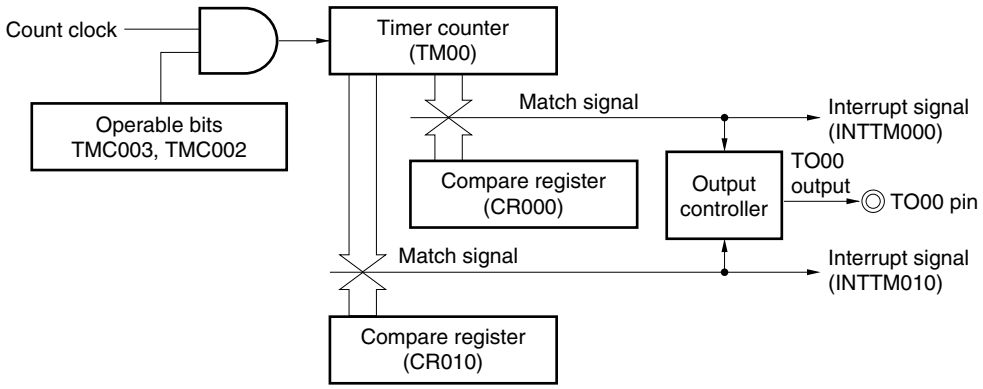
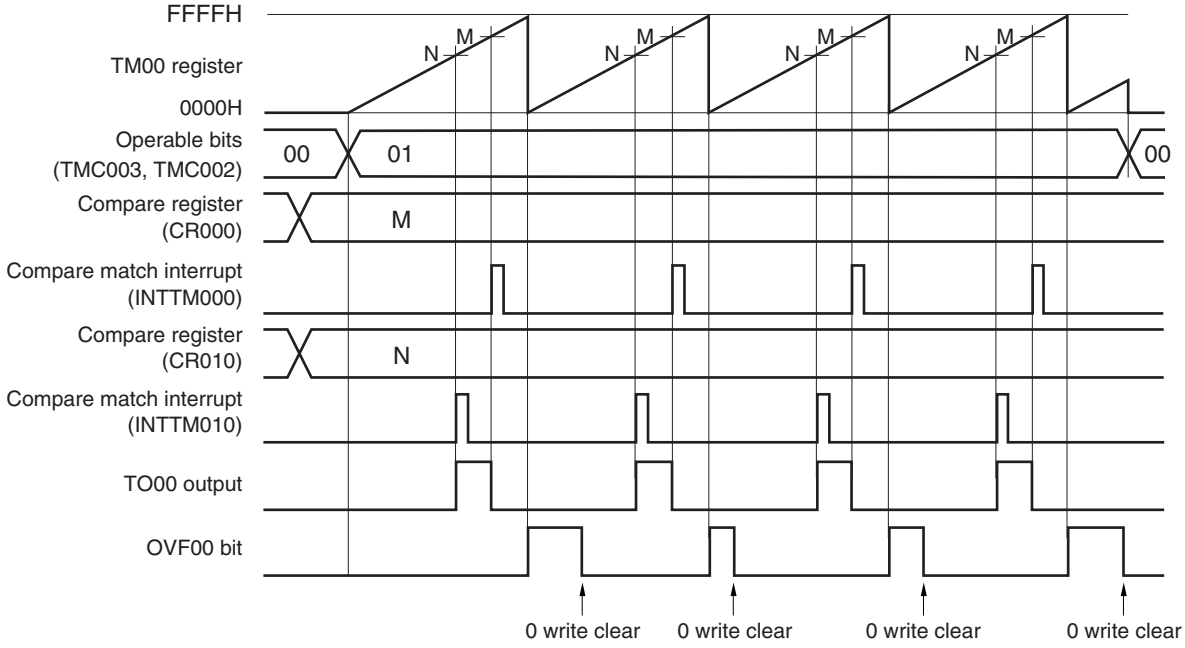


Figure 6-34. Timing Example of Free-Running Timer Mode
(CR000: Compare Register, CR010: Compare Register)

• TOC00 = 13H, PRM00 = 00H, CRC00 = 00H, TMC00 = 04H



This is an application example where two compare registers are used in the free-running timer mode.
The TO00 output level is reversed each time the count value of TM00 matches the set value of CR000 or CR010.
When the count value matches the register value, the INTTM000 or INTTM010 signal is generated.

(2) Free-running timer mode operation
(CR000: compare register, CR010: capture register)

Figure 6-35. Block Diagram of Free-Running Timer Mode
(CR000: Compare Register, CR010: Capture Register)

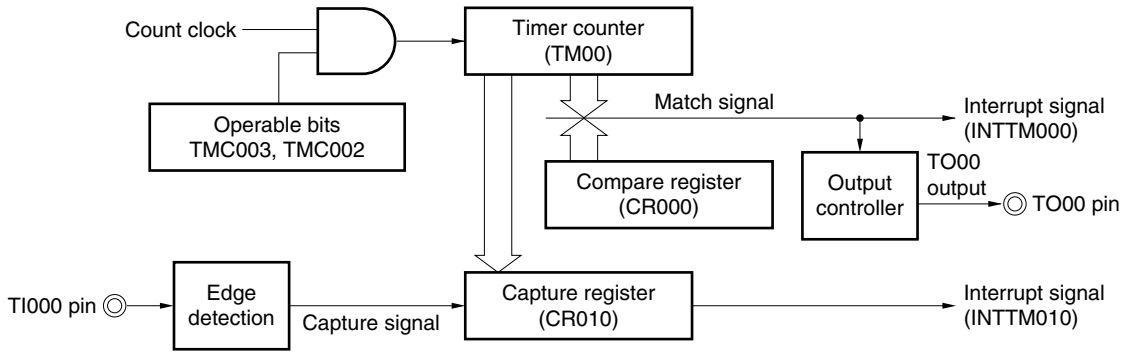
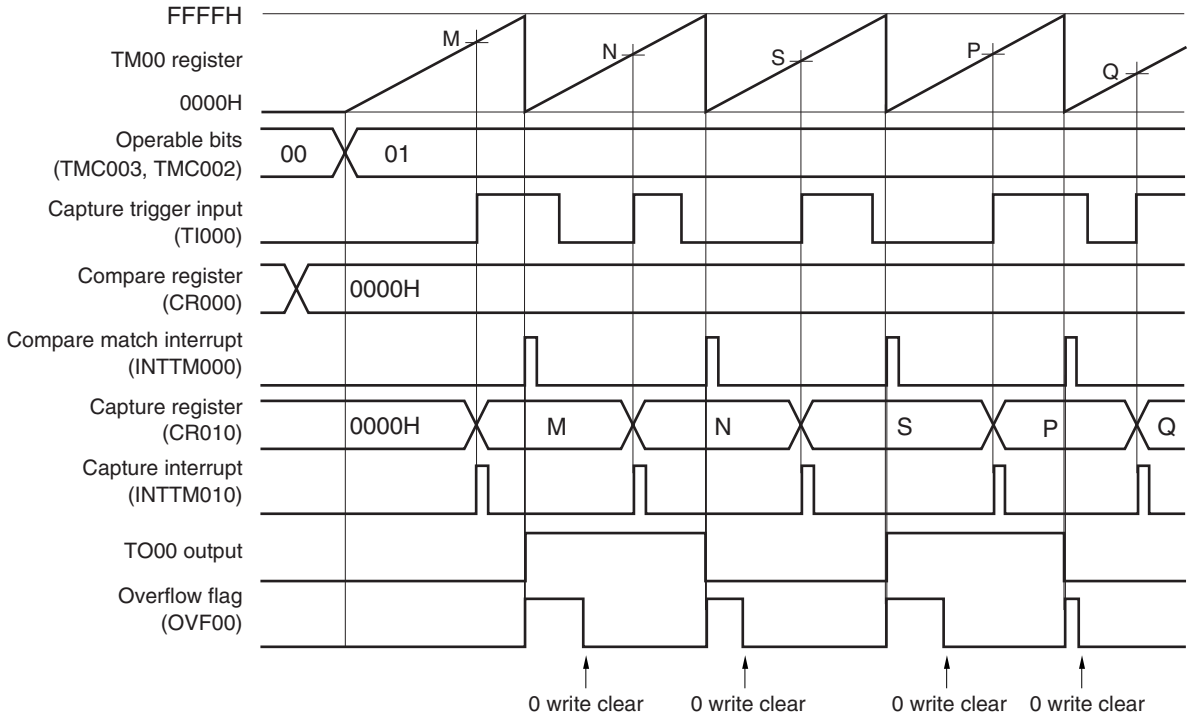


Figure 6-36. Timing Example of Free-Running Timer Mode (CR000: Compare Register, CR010: Capture Register)

• TOC00 = 13H, PRM00 = 10H, CRC00 = 04H, TMC00 = 04H

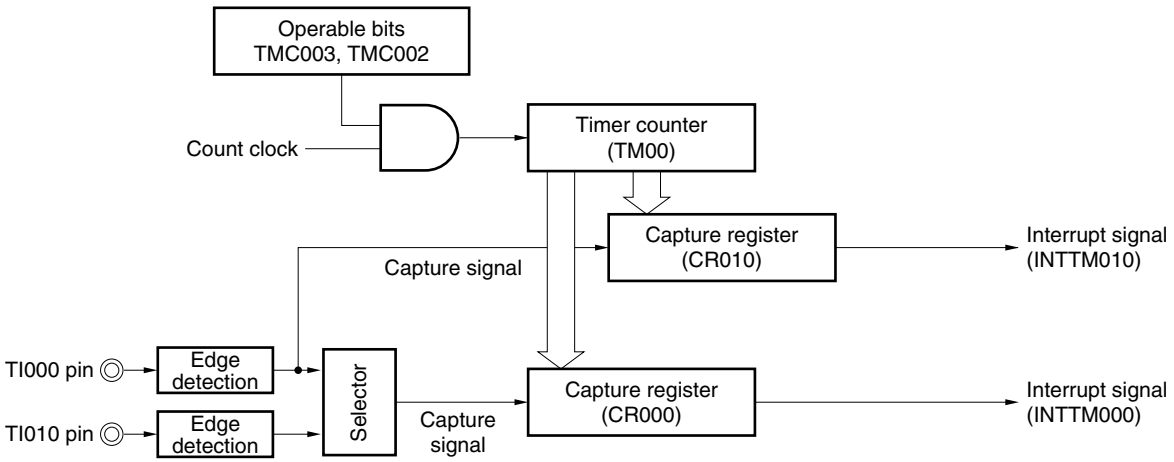


This is an application example where a compare register and a capture register are used at the same time in the free-running timer mode.

In this example, the INTTM000 signal is generated and the TO00 output is reversed each time the count value of TM00 matches the set value of CR000 (compare register). In addition, the INTTM010 signal is generated and the count value of TM00 is captured to CR010 each time the valid edge of the TI000 pin is detected.

(3) Free-running timer mode operation
(CR000: capture register, CR010: capture register)

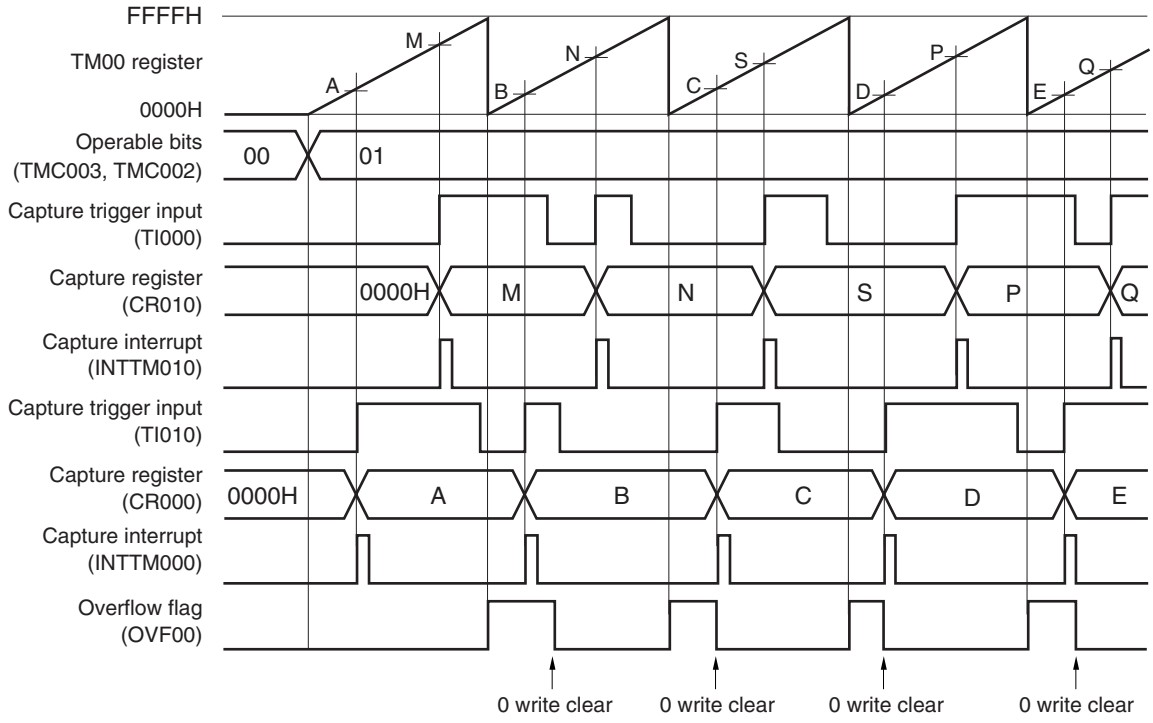
Figure 6-37. Block Diagram of Free-Running Timer Mode
(CR000: Capture Register, CR010: Capture Register)



Remark If both CR000 and CR010 are used as capture registers in the free-running timer mode, the TO00 output level is not inverted.
However, it can be inverted each time the valid edge of the TI000 pin is detected if bit 1 (TMC001) of 16-bit timer mode control register 00 (TMC00) is set to 1.

Figure 6-38. Timing Example of Free-Running Timer Mode (CR000: Capture Register, CR010: Capture Register) (1/2)

(a) TOC00 = 13H, PRM00 = 50H, CRC00 = 05H, TMC00 = 04H

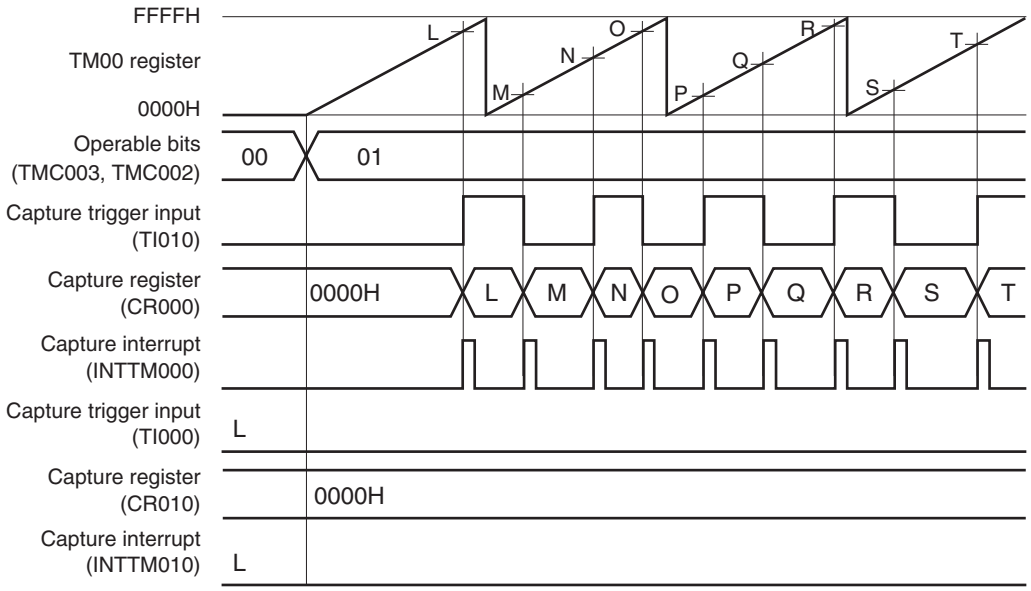


This is an application example where the count values that have been captured at the valid edges of separate capture trigger signals are stored in separate capture registers in the free-running timer mode.

The count value is captured to CR010 when the valid edge of the TI000 pin input is detected and to CR000 when the valid edge of the TI010 pin input is detected.

Figure 6-38. Timing Example of Free-Running Timer Mode (CR000: Capture Register, CR010: Capture Register) (2/2)

(b) TOC00 = 13H, PRM00 = C0H, CRC00 = 05H, TMC00 = 04H

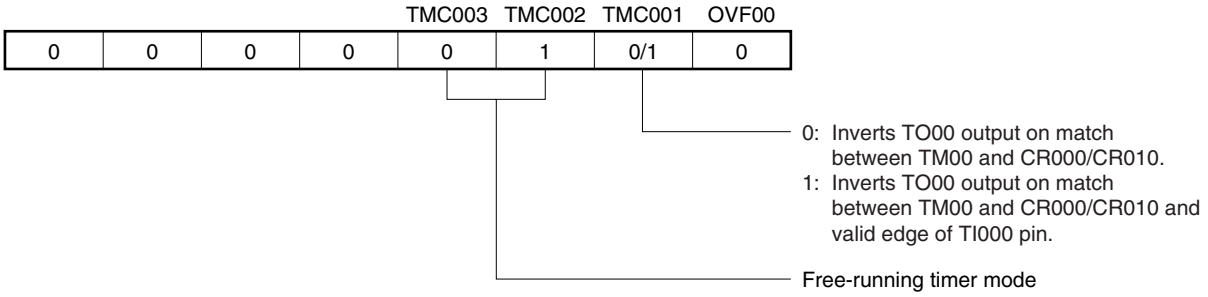


This is an application example where both the edges of the TI010 pin are detected and the count value is captured to CR000 in the free-running timer mode.

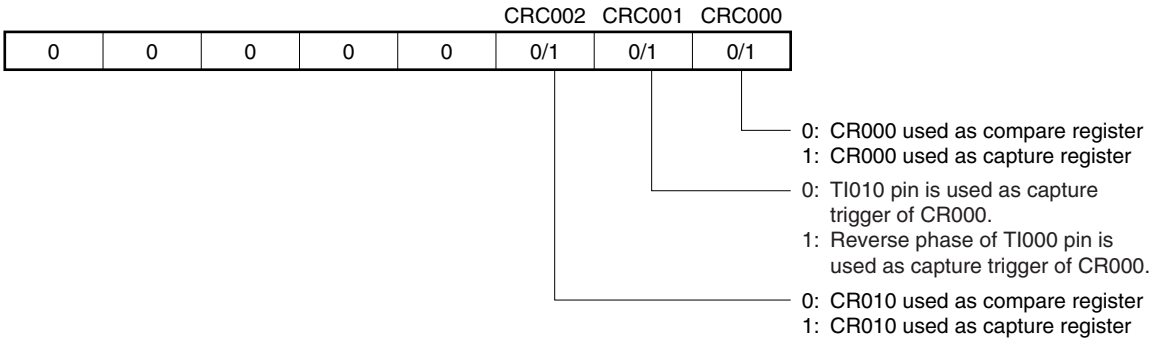
When both CR000 and CR010 are used as capture registers and when the valid edge of only the TI010 pin is to be detected, the count value cannot be captured to CR010.

Figure 6-39. Example of Register Settings in Free-Running Timer Mode (1/2)

(a) 16-bit timer mode control register 00 (TMC00)



(b) Capture/compare control register 00 (CRC00)



(c) 16-bit timer output control register 00 (TOC00)

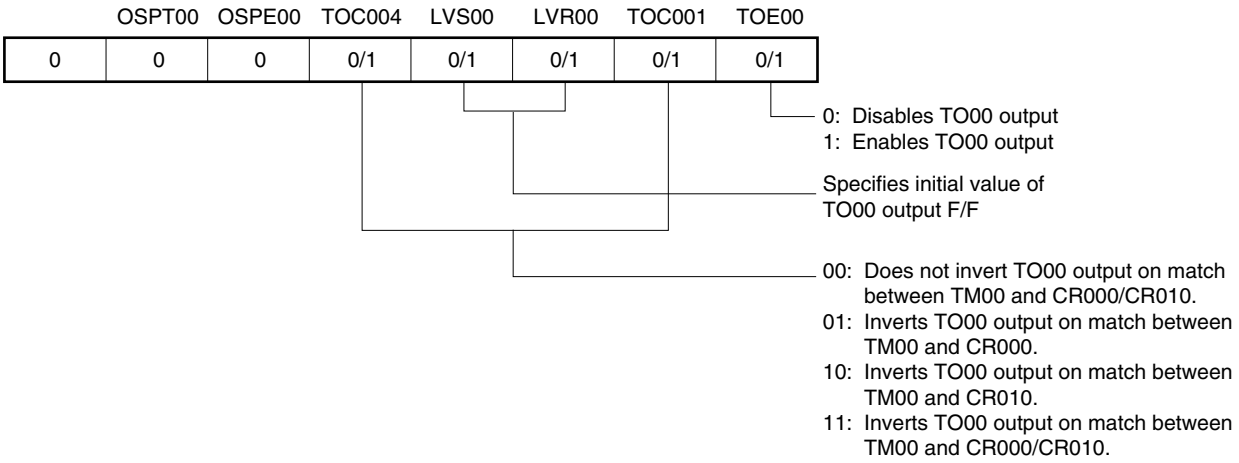
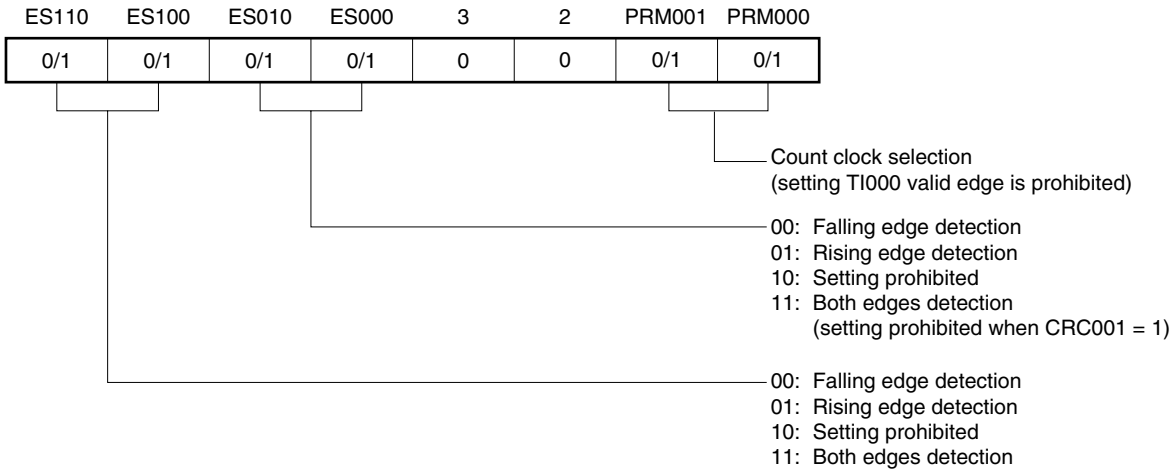


Figure 6-39. Example of Register Settings in Free-Running Timer Mode (2/2)

(d) Prescaler mode register 00 (PRM00)



(e) 16-bit timer counter 00 (TM00)

By reading TM00, the count value can be read.

(f) 16-bit capture/compare register 000 (CR000)

When this register is used as a compare register and when its value matches the count value of TM00, an interrupt signal (INTTM000) is generated. The count value of TM00 is not cleared.

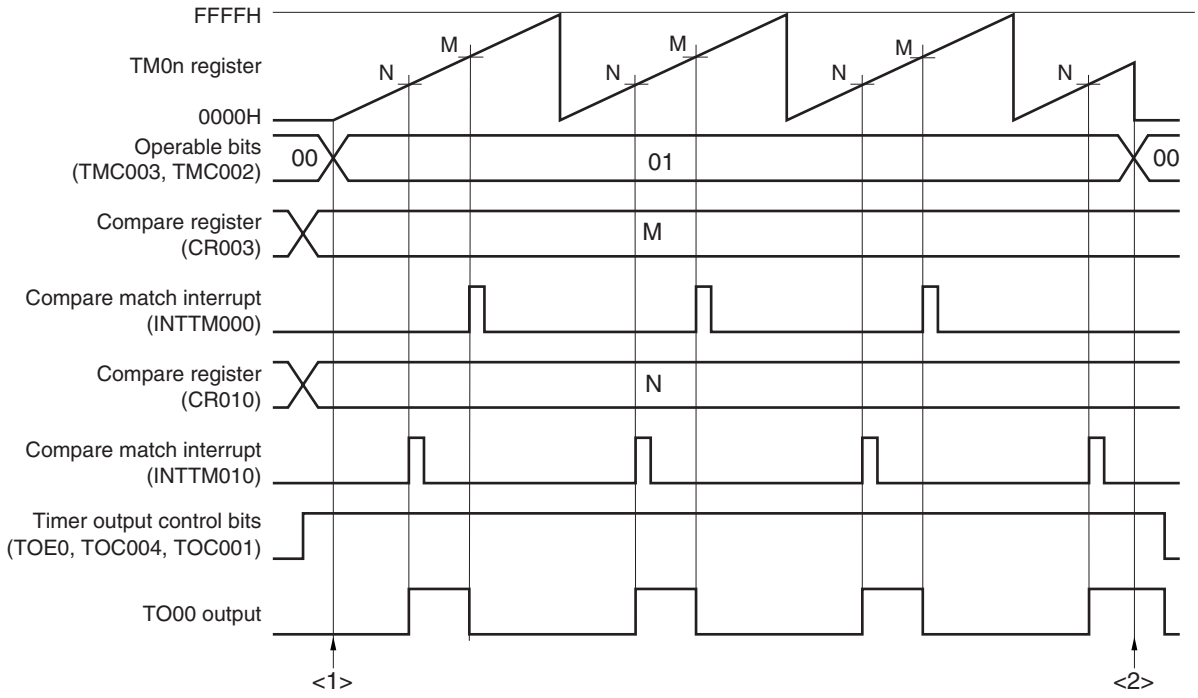
To use this register as a capture register, select either the TI000 or TI010 pin input as a capture trigger. When the valid edge of the capture trigger is detected, the count value of TM00 is stored in CR000.

(g) 16-bit capture/compare register 010 (CR010)

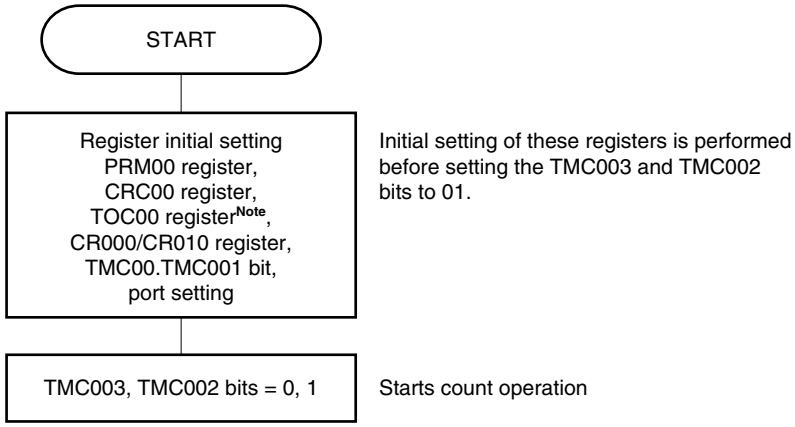
When this register is used as a compare register and when its value matches the count value of TM00, an interrupt signal (INTTM010) is generated. The count value of TM00 is not cleared.

When this register is used as a capture register, the TI000 pin input is used as a capture trigger. When the valid edge of the capture trigger is detected, the count value of TM00 is stored in CR010.

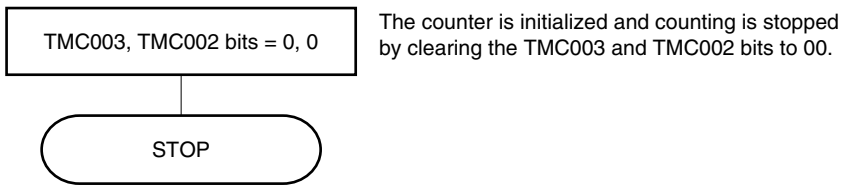
Figure 6-40. Example of Software Processing in Free-Running Timer Mode



<1> Count operation start flow



<2> Count operation stop flow



Note Care must be exercised when setting TOC00. For details, see **6.3 (3) 16-bit timer output control register 00 (TOC00)**.

6.4.6 PPG output operation

A square wave having a pulse width set in advance by CR010 is output from the TO00 pin as a PPG (Programmable Pulse Generator) signal during a cycle set by CR000 when bits 3 and 2 (TMC003 and TMC002) of 16-bit timer mode control register 00 (TMC00) are set to 11 (clear & start upon a match between TM00 and CR000).

The pulse cycle and duty factor of the pulse generated as the PPG output are as follows.

- Pulse cycle = (Set value of CR000 + 1) × Count clock cycle
- Duty = (Set value of CR010 + 1) / (Set value of CR000 + 1)

Caution To change the duty factor (value of CR010) during operation, see 6.5.1 Rewriting CR010 during TM00 operation.

Remarks 1. For the setting of I/O pins, see **6.3 (5) Port mode registers 0 and 1 (PM0, PM1)**.

2. For how to enable the INTTM000 signal interrupt, see **CHAPTER 18 INTERRUPT FUNCTIONS**.

Figure 6-41. Block Diagram of PPG Output Operation

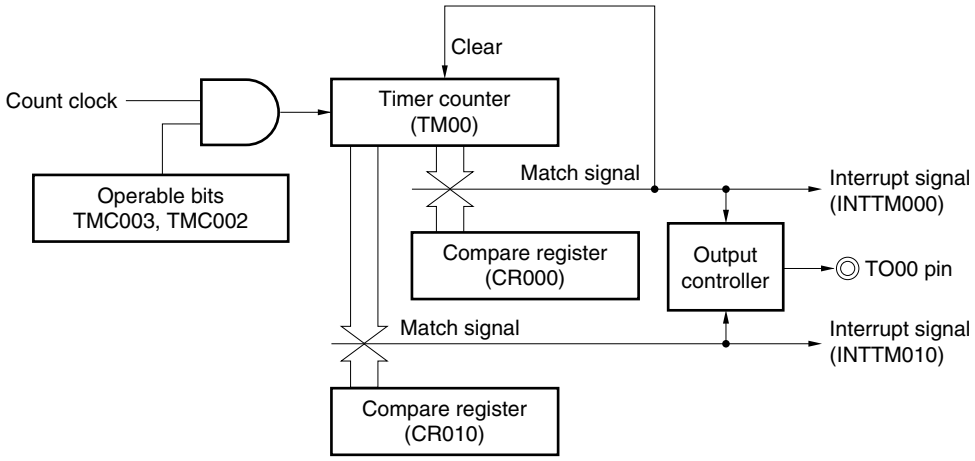
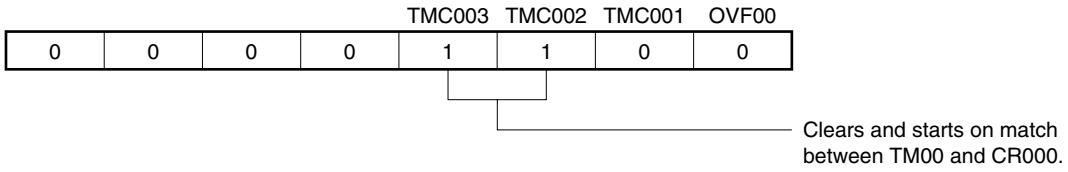
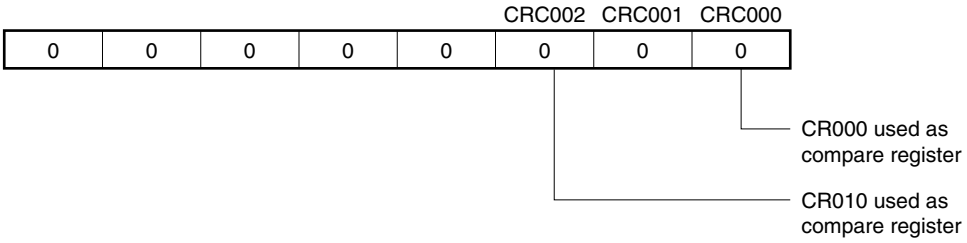


Figure 6-42. Example of Register Settings for PPG Output Operation

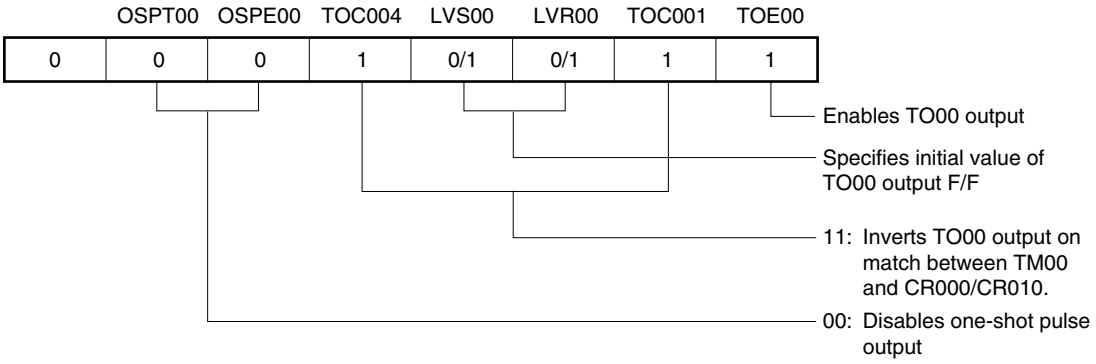
(a) 16-bit timer mode control register 00 (TMC00)



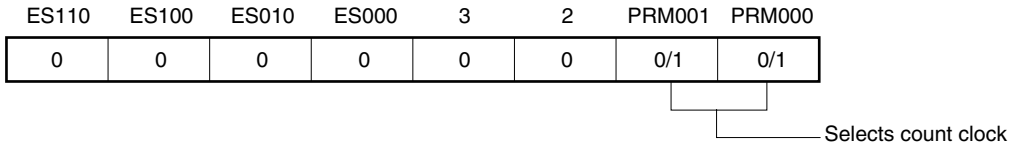
(b) Capture/compare control register 00 (CRC00)



(c) 16-bit timer output control register 00 (TOC00)



(d) Prescaler mode register 00 (PRM00)



(e) 16-bit timer counter 00 (TM00)

By reading TM00, the count value can be read.

(f) 16-bit capture/compare register 000 (CR000)

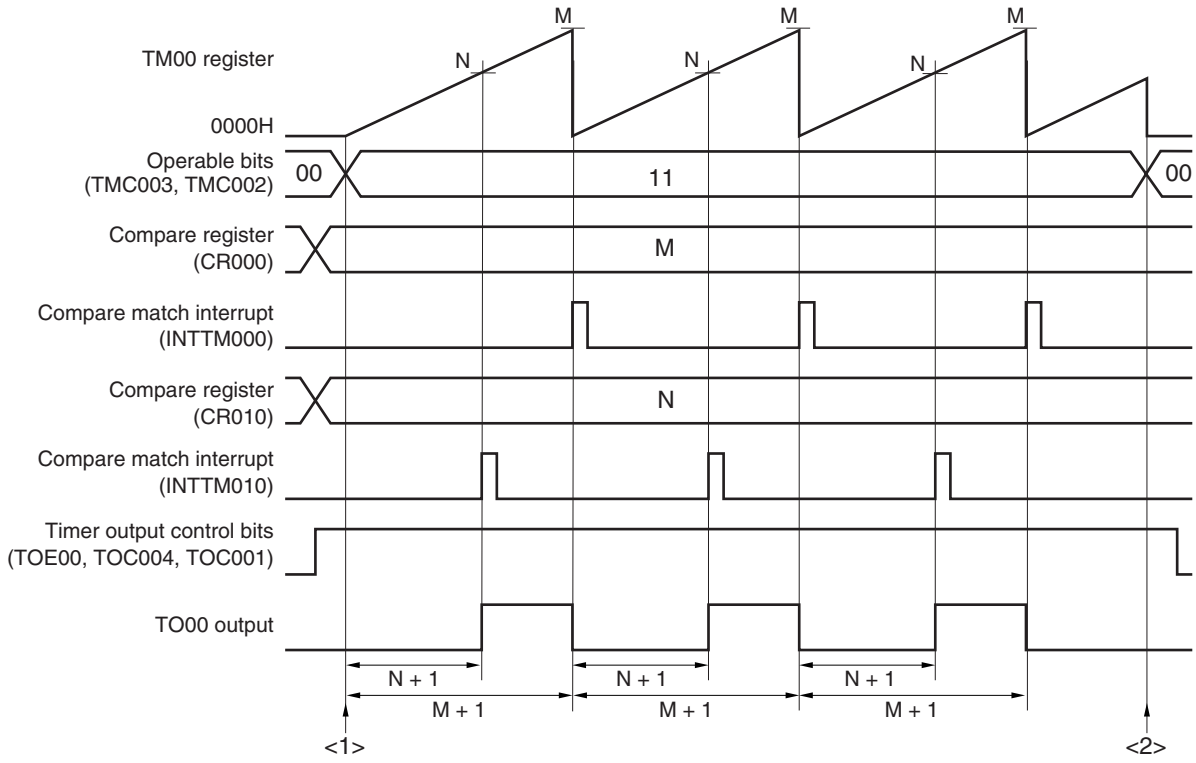
An interrupt signal (INTTM000) is generated when the value of this register matches the count value of TM00. The count value of TM00 is cleared.

(g) 16-bit capture/compare register 010 (CR010)

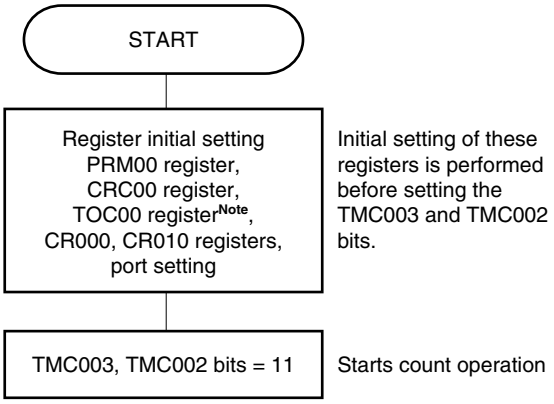
An interrupt signal (INTTM010) is generated when the value of this register matches the count value of TM00. The count value of TM00 is not cleared.

Caution Set values to CR000 and CR010 such that the condition 0000H ≤ CR010 < CR000 ≤ FFFFH is satisfied.

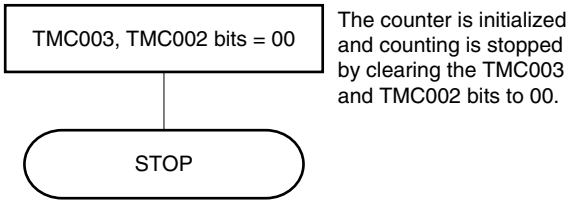
Figure 6-43. Example of Software Processing for PPG Output Operation



<1> Count operation start flow



<2> Count operation stop flow



Note Care must be exercised when setting TOC00. For details, see **6.3 (3) 16-bit timer output control register 00 (TOC00)**.

Remark PPG pulse cycle = (M + 1) × Count clock cycle
PPG duty = (N + 1)/(M + 1)

6.4.7 One-shot pulse output operation

A one-shot pulse can be output by setting bits 3 and 2 (TMC003 and TMC002) of the 16-bit timer mode control register 00 (TMC00) to 01 (free-running timer mode) or to 10 (clear & start mode entered by the TI000 pin valid edge) and setting bit 5 (OSPE00) of 16-bit timer output control register 00 (TOC00) to 1.

When bit 6 (OSPT00) of TOC00 is set to 1 or when the valid edge is input to the TI000 pin during timer operation, clearing & starting of TM00 is triggered, and a pulse of the difference between the values of CR000 and CR010 is output only once from the TO00 pin.

Cautions 1. Do not input the trigger again (setting OSPT00 to 1 or detecting the valid edge of the TI000 pin) while the one-shot pulse is output. To output the one-shot pulse again, generate the trigger after the current one-shot pulse output has completed.

2. To use only the setting of OSPT00 to 1 as the trigger of one-shot pulse output, do not change the level of the TI000 pin or its alternate function port pin. Otherwise, the pulse will be unexpectedly output.

Remarks 1. For the setting of the I/O pins, see **6.3 (5) Port mode registers 0 and 1 (PM0, PM1)**.

2. For how to enable the INTTM000 signal interrupt, see **CHAPTER 18 INTERRUPT FUNCTIONS**.

Figure 6-44. Block Diagram of One-Shot Pulse Output Operation

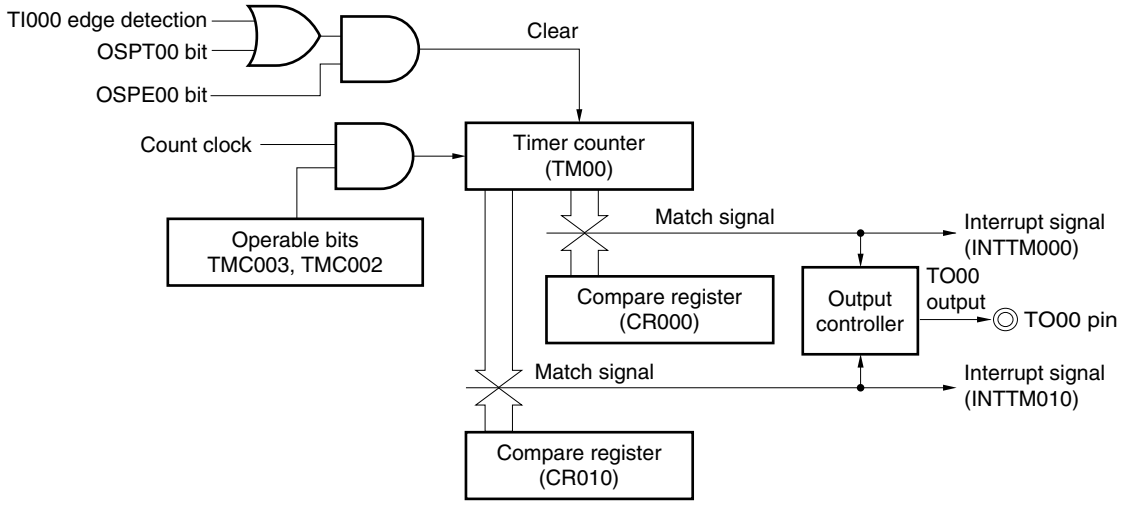
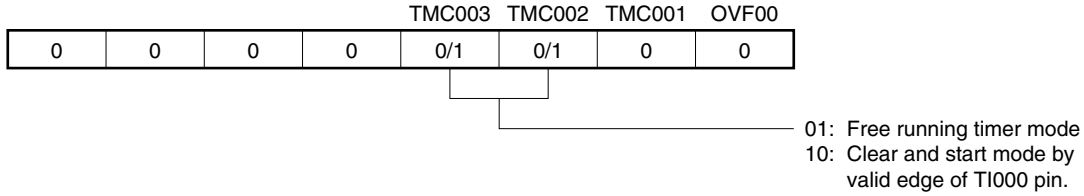
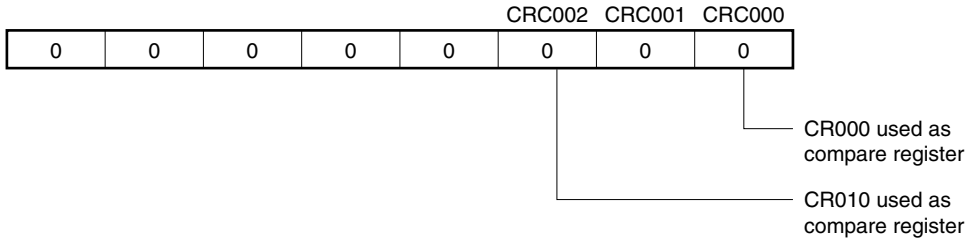


Figure 6-45. Example of Register Settings for One-Shot Pulse Output Operation (1/2)

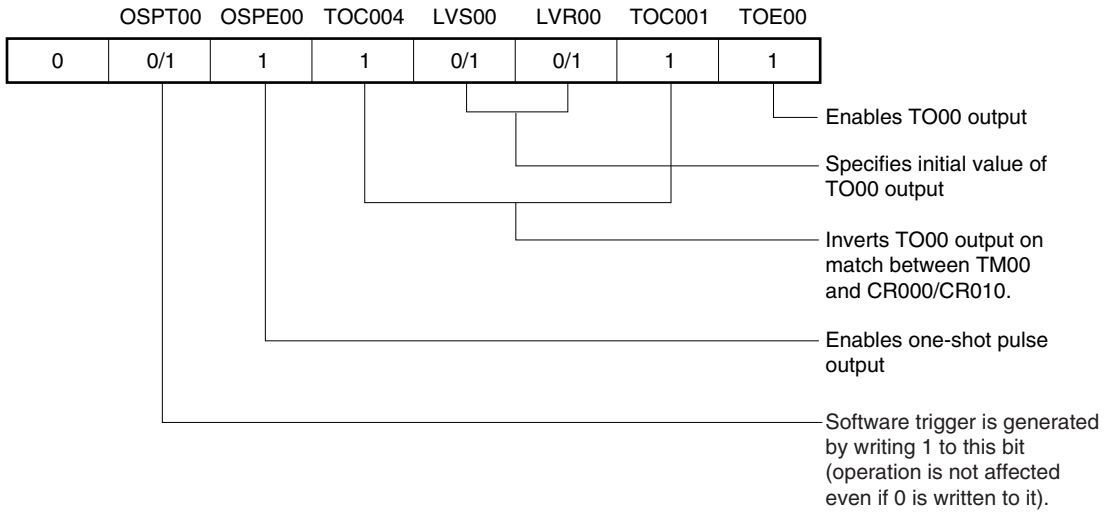
(a) 16-bit timer mode control register 00 (TMC00)



(b) Capture/compare control register 00 (CRC00)



(c) 16-bit timer output control register 00 (TOC00)



(d) Prescaler mode register 00 (PRM00)

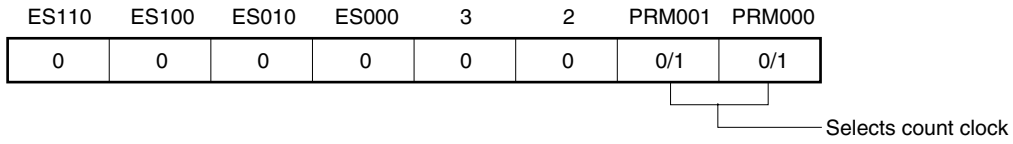


Figure 6-45. Example of Register Settings for One-Shot Pulse Output Operation (2/2)

(e) 16-bit timer counter 00 (TM00)

By reading TM00, the count value can be read.

(f) 16-bit capture/compare register 000 (CR000)

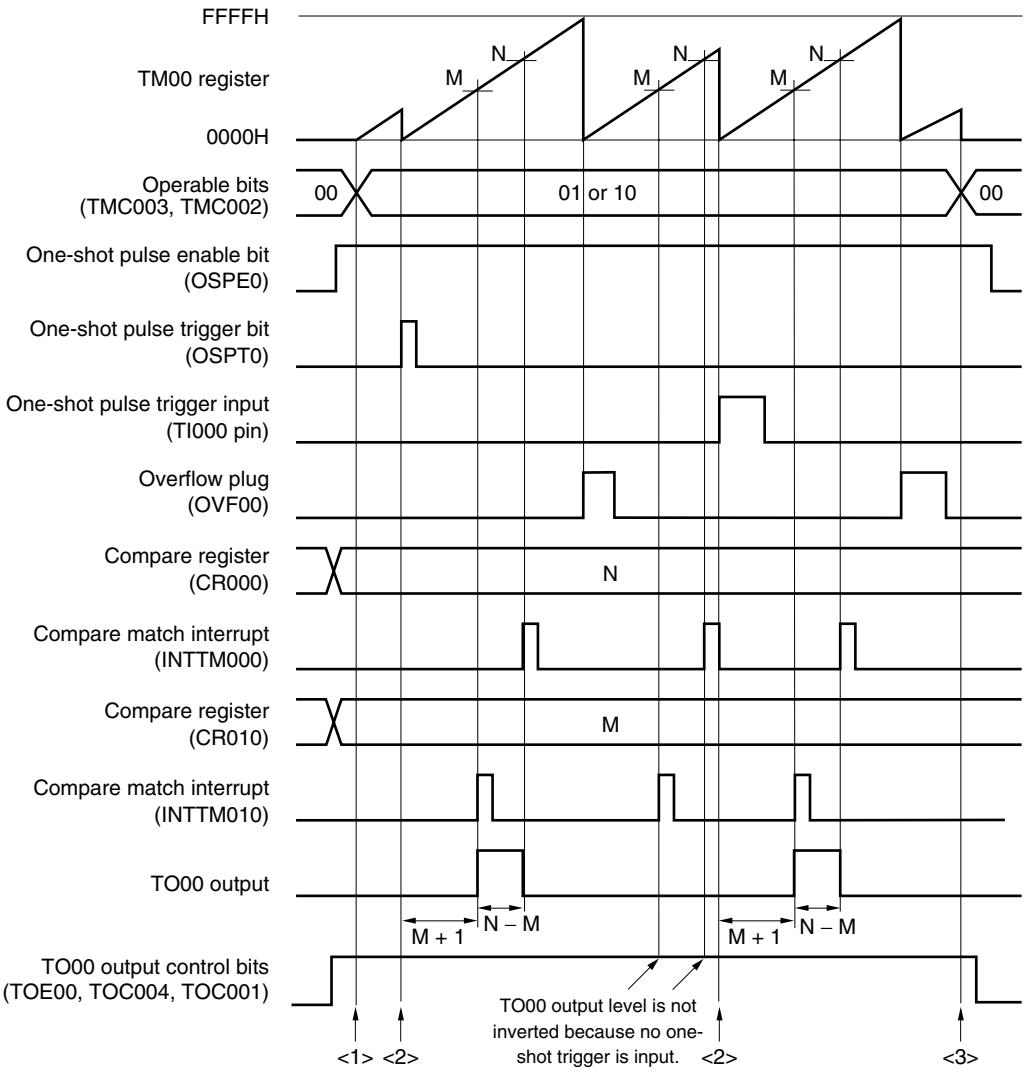
This register is used as a compare register when a one-shot pulse is output. When the value of TM00 matches that of CR000, an interrupt signal (INTTM000) is generated and the TO00 output level is inverted.

(g) 16-bit capture/compare register 010 (CR010)

This register is used as a compare register when a one-shot pulse is output. When the value of TM00 matches that of CR010, an interrupt signal (INTTM010) is generated and the TO00 output level is inverted.

Caution Do not set the same value to CR000 and CR010.

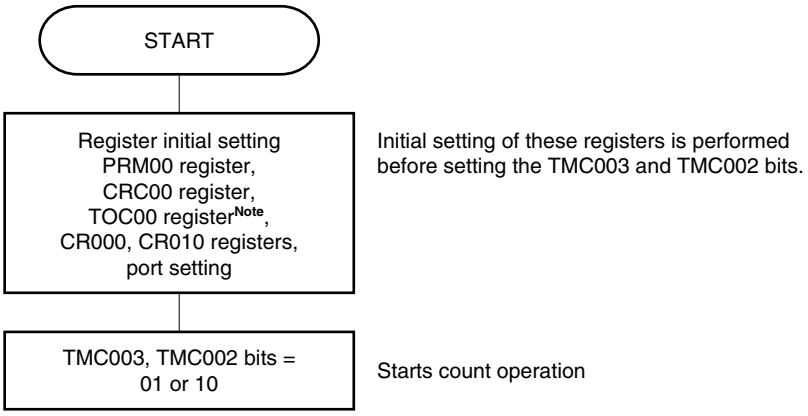
Figure 6-46. Example of Software Processing for One-Shot Pulse Output Operation (1/2)



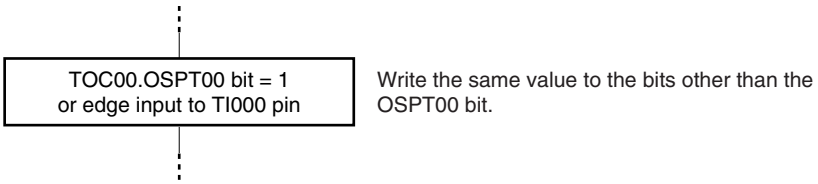
- Time from when the one-shot pulse trigger is input until the one-shot pulse is output
  = (M + 1) × Count clock cycle
- One-shot pulse output active level width
  = (N − M) × Count clock cycle

Figure 6-46. Example of Software Processing for One-Shot Pulse Output Operation (2/2)

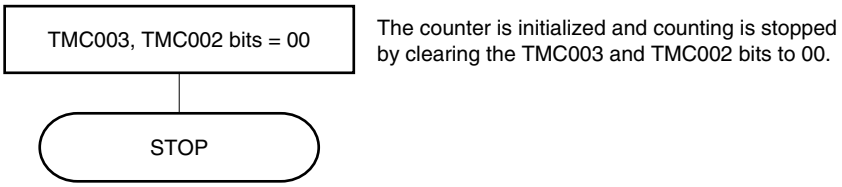
<1> Count operation start flow



<2> One-shot trigger input flow



<3> Count operation stop flow



Note Care must be exercised when setting TOC00. For details, see **6.3 (3) 16-bit timer output control register 00 (TOC00)**.

6.4.8 Pulse width measurement operation

TM00 can be used to measure the pulse width of the signal input to the TI000 and TI010 pins.

Measurement can be accomplished by operating the 16-bit timer/event counter 00 in the free-running timer mode or by restarting the timer in synchronization with the signal input to the TI000 pin.

When an interrupt is generated, read the value of the valid capture register and measure the pulse width. Check bit 0 (OVF00) of 16-bit timer mode control register 00 (TMC00). If it is set (to 1), clear it to 0 by software.

Figure 6-47. Block Diagram of Pulse Width Measurement (Free-Running Timer Mode)

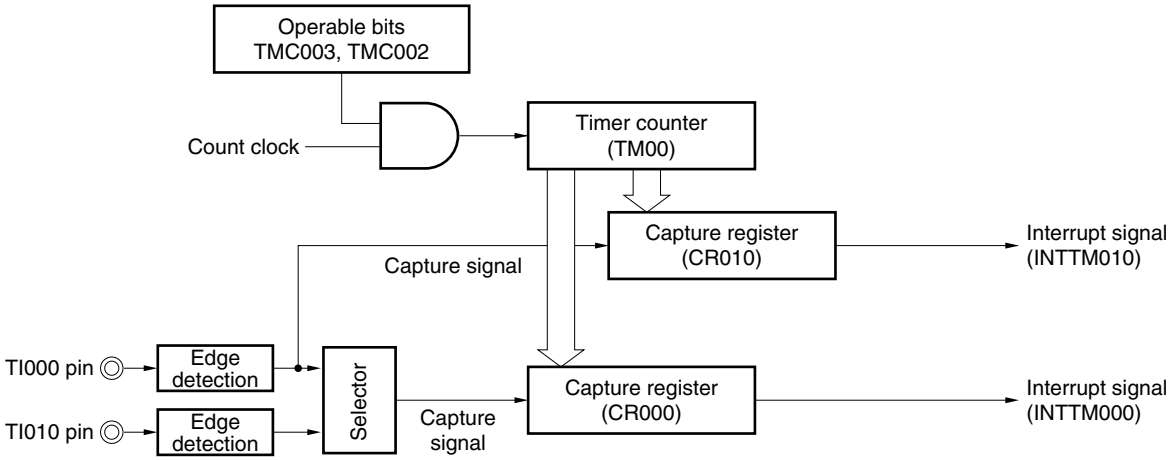
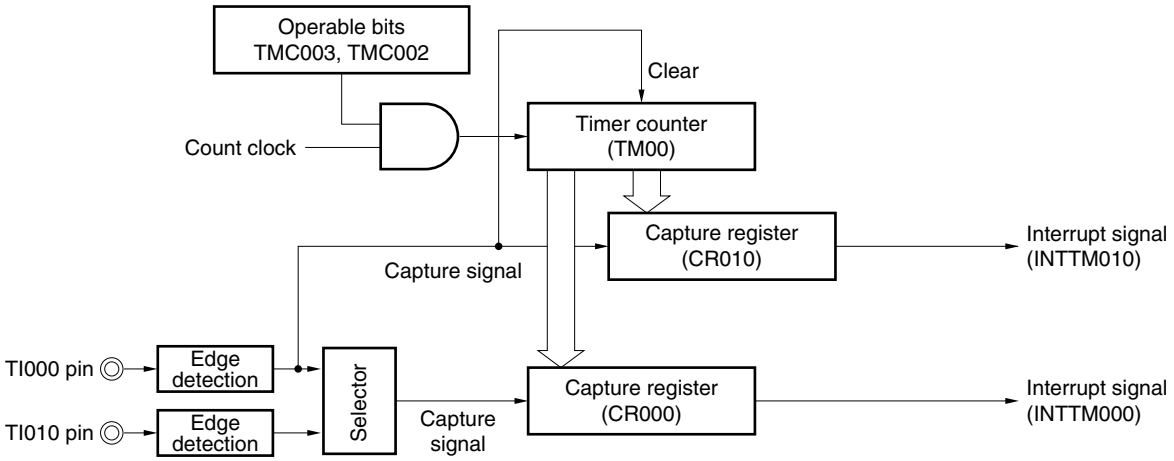


Figure 6-48. Block Diagram of Pulse Width Measurement (Clear & Start Mode Entered by TI000 Pin Valid Edge Input)



A pulse width can be measured in the following three ways.

- Measuring the pulse width by using two input signals of the TI000 and TI010 pins (free-running timer mode)
- Measuring the pulse width by using one input signal of the TI000 pin (free-running timer mode)
- Measuring the pulse width by using one input signal of the TI000 pin (clear & start mode entered by the TI000 pin valid edge input)

Remarks 1. For the setting of the I/O pins, see **6.3 (5) Port mode registers 0 and 1 (PM0, PM1)**.

2. For how to enable the INTTM000 signal interrupt, see **CHAPTER 18 INTERRUPT FUNCTIONS**.

(1) Measuring the pulse width by using two input signals of the TI000 and TI010 pins (free-running timer mode)

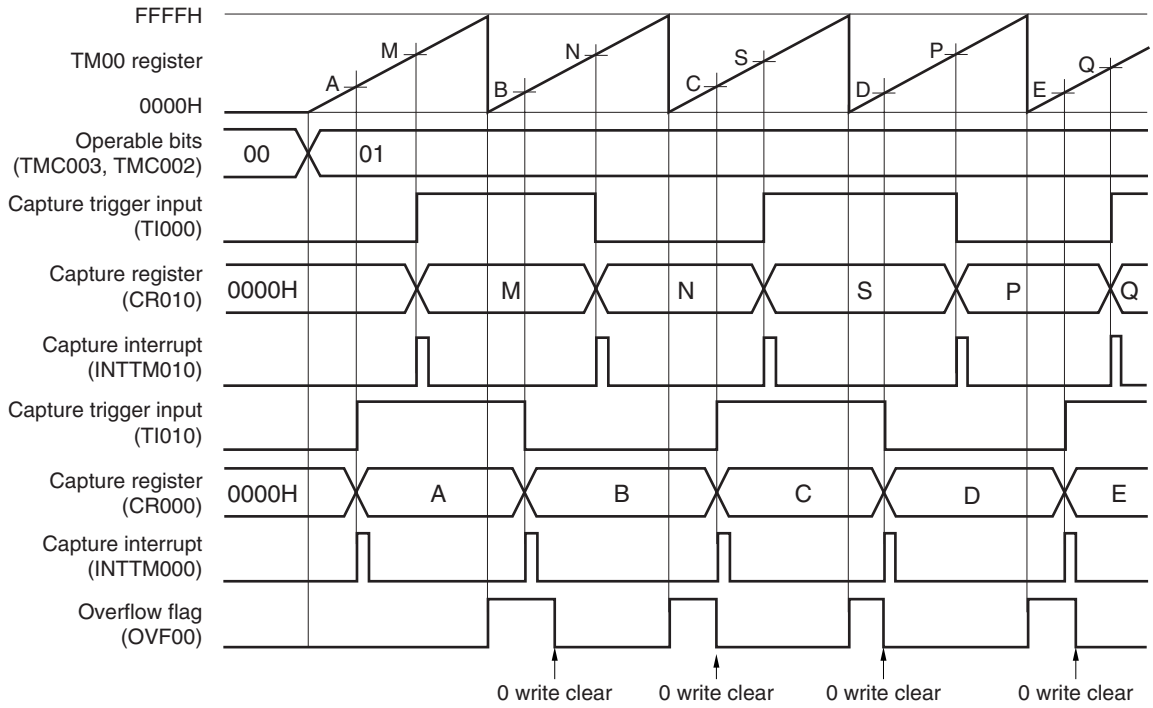
Set the free-running timer mode (TMC003 and TMC002 = 01). When the valid edge of the TI000 pin is detected, the count value of TM00 is captured to CR010. When the valid edge of the TI010 pin is detected, the count value of TM00 is captured to CR000. Specify detection of both the edges of the TI000 and TI010 pins.

By this measurement method, the previous count value is subtracted from the count value captured by the edge of each input signal. Therefore, save the previously captured value to a separate register in advance.

If an overflow occurs, the value becomes negative if the previously captured value is simply subtracted from the current captured value and, therefore, a borrow occurs (bit 0 (CY) of the program status word (PSW) is set to 1). If this happens, ignore CY and take the calculated value as the pulse width. In addition, clear bit 0 (OVF00) of 16-bit timer mode control register 00 (TMC00) to 0.

Figure 6-49. Timing Example of Pulse Width Measurement (1)

• TMC00 = 04H, PRM00 = F0H, CRC00 = 05H



(2) Measuring the pulse width by using one input signal of the TI000 pin (free-running mode)

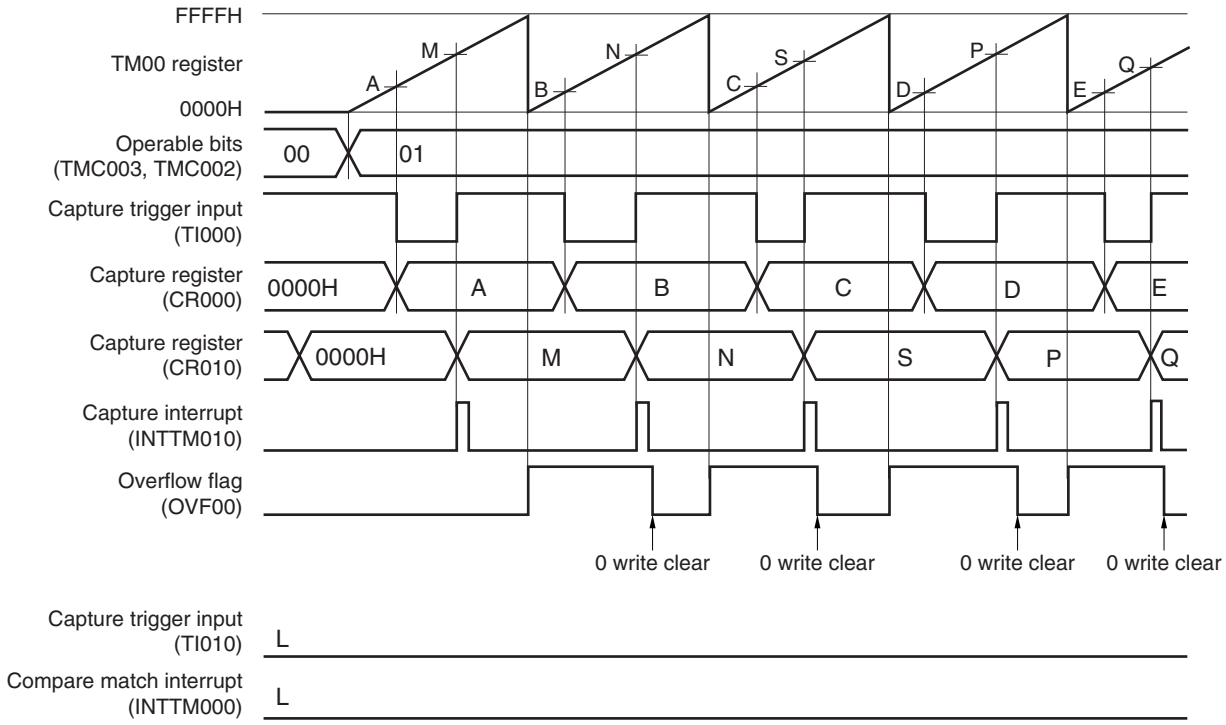
Set the free-running timer mode (TMC003 and TMC002 = 01). The count value of TM00 is captured to CR000 in the phase reverse to the valid edge detected on the TI000 pin. When the valid edge of the TI000 pin is detected, the count value of TM00 is captured to CR010.

By this measurement method, values are stored in separate capture registers when a width from one edge to another is measured. Therefore, the capture values do not have to be saved. By subtracting the value of one capture register from that of another, a high-level width, low-level width, and cycle are calculated.

If an overflow occurs, the value becomes negative if one captured value is simply subtracted from another and, therefore, a borrow occurs (bit 0 (CY) of the program status word (PSW) is set to 1). If this happens, ignore CY and take the calculated value as the pulse width. In addition, clear bit 0 (OVF00) of 16-bit timer mode control register 00 (TMC00) to 0.

Figure 6-50. Timing Example of Pulse Width Measurement (2)

• TMC00 = 04H, PRM00 = 10H, CRC00 = 07H



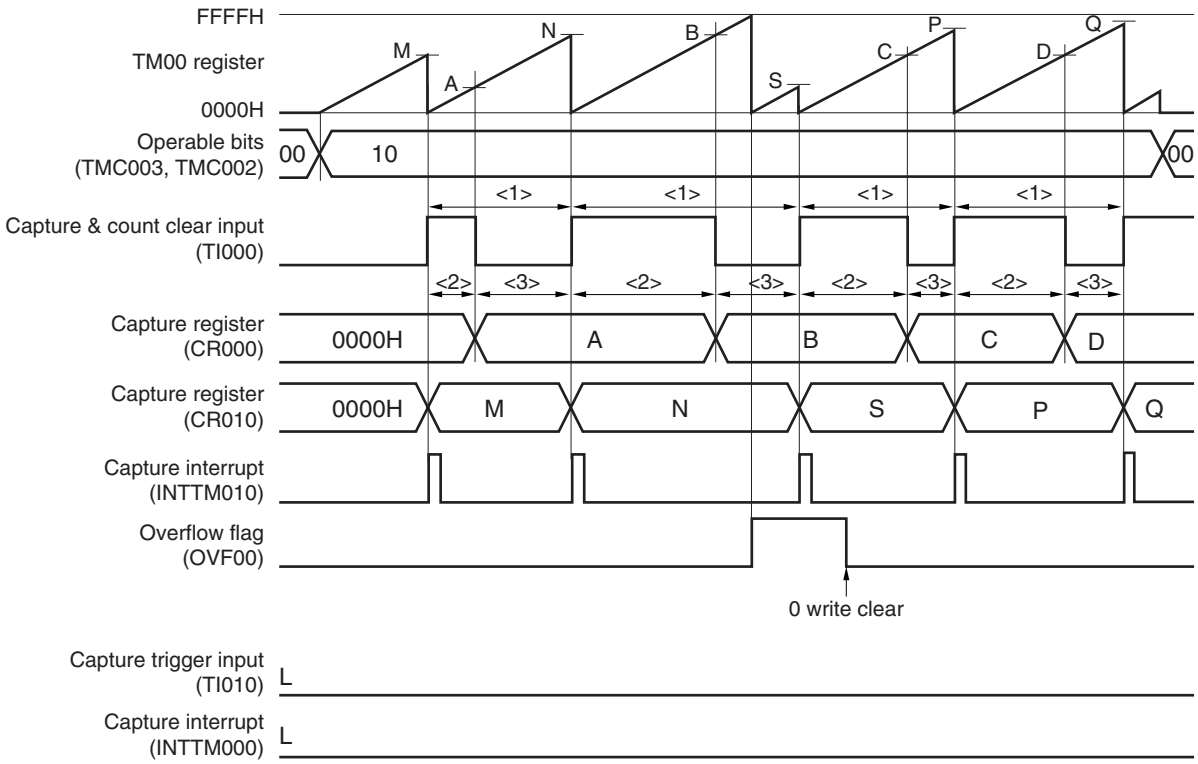
(3) Measuring the pulse width by using one input signal of the TI000 pin (clear & start mode entered by the TI000 pin valid edge input)

Set the clear & start mode entered by the TI000 pin valid edge (TMC003 and TMC002 = 10). The count value of TM00 is captured to CR000 in the phase reverse to the valid edge of the TI000 pin, and the count value of TM00 is captured to CR010 and TM00 is cleared (0000H) when the valid edge of the TI000 pin is detected. Therefore, a cycle is stored in CR010 if TM00 does not overflow.

If an overflow occurs, take the value that results from adding 10000H to the value stored in CR010 as a cycle. Clear bit 0 (OVF00) of 16-bit timer mode control register 00 (TMC00) to 0.

Figure 6-51. Timing Example of Pulse Width Measurement (3)

• TMC00 = 08H, PRM00 = 10H, CRC00 = 07H



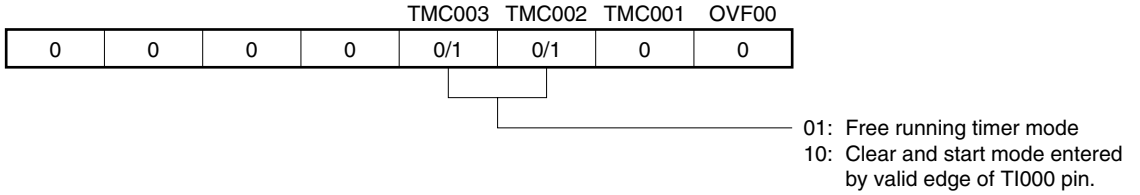
<1> Pulse cycle = (10000H × Number of times OVF00 bit is set to 1 + Captured value of CR010) × Count clock cycle

<2> High-level pulse width = (10000H × Number of times OVF00 bit is set to 1 + Captured value of CR000) × Count clock cycle

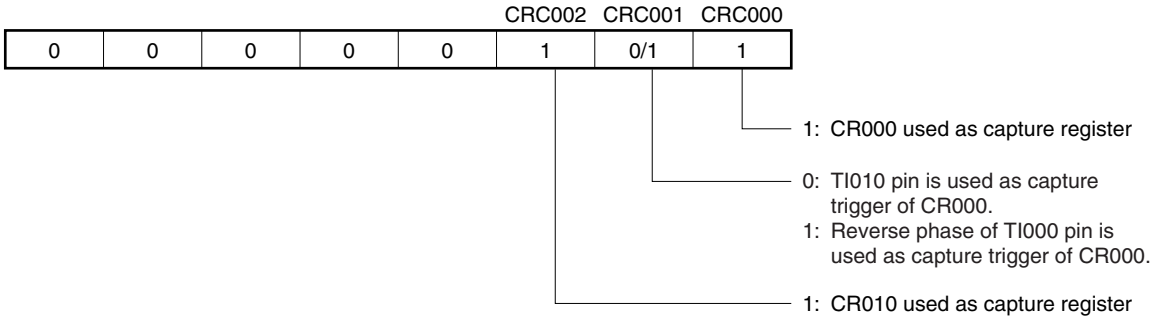
<3> Low-level pulse width = (Pulse cycle − High-level pulse width)

Figure 6-52. Example of Register Settings for Pulse Width Measurement (1/2)

(a) 16-bit timer mode control register 00 (TMC00)



(b) Capture/compare control register 00 (CRC00)



(c) 16-bit timer output control register 00 (TOC00)

	OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00
0	0	0	0	0	0	0	0

(d) Prescaler mode register 00 (PRM00)

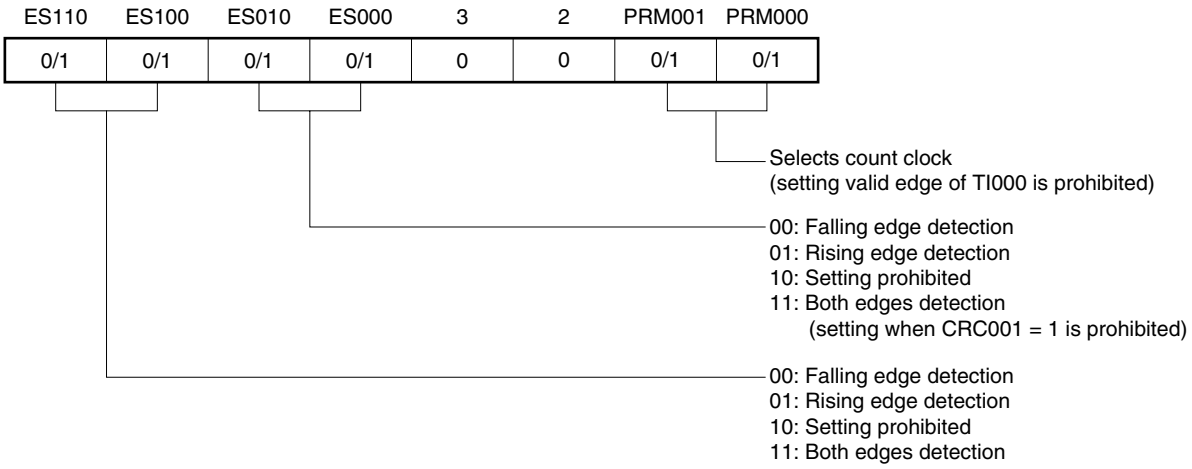


Figure 6-52. Example of Register Settings for Pulse Width Measurement (2/2)

(e) 16-bit timer counter 00 (TM00)

By reading TM00, the count value can be read.

(f) 16-bit capture/compare register 000 (CR000)

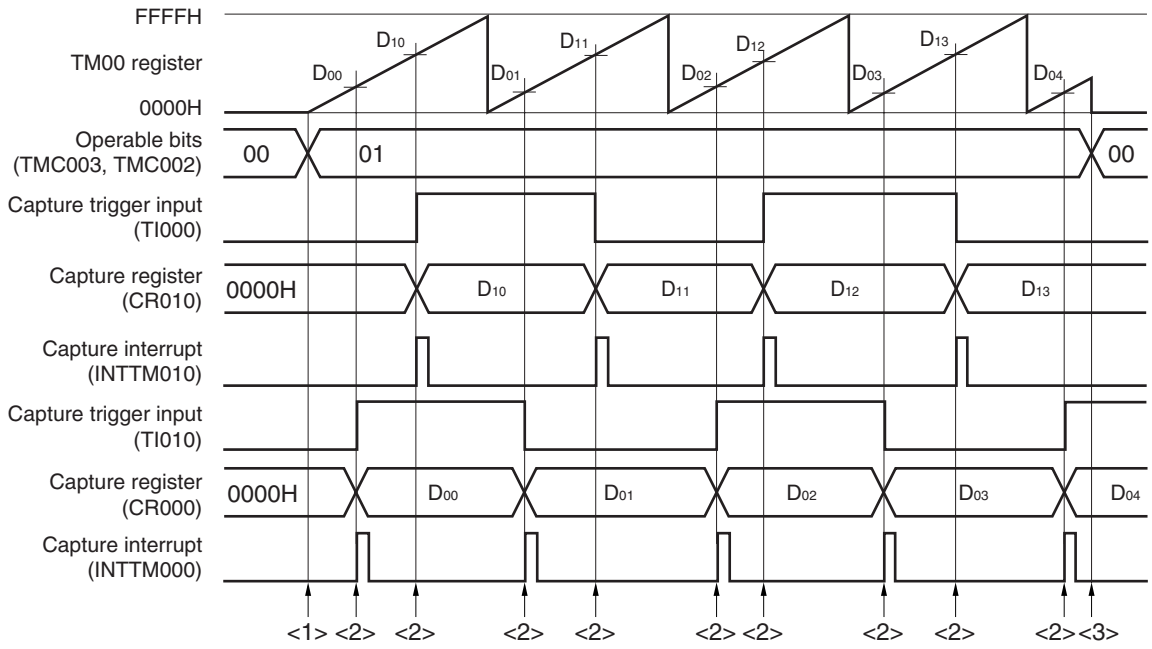
This register is used as a capture register. Either the TI000 or TI010 pin is selected as a capture trigger. When a specified edge of the capture trigger is detected, the count value of TM00 is stored in CR000.

(g) 16-bit capture/compare register 010 (CR010)

This register is used as a capture register. The signal input to the TI000 pin is used as a capture trigger. When the capture trigger is detected, the count value of TM00 is stored in CR010.

Figure 6-53. Example of Software Processing for Pulse Width Measurement (1/2)

(a) Example of free-running timer mode



(b) Example of clear & start mode entered by TI000 pin valid edge

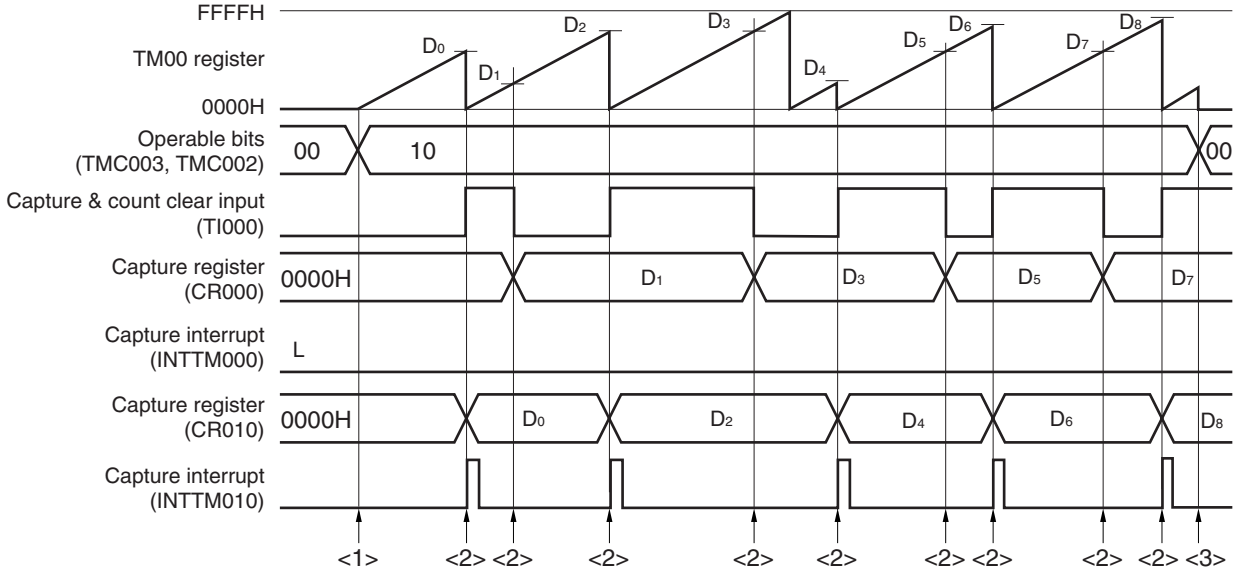
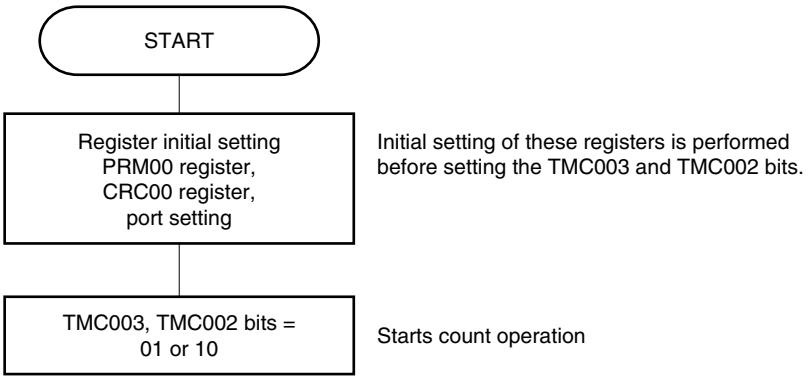
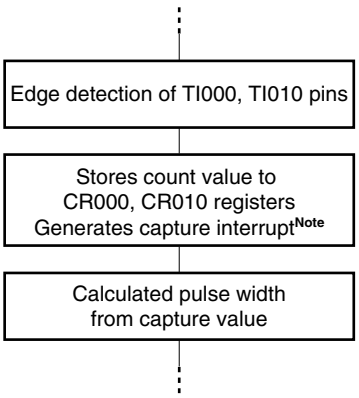


Figure 6-53. Example of Software Processing for Pulse Width Measurement (2/2)

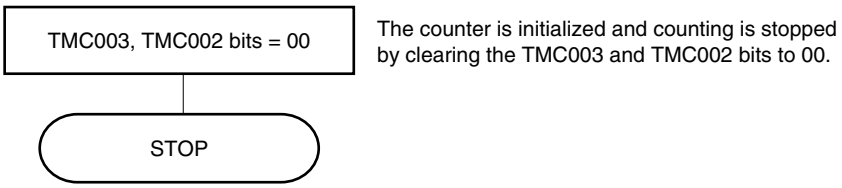
<1> Count operation start flow



<2> Capture trigger input flow



<3> Count operation stop flow



Note The capture interrupt signal (INTTM000) is not generated when the reverse-phase edge of the TI000 pin input is selected to the valid edge of CR000.

6.5 Special Use of TM00

6.5.1 Rewriting CR010 during TM00 operation

In principle, rewriting CR000 and CR010 of the 78K0/Kx2-A microcontrollers when they are used as compare registers is prohibited while TM00 is operating (TMC003 and TMC002 = other than 00).

However, the value of CR010 can be changed, even while TM00 is operating, using the following procedure if CR010 is used for PPG output and the duty factor is changed (when setting CR010 to a smaller or larger value than the current value, rewrite the CR010 value immediately after a match between CR010 and TM00 or between CR000 and TM00. When CR010 is rewritten immediately before a match between CR010 and TM00 or between CR000 and TM00, an unexpected operation may be performed).

Procedure for changing value of CR010

<1> Disable interrupt INTTM010 (TMMK010 = 1).
<2> Disable reversal of the timer output when the value of TM00 matches that of CR010 (TOC004 = 0).
<3> Change the value of CR010.
<4> Wait for one cycle of the count clock of TM00.
<5> Enable reversal of the timer output when the value of TM00 matches that of CR010 (TOC004 = 1).
<6> Clear the interrupt flag of INTTM010 (TMIF010 = 0) to 0.
<7> Enable interrupt INTTM010 (TMMK010 = 0).

Remark For TMIF010 and TMMK010, see **CHAPTER 18 INTERRUPT FUNCTIONS**.

6.5.2 Setting LVS00 and LVR00

(1) Usage of LVS00 and LVR00

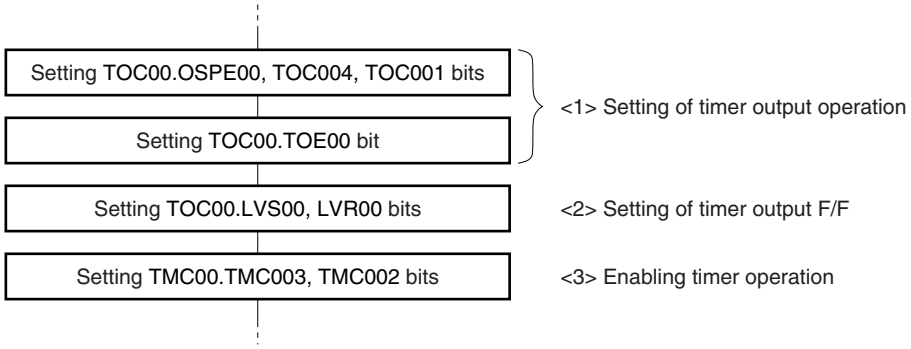
LVS00 and LVR00 are used to set the default value of the TO00 output and to invert the timer output without enabling the timer operation (TMC003 and TMC002 = 00). Clear LVS00 and LVR00 to 00 (default value: low-level output) when software control is unnecessary.

LVS00	LVR00	Timer Output Status
0	0	Not changed (low-level output)
0	1	Cleared (low-level output)
1	0	Set (high-level output)
1	1	Setting prohibited

(2) Setting LVS00 and LVR00

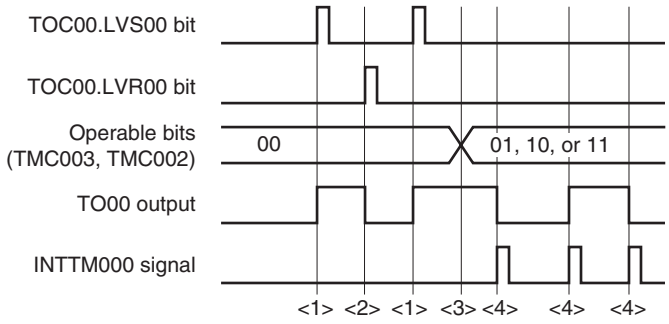
Set LVS00 and LVR00 using the following procedure.

Figure 6-54. Example of Flow for Setting LVS00 and LVR00 Bits



Caution Be sure to set LVS00 and LVR00 following steps <1>, <2>, and <3> above. Step <2> can be performed after <1> and before <3>.

Figure 6-55. Timing Example of LVR00 and LVS00



<1> The TO00 output goes high when LVS00 and LVR00 = 10.

<2> The TO00 output goes low when LVS00 and LVR00 = 01 (the pin output remains unchanged from the high level even if LVS00 and LVR00 are cleared to 00).

<3> The timer starts operating when TMC003 and TMC002 are set to 01, 10, or 11. Because LVS00 and LVR00 were set to 10 before the operation was started, the TO00 output starts from the high level. After the timer starts operating, setting LVS00 and LVR00 is prohibited until TMC003 and TMC002 = 00 (disabling the timer operation).

<4> The TO00 output level is inverted each time an interrupt signal (INTTM000) is generated.

6.6 Cautions for 16-Bit Timer/Event Counter 00

(1) Restrictions for each channel of 16-bit timer/event counter 00

Table 6-3 shows the restrictions for each channel.

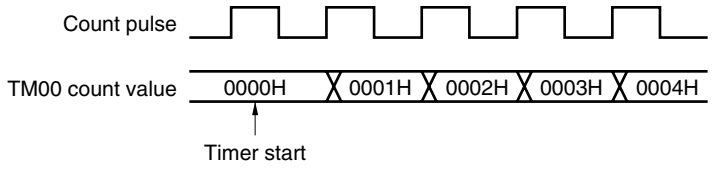
Table 6-3. Restrictions for Each Channel of 16-Bit Timer/Event Counter 00

Operation	Restriction
As interval timer	–
As square wave output	
As external event counter	
As clear & start mode entered by TI000 pin valid edge input	Using timer output (TO00) is prohibited when detection of the valid edge of the TI010 pin is used. (TOC00 = 00H)
As free-running timer	–
As PPG output	0000H ≤ CP010 < CR000 ≤ FFFFH
As one-shot pulse output	Setting the same value to CR000 and CP010 is prohibited.
As pulse width measurement	Using timer output (TO00) is prohibited (TOC00 = 00H)

(2) Timer start errors

An error of up to one clock may occur in the time required for a match signal to be generated after timer start. This is because counting TM00 is started asynchronously to the count pulse.

Figure 6-56. Start Timing of TM00 Count



(3) Setting of CR000 and CR010 (clear & start mode entered upon a match between TM00 and CR000)

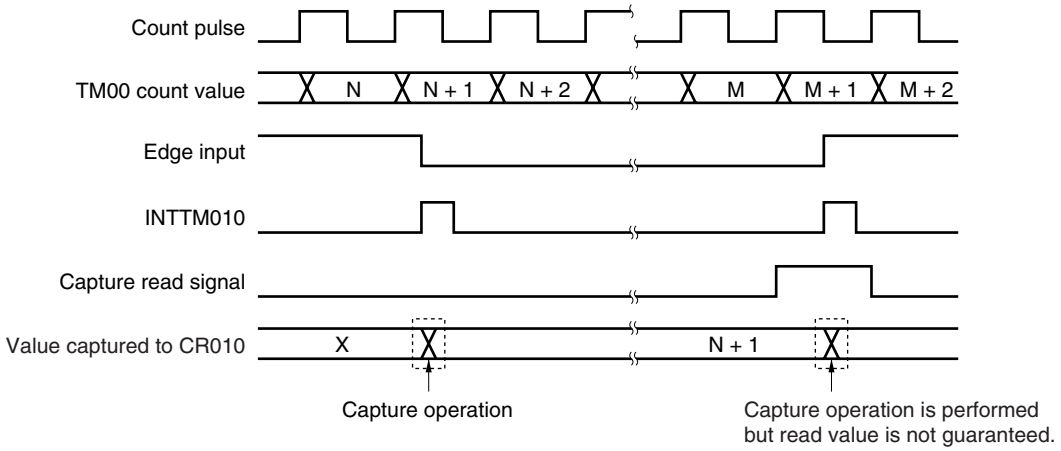
Set a value other than 0000H to CR000 and CR010 (TM00 cannot count one pulse when it is used as an external event counter).

(4) Timing of holding data by capture register

(a) When the valid edge is input to the TI000/TI010 pin and the reverse phase of the TI000 pin is detected while CR000/CR010 is read, CR010 performs a capture operation but the read value of CR000/CR010 is not guaranteed. At this time, an interrupt signal (INTTM000/INTTM010) is generated when the valid edge of the TI000/TI010 pin is detected (the interrupt signal is not generated when the reverse-phase edge of the TI000 pin is detected).

When the count value is captured because the valid edge of the TI000/TI010 pin was detected, read the value of CR000/CR010 after INTTM000/INTTM010 is generated.

Figure 6-57. Timing of Holding Data by Capture Register



(b) The values of CR000 and CR010 are not guaranteed after 16-bit timer/event counter 00 stops.

(5) Setting valid edge

Set the valid edge of the TI000 pin while the timer operation is stopped (TMC003 and TMC002 = 00). Set the valid edge by using ES000 and ES001.

(6) Re-triggering one-shot pulse

Make sure that the trigger is not generated while an active level is being output in the one-shot pulse output mode. Be sure to input the next trigger after the current active level is output.

(7) Operation of OVF00 flag

(a) Setting OVF00 flag (1)

The OVF00 flag is set to 1 in the following case, as well as when TM00 overflows.

Select the clear & start mode entered upon a match between TM00 and CR000.

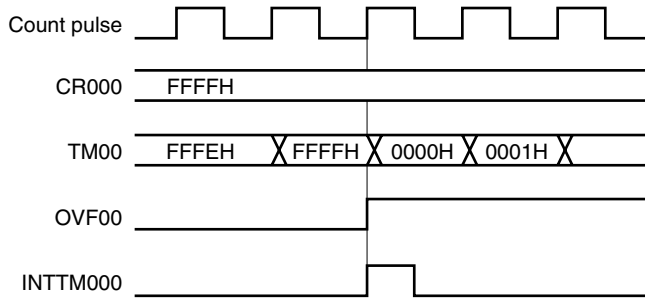
↓

Set CR000 to FFFFH.

↓

When TM00 matches CR000 and TM00 is cleared from FFFFH to 0000H

Figure 6-58. Operation Timing of OVF00 Flag



(b) Clearing OVF00 flag

Even if the OVF00 flag is cleared to 0 after TM00 overflows and before the next count clock is counted (before the value of TM00 becomes 0001H), it is set to 1 again and clearing is invalid.

(8) One-shot pulse output

One-shot pulse output operates correctly in the free-running timer mode or the clear & start mode entered by the TI000 pin valid edge. The one-shot pulse cannot be output in the clear & start mode entered upon a match between TM00 and CR000.

(9) Capture operation

(a) When valid edge of TI000 is specified as count clock

When the valid edge of TI000 is specified as the count clock, the capture register for which TI000 is specified as a trigger does not operate correctly.

(b) Pulse width to accurately capture value by signals input to TI010 and TI000 pins

To accurately capture the count value, the pulse input to the TI000 and TI010 pins as a capture trigger must be wider than two count clocks selected by PRM00 (see **Figure 6-7**).

(c) Generation of interrupt signal

The capture operation is performed at the falling edge of the count clock but the interrupt signals (INTTM000 and INTTM010) are generated at the rising edge of the next count clock (see **Figure 6-7**).

(d) Note when CRC001 (bit 1 of capture/compare control register 00 (CRC00)) is set to 1

When the count value of the TM00 register is captured to the CR000 register in the phase reverse to the signal input to the TI000 pin, the interrupt signal (INTTM000) is not generated after the count value is captured. If the valid edge is detected on the TI010 pin during this operation, the capture operation is not performed but the INTTM000 signal is generated as an external interrupt signal. Mask the INTTM000 signal when the external interrupt is not used.

(10) Edge detection

(a) Specifying valid edge after reset

If the operation of the 16-bit timer/event counter 00 is enabled after reset and while the TI000 or TI010 pin is at high level and when the rising edge or both the edges are specified as the valid edge of the TI000 or TI010 pin, then the high level of the TI000 or TI010 pin is detected as the rising edge. Note this when the TI000 or TI010 pin is pulled up. However, the rising edge is not detected when the operation is once stopped and then enabled again.

(b) Sampling clock for eliminating noise

The sampling clock for eliminating noise differs depending on whether the valid edge of TI000 is used as the count clock or capture trigger. In the former case, the sampling clock is fixed to f_{PRS}. In the latter, the count clock selected by PRM00 is used for sampling.

When the signal input to the TI000 pin is sampled and the valid level is detected two times in a row, the valid edge is detected. Therefore, noise having a short pulse width can be eliminated (see **Figure 6-7**).

(11) Timer operation

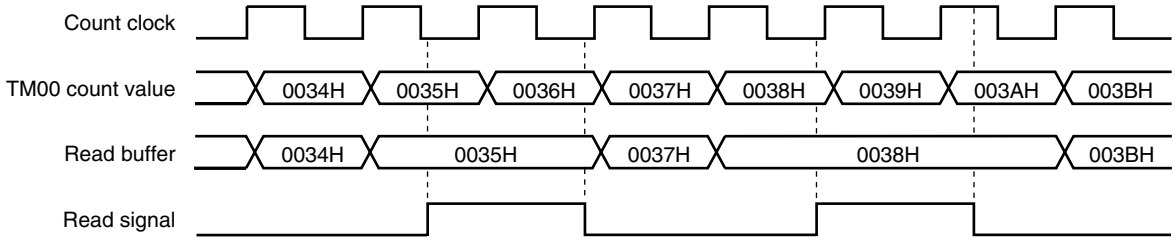
The signal input to the TI000/TI010 pin is not acknowledged while the timer is stopped, regardless of the operation mode of the CPU.

Remark f_{PRS}: Peripheral hardware clock frequency

(12) Reading of 16-bit timer counter 00 (TM00)

TM00 can be read without stopping the actual counter, because the count values captured to the buffer are fixed when it is read. The buffer, however, may not be updated when it is read immediately before the counter counts up, because the buffer is updated at the timing the counter counts up.

Figure 6-59. 16-bit Timer Counter 00 (TM00) Read Timing



CHAPTER 7 8-BIT TIMER/EVENT COUNTERS 50 AND 51

7.1 Functions of 8-Bit Timer/Event Counters 50 and 51

8-bit timer/event counters 50 and 51 are mounted onto all 78K0/Kx2-A microcontrollers..
8-bit timer/event counters 50 and 51 have the following functions.

- Interval timer
- External event counter
- Square-wave output
- PWM output

7.2 Configuration of 8-Bit Timer/Event Counters 50 and 51

8-bit timer/event counters 50 and 51 include the following hardware.

Table 7-1. Configuration of 8-Bit Timer/Event Counters 50 and 51

Item	Configuration
Timer register	8-bit timer counter 5n (TM5n)
Register	8-bit timer compare register 5n (CR5n)
Timer input	TI5n[Note]
Timer output	TO5n[Note]
Control registers	Timer clock selection register 5n (TCL5n) 8-bit timer mode control register 5n (TMC5n) Port mode register 1, 3 (PM1, PM3)[Note] Port register 1, 3 (P1, P3)[Note]

Note The port pins with which the I/O pins for 8-bit timer/event counters 50 and 51 are shared differ depending on the product as follows:

- 78K0/KB2-A: Shared with the pins of port 1
- 78K0/KC2-A: Shared with the pins of port 3

Remark n = 0, 1

Figures 7-1 and 7-2 show the block diagrams of 8-bit timer/event counters 50 and 51.

Figure 7-1. Block Diagram of 8-Bit Timer/Event Counter 50

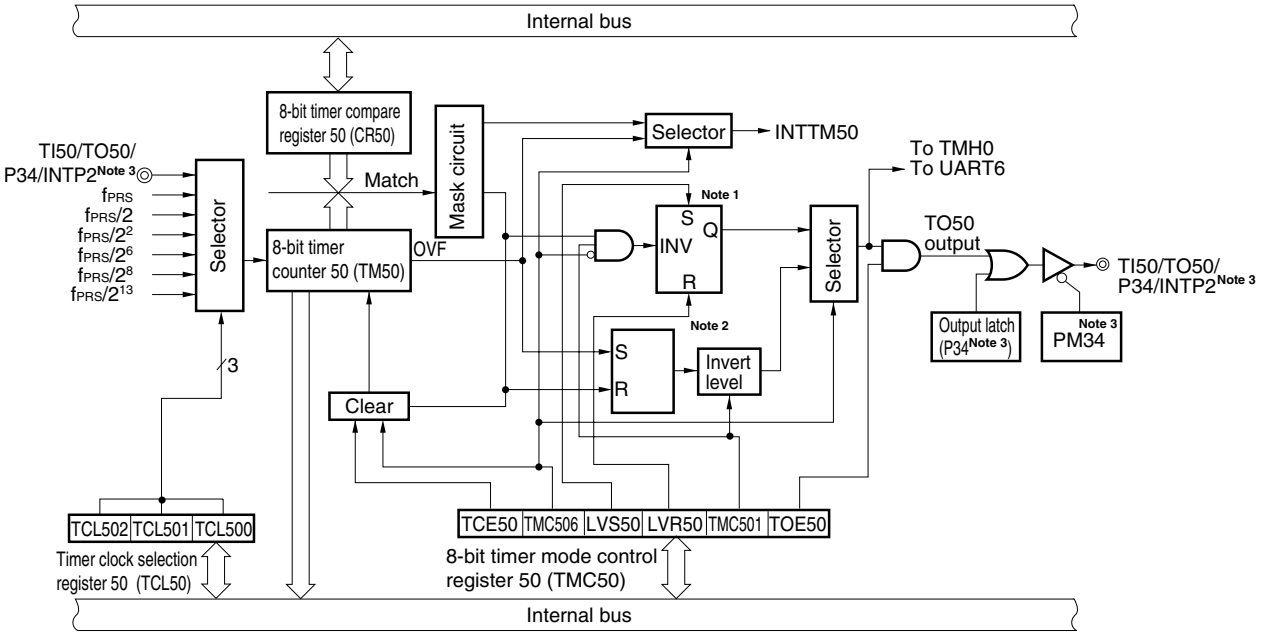
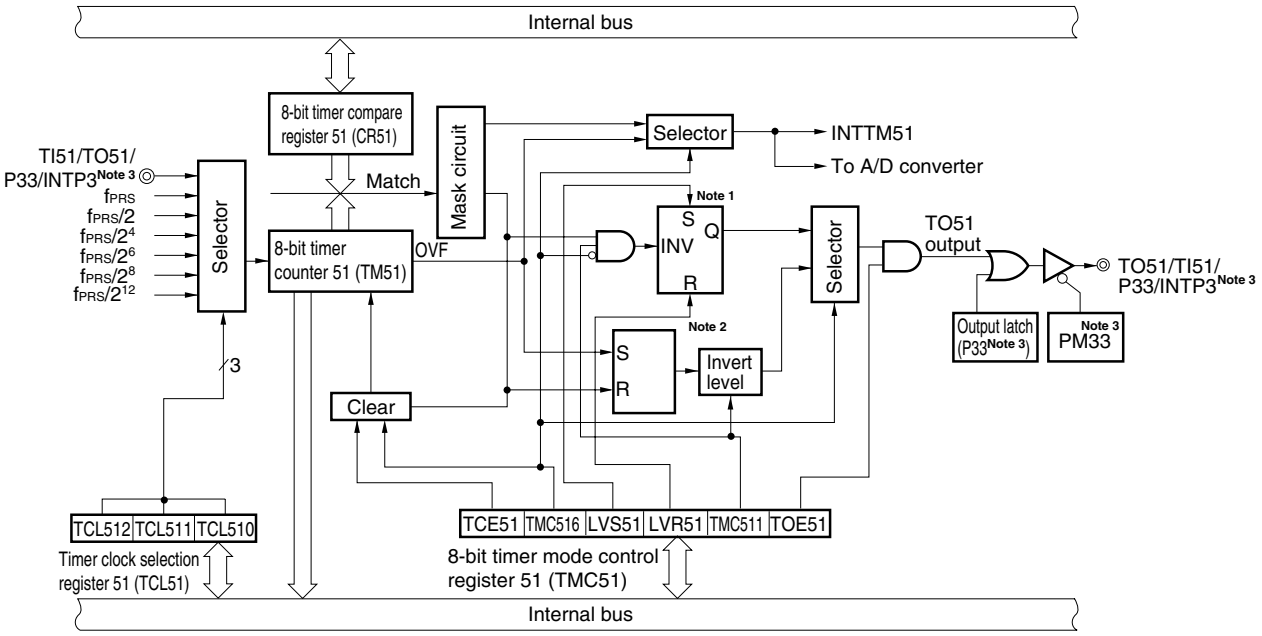


Figure 7-2. Block Diagram of 8-Bit Timer/Event Counter 51



Notes 1. Timer output F/F

2. PWM output F/F

3. 78K0/KB2-A: TI50/TO50/RxD6/P11, output latch (P11), PM11,
TI51/TO51/TxD6/P10, output latch (P10), PM10
78K0/KC2-A: TI50/TO50/INTP2/P34, output latch (P34), PM34,
TI51/TO51/INTP3/P33, output latch (P33), PM33

(1) 8-bit timer counter 5n (TM5n)

TM5n is an 8-bit register that counts the count pulses and is read-only.

The counter is incremented in synchronization with the rising edge of the count clock.

Figure 7-3. Format of 8-Bit Timer Counter 5n (TM5n)

Address: FF16H (TM50), FF1FH (TM51) After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
TM5n (n = 0, 1)								

In the following situations, the count value is cleared to 00H.

<1> Reset signal generation

<2> When TCE5n is cleared

<3> When TM5n and CR5n match in the mode in which clear & start occurs upon a match of the TM5n and CR5n.

(2) 8-bit timer compare register 5n (CR5n)

CR5n can be read and written by an 8-bit memory manipulation instruction.

Except in PWM mode, the value set in CR5n is constantly compared with the 8-bit timer counter 5n (TM5n) count value, and an interrupt request (INTTM5n) is generated if they match.

In the PWM mode, TO5n output becomes inactive when the values of TM5n and CR5n match, but no interrupt is generated.

The value of CR5n can be set within 00H to FFH.

Reset signal generation clears CR5n to 00H.

Figure 7-4. Format of 8-Bit Timer Compare Register 5n (CR5n)

Address: FF17H (CR50), FF41H (CR51) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CR5n (n = 0, 1)								

Cautions 1. In the mode in which clear & start occurs on a match of TM5n and CR5n (TMC5n6 = 0), do not write other values to CR5n during operation.

2. In PWM mode, make the CR5n rewrite period 3 count clocks of the count clock (clock selected by TCL5n) or more.

Remark n = 0, 1

7.3 Registers Controlling 8-Bit Timer/Event Counters 50 and 51

The following four registers are used to control 8-bit timer/event counters 50 and 51.

- Timer clock selection register 5n (TCL5n)
- 8-bit timer mode control register 5n (TMC5n)
- Port mode register 1 (PM1) or port mode register 3 (PM3)[Note]
- Port register 1 (P1) or port register 3 (P3)[Note]

Note The port pins with which the I/O pins for 8-bit timer/event counters 50 and 51 are shared differ depending on the product as follows:
- 78K0/KB2-A: Shared with the pins of port 1
- 78K0/KC2-A: Shared with the pins of port 3

(1) Timer clock selection register 5n (TCL5n)

This register sets the count clock of 8-bit timer/event counter 5n and the valid edge of the TI5n pin input.

TCL5n can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears TCL5n to 00H.

Remark n = 0, 1

Figure 7-5. Format of Timer Clock Selection Register 50 (TCL50)

Address: FF6AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TCL50	0	0	0	0	0	TCL502	TCL501	TCL500

TCL502	TCL501	TCL500	Count clock selection[Note 1]				
				f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	0	TI50 pin falling edge[Note 2]				
0	0	1	TI50 pin rising edge[Note 2]				
0	1	0	f_{PRS}[Note 3]	2 MHz	5 MHz	10 MHz	20 MHz[Note 4]
0	1	1	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz
1	0	0	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz
1	0	1	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	156.25 kHz	312.5 kHz
1	1	0	$f_{PRS}/2^8$	7.81 kHz	19.53 kHz	39.06 kHz	78.13 kHz
1	1	1	$f_{PRS}/2^{13}$	0.24 kHz	0.61 kHz	1.22 kHz	2.44 kHz

Notes 1. The frequency that can be used for the peripheral hardware clock (f_{PRS}) differs depending on the power supply voltage.

Supply Voltage	Use frequency range of peripheral hardware clock (f_{PRS})
$2.7\ V \le V_{DD} \le 5.5\ V$	$f_{PRS} \le 20$ MHz
$1.8\ V \le V_{DD} < 2.7\ V$	$f_{PRS} \le 5$ MHz

(The values shown in the table above are those when $f_{PRS} = f_{XH}$ (XSEL = 1).)

2. Do not start timer operation with the external clock from the TI50 pin when the internal high-speed oscillation clock and high-speed system clock are stopped while the CPU operates with the subsystem clock, or when in the STOP mode.

3. If the peripheral hardware clock (f_{PRS}) operates on the internal high-speed oscillation clock (f_{RH}) (XSEL = 0), when $1.8\ V \le V_{DD} < 2.7\ V$, the setting of TCL502, TCL501, TCL500 = 0, 1, 0 (count clock: f_{PRS}) is prohibited.

4. This is settable only if $4.0\ V \le V_{DD} \le 5.5\ V$.

Cautions 1. When rewriting TCL50 to other data, stop the timer operation beforehand.

2. Be sure to clear bits 3 to 7 to "0".

Remark f_{PRS}: Peripheral hardware clock frequency

Figure 7-6. Format of Timer Clock Selection Register 51 (TCL51)

Address: FF8CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TCL51	0	0	0	0	0	TCL512	TCL511	TCL510

TCL512	TCL511	TCL510	Count clock selection[Note 1]				
				f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	0	TI51 pin falling edge[Note 2]				
0	0	1	TI51 pin rising edge[Note 2]				
0	1	0	f_{PRS}[Note 3]	2 MHz	5 MHz	10 MHz	20 MHz[Note 4]
0	1	1	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz
1	0	0	$f_{PRS}/2^4$	125 kHz	312.5 kHz	625 kHz	1.25 MHz
1	0	1	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	156.25 kHz	312.5 kHz
1	1	0	$f_{PRS}/2^8$	7.81 kHz	19.53 kHz	39.06 kHz	78.13 kHz
1	1	1	$f_{PRS}/2^{12}$	0.49 kHz	1.22 kHz	2.44 kHz	4.88 kHz

Notes 1. The frequency that can be used for the peripheral hardware clock (f_{PRS}) differs depending on the power supply voltage.

Supply Voltage	Use frequency range of peripheral hardware clock (f_{PRS})
$2.7\ V \le V_{DD} \le 5.5\ V$	$f_{PRS} \le 20$ MHz
$1.8\ V \le V_{DD} < 2.7\ V$	$f_{PRS} \le 5$ MHz

(The values shown in the table above are those when $f_{PRS} = f_{XH}$ (XSEL = 1).)

2. Do not start timer operation with the external clock from the TI51 pin when the internal high-speed oscillation clock and high-speed system clock are stopped while the CPU operates with the subsystem clock, or when in the STOP mode.

3. If the peripheral hardware clock (f_{PRS}) operates on the internal high-speed oscillation clock (f_{RH}) (XSEL = 0), when $1.8\ V \le V_{DD} < 2.7\ V$, the setting of TCL512, TCL511, TCL510 = 0, 1, 0 (count clock: f_{PRS}) is prohibited.

4. This is settable only if $4.0\ V \le V_{DD} \le 5.5\ V$.

Cautions 1. When rewriting TCL51 to other data, stop the timer operation beforehand.

2. Be sure to clear bits 3 to 7 to "0".

Remark f_{PRS}: Peripheral hardware clock frequency

(2) 8-bit timer mode control register 5n (TMC5n)

TMC5n is a register that performs the following five types of settings.

<1> 8-bit timer counter 5n (TM5n) count operation control
<2> 8-bit timer counter 5n (TM5n) operating mode selection
<3> Timer output F/F (flip flop) status setting
<4> Active level selection in timer F/F control or PWM (free-running) mode.
<5> Timer output control

TMC5n can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Remark n = 0, 1

Figure 7-7. Format of 8-Bit Timer Mode Control Register 50 (TMC50)

Address: FF6BH After reset: 00H R/W[Note]

Symbol	<7>	6	5	4	<3>	<2>	1	<0>
TMC50	TCE50	TMC506	0	0	LVS50	LVR50	TMC501	TOE50

TCE50	TM50 count operation control
0	After clearing to 0, count operation disabled (counter stopped)
1	Count operation start

TMC506	TM50 operating mode selection
0	Mode in which clear & start occurs on a match between TM50 and CR50
1	PWM (free-running) mode

LVS50	LVR50	Timer output F/F status setting
0	0	No change
0	1	Timer output F/F clear (0) (default value of TO50 output: low level)
1	0	Timer output F/F set (1) (default value of TO50 output: high level)
1	1	Setting prohibited

TMC501	In other modes (TMC506 = 0)	In PWM mode (TMC506 = 1)
	Timer F/F control	Active level selection
0	Inversion operation disabled	Active-high
1	Inversion operation enabled	Active-low

TOE50	Timer output control
0	Output disabled (TO50 output is low level)
1	Output enabled

Note Bits 2 and 3 are write-only.

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

Figure 7-8. Format of 8-Bit Timer Mode Control Register 51 (TMC51)

Address: FF43H After reset: 00H R/W[Note]

Symbol	<7>	6	5	4	<3>	<2>	1	<0>
TMC51	TCE51	TMC516	0	0	LVS51	LVR51	TMC511	TOE51

TCE51	TM51 count operation control
0	After clearing to 0, count operation disabled (counter stopped)
1	Count operation start

TMC516	TM51 operating mode selection
0	Mode in which clear & start occurs on a match between TM51 and CR51
1	PWM (free-running) mode

LVS51	LVR51	Timer output F/F status setting
0	0	No change
0	1	Timer output F/F clear (0) (default value of TO51 output: low)
1	0	Timer output F/F set (1) (default value of TO51 output: high)
1	1	Setting prohibited

TMC511	In other modes (TMC516 = 0)	In PWM mode (TMC516 = 1)
	Timer F/F control	Active level selection
0	Inversion operation disabled	Active-high
1	Inversion operation enabled	Active-low

TOE51	Timer output control
0	Output disabled (TO51 output is low level)
1	Output enabled

Note Bits 2 and 3 are write-only.

Cautions 1. The settings of LVS5n and LVR5n are valid in other than PWM mode.

2. Perform <1> to <4> below in the following order, not at the same time.

<1> Set TMC5n1, TMC5n6: Operation mode setting

<2> Set TOE5n to enable output: Timer output enable

<3> Set LVS5n, LVR5n (see Caution 1): Timer F/F setting

<4> Set TCE5n

3. When TCE5n = 1, setting the other bits of TMC5n is prohibited.

4. The actual pin outputs are determined depending on PM10, PM11. P00, and P01 in 78K0/KB2-A, and PM33, PM34, P33, and P34 78K0/KC2-A, besides TO5n output.

Remarks 1. In PWM mode, PWM output is made inactive by clearing TCE5n to 0.

2. If LVS5n and LVR5n are read, the value is 0.

3. The values of the TMC5n6, LVS5n, LVR5n, TMC5n1, and TOE5n bits are reflected at the TO5n output regardless of the value of TCE5n.

4. n = 0, 1

(3) Port mode registers 1 and 3 (PM1, PM3)

These registers set port 1 and 3 input/output in 1-bit units.

PM1 and PM3 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

- 78K0/KB2-A

  When using the P11/TI50/TO50/RxD6 and P10/TI51/TO51/TxD6 pins for timer output, set PM11, PM10 and the output latches of P11, P10 to 0.

  When using the P11/TO50/TI50/RxD6, P10/TO51/TI51/TxD6 pins for timer input, set PM11 and PM10 to 1. At this time, the output latches of P11 and P10 may be 0 or 1.

Figure 7-9. Format of Port Mode Register 1 (PM1)

Address: FF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	1	1	1	1	PM13	PM12	PM11	PM10

PM1n	P1n pin I/O mode selection (n = 0 to 3)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

- 78K0/KC2-A

  When using the P34/TO50/TI50/INTP2 and P33/TO51/TI51/INTP3 pins for timer output, set PM34, PM33 and the output latches of P34, P33 to 0.

  When using the P34/TO50/TI50/INTP2 and P33/TO51/TI51/INTP3 pins for timer input, set PM34 and PM33 to 1. At this time, the output latches of P34 and P33 may be 0 or 1.

Figure 7-10. Format of Port Mode Register 3 (PM3)

Address: FF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	1	PM35	PM34	PM33	PM32	PM31	1

PM3n	P1n pin I/O mode selection (n = 1 to 5)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

7.4 Operations of 8-Bit Timer/Event Counters 50 and 51

7.4.1 Operation as interval timer

8-bit timer/event counter 5n operates as an interval timer that generates interrupt requests repeatedly at intervals of the count value preset to 8-bit timer compare register 5n (CR5n).

When the count value of 8-bit timer counter 5n (TM5n) matches the value set to CR5n, counting continues with the TM5n value cleared to 0 and an interrupt request signal (INTTM5n) is generated.

The count clock of TM5n can be selected with bits 0 to 2 (TCL5n0 to TCL5n2) of timer clock selection register 5n (TCL5n).

Setting

<1> Set the registers.
- TCL5n: Select the count clock.
- CR5n: Compare value
- TMC5n: Stop the count operation, select the mode in which clear & start occurs on a match of TM5n and CR5n.
  (TMC5n = 0000×××0B × = Don't care)

<2> After TCE5n = 1 is set, the count operation starts.

<3> If the values of TM5n and CR5n match, INTTM5n is generated (TM5n is cleared to 00H).

<4> INTTM5n is generated repeatedly at the same interval.
Set TCE5n to 0 to stop the count operation.

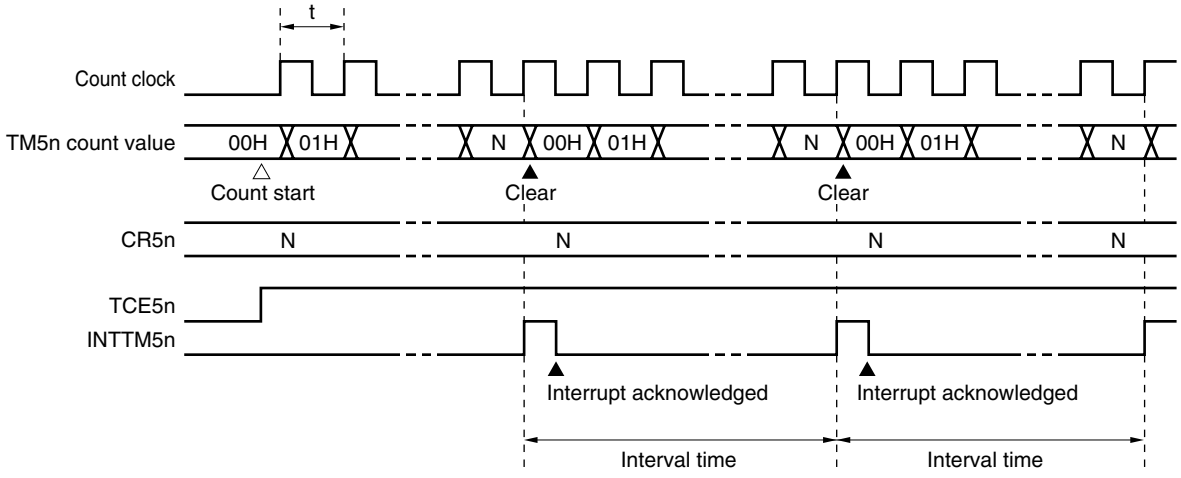
Caution Do not write other values to CR5n during operation.

Remarks 1. For how to enable the INTTM5n signal interrupt, see **CHAPTER 18 INTERRUPT FUNCTIONS**.

2. n = 0, 1

Figure 7-11. Interval Timer Operation Timing (1/2)

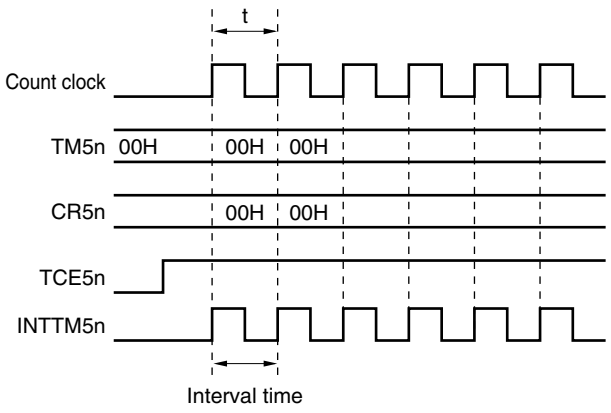
(a) Basic operation



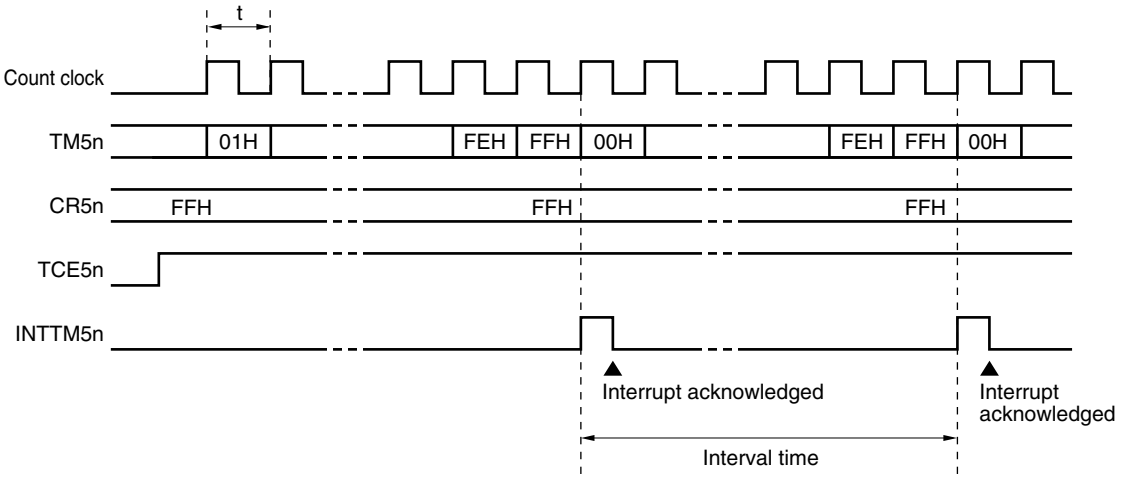
Remark Interval time = (N + 1) × t
N = 01H to FFH
n = 0, 1

Figure 7-11. Interval Timer Operation Timing (2/2)

(b) When CR5n = 00H



(c) When CR5n = FFH



Remark n = 0, 1

7.4.2 Operation as external event counter

The external event counter counts the number of external clock pulses to be input to the TI5n pin by 8-bit timer counter 5n (TM5n).

TM5n is incremented each time the valid edge specified by timer clock selection register 5n (TCL5n) is input. Either the rising or falling edge can be selected.

When the TM5n count value matches the value of 8-bit timer compare register 5n (CR5n), TM5n is cleared to 0 and an interrupt request signal (INTTM5n) is generated.

Whenever the TM5n value matches the value of CR5n, INTTM5n is generated.

Setting

<1> Set each register.

- Set the port mode register (PM11, PM10, PM34, PM33)[Note] to 1.
- TCL5n: Select TI5n pin input edge.
  TI5n pin falling edge → TCL5n = 00H
  TI5n pin rising edge → TCL5n = 01H
- CR5n: Compare value
- TMC5n: Stop the count operation, select the mode in which clear & start occurs on match of TM5n and CR5n, disable the timer F/F inversion operation, disable timer output.
  (TMC5n = 00000000B)

<2> When TCE5n = 1 is set, the number of pulses input from the TI5n pin is counted.

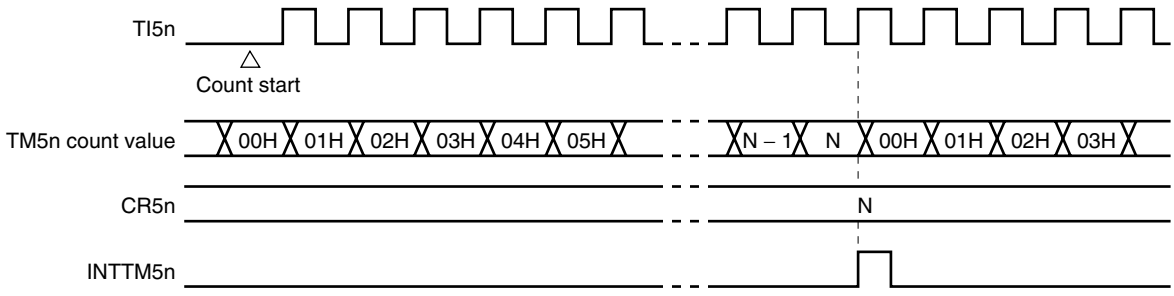
<3> When the values of TM5n and CR5n match, INTTM5n is generated (TM5n is cleared to 00H).

<4> After these settings, INTTM5n is generated each time the values of TM5n and CR5n match.

Note 8-bit timer/event counter 50: PM11 (In case of 78K0/KB2-A)
PM34 (In case of 78K0/KC2-A)
8-bit timer/event counter 51: PM10 (In case of 78K0/KB2-A)
PM33 (In case of 78K0/KC2-A)

Remark For how to enable the INTTM5n signal interrupt, see **CHAPTER 18 INTERRUPT FUNCTIONS**.

Figure 7-12. External Event Counter Operation Timing (with Rising Edge Specified)



Remark N = 00H to FFH
n = 0, 1

7.4.3 Square-wave output operation

A square wave with any selected frequency is output at intervals determined by the value preset to 8-bit timer compare register 5n (CR5n).

The TO5n pin output status is inverted at intervals determined by the count value preset to CR5n by setting bit 0 (TOE5n) of 8-bit timer mode control register 5n (TMC5n) to 1. This enables a square wave with any selected frequency to be output (duty = 50%).

Setting

<1> Set each register.

- Clear the port output latch (P10, P11, P33, P34)[Note] and port mode register (PM10, PM11, PM33, PM34)[Note] to 0.
- TCL5n: Select the count clock.
- CR5n: Compare value
- TMC5n: Stop the count operation, select the mode in which clear & start occurs on a match of TM5n and CR5n.

LVS5n	LVR5n	Timer Output F/F Status Setting
0	1	Timer output F/F clear (0) (default value of TO5n output: low level)
1	0	Timer output F/F set (1) (default value of TO5n output: high level)

Timer output enabled
(TMC5n = 00001011B or 00000111B)

<2> After TCE5n = 1 is set, the count operation starts.

<3> The timer output F/F is inverted by a match of TM5n and CR5n. After INTTM5n is generated, TM5n is cleared to 00H.

<4> After these settings, the timer output F/F is inverted at the same interval and a square wave is output from TO5n.

The frequency is as follows.

- Frequency = 1/2t (N + 1)
  (N: 00H to FFH)

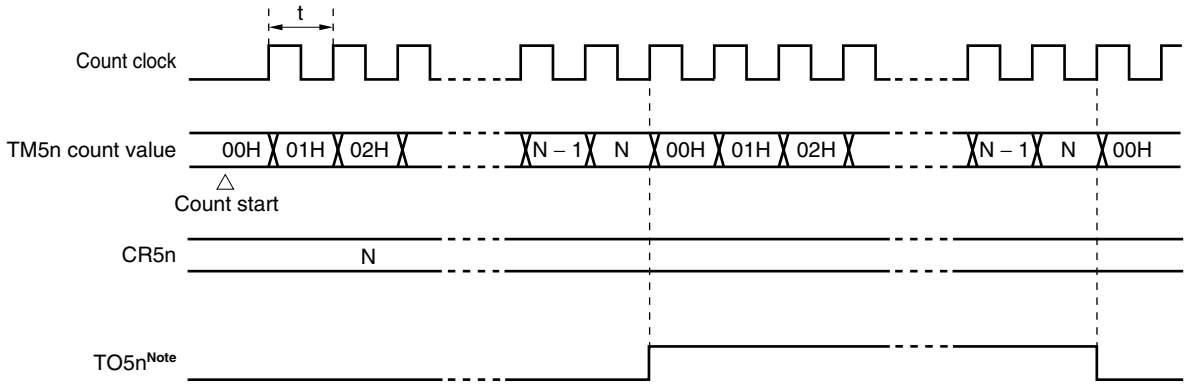
Note 8-bit timer/event counter 50: P11, PM11 (In case of 78K0/KB2-A)
P34, PM34 (In case of 78K0/KC2-A)
8-bit timer/event counter 51: P10, PM10 (In case of 78K0/KB2-A))
P33, PM33 (In case of 78K0/KC2-A)

Caution Do not write other values to CR5n during operation.

Remarks 1. For how to enable the INTTM5n signal interrupt, see **CHAPTER 18 INTERRUPT FUNCTIONS**.

2. n = 0, 1

Figure 7-13. Square-Wave Output Operation Timing



Note The initial value of TO5n output can be set by bits 2 and 3 (LVR5n, LVS5n) of 8-bit timer mode control register 5n (TMC5n).

7.4.4 PWM output operation

8-bit timer/event counter 5n operates as a PWM output when bit 6 (TMC5n6) of 8-bit timer mode control register 5n (TMC5n) is set to 1.

The duty pulse determined by the value set to 8-bit timer compare register 5n (CR5n) is output from TO5n.

Set the active level width of the PWM pulse to CR5n; the active level can be selected with bit 1 (TMC5n1) of TMC5n.

The count clock can be selected with bits 0 to 2 (TCL5n0 to TCL5n2) of timer clock selection register 5n (TCL5n).

PWM output can be enabled/disabled with bit 0 (TOE5n) of TMC5n.

Caution In PWM mode, make the CR5n rewrite period 3 count clocks of the count clock (clock selected by TCL5n) or more.

Remark n = 0, 1

(1) PWM output basic operation

Setting

<1> Set each register.

- Clear the port output latch (P10, P11, P33, P34)[Note] and port mode register (PM10, PM11, PM33, PM34)[Note] to 0.
- TCL5n: Select the count clock.
- CR5n: Compare value
- TMC5n: Stop the count operation, select PWM mode.
  The timer output F/F is not changed.

TMC5n1	Active Level Selection
0	Active-high
1	Active-low

Timer output enabled
(TMC5n = 01000001B or 01000011B)

<2> The count operation starts when TCE5n = 1.
Clear TCE5n to 0 to stop the count operation.

Note 8-bit timer/event counter 50: P11, PM11 (In case of 78K0/KB2-A)
P34, PM34 (In case of 78K0/KC2-A)
8-bit timer/event counter 51: P10, PM10 (In case of 78K0/KB2-A)
P33, PM33 (In case of 78K0/KC2-A)

PWM output operation

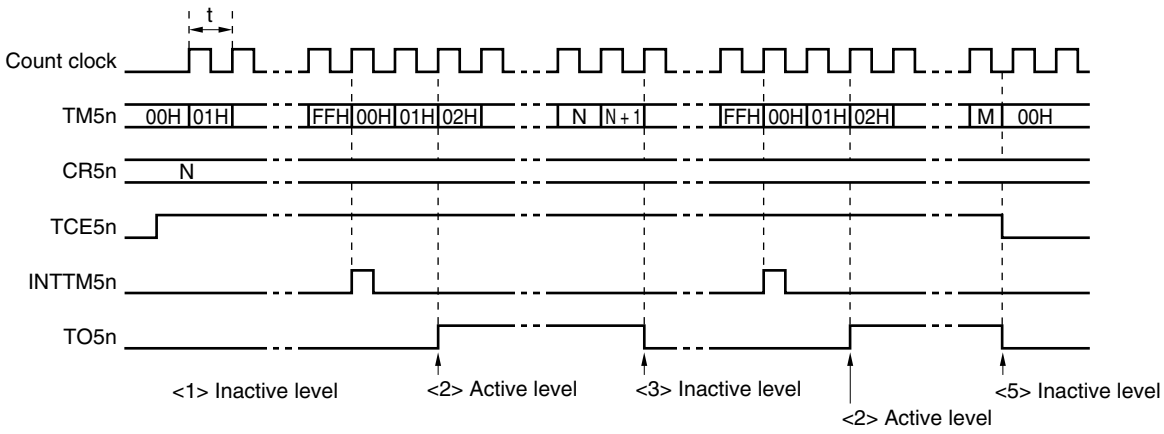
<1> PWM output (TO5n output) outputs an inactive level until an overflow occurs.
<2> When an overflow occurs, the active level is output. The active level is output until CR5n matches the count value of 8-bit timer counter 5n (TM5n).
<3> After the CR5n matches the count value, the inactive level is output until an overflow occurs again.
<4> Operations <2> and <3> are repeated until the count operation stops.
<5> When the count operation is stopped with TCE5n = 0, PWM output becomes inactive.
For details of timing, see **Figures 7-14** and **7-15**.
The cycle, active-level width, and duty are as follows.

- Cycle = 2^8t
- Active-level width = Nt
- Duty = N/2^8
  (N = 00H to FFH)

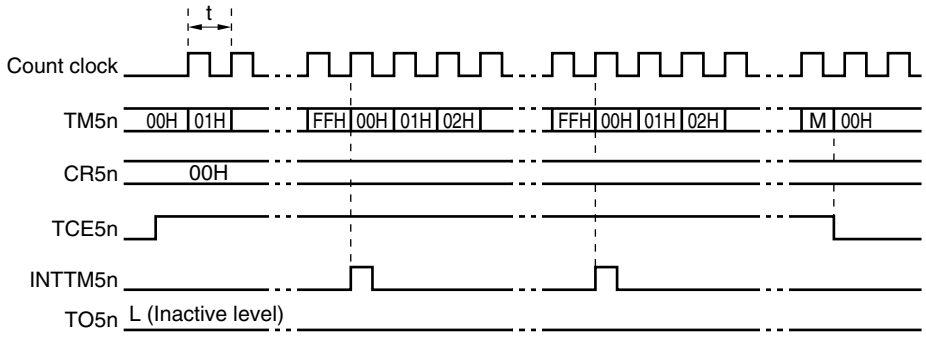
Remark n = 0, 1

Figure 7-14. PWM Output Operation Timing

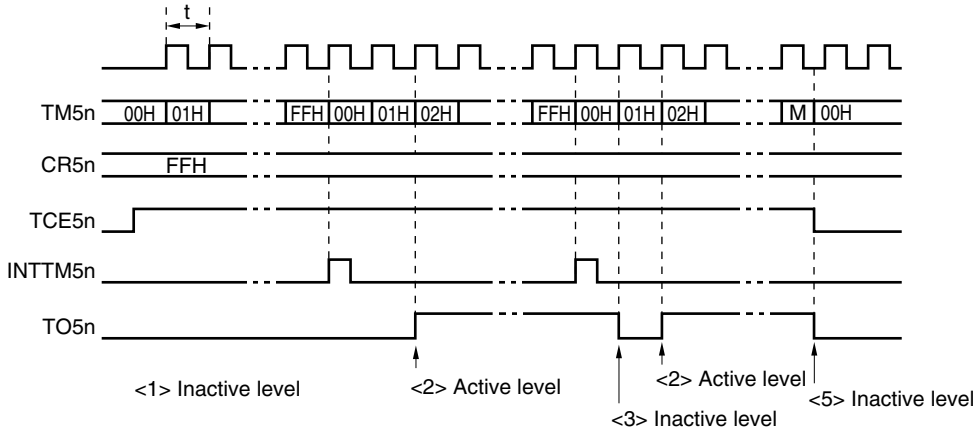
(a) Basic operation (active level = H)



(b) CR5n = 00H



(c) CR5n = FFH



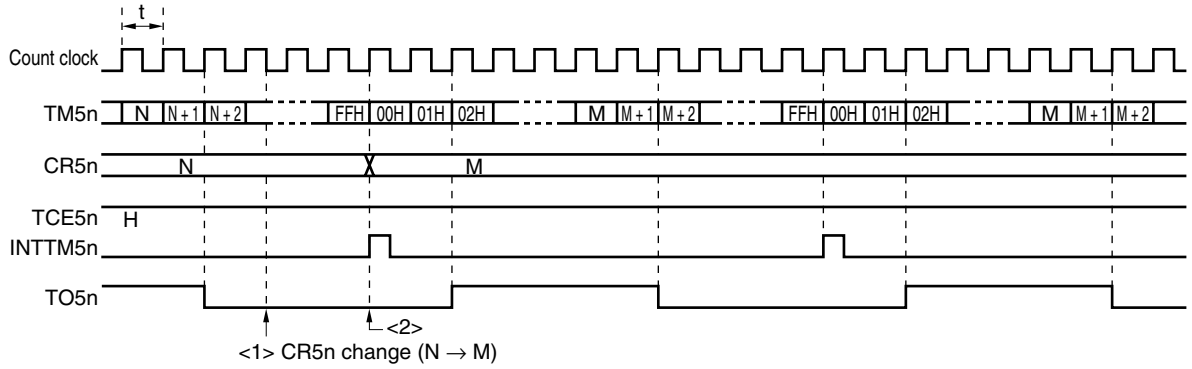
Remarks 1. <1> to <3> and <5> in Figure 7-14 (a) and (c) correspond to <1> to <3> and <5> in PWM output operation in **7.4.4 (1) PWM output basic operation**.

2. n = 0, 1

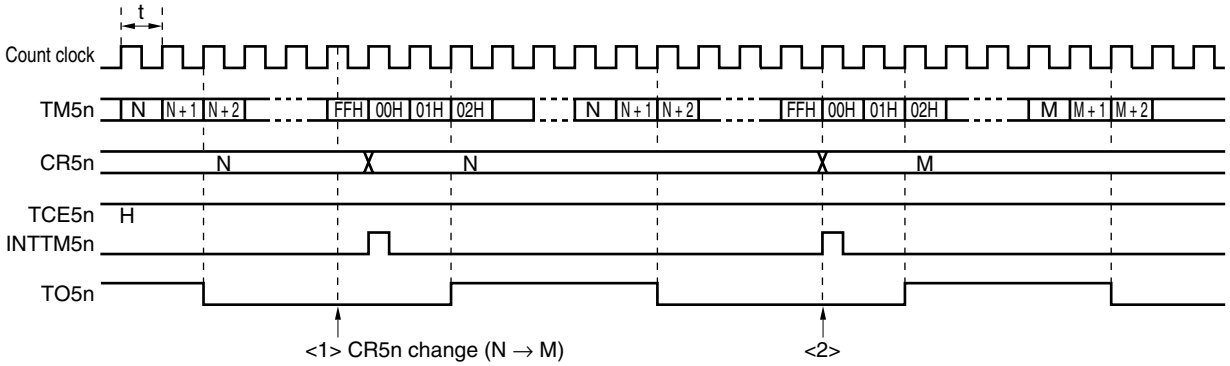
(2) Operation with CR5n changed

Figure 7-15. Timing of Operation with CR5n Changed

(a) CR5n value is changed from N to M before clock rising edge of FFH
→ Value is transferred to CR5n at overflow immediately after change.



(b) CR5n value is changed from N to M after clock rising edge of FFH
→ Value is transferred to CR5n at second overflow.



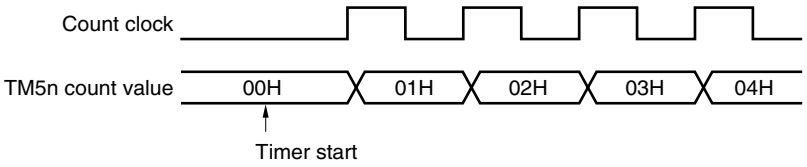
Caution When reading from CR5n between <1> and <2> in Figure 7-15, the value read differs from the actual value (read value: M, actual value of CR5n: N).

7.5 Cautions for 8-Bit Timer/Event Counters 50 and 51

(1) Timer start error

An error of up to one clock may occur in the time required for a match signal to be generated after timer start. This is because 8-bit timer counters 50 and 51 (TM50, TM51) are started asynchronously to the count clock.

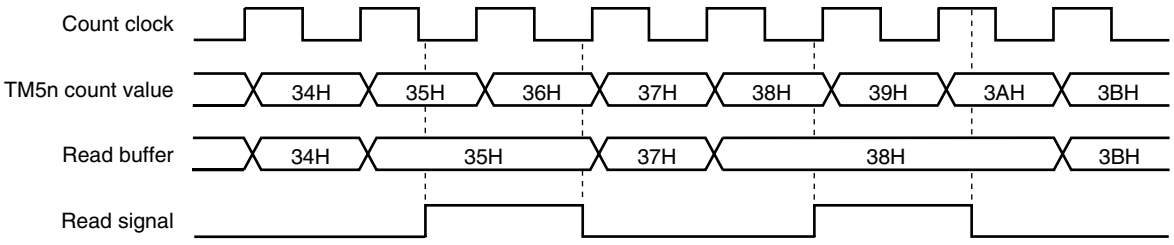
Figure 7-16. 8-Bit Timer Counter 5n (TM5n) Start Timing



(2) Reading of 8-bit timer counter 5n (TM5n)

TM5n can be read without stopping the actual counter, because the count values captured to the buffer are fixed when it is read. The buffer, however, may not be updated when it is read immediately before the counter counts up, because the buffer is updated at the timing the counter counts up.

Figure 7-17. 8-bit Timer Counter 5n (TM5n) Read Timing



Remark n = 0, 1

CHAPTER 8 8-BIT TIMERS H0 AND H1

8.1 Functions of 8-Bit Timers H0 and H1

8-bit timers H0 and H1 are mounted onto all 78K0/Kx2-A microcontrollers.
8-bit timers H0 and H1 have the following functions.

- Interval timer
- Square-wave output
- PWM output
- Carrier generator (8-bit timer H1 only)

8.2 Configuration of 8-Bit Timers H0 and H1

8-bit timers H0 and H1 include the following hardware.

Table 8-1. Configuration of 8-Bit Timers H0 and H1

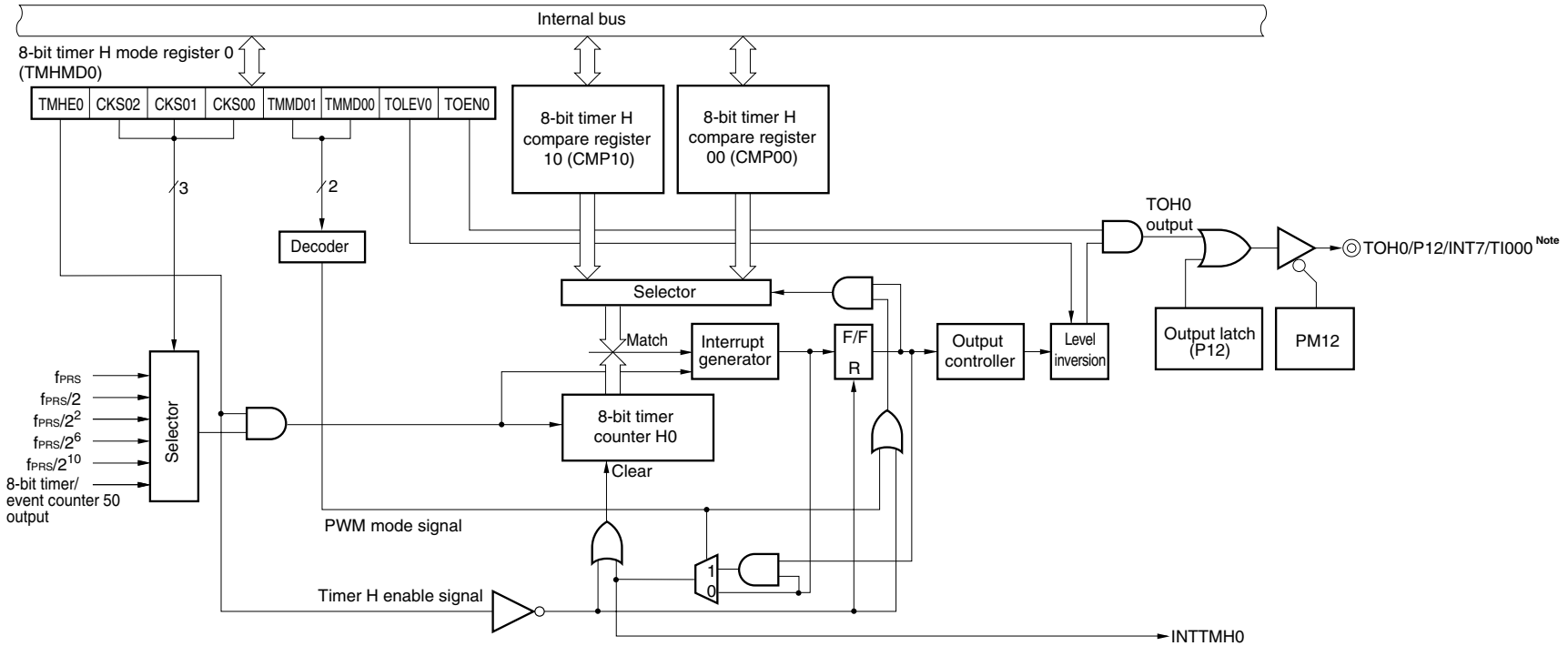
Item	Configuration
Timer register	8-bit timer counter Hn
Registers	8-bit timer H compare register 0n (CMP0n) 8-bit timer H compare register 1n (CMP1n)
Timer output	TOHn, output controller
Control registers	8-bit timer H mode register n (TMHMDn) 8-bit timer H carrier control register 1 (TMCYC1)[Note] Port mode register 1 (PM1) Port register 1 (P1)

Note 8-bit timer H1 only

Remark n = 0, 1

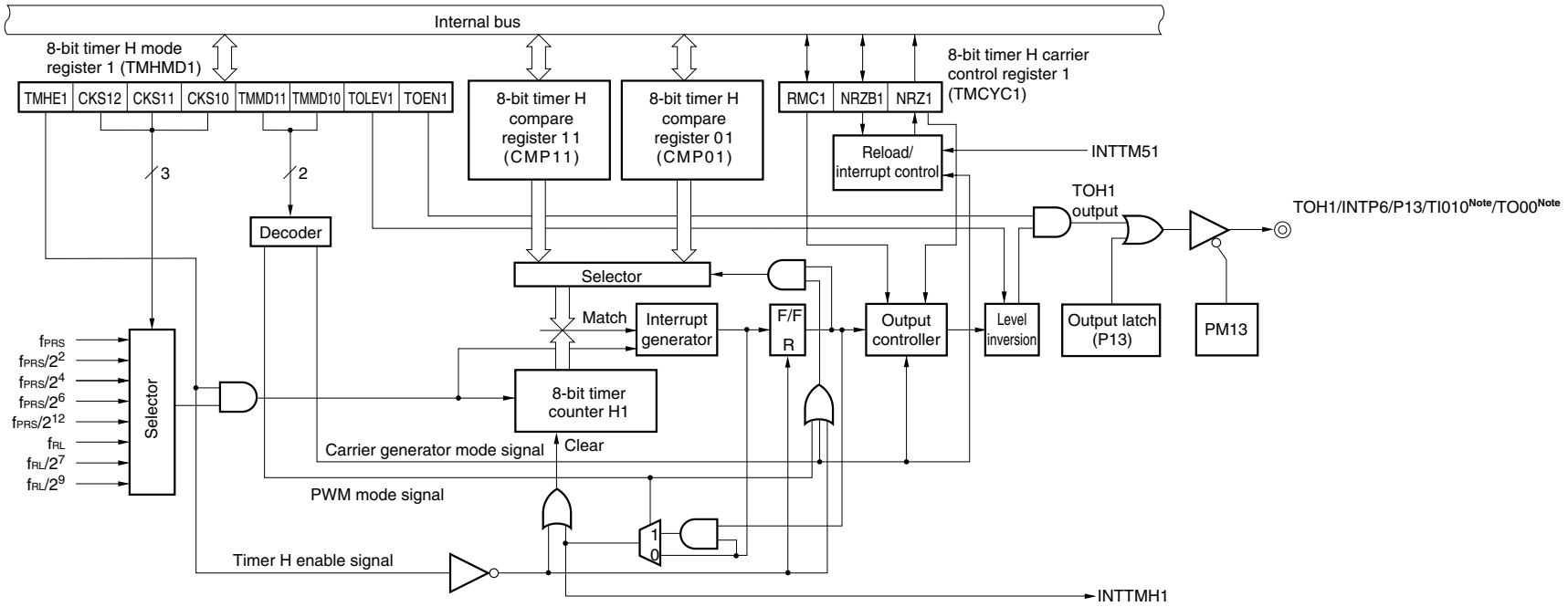
Figures 8-1 and 8-2 show the block diagrams.

Figure 8-1. Block Diagram of 8-Bit Timer H1



Note 78K0/KB2-A only.

Figure 8-2. Block Diagram of 8-Bit Timer H1



Note 78K0/KB2-A only.

(1) 8-bit timer H compare register 0n (CMP0n)

This register can be read or written by an 8-bit memory manipulation instruction. This register is used in all of the timer operation modes.

This register constantly compares the value set to CMP0n with the count value of the 8-bit timer counter Hn and, when the two values match, generates an interrupt request signal (INTTMHn) and inverts the output level of TOHn.

Rewrite the value of CMP0n while the timer is stopped (TMHEn = 0).

A reset signal generation clears this register to 00H.

Figure 8-3. Format of 8-Bit Timer H Compare Register 0n (CMP0n)

Address: FF0EH (CMP00), FF1AH (CMP01) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CMP0n (n = 0, 1)								

Caution CMP0n cannot be rewritten during timer count operation. CMP0n can be refreshed (the same value is written) during timer count operation.

(2) 8-bit timer H compare register 1n (CMP1n)

This register can be read or written by an 8-bit memory manipulation instruction. This register is used in the PWM output mode and carrier generator mode.

In the PWM output mode, this register constantly compares the value set to CMP1n with the count value of the 8-bit timer counter Hn and, when the two values match, inverts the output level of TOHn. No interrupt request signal is generated.

In the carrier generator mode, the CMP1n register always compares the value set to CMP1n with the count value of the 8-bit timer counter Hn and, when the two values match, generates an interrupt request signal (INTTMHn). At the same time, the count value is cleared.

CMP1n can be refreshed (the same value is written) and rewritten during timer count operation.

If the value of CMP1n is rewritten while the timer is operating, the new value is latched and transferred to CMP1n when the count value of the timer matches the old value of CMP1n, and then the value of CMP1n is changed to the new value. If matching of the count value and the CMP1n value and writing a value to CMP1n conflict, the value of CMP1n is not changed.

A reset signal generation clears this register to 00H.

Figure 8-4. Format of 8-Bit Timer H Compare Register 1n (CMP1n)

Address: FF0DH (CMP10), FF1BH (CMP11) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CMP1n (n = 0, 1)								

Caution In the PWM output mode and carrier generator mode, be sure to set CMP1n when starting the timer count operation (TMHEn = 1) after the timer count operation was stopped (TMHEn = 0) (be sure to set again even if setting the same value to CMP1n).

Remark n = 0, 1

8.3 Registers Controlling 8-Bit Timers H0 and H1

The following four registers are used to control 8-bit timers H0 and H1.

- 8-bit timer H mode register n (TMHMDn)
- 8-bit timer H carrier control register 1 (TMCYC1)[Note]
- Port mode register 1 (PM1)
- Port register 1 (P1)

Note 8-bit timer H1 only.

(1) 8-bit timer H mode register n (TMHMDn)

This register controls the mode of timer H.

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

Reset signal generation clears this register to 00H.

Remark n = 0, 1

Figure 8-5. Format of 8-Bit Timer H Mode Register 0 (TMHMD0)

Address: FF69H After reset: 00H R/W

	<7>	6	5	4	3	2	<1>	<0>
TMHMD0	TMHE0	CKS02	CKS01	CKS00	TMMD01	TMMD00	TOLEV0	TOEN0

TMHE0	Timer operation enable
0	Stops timer count operation (counter is cleared to 0)
1	Enables timer count operation (count operation started by inputting clock)

CKS02	CKS01	CKS00	Count clock selection[Note 1]				
				f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	0	f_{PRS}[Note 2]	2 MHz	5 MHz	10 MHz	20 MHz[Note 3]
0	0	1	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz
0	1	0	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz
0	1	1	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	156.25 kHz	312.5 kHz
1	0	0	$f_{PRS}/2^{10}$	1.95 kHz	4.88 kHz	9.77 kHz	19.54 kHz
1	0	1	TM50 output[Note 4]				
Other than above			Setting prohibited				

TMMD01	TMMD00	Timer operation mode
0	0	Interval timer mode
1	0	PWM output mode
Other than above		Setting prohibited

TOLEV0	Timer output level control (in default mode)
0	Low level
1	High level

TOEN0	Timer output control
0	Disables output
1	Enables output

Note 1. The frequency that can be used for the peripheral hardware clock (f_{PRS}) differs depending on the power supply voltage.

Supply Voltage	Use frequency range of peripheral hardware clock (f_{PRS})
$2.7\ V \le V_{DD} \le 5.5\ V$	$f_{PRS} \le 20$ MHz
$1.8\ V \le V_{DD} < 2.7\ V$	$f_{PRS} \le 5$ MHz

(The values shown in the table above are those when $f_{PRS} = f_{XH}$ (XSEL = 1).)

Notes 2. If the peripheral hardware clock (f_{PRS}) operates on the internal high-speed oscillation clock (f_{RH}) (XSEL = 0), when 1.8 V ≤ V_{DD} < 2.7 V, the setting of CKS02 = CKS01 = CKS00 = 0 (count clock: f_{PRS}) is prohibited.

3. This is settable only if 4.0 V ≤ V_{DD} ≤ 5.5 V.

4. Note the following points when selecting the TM50 output as the count clock.

- Mode in which the count clock is cleared and started upon a match of TM50 and CR50 (TMC506 = 0)

  Start the operation of 8-bit timer/event counter 50 first and then enable the timer F/F inversion operation (TMC501 = 1).
- PWM mode (TMC506 = 1)

  Start the operation of 8-bit timer/event counter 50 first and then set the count clock to make the duty = 50%.

It is not necessary to enable (TOE50 = 1) TO50 output in any mode.

Cautions 1. When TMHE0 = 1, setting the other bits of TMHMD0 is prohibited. However, TMHMD0 can be refreshed (the same value is written).

2. In the PWM output mode, be sure to set the 8-bit timer H compare register 10 (CMP10) when starting the timer count operation (TMHE0 = 1) after the timer count operation was stopped (TMHE0 = 0) (be sure to set again even if setting the same value to CMP10).

3. The actual P12/INTP7/TOH0/TI000[Note] pin output is determined depending on PM12 and P12, besides TOH0 output.

Note 78K0/KB2-A only.

Remarks 1. f_{PRS}: Peripheral hardware clock frequency

2. TMC506: Bit 6 of 8-bit timer mode control register 50 (TMC50)
TMC501: Bit 1 of TMC50

Figure 8-6. Format of 8-Bit Timer H Mode Register 1 (TMHMD1)

Address: FF6CH After reset: 00H R/W

	<7>	6	5	4	3	2	<1>	<0>
TMHMD1	TMHE1	CKS12	CKS11	CKS10	TMMD11	TMMD10	TOLEV1	TOEN1

TMHE1	Timer operation enable
0	Stops timer count operation (counter is cleared to 0)
1	Enables timer count operation (count operation started by inputting clock)

CKS12	CKS11	CKS10	Count clock selection[Note 1]				
				f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	0	f_{PRS}[Note 2]	2 MHz	5 MHz	10 MHz	20 MHz[Note 3]
0	0	1	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz
0	1	0	$f_{PRS}/2^4$	125 kHz	312.5 kHz	625 kHz	1.25 MHz
0	1	1	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	156.25 kHz	312.5 kHz
1	0	0	$f_{PRS}/2^{12}$	0.49 kHz	1.22 kHz	2.44 kHz	4.88 kHz
1	0	1	$f_{RL}/2^7$	1.88 kHz (TYP.)			
1	1	0	$f_{RL}/2^9$	0.47 kHz (TYP.)			
1	1	1	f_{RL}	240 kHz (TYP.)			

TMMD11	TMMD10	Timer operation mode
0	0	Interval timer mode
0	1	Carrier generator mode
1	0	PWM output mode
1	1	Setting prohibited

TOLEV1	Timer output level control (in default mode)
0	Low level
1	High level

TOEN1	Timer output control
0	Disables output
1	Enables output

Note 1. The frequency that can be used for the peripheral hardware clock (f_{PRS}) differs depending on the power supply voltage.

Supply Voltage	Use frequency range of peripheral hardware clock (f_{PRS})
$2.7\ V \le V_{DD} \le 5.5\ V$	$f_{PRS} \le 20$ MHz
$1.8\ V \le V_{DD} < 2.7\ V$	$f_{PRS} \le 5$ MHz

(The values shown in the table above are those when $f_{PRS} = f_{XH}$ (XSEL = 1).)

Notes 2. If the peripheral hardware clock (f_{PRS}) operates on the internal high-speed oscillation clock (f_{RH}) (XSEL = 0), when 1.8 V ≤ V_{DD} < 2.7 V, the setting of CKS12 = CKS11 = CKS10 = 0 (count clock: f_{PRS}) is prohibited.

3. This is settable only if 4.0 V ≤ V_{DD} ≤ 5.5 V.

Cautions 1. When TMHE1 = 1, setting the other bits of TMHMD1 is prohibited. However, TMHMD1 can be refreshed (the same value is written).

2. In the PWM output mode and carrier generator mode, be sure to set the 8-bit timer H compare register 11 (CMP11) when starting the timer count operation (TMHE1 = 1) after the timer count operation was stopped (TMHE1 = 0) (be sure to set again even if setting the same value to CMP11).

3. When the carrier generator mode is used, set so that the count clock frequency of TMH1 becomes more than 6 times the count clock frequency of TM51.

4. The actual P13/TOH1/INTP6/TI010[Note]/TO00[Note] pin output is determined depending on PM13 and P13, besides TOH1 output.

Note 78K0/KB2-A only.

Remarks 1. f_{PRS}: Peripheral hardware clock frequency

2. f_{RL}: Internal low-speed oscillation clock frequency

(2) 8-bit timer H carrier control register 1 (TMCYC1)

This register controls the remote control output and carrier pulse output status of 8-bit timer H1.

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

Reset signal generation clears this register to 00H.

Figure 8-7. Format of 8-Bit Timer H Carrier Control Register 1 (TMCYC1)

Address: FF6DH After reset: 00H R/W[Note]

	7	6	5	4	3	2	1	<0>
TMCYC1	0	0	0	0	0	RMC1	NRZB1	NRZ1

RMC1	NRZB1	Remote control output
0	0	Low-level output
0	1	High-level output at rising edge of INTTM51 signal input
1	0	Low-level output
1	1	Carrier pulse output at rising edge of INTTM51 signal input

NRZ1	Carrier pulse output status flag
0	Carrier output disabled status (low-level status)
1	Carrier output enabled status (RMC1 = 1: Carrier pulse output, RMC1 = 0: High-level status)

Note Bit 0 is read-only.

Caution Do not rewrite RMC1 when TMHE = 1. However, TMCYC1 can be refreshed (the same value is written).

(3) Port mode register 1 (PM1)

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

When using the P12/TOH0/INTP7/TI000[Note] and P13/TOH1/INTP6/TI010[Note]/TO00[Note] pins for timer output, clear PM12 and PM13 and the output latches of P12 and P13 to 0.

PM1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Note 78K0/KB2-A only.

Figure 8-8. Format of Port Mode Register 1 (PM1)

Address: FF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	1	1	1	1	PM13	PM12	PM11	PM10

PM1n	P1n pin I/O mode selection (n = 0 to 3)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

8.4 Operation of 8-Bit Timers H0 and H1

8.4.1 Operation as interval timer/square-wave output

When the 8-bit timer counter Hn and compare register 0n (CMP0n) match, an interrupt request signal (INTTMHn) is generated and the 8-bit timer counter Hn is cleared to 00H.

Compare register 1n (CMP1n) is not used in interval timer mode. Since a match of the 8-bit timer counter Hn and the CMP1n register is not detected even if the CMP1n register is set, timer output is not affected.

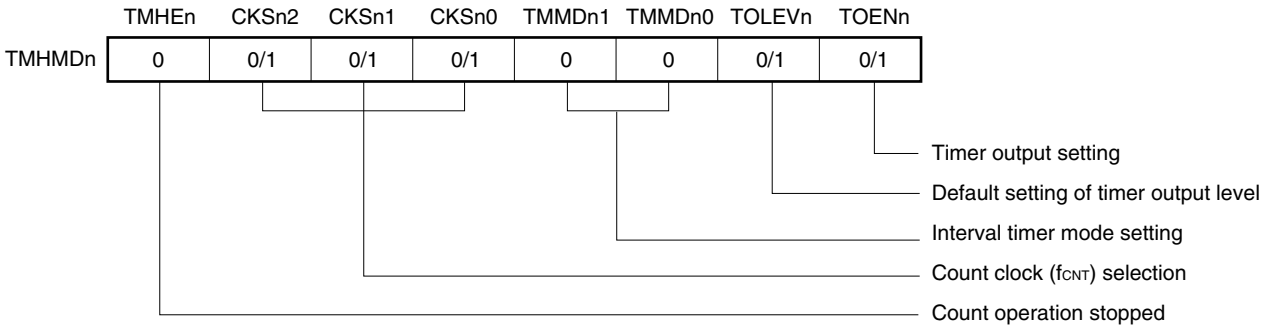
By setting bit 0 (TOENn) of timer H mode register n (TMHMDn) to 1, a square wave of any frequency (duty = 50%) is output from TOHn.

Setting

<1> Set each register.

Figure 8-9. Register Setting During Interval Timer/Square-Wave Output Operation

(i) Setting timer H mode register n (TMHMDn)



(ii) CMP0n register setting

The interval time is as follows if N is set as a comparison value.

- Interval time = (N +1)/f_{CNT}

<2> Count operation starts when TMHEn = 1.

<3> When the values of the 8-bit timer counter Hn and the CMP0n register match, the INTTMHn signal is generated and the 8-bit timer counter Hn is cleared to 00H.

<4> Subsequently, the INTTMHn signal is generated at the same interval. To stop the count operation, clear TMHEn to 0.

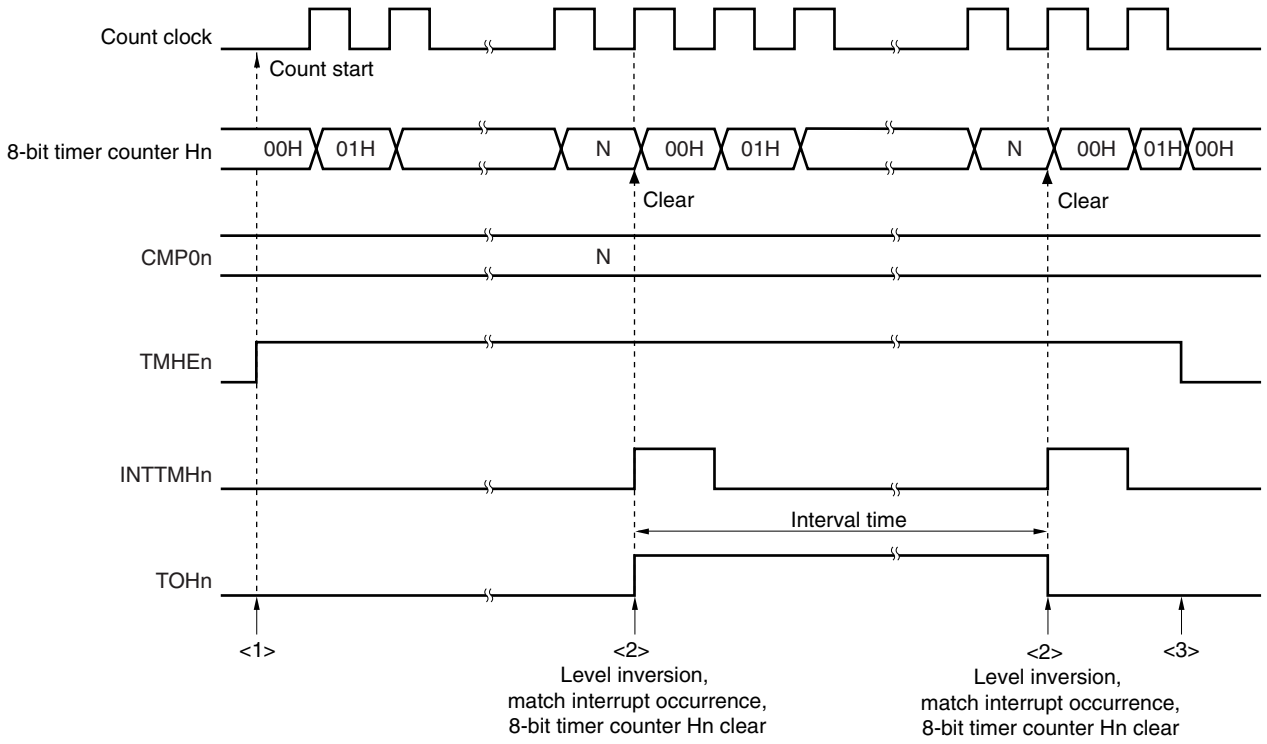
Remarks 1. For the setting of the output pin, see **8.3 (3) Port mode register 1 (PM1)**.

2. For how to enable the INTTMHn signal interrupt, see **CHAPTER 18 INTERRUPT FUNCTIONS**.

3. n = 0, 1

Figure 8-10. Timing of Interval Timer/Square-Wave Output Operation (1/2)

(a) Basic operation (Operation When 01H ≤ CMP0n ≤ FEH)



<1> The count operation is enabled by setting the TMHEn bit to 1. The count clock starts counting no more than 1 clock after the operation is enabled.

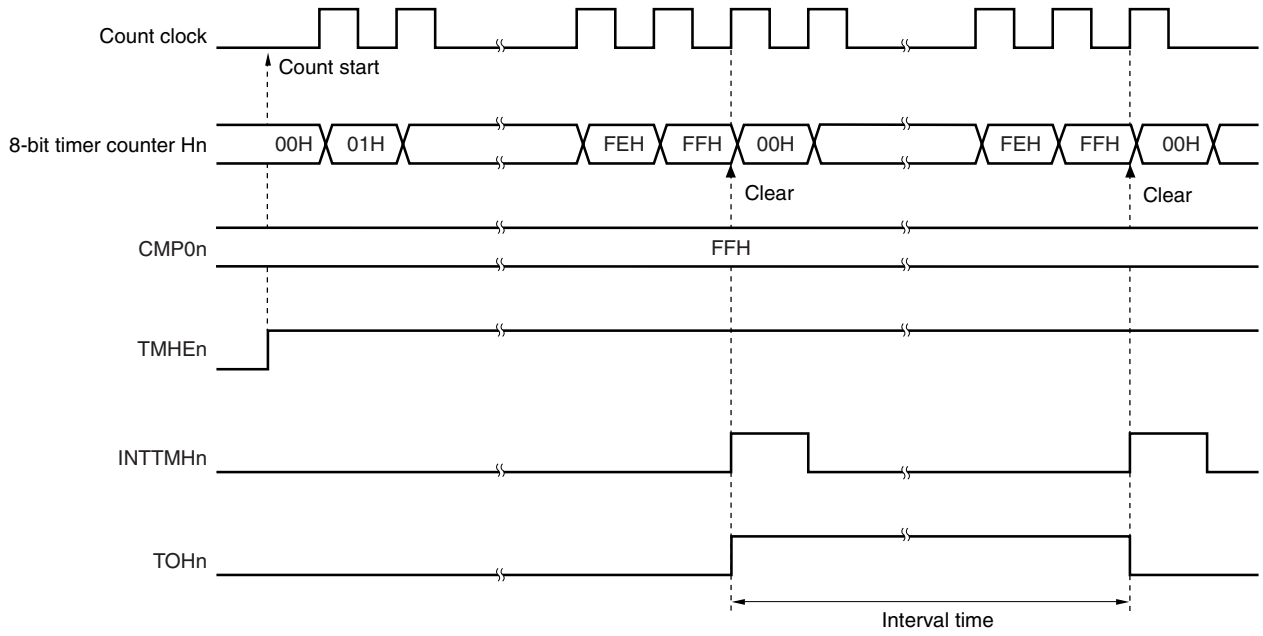
<2> When the value of the 8-bit timer counter Hn matches the value of the CMP0n register, the value of the timer counter is cleared, and the level of the TOHn output is inverted. In addition, the INTTMHn signal is output at the rising edge of the count clock.

<3> If the TMHEn bit is cleared to 0 while timer H is operating, the INTTMHn signal and TOHn output are set to the default level. If they are already at the default level before the TMHEn bit is cleared to 0, then that level is maintained.

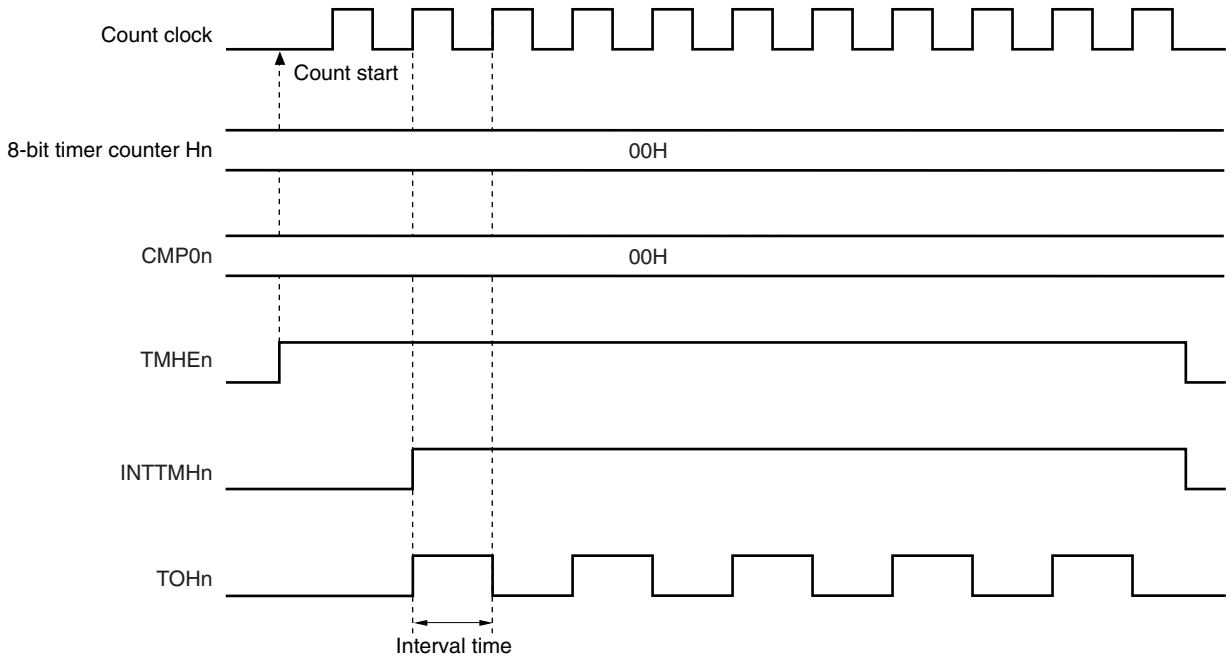
Remark n = 0, 1
01H ≤ N ≤ FEH

Figure 8-10. Timing of Interval Timer/Square-Wave Output Operation (2/2)

(b) Operation when CMP0n = FFH



(c) Operation when CMP0n = 00H



Remark n = 0, 1

8.4.2 Operation as PWM output

In PWM output mode, a pulse with an arbitrary duty and arbitrary cycle can be output.

The 8-bit timer compare register 0n (CMP0n) controls the cycle of timer output (TOHn). Rewriting the CMP0n register during timer operation is prohibited.

The 8-bit timer compare register 1n (CMP1n) controls the duty of timer output (TOHn). Rewriting the CMP1n register during timer operation is possible.

The operation in PWM output mode is as follows.

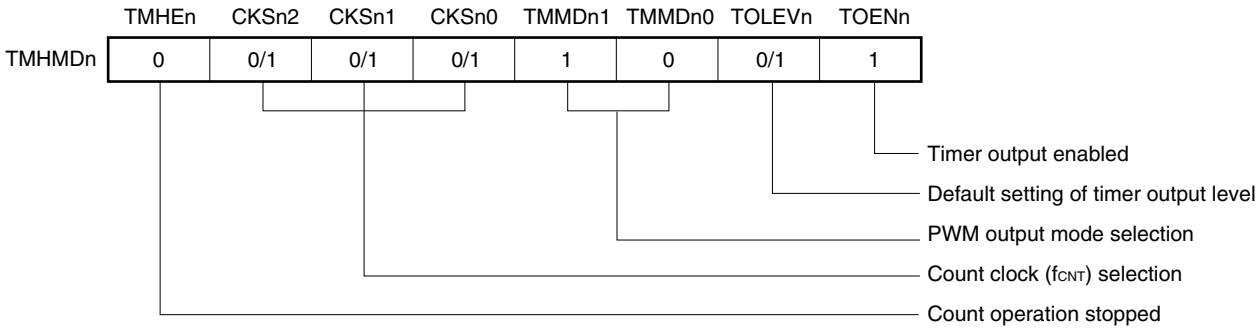
PWM output (TOHn output) outputs an active level and 8-bit timer counter Hn is cleared to 0 when 8-bit timer counter Hn and the CMP0n register match after the timer count is started. PWM output (TOHn output) outputs an inactive level when 8-bit timer counter Hn and the CMP1n register match.

Setting

<1> Set each register.

Figure 8-11. Register Setting in PWM Output Mode

(i) Setting timer H mode register n (TMHMDn)



(ii) Setting CMP0n register

- Compare value (N): Cycle setting

(iii) Setting CMP1n register

- Compare value (M): Duty setting

Remarks 1. n = 0, 1

2. 00H ≤ CMP1n (M) < CMP0n (N) ≤ FFH

<2> The count operation starts when TMHEn = 1.

<3> The CMP0n register is the compare register that is to be compared first after counter operation is enabled. When the values of the 8-bit timer counter Hn and the CMP0n register match, the 8-bit timer counter Hn is cleared, an interrupt request signal (INTTMHn) is generated, and an active level is output. At the same time, the compare register to be compared with the 8-bit timer counter Hn is changed from the CMP0n register to the CMP1n register.

<4> When the 8-bit timer counter Hn and the CMP1n register match, an inactive level is output and the compare register to be compared with the 8-bit timer counter Hn is changed from the CMP1n register to the CMP0n register. At this time, the 8-bit timer counter Hn is not cleared and the INTTMHn signal is not generated.

<5> By performing procedures <3> and <4> repeatedly, a pulse with an arbitrary duty can be obtained.

<6> To stop the count operation, set TMHEn = 0.

If the setting value of the CMP0n register is N, the setting value of the CMP1n register is M, and the count clock frequency is f_{CNT}, the PWM pulse output cycle and duty are as follows.

- PWM pulse output cycle = (N + 1)/f_{CNT}
- Duty = (M + 1)/(N + 1)

Cautions 1. The set value of the CMP1n register can be changed while the timer counter is operating. However, this takes a duration of three operating clocks (signal selected by the CKSn2 to CKSn0 bits of the TMHMDn register) from when the value of the CMP1n register is changed until the value is transferred to the register.

2. Be sure to set the CMP1n register when starting the timer count operation (TMHEn = 1) after the timer count operation was stopped (TMHEn = 0) (be sure to set again even if setting the same value to the CMP1n register).

3. Make sure that the CMP1n register setting value (M) and CMP0n register setting value (N) are within the following range.

00H ≤ CMP1n (M) < CMP0n (N) ≤ FFH

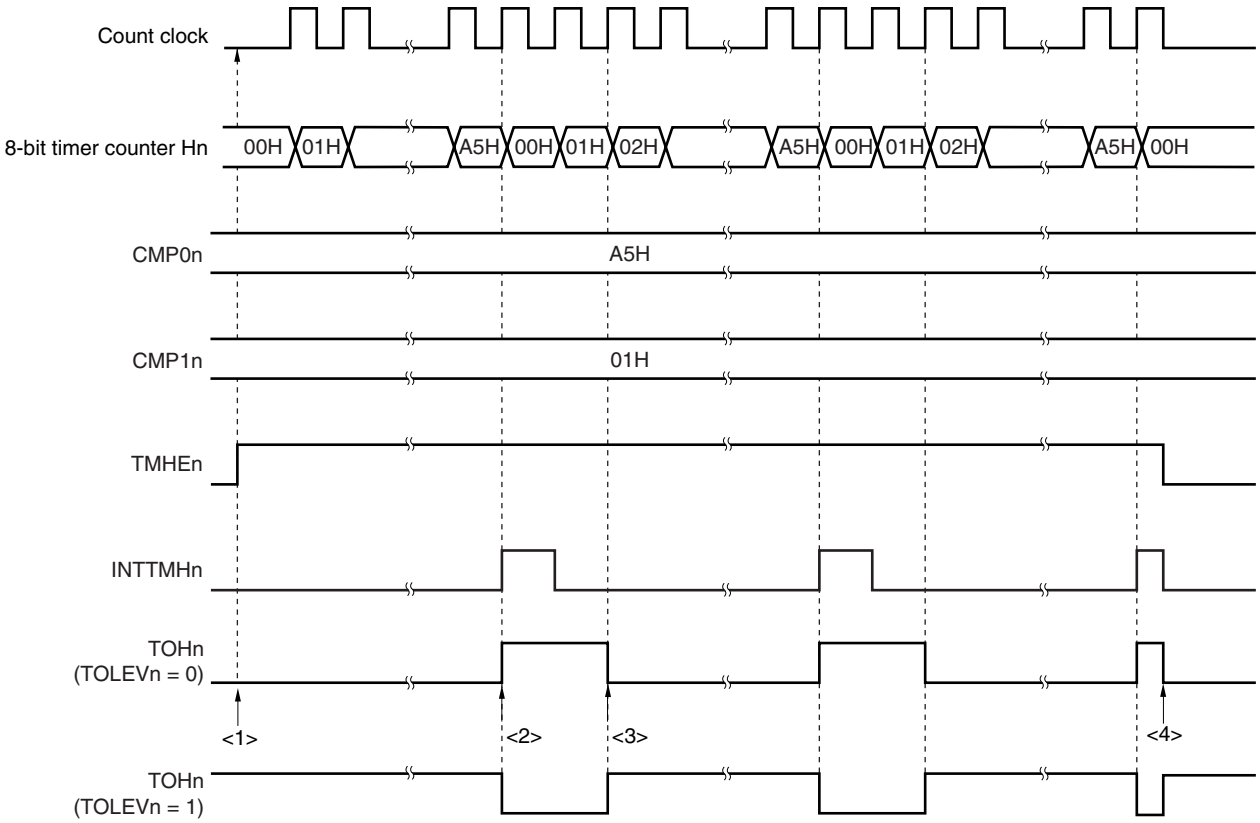
Remarks 1. For the setting of the output pin, see **8.3 (3) Port mode register 1 (PM1)**.

2. For details on how to enable the INTTMHn signal interrupt, see **CHAPTER 18 INTERRUPT FUNCTIONS**.

3. n = 0, 1

Figure 8-12. Operation Timing in PWM Output Mode (1/4)

(a) Basic operation



<1> The count operation is enabled by setting the TMHEn bit to 1. Start the 8-bit timer counter Hn by masking one count clock to count up. At this time, PWM output outputs an inactive level.

<2> When the values of the 8-bit timer counter Hn and the CMP0n register match, an active level is output. At this time, the value of the 8-bit timer counter Hn is cleared, and the INTTMHn signal is output.

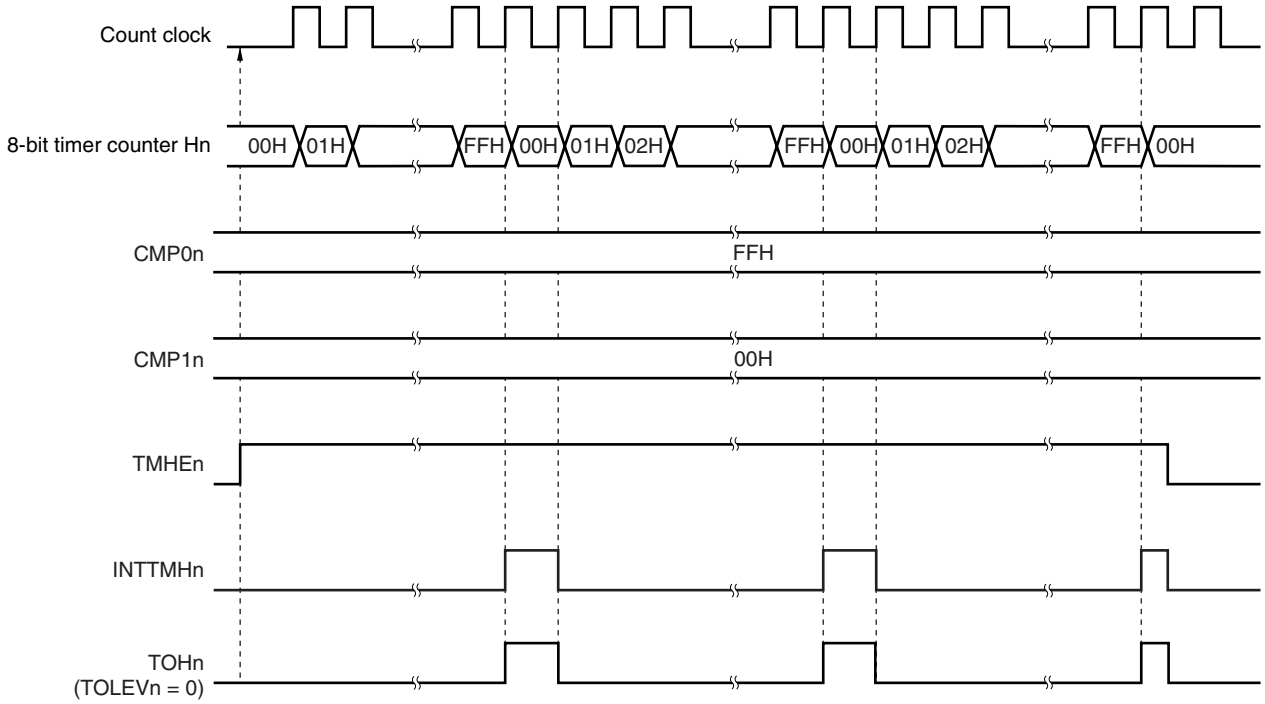
<3> When the values of the 8-bit timer counter Hn and the CMP1n register match, an inactive level is output. At this time, the 8-bit timer counter value is not cleared and the INTTMHn signal is not output.

<4> Clearing the TMHEn bit to 0 during timer Hn operation sets the INTTMHn signal to the default and PWM output to an inactive level.

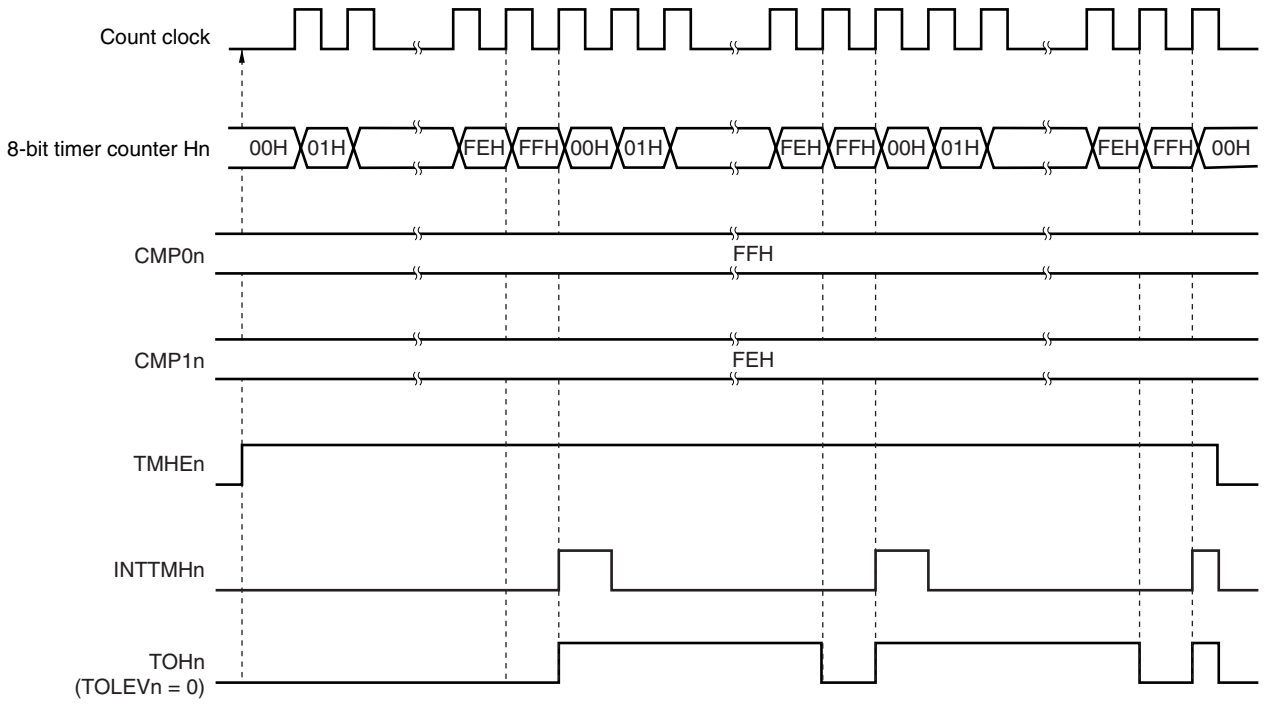
Remark n = 0, 1

Figure 8-12. Operation Timing in PWM Output Mode (2/4)

(b) Operation when CMP0n = FFH, CMP1n = 00H



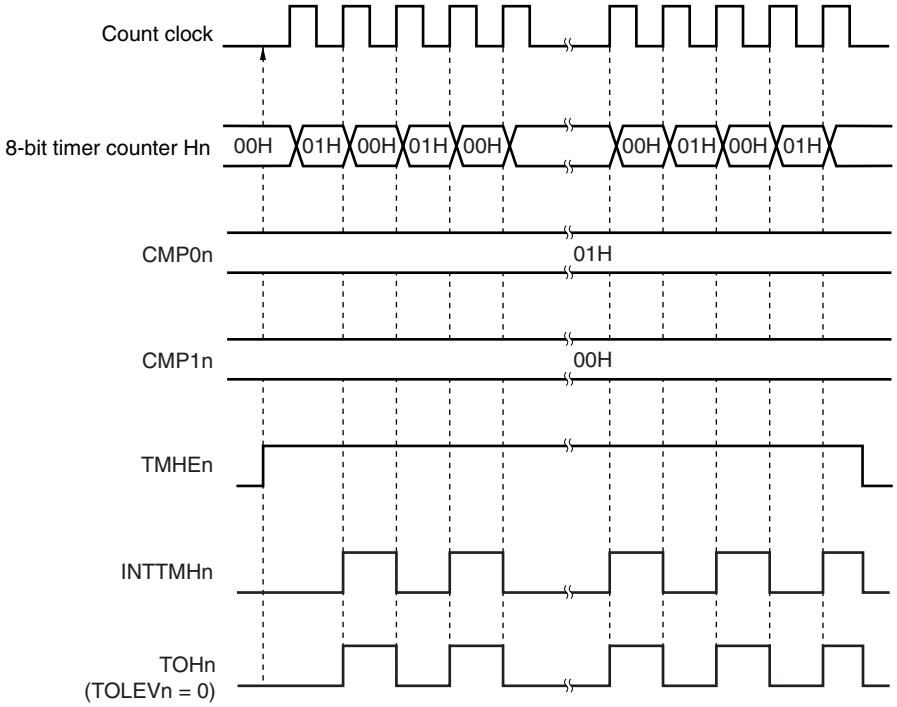
(c) Operation when CMP0n = FFH, CMP1n = FEH



Remark n = 0, 1

Figure 8-12. Operation Timing in PWM Output Mode (3/4)

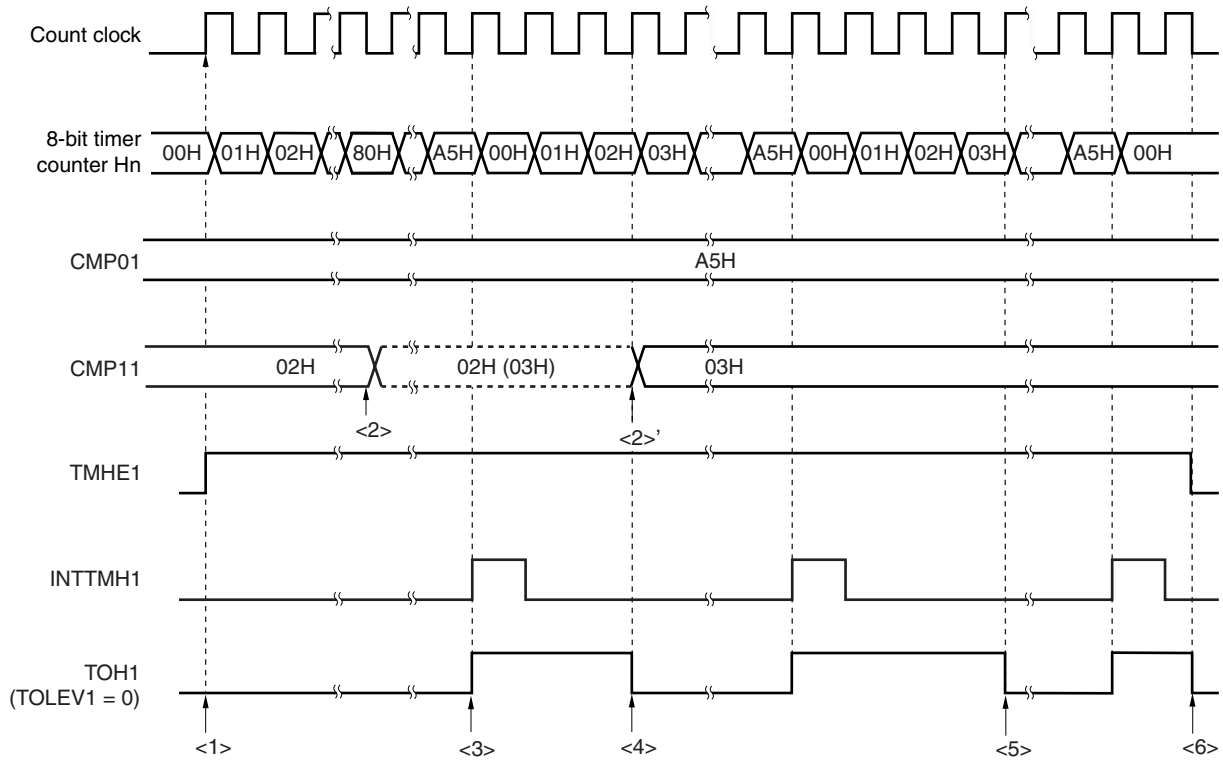
(d) Operation when CMP0n = 01H, CMP1n = 00H



Remark n = 0, 1

Figure 8-12. Operation Timing in PWM Output Mode (4/4)

(e) Operation by changing CMP1n (CMP1n = 02H → 03H, CMP0n = A5H)



<1> The count operation is enabled by setting TMHEn = 1. Start the 8-bit timer counter Hn by masking one count clock to count up. At this time, PWM output outputs an inactive level.

<2> The CMP1n register value can be changed during timer counter operation. This operation is asynchronous to the count clock.

<3> When the values of the 8-bit timer counter Hn and the CMP0n register match, the value of the 8-bit timer counter Hn is cleared, an active level is output, and the INTTMHn signal is output.

<4> If the CMP1n register value is changed, the value is latched and not transferred to the register. When the values of the 8-bit timer counter Hn and the CMP1n register before the change match, the value is transferred to the CMP1n register and the CMP1n register value is changed (<2>').

However, three count clocks or more are required from when the CMP1n register value is changed to when the value is transferred to the register. If a match signal is generated within three count clocks, the changed value cannot be transferred to the register.

<5> When the values of the 8-bit timer counter Hn and the CMP1n register after the change match, an inactive level is output. The 8-bit timer counter Hn is not cleared and the INTTMHn signal is not generated.

<6> Clearing the TMHEn bit to 0 during timer Hn operation sets the INTTMHn signal to the default and PWM output to an inactive level.

Remark n = 0, 1

8.4.3 Carrier generator operation (8-bit timer H1 only)

In the carrier generator mode, the 8-bit timer H1 is used to generate the carrier signal of an infrared remote controller, and the 8-bit timer/event counter 51 is used to generate an infrared remote control signal (time count).

The carrier clock generated by the 8-bit timer H1 is output in the cycle set by the 8-bit timer/event counter 51.

In carrier generator mode, the output of the 8-bit timer H1 carrier pulse is controlled by the 8-bit timer/event counter 51, and the carrier pulse is output from the TOH1 output.

(1) Carrier generation

In carrier generator mode, the 8-bit timer H compare register 01 (CMP01) generates a low-level width carrier pulse waveform and the 8-bit timer H compare register 11 (CMP11) generates a high-level width carrier pulse waveform.

Rewriting the CMP11 register during the 8-bit timer H1 operation is possible but rewriting the CMP01 register is prohibited.

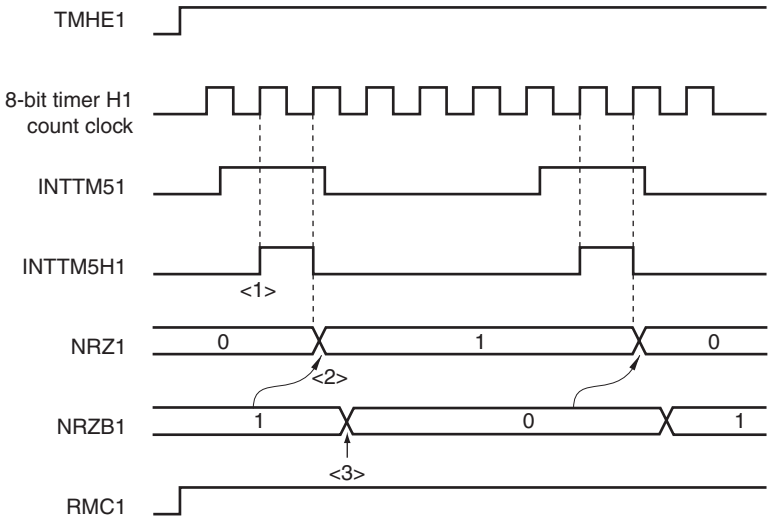
(2) Carrier output control

Carrier output is controlled by the interrupt request signal (INTTM51) of the 8-bit timer/event counter 51 and the NRZB1 and RMC1 bits of the 8-bit timer H carrier control register (TMCYC1). The relationship between the outputs is shown below.

RMC1 Bit	NRZB1 Bit	Output
0	0	Low-level output
0	1	High-level output at rising edge of INTTM51 signal input
1	0	Low-level output
1	1	Carrier pulse output at rising edge of INTTM51 signal input

To control the carrier pulse output during a count operation, the NRZ1 and NRZB1 bits of the TMCYC1 register have a master and slave bit configuration. The NRZ1 bit is read-only but the NRZB1 bit can be read and written. The INTTM51 signal is synchronized with the 8-bit timer H1 count clock and is output as the INTTM5H1 signal. The INTTM5H1 signal becomes the data transfer signal of the NRZ1 bit, and the NRZB1 bit value is transferred to the NRZ1 bit. The timing for transfer from the NRZB1 bit to the NRZ1 bit is as shown below.

Figure 8-13. Transfer Timing



<1> The INTTM51 signal is synchronized with the count clock of the 8-bit timer H1 and is output as the INTTM5H1 signal.

<2> The value of the NRZB1 bit is transferred to the NRZ1 bit at the second clock from the rising edge of the INTTM5H1 signal.

<3> Write the next value to the NRZB1 bit in the interrupt servicing program that has been started by the INTTM5H1 interrupt or after timing has been checked by polling the interrupt request flag. Write data to count the next time to the CR51 register.

Cautions 1. Do not rewrite the NRZB1 bit again until at least the second clock after it has been rewritten, or else the transfer from the NRZB1 bit to the NRZ1 bit is not guaranteed.

2. When the 8-bit timer/event counter 51 is used in the carrier generator mode, an interrupt is generated at the timing of <1>. When the 8-bit timer/event counter 51 is used in a mode other than the carrier generator mode, the timing of the interrupt generation differs.

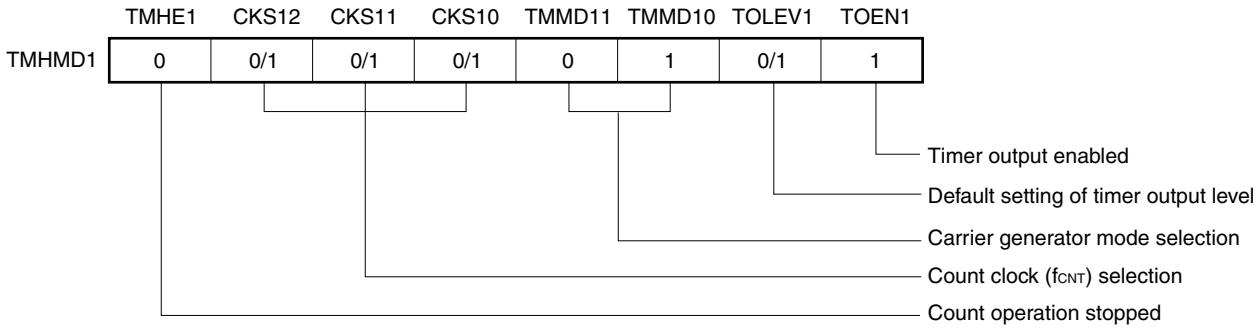
Remark INTTM5H1 is an internal signal and not an interrupt source.

Setting

<1> Set each register.

Figure 8-14. Register Setting in Carrier Generator Mode

(i) Setting 8-bit timer H mode register 1 (TMHMD1)



(ii) CMP01 register setting

- Compare value

(iii) CMP11 register setting

- Compare value

(iv) TMCYC1 register setting

- RMC1 = 1 ... Remote control output enable bit
- NRZB1 = 0/1 ... carrier output enable bit

(v) TCL51 and TMC51 register setting

- See **7.3 Registers Controlling 8-Bit Timer/Event Counters 50 and 51**.

<2> When TMHE1 = 1, the 8-bit timer H1 starts counting.

<3> When TCE51 of the 8-bit timer mode control register 51 (TMC51) is set to 1, the 8-bit timer/event counter 51 starts counting.

<4> After the count operation is enabled, the first compare register to be compared is the CMP01 register. When the count value of the 8-bit timer counter H1 and the CMP01 register value match, the INTTMH1 signal is generated, the 8-bit timer counter H1 is cleared. At the same time, the compare register to be compared with the 8-bit timer counter H1 is switched from the CMP01 register to the CMP11 register.

<5> When the count value of the 8-bit timer counter H1 and the CMP11 register value match, the INTTMH1 signal is generated, the 8-bit timer counter H1 is cleared. At the same time, the compare register to be compared with the 8-bit timer counter H1 is switched from the CMP11 register to the CMP01 register.

<6> By performing procedures <4> and <5> repeatedly, a carrier clock is generated.

<7> The INTTM51 signal is synchronized with count clock of the 8-bit timer H1 and output as the INTTM5H1 signal. The INTTM5H1 signal becomes the data transfer signal for the NRZB1 bit, and the NRZB1 bit value is transferred to the NRZ1 bit.

<8> Write the next value to the NRZB1 bit in the interrupt servicing program that has been started by the INTTM5H1 interrupt or after timing has been checked by polling the interrupt request flag. Write data to count the next time to the CR51 register.

<9> When the NRZ1 bit is high level, a carrier clock is output by TOH1 output.

<10> By performing the procedures above, an arbitrary carrier clock is obtained. To stop the count operation, clear TMHE1 to 0.

If the setting value of the CMP01 register is N, the setting value of the CMP11 register is M, and the count clock frequency is f_{CNT}, the carrier clock output cycle and duty are as follows.

- Carrier clock output cycle = (N + M + 2)/f_{CNT}
- Duty = High-level width/carrier clock output width = (M + 1)/(N + M + 2)

Cautions 1. Be sure to set the CMP11 register when starting the timer count operation (TMHE1 = 1) after the timer count operation was stopped (TMHE1 = 0) (be sure to set again even if setting the same value to the CMP11 register).

2. Set so that the count clock frequency of TMH1 becomes more than 6 times the count clock frequency of TM51.

3. Set the values of the CMP01 and CMP11 registers in a range of 01H to FFH.

4. The set value of the CMP11 register can be changed while the timer counter is operating. However, it takes the duration of three operating clocks (signal selected by the CKS12 to CKS10 bits of the TMHMD1 register) since the value of the CMP11 register has been changed until the value is transferred to the register.

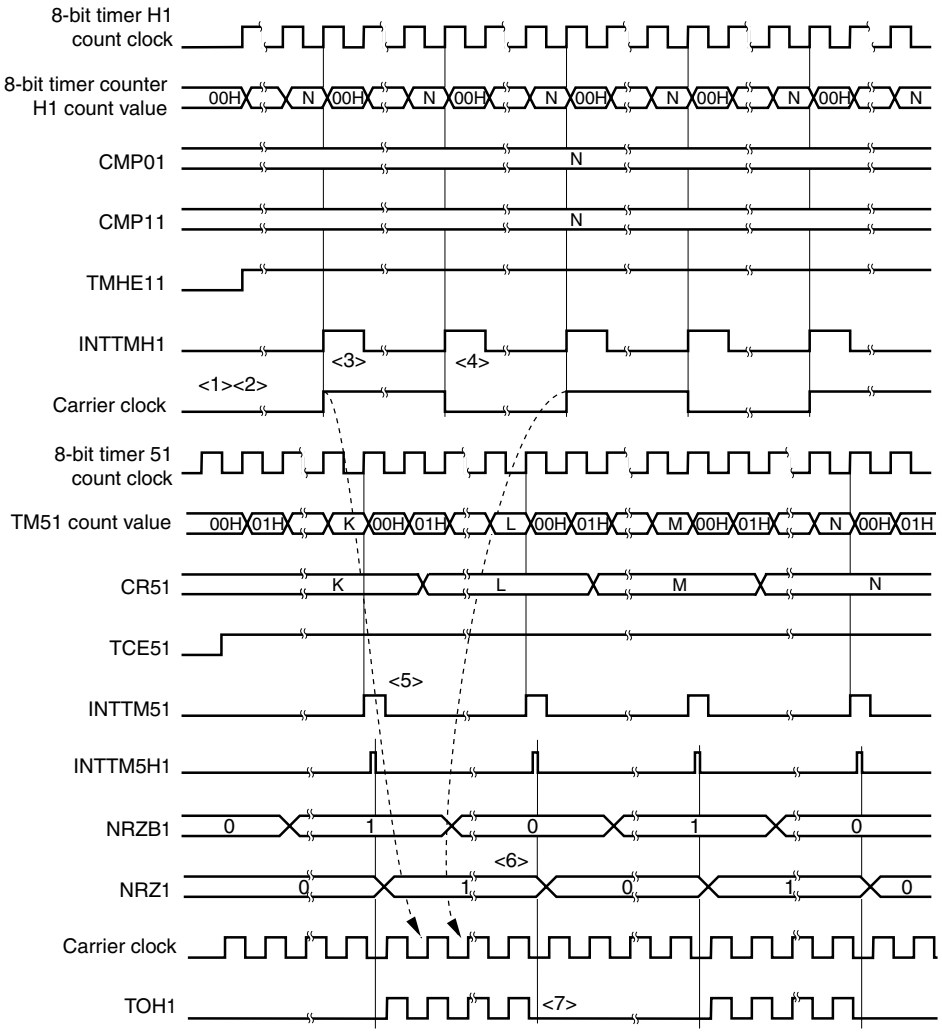
5. Be sure to set the RMC1 bit before the count operation is started.

Remarks 1. For the setting of the output pin, see **8.3 (3) Port mode register 1 (PM1)**.

2. For how to enable the INTTMH1 signal interrupt, see **CHAPTER 18 INTERRUPT FUNCTIONS**.

Figure 8-15. Carrier Generator Mode Operation Timing (1/3)

(a) Operation when CMP01 = N, CMP11 = N



<1> When TMHE1 = 0 and TCE51 = 0, the 8-bit timer counter H1 operation is stopped.

<2> When TMHE1 = 1 is set, the 8-bit timer counter H1 starts a count operation. At that time, the carrier clock remains default.

<3> When the count value of the 8-bit timer counter H1 matches the CMP01 register value, the first INTTMH1 signal is generated, the carrier clock signal is inverted, and the compare register to be compared with the 8-bit timer counter H1 is switched from the CMP01 register to the CMP11 register. The 8-bit timer counter H1 is cleared to 00H.

<4> When the count value of the 8-bit timer counter H1 matches the CMP11 register value, the INTTMH1 signal is generated, the carrier clock signal is inverted, and the compare register to be compared with the 8-bit timer counter H1 is switched from the CMP11 register to the CMP01 register. The 8-bit timer counter H1 is cleared to 00H. By performing procedures <3> and <4> repeatedly, a carrier clock with duty fixed to 50% is generated.

<5> When the INTTM51 signal is generated, it is synchronized with the 8-bit timer H1 count clock and is output as the INTTM5H1 signal.

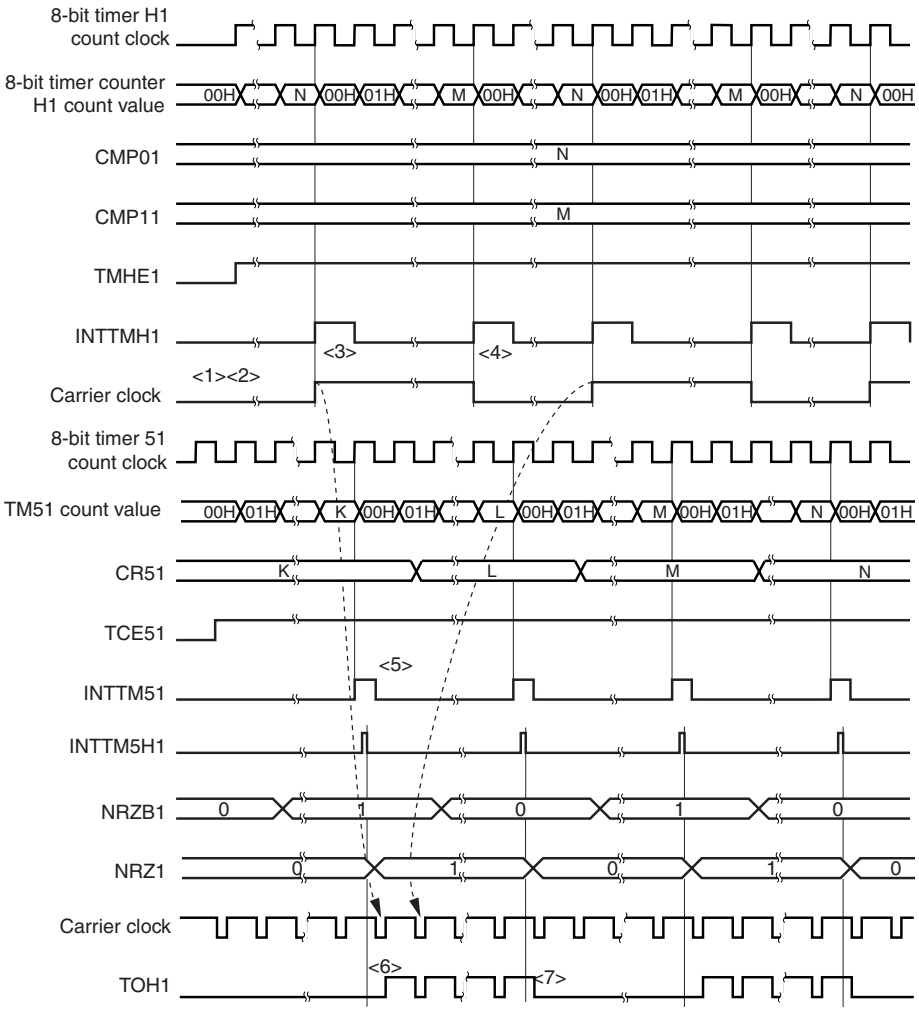
<6> The INTTM5H1 signal becomes the data transfer signal for the NRZB1 bit, and the NRZB1 bit value is transferred to the NRZ1 bit.

<7> When NRZ1 = 0 is set, the TOH1 output becomes low level.

Remark INTTM5H1 is an internal signal and not an interrupt source.

Figure 8-15. Carrier Generator Mode Operation Timing (2/3)

(b) Operation when CMP01 = N, CMP11 = M



<1> When TMHE1 = 0 and TCE51 = 0, the 8-bit timer counter H1 operation is stopped.

<2> When TMHE1 = 1 is set, the 8-bit timer counter H1 starts a count operation. At that time, the carrier clock remains default.

<3> When the count value of the 8-bit timer counter H1 matches the CMP01 register value, the first INTTMH1 signal is generated, the carrier clock signal is inverted, and the compare register to be compared with the 8-bit timer counter H1 is switched from the CMP01 register to the CMP11 register. The 8-bit timer counter H1 is cleared to 00H.

<4> When the count value of the 8-bit timer counter H1 matches the CMP11 register value, the INTTMH1 signal is generated, the carrier clock signal is inverted, and the compare register to be compared with the 8-bit timer counter H1 is switched from the CMP11 register to the CMP01 register. The 8-bit timer counter H1 is cleared to 00H. By performing procedures <3> and <4> repeatedly, a carrier clock with duty fixed to other than 50% is generated.

<5> When the INTTM51 signal is generated, it is synchronized with the 8-bit timer H1 count clock and is output as the INTTM5H1 signal.

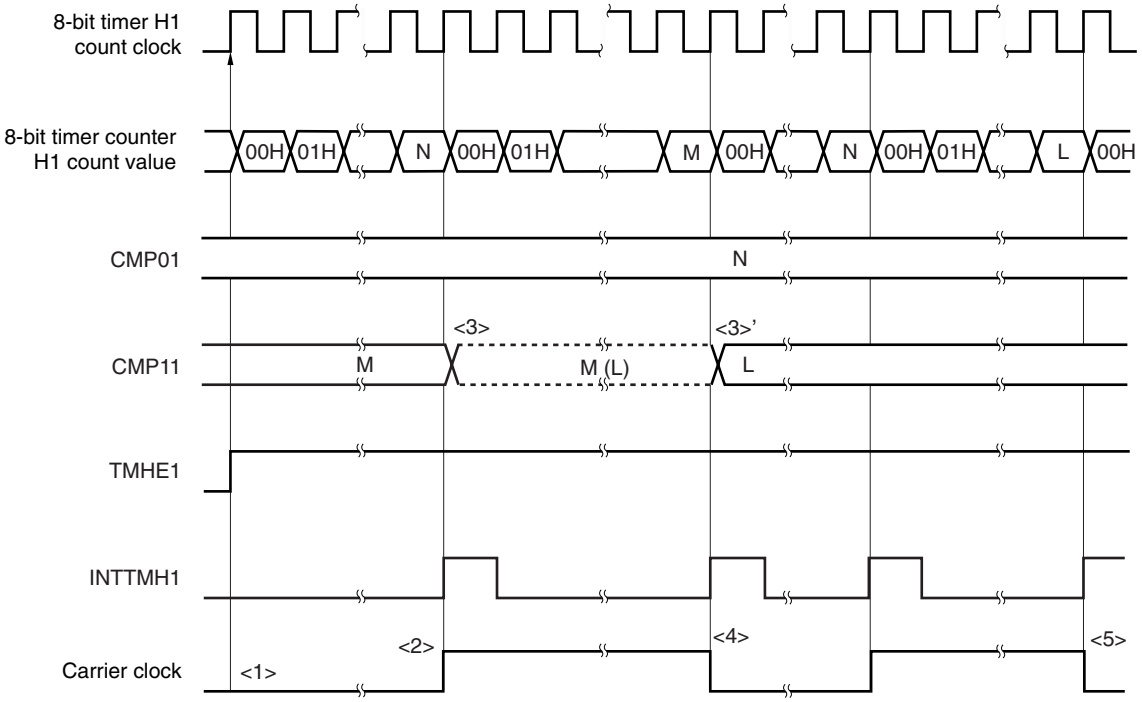
<6> A carrier signal is output at the first rising edge of the carrier clock if NRZ1 is set to 1.

<7> When NRZ1 = 0, the TOH1 output is held at the high level and is not changed to low level while the carrier clock is high level (from <6> and <7>, the high-level width of the carrier clock waveform is guaranteed).

Remark INTTM5H1 is an internal signal and not an interrupt source.

Figure 8-15. Carrier Generator Mode Operation Timing (3/3)

(c) Operation when CMP11 is changed



<1> When TMHE1 = 1 is set, the 8-bit timer H1 starts a count operation. At that time, the carrier clock remains default.

<2> When the count value of the 8-bit timer counter H1 matches the value of the CMP01 register, the INTTMH1 signal is output, the carrier signal is inverted, and the timer counter is cleared to 00H. At the same time, the compare register whose value is to be compared with that of the 8-bit timer counter H1 is changed from the CMP01 register to the CMP11 register.

<3> The CMP11 register is asynchronous to the count clock, and its value can be changed while the 8-bit timer H1 is operating. The new value (L) to which the value of the register is to be changed is latched. When the count value of the 8-bit timer counter H1 matches the value (M) of the CMP11 register before the change, the CMP11 register is changed (<3>').

However, it takes three count clocks or more since the value of the CMP11 register has been changed until the value is transferred to the register. Even if a match signal is generated before the duration of three count clocks elapses, the new value is not transferred to the register.

<4> When the count value of 8-bit timer counter H1 matches the value (M) of the CMP1 register before the change, the INTTMH1 signal is output, the carrier signal is inverted, and the timer counter is cleared to 00H. At the same time, the compare register whose value is to be compared with that of the 8-bit timer counter H1 is changed from the CMP11 register to the CMP01 register.

<5> The timing at which the count value of the 8-bit timer counter H1 and the CMP11 register value match again is indicated by the value after the change (L).

CHAPTER 9 REAL-TIME COUNTER

Item	78K0/KB2-A	78K0/KC2-A
	30 pins	48 pins
Real-time counter	–	√ (RTC output : 2)

Remark √ : Mounted, −: Not mounted

9.1 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

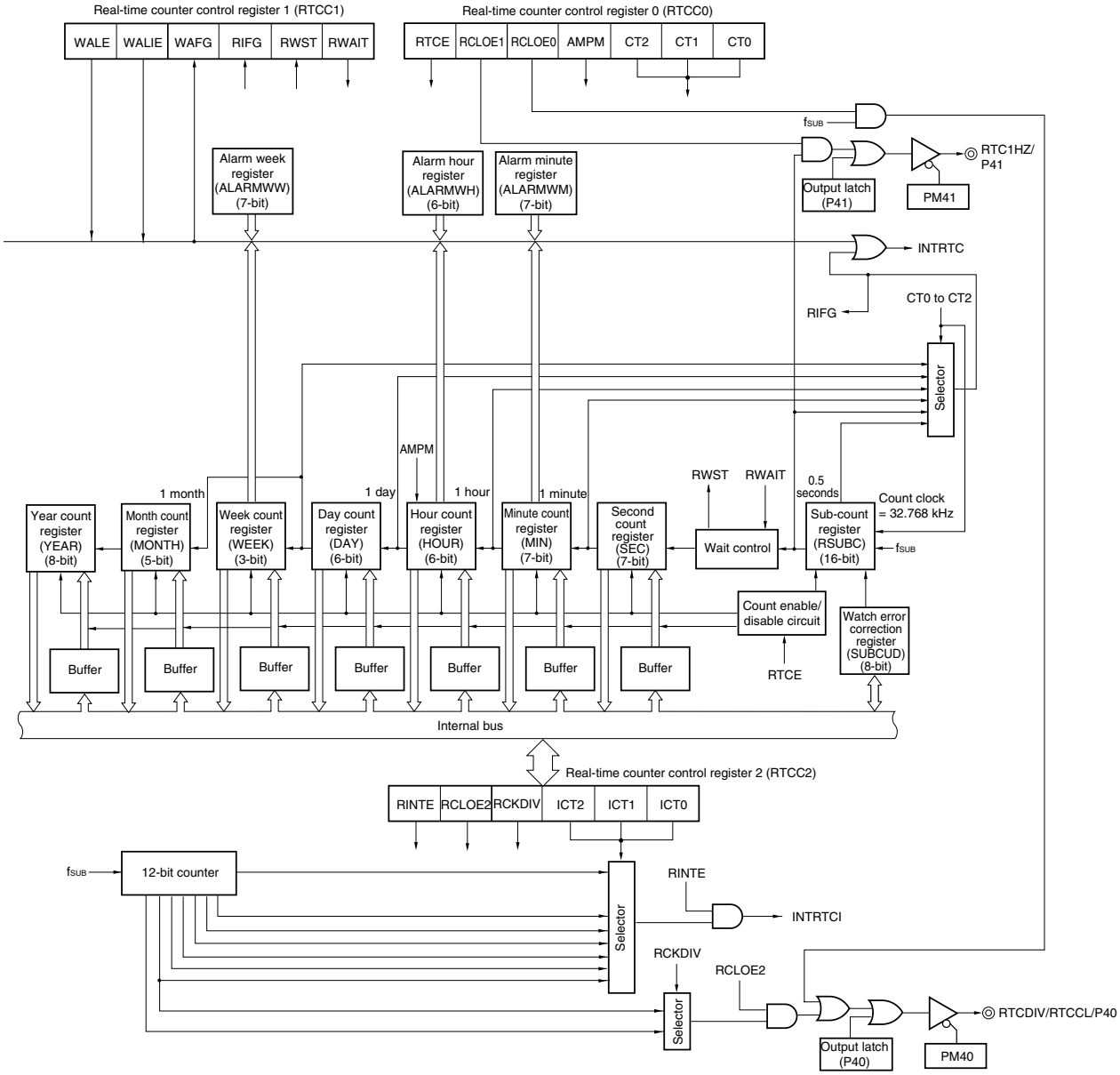
9.2 Configuration of Real-Time Counter

The real-time counter includes the following hardware.

Table 9-1. Configuration of Real-Time Counter

Item	Configuration
Control registers	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 9-1. Block Diagram of Real-Time Counter



9.3 Registers Controlling Real-Time Counter

The real-time counter is controlled by the following 16 registers.

(1) 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 9-2. Format of Real-Time Counter Control Register 0 (RTCC0)

Address: FF7DH After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	3	2	1	0
RTCC0	RTCE	0	RCLOE1	RCLOE0	AMPM	CT2	CT1	CT0

RTCE	Real-time counter operation control
0	Stops counter operation.
1	Starts counter operation.

RCLOE1	RTC1HZ pin output control
0	Disables output of RTC1HZ pin (1 Hz).
1	Enables output of RTC1HZ pin (1 Hz).

RCLOE0[Note]	RTCCL pin output control
0	Disables output of RTCCL pin (32.768 kHz).
1	Enables output of RTCCL pin (32.768 kHz).

AMPM	Selection of 12-/24-hour system
0	12-hour system (a.m. and p.m. are displayed.)
1	24-hour system
• To change the value of AMPM, set RWAIT (bit 0 of RTCC1) to 1, and re-set the hour count register (HOUR). • Table 9-2 shows the displayed time digits that are displayed.	

CT2	CT1	CT0	Constant-period interrupt (INTRTC) selection
0	0	0	Does not use constant-period interrupt function.
0	0	1	Once per 0.5 s (synchronized with second count up)
0	1	0	Once per 1 s (same time as second count up)
0	1	1	Once per 1 m (second 00 of every minute)
1	0	0	Once per 1 hour (minute 00 and second 00 of every hour)
1	0	1	Once per 1 day (hour 00, minute 00, and second 00 of every day)
1	1	×	Once per 1 month (Day 1, hour 00 a.m., minute 00, and second 00 of every month)
When changing the values of CT2 to CT0 , rewrite the values of CT2 to CT0 after disabling interrupt servicing INTRTC. Furthermore, after rewriting the values of CT2 to CT0, enable INTRTC after clearing the WAFG flag, RIFG flag, and RTCIF flags.			

Note RCLOE0 and RCLOE2 must not be enabled at the same time.

Caution If RCLOE0 and RCLOE1 are changed when RTCE = 1, a pulse with a narrow width may be generated on the 32.768 kHz and 1 Hz output signals.

Remark ×: don't care

(2) 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 9-3. Format of Real-Time Counter Control Register 1 (RTCC1) (1/2)

Address: FF7EH After reset: 00H R/W

Symbol	<7>	<6>	5	<4>	<3>	2	<1>	<0>
RTCC1	WALE[Note]	WALIE	0	WAFG	RIFG	0	RWST	RWAIT

WALE[Note]	Alarm operation control
0	Match operation is invalid.
1	Match operation is valid.
When setting each alarm register (WALIE flag of RTCC1, the ALARMWM register, the ALARMWH register, and the ALARMWW register), set match operation to be disable (“0”) for the WALE bit.	

WALIE	Control of alarm interrupt (INTRTC) function operation
0	Does not generate interrupt on matching of alarm.
1	Generates interrupt on matching of alarm.

WAFG	Alarm detection status flag
0	Alarm mismatch
1	Detection of matching of alarm
This is a status flag that indicates detection of matching with the alarm. It is valid only when WALE = 1 and is set to “1” one clock (32.768 kHz) after matching of the alarm is detected. This flag is cleared when “0” is written to it. Writing “1” to it is invalid.	

Note Rewrite the WALE bit after disabling interrupt servicing INTRTC. Furthermore, after rewriting the WALE bit, enable INTRTC after clearing the WAFG flag, RIFG flag, and RTCIF flags.

Figure 9-3. 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 register is 1.	

RWAIT	Wait control of real-time counter
0	Sets counter operation.
1	Stops SEC to YEAR counters. Mode to read or write counter value
This bit controls the operation of the counter. Be sure to write “1” to it to read or write the counter value. Because RSUBC continues operation, complete reading or writing of it in 1 second, and clear this bit back to 0. When RWAIT = 1, it takes up to 1 clock (32.768 kHz) until the counter value can be read or written. If RSUBC overflows when RWAIT = 1, it counts up after RWAIT = 0. If the second count register is written, however, it does not count up because RSUBC is cleared.	

Caution The RIFG and WAFG flags may be cleared when the RTCC1 register is written by using a 1-bit manipulation instruction. Use, therefore, an 8-bit manipulation instruction in order to write to the RTCC1 register. To prevent the RIFG and WAFG flags from being cleared during writing, disable writing by setting “1” to the corresponding bit. When the value may be rewritten because the RIFG and WAFG flags are not being used, the RTCC1 register may be written by using a 1-bit manipulation instruction.

Remark Fixed-cycle interrupts and alarm match interrupts use the same interrupt source (INTRTC). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC occurrence.

(3) 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 9-4. Format of Real-Time Counter Control Register 2 (RTCC2)

Address: FF7FH 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	$2^6/f_{SUB}$ (1.953125 ms)
1	0	0	1	$2^7/f_{SUB}$ (3.90625 ms)
1	0	1	0	2^8f_{SUB} (7.8125 ms)
1	0	1	1	$2^9/f_{SUB}$ (15.625 ms)
1	1	0	0	$2^{10}/f_{SUB}$ (31.25 ms)
1	1	0	1	$2^{11}/f_{SUB}$ (62.5 ms)
1	1	1	×	$2^{12}/f_{SUB}$ (125 ms)

RCLOE2[Note]	RTCDIV pin output control
0	Output of RTCDIV pin is disabled.
1	Output of RTCDIV pin is enabled.

RCKDIV	Selection of RTCDIV pin output frequency
0	RTCDIV pin outputs 512 Hz (1.95 ms).
1	RTCDIV pin outputs 16.384 kHz (0.061 ms).

Note RCLOE0 and RCLOE2 must not be enabled at the same time.

Cautions 1. Change ICT2, ICT1, and ICT0 when RINTE = 0.

2. When the output from RTCDIV pin is stopped, the output continues after a maximum of two clocks of f_{SUB} and enters the low level. While 512 Hz is output, and when the output is stopped immediately after entering the high level, a pulse of at least one clock width of f_{SUB} may be generated.

Remark f_{SUB} **:** Subsystem clock frequency

(4) 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 9-5. Format of Sub-Count Register (RSUBC)

Address: FF70H, FF71H After reset: 0000H R

	FFB1H								FFB0H							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSUBC																

(5) Second count register (SEC)

The SEC register is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of seconds.

It counts up when the sub-counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Set a decimal value of 00 to 59 to this register in BCD code. If a value outside the range is set, the register value returns to the normal value after 1 period.

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

Reset signal generation clears this register to 00H.

Figure 9-6. Format of Second Count Register (SEC)

Address: FF72H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SEC	0	SEC40	SEC20	SEC10	SEC8	SEC4	SEC2	SEC1

(6) Minute count register (MIN)

The MIN register is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of minutes.

It counts up when the second counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Set a decimal value of 00 to 59 to this register in BCD code. If a value outside the range is set, the register value returns to the normal value after 1 period.

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

Reset signal generation clears this register to 00H.

Figure 9-7. Format of Minute Count Register (MIN)

Address: FF73H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MIN	0	MIN40	MIN20	MIN10	MIN8	MIN4	MIN2	MIN1

(7) Hour count register (HOUR)

The HOUR register is an 8-bit register that takes a value of 00 to 23 or 01 to 12, 21 to 32 (decimal) and indicates the count value of hours.

It counts up when the minute counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Set a decimal value of 00 to 23, 01 to 12, or 21 to 32 by using BCD code. If a value outside the range is set, the register value returns to the normal value after 1 period.

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

Reset signal generation clears this register to 12H.

However, the value of this register is 00H if the AMPM bit is set to 1 after reset.

Figure 9-8. Format of Hour Count Register (HOUR)

Address: FF74H After reset: 12H R/W

Symbol	7	6	5	4	3	2	1	0
HOUR	0	0	HOUR20	HOUR10	HOUR8	HOUR4	HOUR2	HOUR1

Caution Bit 5 (HOUR20) of HOUR indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).

Table 9-2. Displayed Time Digits

24-Hour Display (AMPM bit = 1)		12-Hour Display (AMPM bit = 0)	
Time	HOUR Register	Time	HOUR Register
0	00H	0 a.m.	12H
1	01H	1 a.m.	01H
2	02H	2 a.m.	02H
3	03H	3 a.m.	03H
4	04H	4 a.m.	04H
5	05H	5 a.m.	05H
6	06H	6 a.m.	06H
7	07H	7 a.m.	07H
8	08H	8 a.m.	08H
9	09H	9 a.m.	09H
10	10H	10 a.m.	10H
11	11H	11 a.m.	11H
12	12H	0 p.m.	32H
13	13H	1 p.m.	21H
14	14H	2 p.m.	22H
15	15H	3 p.m.	23H
16	16H	4 p.m.	24H
17	17H	5 p.m.	25H
18	18H	6 p.m.	26H
19	19H	7 p.m.	27H
20	20H	8 p.m.	28H
21	21H	9 p.m.	29H
22	22H	10 p.m.	30H
23	23H	11 p.m.	31H

The HOUR register value is set to 12-hour display when the AMPM bit is “0” and to 24-hour display when the AMPM bit is “1”.

In 12-hour display, the fifth bit of the HOUR register displays 0 for AM and 1 for PM.

(8) Day count register (DAY)

The DAY register is an 8-bit register that takes a value of 1 to 31 (decimal) and indicates the count value of days.

It counts up when the hour counter overflows.

This counter counts as follows.

- 01 to 31 (January, March, May, July, August, October, December)
- 01 to 30 (April, June, September, November)
- 01 to 29 (February, leap year)
- 01 to 28 (February, normal year)

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Set a decimal value of 01 to 31 to this register in BCD code. If a value outside the range is set, the register value returns to the normal value after 1 period.

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

Reset signal generation clears this register to 01H.

Figure 9-9. Format of Day Count Register (DAY)

Address: FF76H After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
DAY	0	0	DAY20	DAY10	DAY8	DAY4	DAY2	DAY1

(9) Week count register (WEEK)

The WEEK register is an 8-bit register that takes a value of 0 to 6 (decimal) and indicates the count value of weekdays.

It counts up in synchronization with the day counter.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Set a decimal value of 00 to 06 to this register in BCD code. If a value outside the range is set, the register value returns to the normal value after 1 period.

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

Reset signal generation clears this register to 00H.

Figure 9-10. Format of Week Count Register (WEEK)

Address: FF75H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
WEEK	0	0	0	0	0	WEEK4	WEEK2	WEEK1

Caution The value corresponding to the month count register or the day count register is not stored in the week count register automatically. After reset release, set the week count register as follow.

Day	WEEK
Sunday	00H
Monday	01H
Tuesday	02H
Wednesday	03H
Thursday	04H
Friday	05H
Saturday	06H

(10) Month count register (MONTH)

The MONTH register is an 8-bit register that takes a value of 1 to 12 (decimal) and indicates the count value of months.

It counts up when the day counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Set a decimal value of 01 to 12 to this register in BCD code. If a value outside the range is set, the register value returns to the normal value after 1 period.

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

Reset signal generation clears this register to 01H.

Figure 9-11. Format of Month Count Register (MONTH)

Address: FF77H After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
MONTH	0	0	0	MONTH10	MONTH8	MONTH4	MONTH2	MONTH1

(11) Year count register (YEAR)

The YEAR register is an 8-bit register that takes a value of 0 to 99 (decimal) and indicates the count value of years.

It counts up when the month 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 the range is set, the register value returns to the normal value after 1 period.

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

Reset signal generation clears this register to 00H.

Figure 9-12. Format of Year Count Register (YEAR)

Address: FF78H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
YEAR	YEAR80	YEAR40	YEAR20	YEAR10	YEAR8	YEAR4	YEAR2	YEAR1

(12) Watch error correction register (SUBCUD)

This register is used to correct the watch with high accuracy when it is slow or fast by changing the value (reference value: 7FFFH) that overflows from the sub-count register (RSUBC) to the second count register.

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

Reset signal generation clears this register to 00H.

Figure 9-13. Format of Watch Error Correction Register (SUBCUD)

Address: FF79H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SUBCUD	DEV	F6	F5	F4	F3	F2	F1	F0

DEV	Setting of watch error correction timing
0	Corrects watch error when the second digits are at 00, 20, or 40 (every 20 seconds).
1	Corrects watch error only when the second digits are at 00 (every 60 seconds).

F6	Setting of watch error correction value
0	Increases by {(F5, F4, F3, F2, F1, F0) – 1} × 2.
1	Decreases by {(/F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2.
When (F6, F5, F4, F3, F2, F1, F0) = (*, 0, 0, 0, 0, 0, *), the watch error is not corrected. * is 0 or 1. /F5 to /F0 are the inverted values of the corresponding bits (000011 when 111100). Range of correction value: (when F6 = 0) 2, 4, 6, 8, … , 120, 122, 124 (when F6 = 1) –2, –4, –6, –8, … , –120, –122, –124	

The range of value that can be corrected by using the watch error correction register (SUBCUD) is shown below.

	DEV = 0 (correction every 20 seconds)	DEV = 1 (correction every 60 seconds)
Correctable range	–189.2 ppm to 189.2 ppm	–63.1 ppm to 63.1 ppm
Maximum excludes quantization error	± 1.53 ppm	± 0.51 ppm
Minimum resolution	± 3.05 ppm	± 1.02 ppm

Remark Set DEV to 0 when the correction range is –63.1 ppm or less, or 63.1 ppm or more.

Caution Rewrite the SUBCUD register after disabling interrupt servicing INTRTC. Furthermore, after rewriting the SUBCUD register, enable INTRTC after clearing the WAFG flag, RIFG flag, and RTCIF flags.

(13) 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 9-14. Format of Alarm Minute Register (ALARMWM)

Address: FF7AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWM	0	WM40	WM20	WM10	WM8	WM4	WM2	WM1

(14) Alarm hour register (ALARMWH)

This register is used to set hours of alarm.

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

Reset signal generation clears this register to 12H.

However, the value of this register is 00H if the AMPM bit is set to 1 after reset.

Caution Set a decimal value of 00 to 23, 01 to 12, or 21 to 32 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

Figure 9-15. Format of Alarm Hour Register (ALARMWH)

Address: FF7BH 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).

(15) Alarm week register (ALARMWW)

This register is used to set date of alarm.

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

Reset signal generation clears this register to 00H.

Figure 9-16 Format of Alarm Week Register (ALARMWW)

Address: FF88H 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

(16) Port mode register 4 (PM4)

This register sets port 4 input/output in 1-bit units.

When using the P40/RTCDIV/RTCCL pin for clock output real-time counter, and P41/RTC1HZ pins for clock output of real-time counter correction, clear PM40 and PM41 and the output latches of P40 and P41 to 0.

PM4 is set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 9-17. Format of Port Mode Register 4 (PM4)

Address: FF24H After reset: FFH R/W

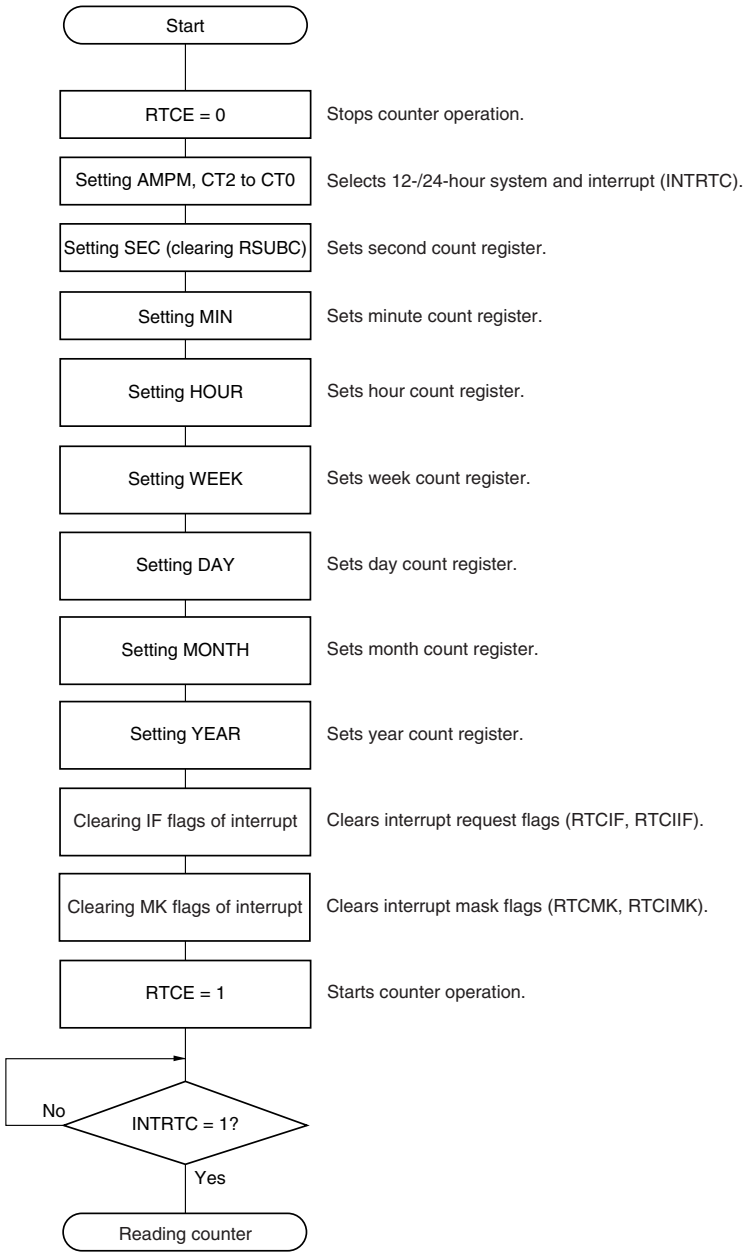
Symbol	7	6	5	4	3	2	1	0
PM4	1	1	1	1	1	PM42	PM41	PM40

PM4n	P4n pin I/O mode selection (n = 0 to 2)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

9.4 Real-Time Counter Operation

9.4.1 Starting operation of real-time counter

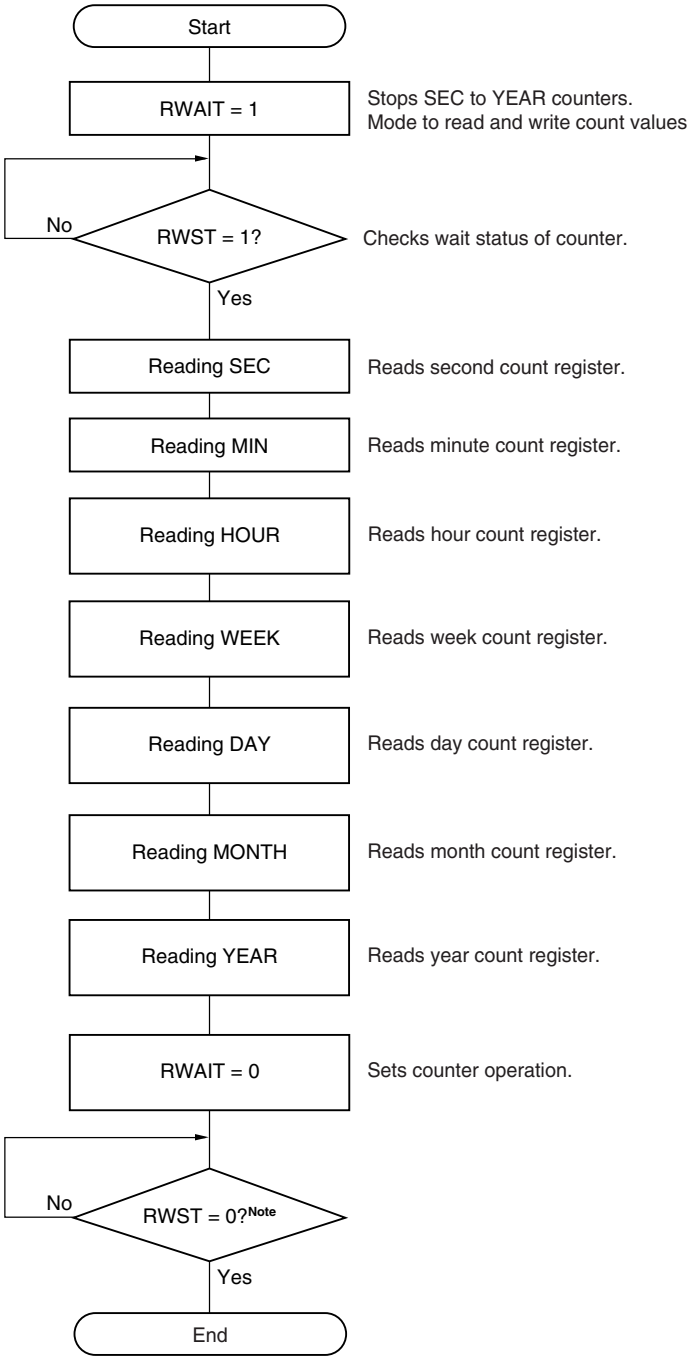
Figure 9-18. Procedure for Starting Operation of Real-Time Counter



9.4.2 Reading/writing real-time counter

Read or write the counter after setting 1 to RWAIT first.

Figure 9-19. Procedure for Reading Real-Time Counter

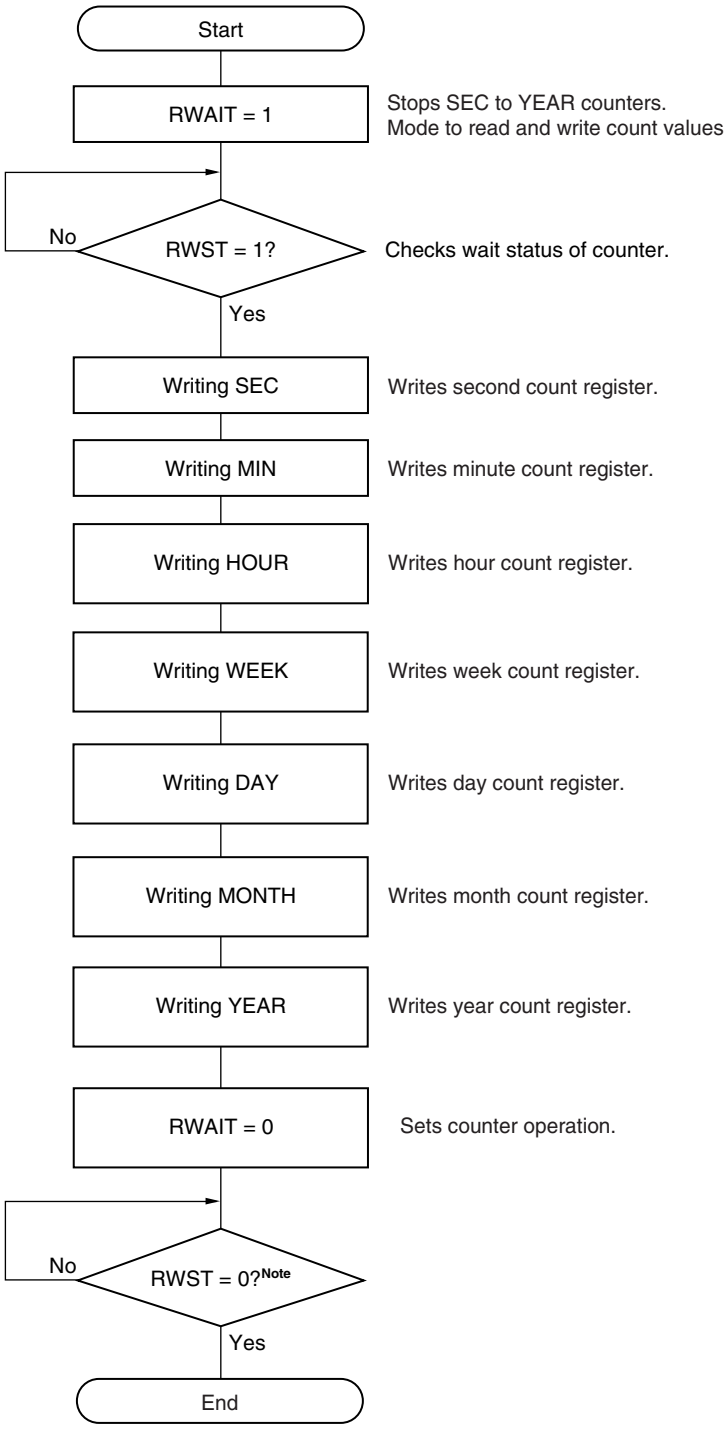


Note Be sure to confirm that RWST = 0 before setting STOP mode.

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 9-20. Procedure for Writing Real-Time Counter



Note Be sure to confirm that RWST = 0 before setting STOP mode.

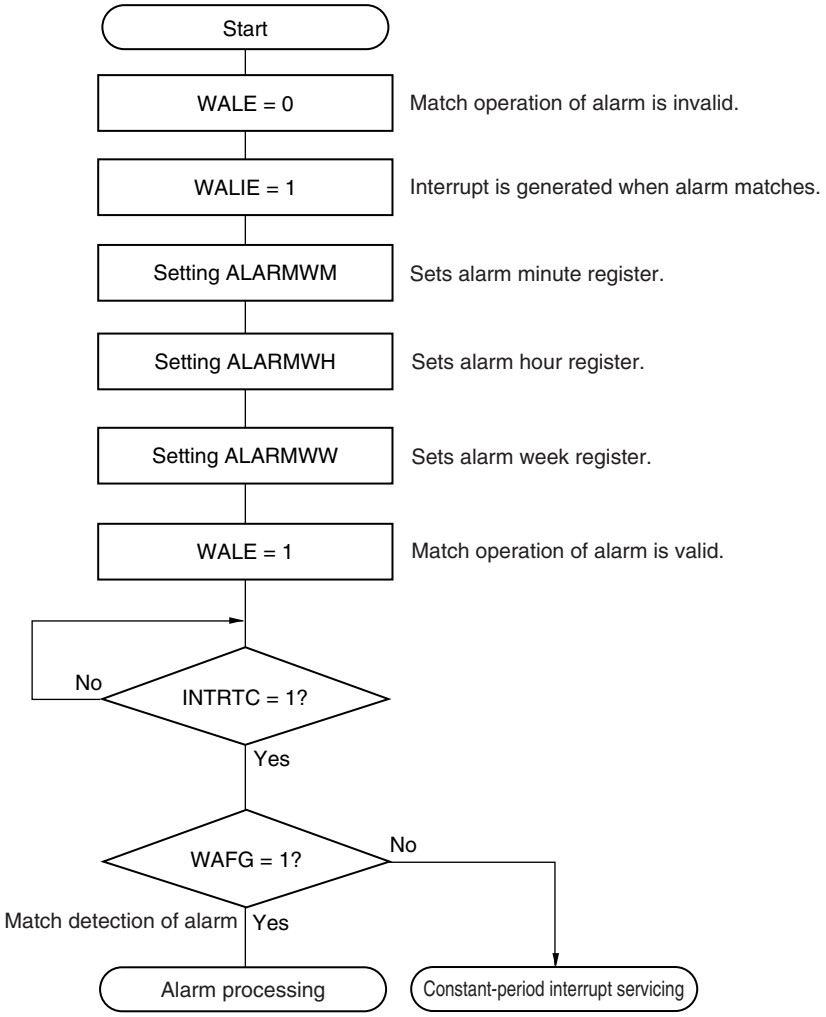
Caution Complete the series of operations of setting RWAIT to 1 to clearing RWAIT to 0 within 1 second.

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.

9.4.3 Setting alarm of real-time counter

Set time of alarm after setting 0 to WALE first.

Figure 9-21. Alarm Setting Procedure



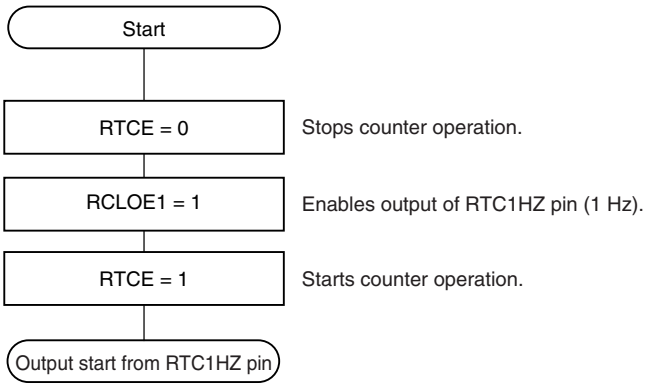
Remarks 1. ALARMWM, ALARMWH, and ALARMWW may be written in any sequence.

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.

9.4.4 1 Hz output of real-time counter

Set output of 1 Hz after setting 0 to RTCE first.

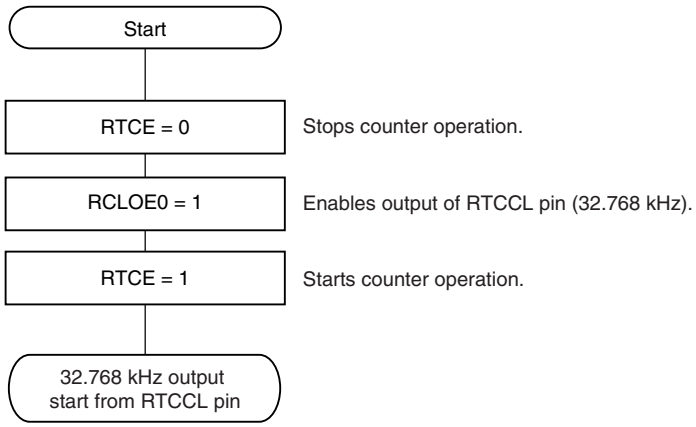
Figure 9-22. 1 Hz Output Setting Procedure



9.4.5 32.768 kHz output of real-time counter

Set output of 32.768 kHz after setting 0 to RTCE first.

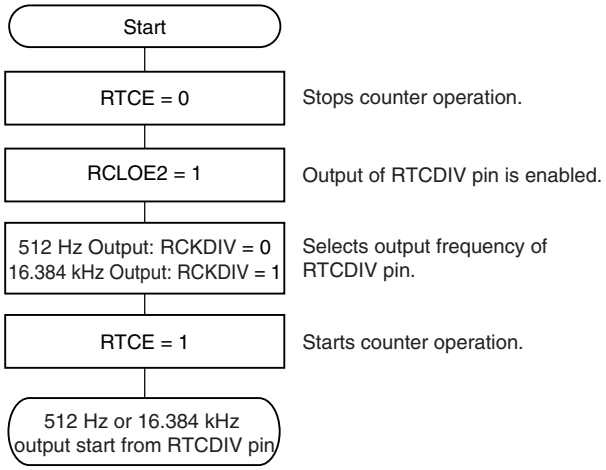
Figure 9-23. 32.768 kHz Output Setting Procedure



9.4.6 512 Hz, 16.384 kHz output of real-time counter

Set outputs of 512 Hz and 16.768 kHz after setting 0 to RTCE first.

Figure 9-24. 512 Hz, 16.384 kHz output Setting Procedure



9.4.7 Example of watch error correction of real-time counter

The watch can be corrected with high accuracy when it is slow or fast, by setting a value to the watch error correction register.

Example of calculating the correction value

The correction value used when correcting the count value of the sub-count register (RSUBC) is calculated by using the following expression.

Set DEV to 0 when the correction range is –63.1 ppm or less, or 63.1 ppm or more.

(When DEV = 0)

Correction value[Note] = Number of correction counts in 1 minute ÷ 3 = (Oscillation frequency ÷ Target frequency – 1) × 32768 × 60 ÷ 3

(When DEV = 1)

Correction value[Note] = Number of correction counts in 1 minute = (Oscillation frequency ÷ Target frequency – 1) × 32768 × 60

Note The correction value is the watch error correction value calculated by using bits 6 to 0 of the watch error correction register (SUBCUD).

(When F6 = 0) Correction value = {(F5, F4, F3, F2, F1, F0) – 1} × 2

(When F6 = 1) Correction value = – {(/F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2

When (F6, F5, F4, F3, F2, F1, F0) is (*, 0, 0, 0, 0, 0, *), watch error correction is not performed. "*" is 0 or 1.

/F5 to /F0 are bit-inverted values (000011 when 111100).

Remarks 1. The correction value is 2, 4, 6, 8, … 120, 122, 124 or –2, –4, –6, –8, … –120, –122, –124.

2. The oscillation frequency is the subsystem clock (f_{SUB}) value.
It can be calculated from the 32 kHz output frequency of the RTCCL pin or the output frequency of the RTC1HZ pin × 32768 when the watch error correction register is set to its initial value (00H).

3. The target frequency is the frequency resulting after correction performed by using the watch error correction register.

Correction example <1>

Example of correcting from 32772.3 Hz to 32768 Hz (32772.3 Hz − 131.2 ppm)

[Measuring the oscillation frequency]

The oscillation frequency[Note] of each product is measured by outputting about 32 kHz from the RTCCL pin or outputting about 1 Hz from the RTC1HZ pin when the watch error correction register is set to its initial value (00H).

Note See **9.4.4 1 Hz output of real-time counter** for the setting procedure of outputting about 1 Hz from the RTC1Hz pin, and **9.4.5 32.768 kHz output of real-time counter** for the setting procedure of outputting about 32 kHz from the RTCCL pin.

[Calculating the correction value]

(When the output frequency from the RTCCL pin is 32772.3 Hz)

If the target frequency is assumed to be 32768 Hz (32772.3 Hz − 131.2 ppm), the correction range for −131.2 ppm is −63.1 ppm or less, so assume DEV to be 0.

The expression for calculating the correction value when DEV is 0 is applied.

$$\begin{aligned}\text{Correction value} &= \text{Number of correction counts in 1 minute} \div 3\\ &= (\text{Oscillation frequency} \div \text{Target frequency} - 1) \times 32768 \times 60 \div 3\\ &= (32772.3 \div 32768 - 1) \times 32768 \times 60 \div 3\\ &= 86\end{aligned}$$

[Calculating the values to be set to (F6 to F0)]

(When the correction value is 86)

If the correction value is 0 or more (when delaying), assume F6 to be 0.

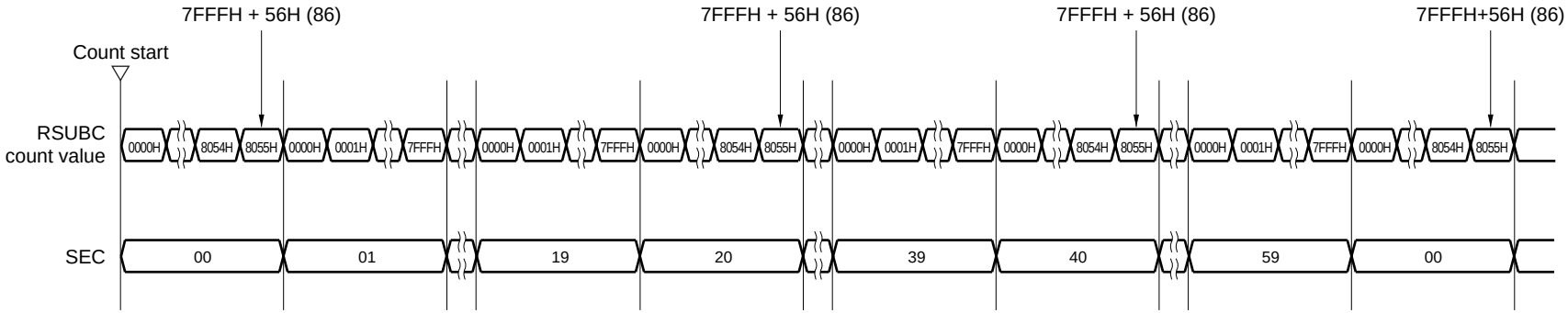
Calculate (F5, F4, F3, F2, F1, F0) from the correction value.

$$\begin{aligned}\{(F5, F4, F3, F2, F1, F0) - 1\} \times 2 &= 86\\ (F5, F4, F3, F2, F1, F0) &= 44\\ (F5, F4, F3, F2, F1, F0) &= (1, 0, 1, 1, 0, 0)\end{aligned}$$

Consequently, when correcting from 32772.3 Hz to 32768 Hz (32772.3 Hz − 131.2 ppm), setting the correction register such that DEV is 0 and the correction value is 86 (bits 6 to 0 of SUBCUD: 0101100) results in 32768 Hz (0 ppm).

Figure 9-25 shows the operation when (DEV, F6, F5, F4, F3, F2, F1, F0) is (0, 0, 1, 0, 1, 1, 0, 0).

Figure 9-25. Operation when (DEV, F6, F5, F4, F3, F2, F1, F0) = (0, 0, 1, 0, 1, 1, 0, 0)



Correction example <2>

Example of correcting from 32767.4 Hz to 32768 Hz (32767.4 Hz + 18.3 ppm)

[Measuring the oscillation frequency]

The oscillation frequency[Note] of each product is measured by outputting about 32 kHz from the RTCCL pin or outputting about 1 Hz from the RTC1HZ pin when the watch error correction register is set to its initial value (00H).

Note See **9.4.4 1 Hz output of real-time counter** for the setting procedure of outputting about 1 Hz from the RTC1Hz pin, and **9.4.5 32.768 kHz output of real-time counter** for the setting procedure of outputting about 32 kHz from the RTCCL pin.

[Calculating the correction value]

(When the output frequency from the RTCCL pin is 0.9999817 Hz)

Oscillation frequency = 32768 × 0.9999817 ≈ 32767.4 Hz

Assume the target frequency to be 32768 Hz (32767.4 Hz + 18.3 ppm) and DEV to be 1.

The expression for calculating the correction value when DEV is 1 is applied.

Correction value = Number of correction counts in 1 minute
= (Oscillation frequency ÷ Target frequency − 1) × 32768 × 60
= (32767.4 ÷ 32768 − 1) × 32768 × 60
= −36

[Calculating the values to be set to (F6 to F0)]

(When the correction value is −36)

If the correction value is 0 or less (when speeding up), assume F6 to be 1.

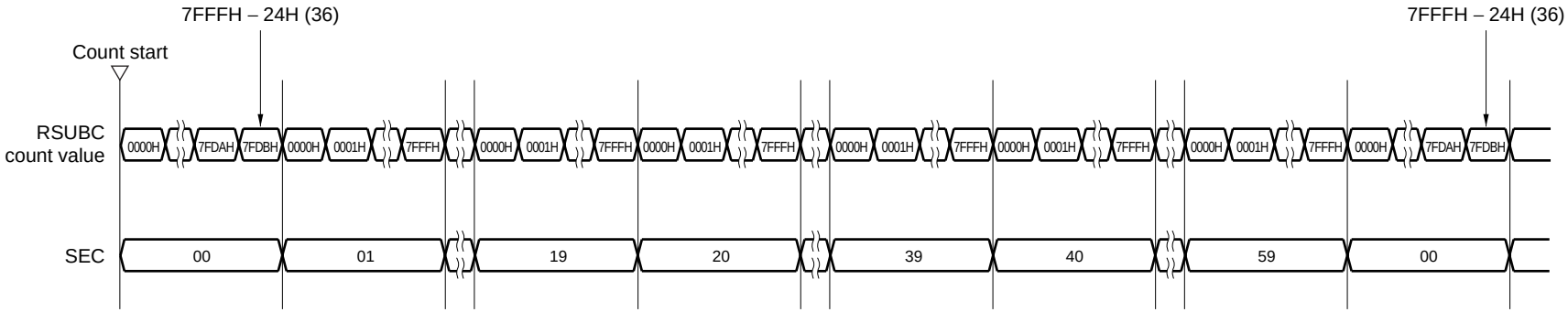
Calculate (F5, F4, F3, F2, F1, F0) from the correction value.

− {(/F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2	= −36
(/F5, /F4, /F3, /F2, /F1, /F0)	= 17
(/F5, /F4, /F3, /F2, /F1, /F0)	= (0, 1, 0, 0, 0, 1)
(F5, F4, F3, F2, F1, F0)	= (1, 0, 1, 1, 1, 0)

Consequently, when correcting from 32767.4 Hz to 32768 Hz (32767.4 Hz + 18.3 ppm), setting the correction register such that DEV is 1 and the correction value is −36 (bits 6 to 0 of SUBCUD: 1101110) results in 32768 Hz (0 ppm).

Figure 9-26 shows the operation when (DEV, F6, F5, F4, F3, F2, F1, F0) is (1, 1, 1, 0, 1, 1, 1, 0).

Figure 9-26. Operation when (DEV, F6, F5, F4, F3, F2, F1, F0) = (1, 1, 1, 0, 1, 1, 1, 0)



CHAPTER 10 WATCHDOG TIMER

10.1 Functions of Watchdog Timer

The watchdog timer is mounted onto all 78K0/Kx2-A microcontrollers.

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
- If the instruction is fetched from an area not set by the IMS register (detection of an invalid check while the CPU hangs up)
- If the CPU accesses an area that is not set by the IMS register (excluding FB00H to FFCFH and FFE0H to FFFFH) by executing a read/write instruction (detection of an abnormal access during a CPU program loop)

When a reset occurs due to the watchdog timer, bit 4 (WDTRF) of the reset control flag register (RESF) is set to 1. For details of RESF, see **CHAPTER 21 RESET FUNCTION**.

10.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

Table 10-1. Configuration of Watchdog Timer

Item	Configuration
Control register	Watchdog timer enable register (WDTE)

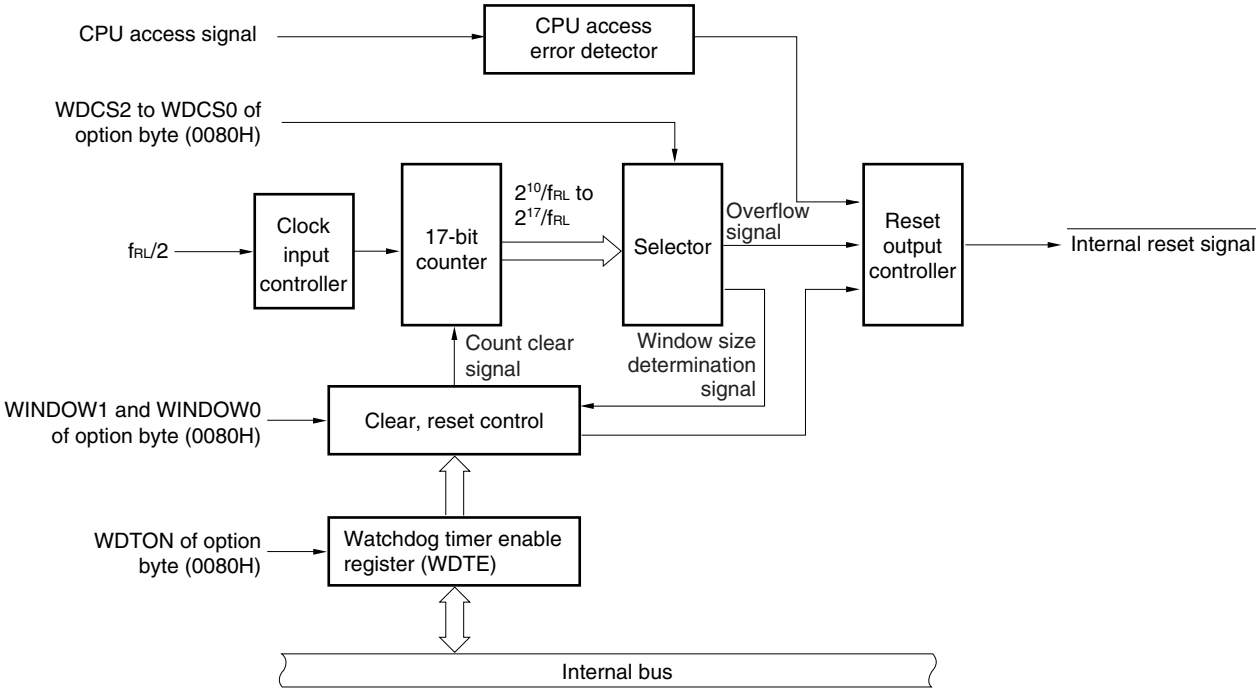
How the counter operation is controlled, overflow time, and window open period are set by the option byte.

Table 10-2. Setting of Option Bytes and Watchdog Timer

Setting of Watchdog Timer	Option Byte (0080H)
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)

Remark For the option byte, see **CHAPTER 24 OPTION BYTE**.

Figure 10-1. Block Diagram of Watchdog Timer



10.3 Register Controlling Watchdog Timer

The watchdog timer is controlled by the watchdog timer enable register (WDTE).

(1) Watchdog timer enable register (WDTE)

Writing ACH to 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 10-2. Format of Watchdog Timer Enable Register (WDTE)

Address: FF99H 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 (0080H). To operate watchdog timer, set WDTON to 1.

WDTON Setting Value	WDTE Reset Value
0 (watchdog timer count operation disabled)	1AH
1 (watchdog timer count operation enabled)	9AH

Cautions 1. If a value other than ACH is written to WDTE, an internal reset signal is generated. If the source clock to the watchdog timer is stopped, however, an internal reset signal is generated when the source clock to the watchdog timer resumes operation.

2. If a 1-bit memory manipulation instruction is executed for WDTE, an internal reset signal is generated. If the source clock to the watchdog timer is stopped, however, an internal reset signal is generated when the source clock to the watchdog timer resumes operation.

3. The value read from WDTE is 9AH/1AH (this differs from the written value (ACH)).

10.4 Operation of Watchdog Timer

10.4.1 Controlling operation of watchdog timer

1. When the watchdog timer is used, its operation is specified by the option byte (0080H).

   - Enable counting operation of the watchdog timer by setting bit 4 (WDTON) of the option byte (0080H) to 1 (the counter starts operating after a reset release) (for details, see **CHAPTER 24**).

WDTON	Operation Control of Watchdog Timer Counter/Illegal Access Detection
0	Counter operation disabled (counting stopped after reset), illegal access detection operation disabled
1	Counter operation enabled (counting started after reset), illegal access detection operation enabled

   - Set an overflow time by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (0080H) (for details, see **10.4.2** and **CHAPTER 24**).
   - Set a window open period by using bits 6 and 5 (WINDOW1 and WINDOW0) of the option byte (0080H) (for details, see **10.4.3** and **CHAPTER 24**).

2. After a reset release, the watchdog timer starts counting.
3. By writing “ACH” to WDTE after the watchdog timer starts counting and before the overflow time set by the option byte, the watchdog timer is cleared and starts counting again.
4. After that, write WDTE the second time or later after a reset release during the window open period. If WDTE is written during a window close period, an internal reset signal is generated.
5. If the overflow time expires without “ACH” written to WDTE, an internal reset signal is generated.
   A internal reset signal is generated in the following cases.

   - If a 1-bit manipulation instruction is executed on the watchdog timer enable register (WDTE)
   - If data other than “ACH” is written to WDTE
   - If the instruction is fetched from an area not set by the IMS register (detection of an invalid check during a CPU program loop)
   - If the CPU accesses an area not set by the IMS register (excluding FB00H to FFCFH and FFE0H to FFFFH) by executing a read/write instruction (detection of an abnormal access during a CPU program loop)

Cautions 1. The first writing to WDTE after a reset release clears the watchdog timer, if it is made before the overflow time regardless of the timing of the writing, 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 $2/f_{RL}$ seconds.

3. The watchdog timer can be cleared immediately before the count value overflows (FFFFH).

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 (LSROSC) of the option byte.

	LSROSC = 0 (Internal Low-Speed Oscillator Can Be Stopped by Software)	LSROSC = 1 (Internal Low-Speed Oscillator Cannot Be Stopped)
In HALT mode	Watchdog timer operation stops.	Watchdog timer operation continues.
In STOP mode		

If LSROSC = 0, the watchdog timer resumes counting after the HALT or STOP mode is released. At this time, the counter is not cleared to 0 but starts counting from the value at which it was stopped.

If oscillation of the internal low-speed oscillator is stopped by setting LSRSTOP (bit 1 of the internal oscillation mode register (RCM) = 1) when LSROSC = 0, the watchdog timer stops operating. At this time, the counter is not cleared to 0.

5. The watchdog timer continues its operation during self-programming and EEPROM emulation of the flash memory. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.

10.4.2 Setting overflow time of watchdog timer

Set the overflow time of the watchdog timer by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (0080H).

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 10-3. Setting of Overflow Time of Watchdog Timer

WDCS2	WDCS1	WDCS0	Overflow Time of Watchdog Timer
0	0	0	$2^{10}/f_{RL}$ (3.88 ms)
0	0	1	$2^{11}/f_{RL}$ (7.76 ms)
0	1	0	$2^{12}/f_{RL}$ (15.52 ms)
0	1	1	$2^{13}/f_{RL}$ (31.03 ms)
1	0	0	$2^{14}/f_{RL}$ (62.06 ms)
1	0	1	$2^{15}/f_{RL}$ (124.12 ms)
1	1	0	$2^{16}/f_{RL}$ (248.24 ms)
1	1	1	$2^{17}/f_{RL}$ (496.48 ms)

Cautions 1. The combination of WDCS2 = WDCS1 = WDCS0 = 0 and WINDOW1 = WINDOW0 = 0 is prohibited.

2. The watchdog timer continues its operation during self-programming and EEPROM emulation of the flash memory. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.

Remarks 1. f_{RL}: Internal low-speed oscillation clock frequency

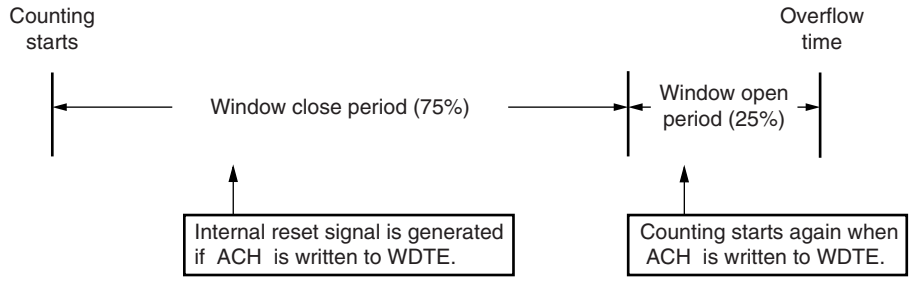
2. ( ): f_{RL} = 264 kHz (MAX.)

10.4.3 Setting window open period of watchdog timer

Set the window open period of the watchdog timer by using bits 6 and 5 (WINDOW1, WINDOW0) of the option byte (0080H). 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 The first writing to WDTE after a reset release clears the watchdog timer, if it is made before the overflow time regardless of the timing of the writing, and the watchdog timer starts counting again.

The window open period to be set is as follows.

Table 10-4. Setting Window Open Period of Watchdog Timer

WINDOW1	WINDOW0	Window Open Period of Watchdog Timer
0	0	25%
0	1	50%
1	0	75%
1	1	100%

Cautions 1. The combination of WDCS2 = WDCS1 = WDCS0 = 0 and WINDOW1 = WINDOW0 = 0 is prohibited.

2. Setting WINDOW1 = WINDOW0 = 0 is prohibited when using the watchdog timer at 1.8 V ≤ V_{DD} < 2.7 V.

3. The watchdog timer continues its operation during self-programming and EEPROM emulation of the flash memory. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.

Remark If the overflow time is set to $2^{11}/f_{RL}$, the window close time and open time are as follows.

(when 2.7 V ≤ V_{DD} ≤ 5.5 V)

	Setting of Window Open Period			
	25%	50%	75%	100%
Window close time	0 to 7.11 ms	0 to 4.74 ms	0 to 2.37 ms	None
Window open time	7.11 to 7.76 ms	4.74 to 7.76 ms	2.37 to 7.76 ms	0 to 7.76 ms

<When window open period is 25%>

- Overflow time:
  $2^{11}/f_{RL}$ (MAX.) = 2^{11}/264 kHz (MAX.) = 7.76 ms
- Window close time:
  0 to $2^{11}/f_{RL}$ (MIN.) × (1 − 0.25) = 0 to 2^{11}/216 kHz (MIN.) × 0.75 = 0 to 7.11 ms
- Window open time:
  $2^{11}/f_{RL}$ (MIN.) × (1 − 0.25) to $2^{11}/f_{RL}$ (MAX.) = 2^{11}/216 kHz (MIN.) × 0.75 to 2^{11}/264 kHz (MAX.)
  = 7.11 to 7.76 ms

CHAPTER 11 CLOCK OUTPUT CONTROLLER

Item	78K0/KB2-A	78K0/KC2-A
	30 pins	48 pins
Clock output	–	√

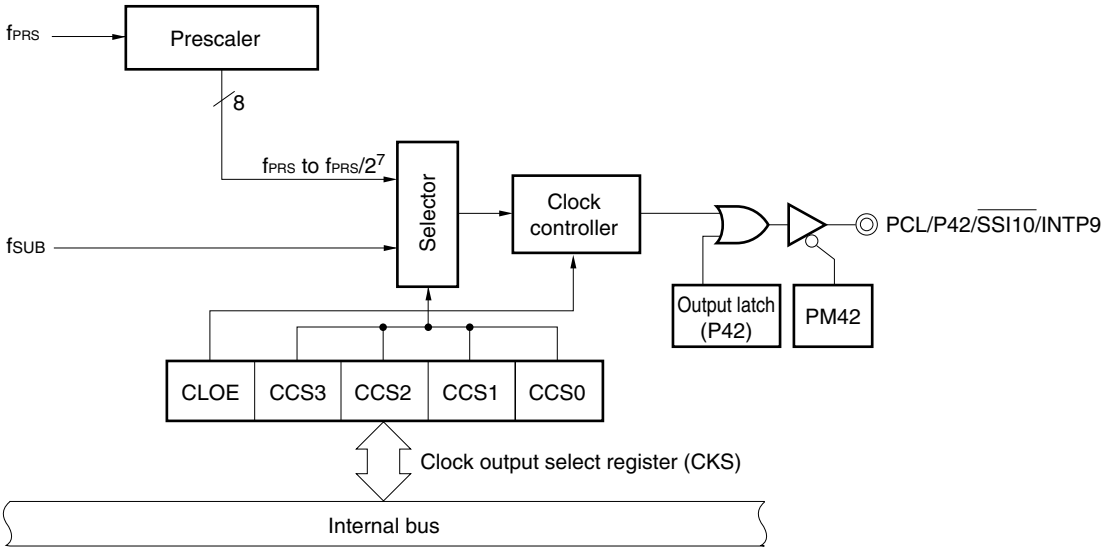
Remark √: Mounted, –: Not mounted

11.1 Functions of Clock Output Controller

The clock output controller is intended for carrier output during remote controlled transmission and clock output for supply to peripheral ICs. The clock selected with the clock output selection register (CKS) is output.

Figure 11-1 shows the block diagram of clock output controller.

Figure 11-1. Block Diagram of Clock Output Controller



11.2 Configuration of Clock Output Controller

The clock output controller includes the following hardware.

Table 11-1. Configuration of Clock Output Controller

Item	Configuration
Control registers	Clock output selection register (CKS) Port mode register 4 (PM4) Port register 4 (P4)

11.3 Registers Controlling Clock Output Controller

The following two registers are used to control the clock output controller.

- Clock output selection register (CKS)
- Port mode register 4 (PM4)

(1) Clock output selection register (CKS)

This register sets output enable/disable for clock output (PCL), and sets the output clock.

CKS is set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears CKS to 00H.

Figure 11-2. Format of Clock Output Selection Register (CKS)

Address: FF40H After reset: 00H R/W

Symbol	7	6	5	<4>	3	2	1	0
CKS	0	0	0	CLOE	CCS3	CCS2	CCS1	CCS0

CLOE	PCL output enable/disable specification
0	Clock division circuit operation stopped. PCL fixed to low level.
1	Clock division circuit operation enabled. PCL output enabled.

CCS3	CCS2	CCS1	CCS0	PCL output clock selection[Note 1]			
					f_{SUB} = 32.768 kHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	0	0	f_{PRS}[Note 2]	–	10 MHz	Setting prohibited[Note 3]
0	0	0	1	$f_{PRS}/2$		5 MHz	10 MHz
0	0	1	0	$f_{PRS}/2^2$		2.5 MHz	5 MHz
0	0	1	1	$f_{PRS}/2^3$		1.25 MHz	2.5 MHz
0	1	0	0	$f_{PRS}/2^4$		625 kHz	1.25 MHz
0	1	0	1	$f_{PRS}/2^5$		312.5 kHz	625 kHz
0	1	1	0	$f_{PRS}/2^6$		156.25 kHz	312.5 kHz
0	1	1	1	$f_{PRS}/2^7$		78.125 kHz	156.25 kHz
1	0	0	0	f_{SUB}	32.768 kHz	–	
Other than above				Setting prohibited			

Notes 1. The frequency that can be used for the peripheral hardware clock (f_{PRS}) differs depending on the power supply voltage.

Supply Voltage	Use frequency range of peripheral hardware clock (f_{PRS})
$2.7\ V \le V_{DD} \le 5.5\ V$	$f_{PRS} \le 20$ MHz
$1.8\ V \le V_{DD} < 2.7\ V$	$f_{PRS} \le 5$ MHz

(The values shown in the table above are those when $f_{PRS} = f_{XH}$ (XSEL = 1).)

2. If the peripheral hardware clock (f_{PRS}) operates on the internal high-speed oscillation clock (XSEL = 0)when $1.8\ V \le V_{DD} < 2.7\ V$, setting CCS3 = CCS2 = CCS1 = CCS0 = 0 (output clock of PCL: f_{PRS}) is prohibited.

3. The PCL output clock prohibits settings if they exceed 10 MHz.

Caution Set CCS3 to CCS0 while the clock output operation is stopped (CLOE = 0).

Remarks 1. f_{PRS}: Peripheral hardware clock frequency

2. f_{SUB}: Subsystem clock frequency

(2) Port mode register 4 (PM4)

This register sets port 4 input/output in 1-bit units.

When using the P42/PCL/SSI10/INTP9 pin for clock output, clear PM42 and the output latches of P42 to 0.

PM4 is set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets PM14 to FFH.

Figure 11-3. Format of Port Mode Register 4 (PM4)

Address: FF24H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM4	1	1	1	1	1	PM42	PM41	PM40

PM4n	P4n pin I/O mode selection (n = 0 to 2)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

11.4 Operations of Clock Output Controller

11.4.1 Operation as clock output

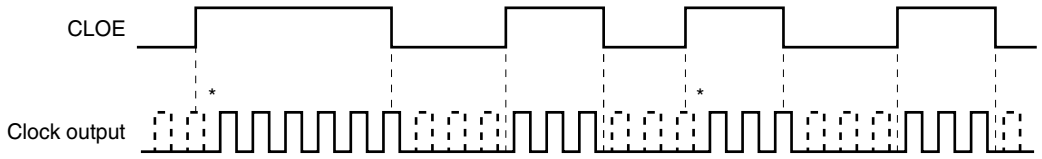
The clock pulse is output as the following procedure.

<1> Select the clock pulse output frequency with bits 0 to 3 (CCS0 to CCS3) of the clock output selection register (CKS) (clock pulse output in disabled status).

<2> Set bit 4 (CLOE) of CKS to 1 to enable clock output.

Remark The clock output controller is designed not to output pulses with a small width during output enable/disable switching of the clock output. As shown in Figure 11-4, 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 11-4. Remote Control Output Application Example



CHAPTER 12 A/D CONVERTER

Items	78K0/KB2-A[Note]	78K0/KC2-A
	30 pins	48 pins
A/D converter	10 ch (ANI0 to ANI5, ANI8 to ANI11)	12 ch (ANI0 to ANI6, ANI8 to ANI11, ANI15)

Note As the AV_{REFM} and AV_{REFP} are not mounted on78K0/KB2-A, replace AV_{REFP} with AV_{REF}, respectively.

12.1 Function of A/D Converter

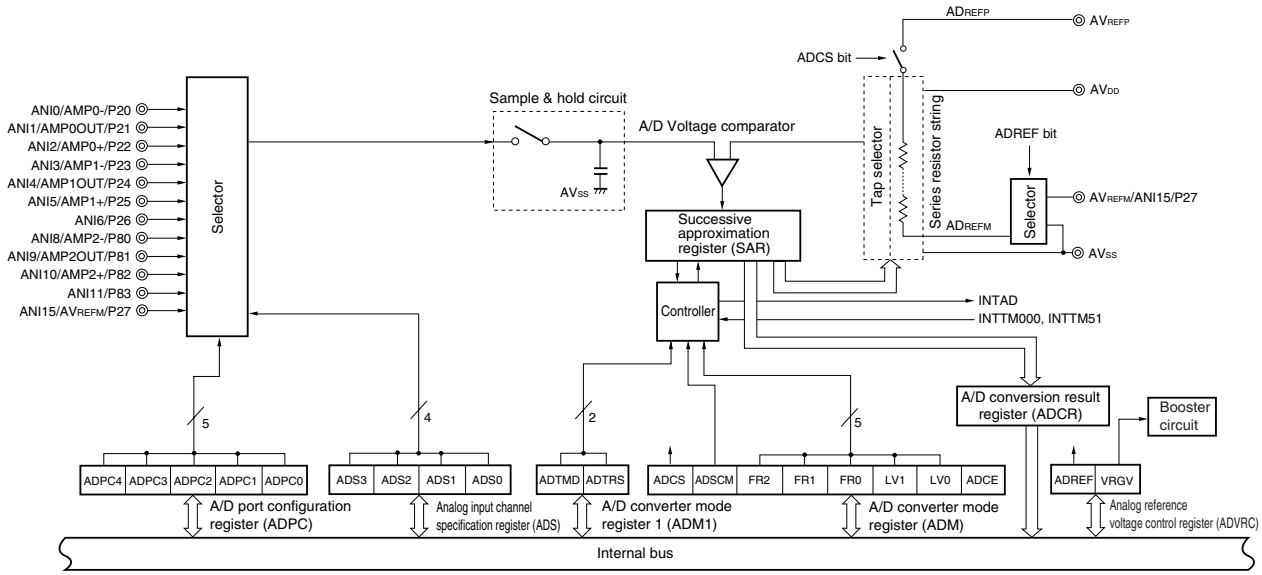
The A/D converter is a 12-bit resolution converter that converts analog input signals into digital values, and is configured to control a total of twelve channels of analog inputs, including up to twelve channels of A/D converter analog inputs (ANI0 to ANI6, ANI8 to ANI11, ANI13).

In products with operational amplifier, ANI1, ANI4, and ANI9 functions alternately as operational amplifier 0, 1, and 2 output (AMP0OUT, AMP1OUT, AMP2OUT). This enables using operational amplifier output as an analog input source.

The operation mode of A/D converter has the following four functions.

- Software trigger mode (continuously convert mode)
- Software trigger mode (single convert mode)
- Timer trigger mode (continuously convert mode)
- Timer trigger mode (single convert mode)

Figure 12-1. Block Diagram of A/D Converter



Remark 78K0/KB2-A: ANI0 to ANI5, ANI8 to ANI11
78K0/KC2-A: ANI0 to ANI6, ANI8 to ANI11, ANI15

12.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

(1) ANI0 to ANI6, ANI8 to ANI11, ANI15 pins

These are the analog input pins of the A/D converter. They input analog signals to be converted into digital signals. Pins other than the one selected as the analog input pin can be used as I/O port pins.

Remark 78K0/KB2-A: ANI0 to ANI5, ANI8 to ANI11
78K0/KC2-A: ANI0 to ANI6, ANI8 to ANI11, ANI15

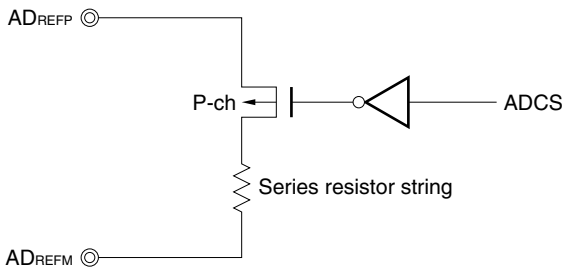
(2) Sample & hold circuit

The sample & hold circuit samples each of the analog input voltages sequentially sent from the input circuit, and sends them to the A/D voltage comparator. This circuit also holds the sampled analog input voltage during A/D conversion.

(3) Series resistor string

The series resistor string is connected between AD_{REFP} and AD_{REFM}[Note], and generates a voltage to be compared with the sampled voltage value.

Figure 12-2. Circuit Configuration of Series Resistor String



Note In the case of 78K0/KB2-A, the series resistor string is connected between AV_{REF} and AV_{SS}.

(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) 12-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 lower 12 bits (the higher 4 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_{DD} pin

This pin inputs a power to the A/D converter. When one or more of the pins of ports 2 and 8 are used as the digital port pins, make AV_{DD} the same potential as V_{DD}.

(10) AV_{SS} pin

This is the ground potential pin of the A/D converter. Always use this pin at the same potential as that of the V_{SS} pin even when the A/D converter is not used.

In the 48-pin versions of 78K0/KC3-A, the ground potential (AV_{SS}) can also be used as the negative reference voltage for the A/D converter (AD_{REFM}). Clear AVREF bit of ADVRC register to 0 when using AV_{SS} as AD_{REFM}.

(11) AD_{REFP} pin Note 1

This pin is used to input the external reference voltage (AV_{REFP}).

The analog signal input to ANI0 to ANI6, ANI8 to ANI11, and ANI15 are converted into a digital signal, based on the voltage Note 2 applied across AV_{REFP} and AV_{REFM}.

(12) AD_{REFM} pin Note 2

This pin is used to input the external reference voltage (AV_{REFM}).

To use AV_{REFM} as the negative reference voltage for the A/D converter (AD_{REFM}), set the ADREF bit of the ADVRC register to 1.

Notes 1. As the AV_{REFM} and AV_{REFP} are not mounted on 78K0/KB2-A, replace AV_{REFP} with AV_{REF}, respectively.

2. In the 78K0/KB2-A, this is the voltage that is applied between AV_{REF} and AV_{SS}.

Remark 78K0/KB2-A: ANI0 to ANI5, ANI8 to ANI11
78K0/KC2-A: ANI0 to ANI6, ANI8 to ANI11, ANI15

12.3 Registers Used in A/D Converter

The A/D converter uses the following eight registers.

- A/D converter mode register (ADM)
- A/D converter mode register 1 (ADM1)
- Analog conference voltage control register (ADVRC)
- 12-bit A/D conversion result register (ADCR)
- 8-bit A/D conversion result register (ADCRH)
- Analog input channel specification register (ADS)
- A/D port configuration register (ADPC)
- Port mode registers 2, 8 (PM2, PM8)

(1) 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 12-3. Format of A/D Converter Mode Register (ADM)

Address: FF38H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
ADM	ADCS	ADSCM	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

ADSCM	Select operation mode of A/D convert
0	Continuously convert mode
1	Single convert mode

ADCE	A/D voltage Comparator operation control[Note 2]
0	Stops A/D voltage comparator operation
1	Enables A/D voltage comparator operation

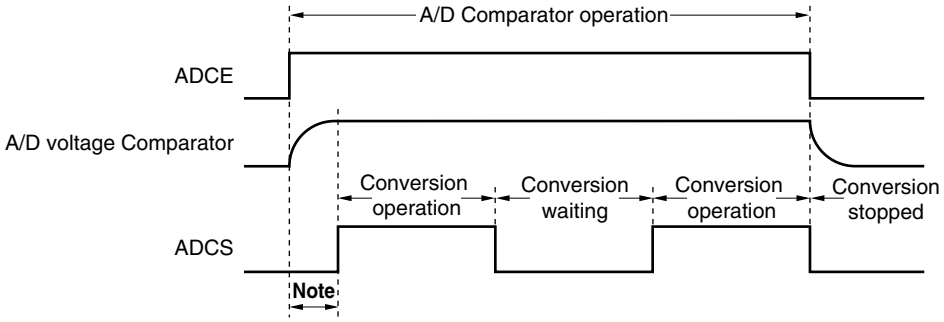
Notes 1. For details of FR2 to FR0, LV1, LV0, and A/D conversion, see **Table 12-2 A/D Conversion Time Selection** .

2. The operation of the A/D voltage comparator is controlled by ADCS and ADCE, and it takes 1 μs from operation start to operation stabilization. Therefore, 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 12-1. 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 (A/D voltage comparator operation, only comparator consumes power)
1	0	Setting prohibited
1	1	Conversion mode (A/D voltage comparator operation)

Figure 12-4. Timing Chart When A/D Voltage Comparator Is Used



Note To stabilize the internal circuit, the time from the rising of the ADCE bit to the falling of the ADCS bit must be 1 μs or longer.

Cautions 1. A/D conversion must be stopped before rewriting bits FR0 to FR2, LV1, and LV0 to values other than the identical data.

2. When using the A/D converter in normal mode 2 (LV1 = 0, LV0 = 1) or low-voltage mode (LV1 = 1, LV0 = 0), enable A/D conversion (by setting ADCE to 1) after starting up the A/D converter voltage booster (by setting VRGV to 1) and waiting for the reference voltage stabilization time (10 μs) to elapse. Note that it takes the A/D voltage comparator 1 μs to stabilize after being enabled. The validity of the data converted first can therefore be improved by enabling A/D conversion (by setting ADCS to 1) at least 1 μs after enabling the A/D voltage comparator. If A/D conversion is enabled without waiting for at least 1 μs, ignore the data converted first.

Table 12-2. A/D Conversion Time Selection

(1) Normal mode 1 : 2.7 V ≤ AV_{DD} ≤ 5.5 V (Stops operation of A/D converter booster circuit)[Note 1]

A/D Converter Mode Register (ADM)					Conversion Time Selection					Conversion Clock (f_{AD})
FR2	FR1	FR0	LV1	LV0		f_{CPU} = 1 MHz	f_{CPU} = 8 MHz	f_{CPU} = 10 MHz	f_{CPU} = 20 MHz	
0	0	0	0	0	240/f_{CPU}	Setting prohibited	30 μs	24 μs	12 μs	f_{CPU}/12
0	0	1			160/f_{CPU}		20 μs	16 μs	8 μs	f_{CPU}/8
0	1	0			120/f_{CPU}		15 μs	12 μs	6 μs	f_{CPU}/6
0	1	1			100/f_{CPU}		12.5 μs	10 μs	5 μs	f_{CPU}/5
1	0	0			80/f_{CPU}		10 μs	8 μs	Setting prohibited	f_{CPU}/4
1	0	1			60/f_{CPU}		7.5 μs	6 μs		f_{CPU}/3
1	1	0			40/f_{CPU}	40 μs	5 μs	Setting prohibited		f_{CPU}/2
1	1	1			20/f_{CPU}	20 μs	Setting prohibited			f_{CPU}

(2) Normal mode 2 : 2.3 V ≤ AV_{DD} ≤ 5.5 V (Operation of A/D converter booster circuit)[Notes 1, 2]

A/D Converter Mode Register (ADM)					Conversion Time Selection					Conversion Clock (f_{AD})
FR2	FR1	FR0	LV1	LV0		f_{CPU} = 1 MHz	f_{CPU} = 8 MHz	f_{CPU} = 10 MHz	f_{CPU} = 20 MHz	
0	0	0	0	1	240/f_{CPU}	Setting prohibited	30 μs	24 μs	12 μs	f_{CPU}/12
0	0	1			160/f_{CPU}		20 μs	16 μs	8 μs	f_{CPU}/8
0	1	0			120/f_{CPU}		15 μs	12 μs	6 μs	f_{CPU}/6
0	1	1			100/f_{CPU}		12.5 μs	10 μs	5 μs	f_{CPU}/5
1	0	0			80/f_{CPU}		10 μs	8 μs	Setting prohibited	f_{CPU}/4
1	0	1			60/f_{CPU}		7.5 μs	6 μs		f_{CPU}/3
1	1	0			40/f_{CPU}	40 μs	5 μs	Setting prohibited		f_{CPU}/2
1	1	1			20/f_{CPU}	20 μs	Setting prohibited			f_{CPU}

(3) Low-voltage mode : 1.8 V ≤ AV_{DD} ≤ 5.5 V (Operation of A/D converter booster circuit)[Notes 1, 2]

A/D Converter Mode Register (ADM)					Conversion Time Selection					Conversion Clock (f_{AD})
FR2	FR1	FR0	LV1	LV0		f_{CPU} = 1 MHz	f_{CPU} = 8 MHz	f_{CPU} = 10 MHz	f_{CPU} = 20 MHz	
0	0	0	1	0	276/f_{CPU}	Setting prohibited	34.5 μs	27.6 μs	13.8 μs	f_{CPU}/12
0	0	1			184/f_{CPU}		23 μs	18.4 μs	9.2 μs	f_{CPU}/8
0	1	0			138/f_{CPU}		17.3 μs	13.8 μs	6.9 μs	f_{CPU}/6
0	1	1			115/f_{CPU}		14.4 μs	11.5 μs	Setting prohibited	f_{CPU}/5
1	0	0			92/f_{CPU}		11.5 μs	9.2 μs		f_{CPU}/4
1	0	1			69/f_{CPU}		8.96 μs	6.9 μs		f_{CPU}/3
1	1	0			46/f_{CPU}	46 μs	Setting prohibited	Setting prohibited		f_{CPU}/2
1	1	1			23/f_{CPU}	23 μs				f_{CPU}

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

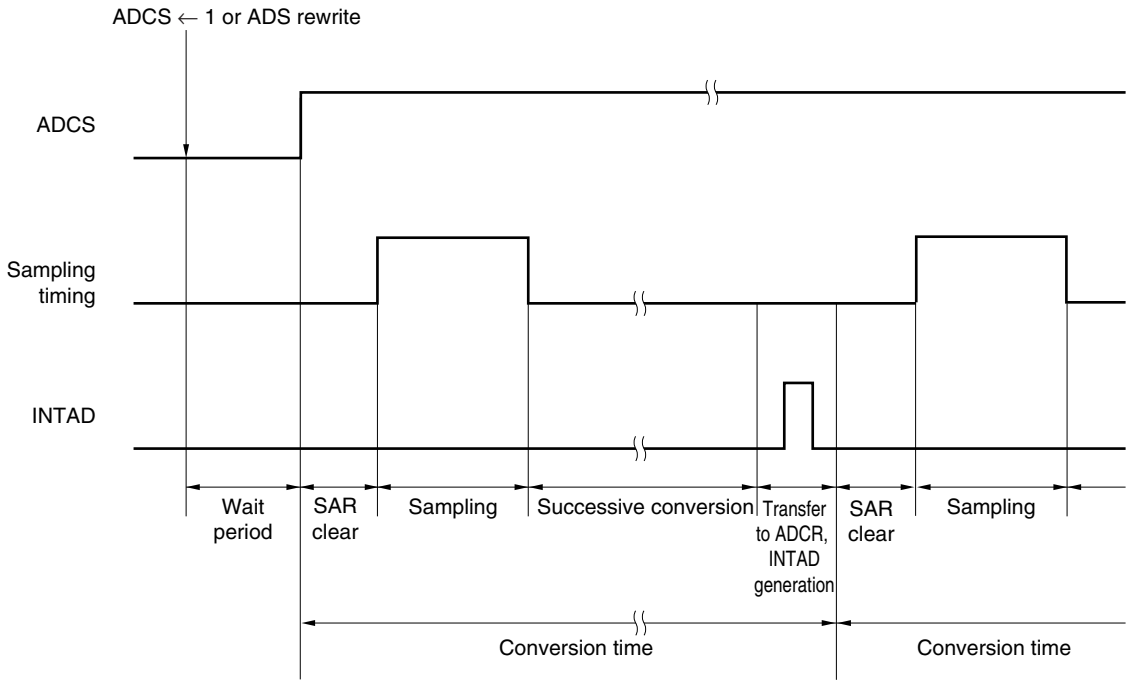
Notes 1. The selectable conversion time differs depending on the used voltage, mode, etc. For details, see **CHAPTER 28 ELECTRICAL SPECIFICATIONS**.

2. When using the A/D converter in Normal mode 2 or low-voltage mode, be sure to enable the A/D converter voltage booster (by setting VRGV to 1).

Caution The CPU clock (f_{CPU}) is supplied to the A/D converter's clock. If f_{CPU} is changed, therefore, the A/D converter's conversion clock (f_{AD}) is also changed. When changing f_{CPU} (by changing the setting of the PCC register or the MCM register), therefore, be sure to stop the A/D converter first (by setting ADCS to 0).

Remark f_{CPU}: CPU clock frequency

Figure 12-5. A/D Converter Sampling and A/D Conversion Timing



(2) A/D converter mode register 1 (ADM1)

This register sets the A/D conversion start trigger.

ADM1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 12-6. Format of A/D Converter Mode Register 1 (ADM1)

Address: FF3AH After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
ADM1	ADTMD	0	0	0	0	0	0	ADTRS

ADTMD	A/D trigger mode selection
0	Software trigger mode
1	Timer trigger mode (hardware trigger mode)

ADTRS	Timer trigger signal selection for A/D conversion
0	INTTM000
1	INTTM51

Caution Rewriting ADM1 during A/D conversion is prohibited. Rewrite it when conversion operation is stopped (ADCS = 0).

(3) Analog conference voltage control register (ADVRC)

This register is used to select the source of the negative reference voltage for the A/D converter.

The electrical characteristics of the A/D converter can be maintained even in low-voltage mode by enabling operation of the A/D converter's input gate voltage booster.ADVRC can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 12-7. Format of Analog Conference Voltage Control Register (ADVRC)

Address: FF2EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADVRC	ADREF Note 1	0	0	0	0	0	VRGV Note 2	0

ADREF Note 1	Negative reference voltage for the A/D converter selection
0	AV_{SS}
1	AV_{REFM} (external conference voltage input)

VRGV Note 2	A/D converter booster circuit operation control
0	Stop operation
1	Enables operation

Notes 1. As the AV_{REFM} is not mounted on 78K0/KB2-A, Clear the ADREF to 0.

2. When using the A/D converter in normal mode 2 (LV1 = 0, LV0 = 1) or low-voltage mode (LV1 = 1, LV0 = 0), enable A/D conversion (by setting ADCE to 1) after starting up the A/D converter voltage booster (by setting VRGV to 1) and waiting for the reference voltage stabilization time (10 μs) to elapse. Note that it takes the A/D voltage comparator 1 μs to stabilize after being enabled. The validity of the data converted first can therefore be improved by enabling A/D conversion (by setting ADCS to 1) at least 1 μs after enabling the A/D voltage comparator. If A/D conversion is enabled without waiting for at least 1 μs, ignore the data converted first.

(4) 12-bit A/D conversion result register (ADCR)

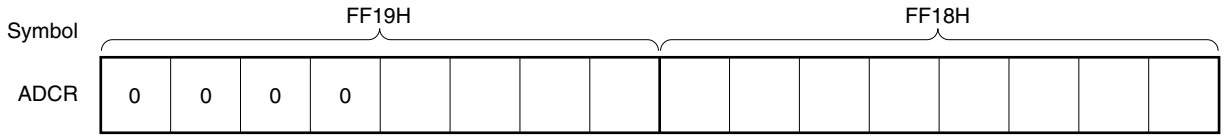
This register is a 16-bit register that stores the A/D conversion result. The higher 4 bits are fixed to 0. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register. The higher 4 bits of the conversion result are stored in FF19H and the lower 8 bits are stored in FF18H.

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

Reset signal generation clears this register to 0000H.

Figure 12-8. Format of 12-Bit A/D Conversion Result Register (ADCR)

Address: FF18H, FF19H After reset: 0000H R



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.

(5) 8-bit A/D conversion result register (ADCRH)

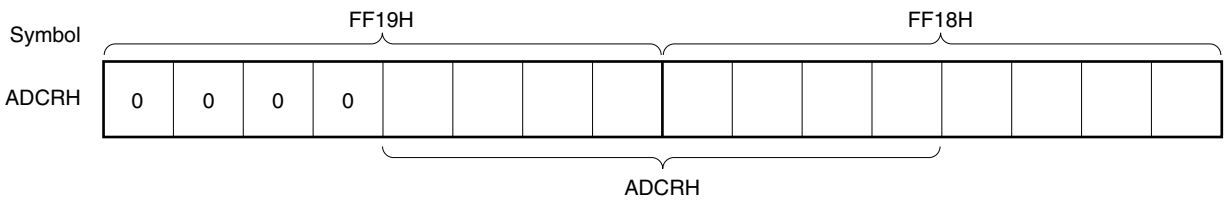
This register is an 8-bit register that stores the A/D conversion result. The higher 8 bits of 12-bit resolution are stored.

ADCRH can be read by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 12-9. Format of 8-Bit A/D Conversion Result Register (ADCRH)

Address: FF19H[Note] After reset: 00H R



Note When address FF19H is read, the data of ADCRH (the lower 4 bits of FF19H and the higher 4 bits of FF18H) is read.

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.

(6) 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 12-10. Format of Analog Input Channel Specification Register (ADS)

Address: FF39H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	0	0	0	0	ADS3	ADS2	ADS1	ADS0

	ADS3	ADS2	ADS1	ADS0	Analog input channel
	0	0	0	0	ANI0
	0	0	0	1	ANI1
	0	0	1	0	ANI2
	0	0	1	1	ANI3
	0	1	0	0	ANI4
	0	1	0	1	ANI5
Note→	0	1	1	0	ANI6
	0	1	1	1	Setting prohibited
	1	0	0	0	ANI8
	1	0	0	1	ANI9
	1	0	1	0	ANI10
	1	0	1	1	ANI11
Note→	1	1	1	1	ANI15
	Other than the above				Setting prohibited

Note In case of 78K0/KB2-A, setting prohibited

Cautions 1. Be sure to clear bits 4 to 7 to "0".

2 Set a channel to be used for A/D conversion in the input mode by using port mode registers 2 and 8 (PM2, PM8).

3. Do not set a pin to be used as a digital I/O pin with ADPC with ADS.

4. When using operational amplifier n, the output signal of operational amplifier n can be used as an analog input.

Remark n = 0 to 2

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

This register switches the ANI0/AMP0-/P20 to ANI6/P26, ANI8/AMP2-/P80 to ANI11/P83, and ANI15/AV_{REFM}/P27 pins to analog input or digital I/O of port.

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

Reset signal generation clears this register to 10H.

Remark 78K0/KB2-A: ANI0 to ANI5, ANI8 to ANI11
78K0/KC2-A: ANI0 to ANI6, ANI8 to ANI11, ANI15

Figure 12-11. Format of A/D Port Configuration Register (ADPC)

Address: FF2FH 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											
						ANI15/AV_{REFM}/P27	ANI11/P83	ANI10/AMP2+/P82	ANI9/$AMP2_{OUT}$/P81	ANI8/AMP2-/P80	ANI6/P26	ANI5/AMP1+/P25	ANI4/$AMP1_{OUT}$/P24	ANI3/AMP1-/P23	ANI2/AMP0+/P22	ANI1/$AMP0_{OUT}$/P21	ANI0/AMP0-/P20
	0	0	0	0	0	A	A	A	A	A	A	A	A	A	A	A	A
	0	0	0	0	1	A	A	A	A	A	A	A	A	A	A	A	D
	0	0	0	1	0	A	A	A	A	A	A	A	A	A	A	D	D
	0	0	0	1	1	A	A	A	A	A	A	A	A	A	D	D	D
	0	0	1	0	0	A	A	A	A	A	A	A	A	D	D	D	D
	0	0	1	0	1	A	A	A	A	A	A	A	D	D	D	D	D
Note →	0	0	1	1	0	A	A	A	A	A	A	D	D	D	D	D	D
	0	0	1	1	1	Setting prohibited											
	0	1	0	0	0	A	A	A	A	A	D	D	D	D	D	D	D
	0	1	0	0	1	A	A	A	A	D	D	D	D	D	D	D	D
	0	1	0	1	0	A	A	A	D	D	D	D	D	D	D	D	D
	0	1	0	1	1	A	A	D	D	D	D	D	D	D	D	D	D
Note →	0	1	1	1	1	A	D	D	D	D	D	D	D	D	D	D	D
	1	0	0	0	0	D	D	D	D	D	D	D	D	D	D	D	D
	Other than the above					Setting prohibited											

Note In the case of 78K0/KB2-A, setting is prohibited.

Cautions 1 Set a channel to be used for A/D conversion in the input mode by using port mode registers 2 and 8 (PM2, PM8).

2. Do not set a pin to be used as a digital I/O pin with ADPC with ADS.

(8) Port mode registers 2 and 8 (PM2, PM8)

When using the ANI0/AMP0-/P20 to ANI6/P26, ANI8/AMP2-/P80 to ANI11/P83, ANI15/AV_{REFM}/P27 pins for analog input port, set PM20 to PM27 and PM80 to PM83 to 1. The output latches of P20 to P27 and P80 to P83 at this time may be 0 or 1.

If PM20 to PM27 and PM80 to PM83 are set to 0, they cannot be used as analog input port pins.

PM2 and PM6 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Caution If a pin is set as an analog input port, not the pin level but "0" is always read.

Remark 78K0/KB2-A : ANI0 to ANI5, ANI8 to ANI11
78K0/KC2-A : ANI0 to ANI6, ANI8 to ANI11, ANI15

Figure 12-12. Format of Port Mode Registers 2 and 8 (PM2, PM8)

<1> μPD78F0590, 78F0591 (30-pin products)

Address: FF22H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM2	1	1	PM25	PM24	PM23	PM22	PM21	PM20

Address: FF28H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM8	1	1	1	1	PM83	PM82	PM81	PM80

<2> μPD78F0592, 78F0593 (48-pin products)

Address: FF22H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20

Address: FF28H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM8	1	1	1	1	PM83	PM82	PM81	PM80

PMmn	Pmn pin I/O mode selection (mn = 20 to 27, 80 to 83)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

The functions of the ANI0/AMP0-/P20 to ANI6/P26, ANI8/AMP2-/P80 to ANI11/P83, ANI15/AV_{REFM}/P27 pins are determined according to the settings of ADPC register, ADS register, PM2 register, PM8 register, OAENn bit, ADREF bit.

Table 12-3. Setting Functions of ANI0/AMP0-/P20, ANI2/AMP0+/P22, ANI3/AMP1-/P23, ANI5/AMP1+/P25, ANI8/AMP2-/P80, and ANI10/AMP2+/P82 Pins

ADPC Register	PM2, PM8 Register	OAENn bit	ADS Register	ANI0/AMP0-/P20, ANI2/AMP0+/P22, ANI3/AMP1-/P23, ANI5/AMP1+/P25, ANI8/AMP2-/P80, ANI10/AMP2+/P82 Pins
Digital I/O selection	Input mode	0	–	Digital input
		1	–	Setting prohibited
	Output mode	0	–	Digital output
		1	–	Setting prohibited
Analog input selection	Input mode	0	Selects ANI.	Analog input (to be converted into digital signal)
			Does not select ANI.	Analog input (not to be converted into digital signal)
		1	Selects ANI.	Setting prohibited
			Does not select ANI.	Operational amplifier input
	Output mode	–	–	Setting prohibited

Table 12-4. Setting Functions of ANI1/AMP0OUT/P21, ANI4/AMP1OUT/P24, ANI9/AMP2OUT/P81 Pins

ADPC Register	PM2, PM8 Register	OAENn bit	ADS Register	ANI1/AMP0OUT/P21, ANI4/AMP1OUT/P24, ANI9/AMP2OUT/P81 Pins
Digital I/O selection	Input mode	0	–	Digital input
		1	–	Setting prohibited
	Output mode	0	–	Digital output
		1	–	Setting prohibited
Analog input selection	Input mode	0	Selects ANI.	Analog input (to be converted into digital signal)
			Does not select ANI.	Analog input (not to be converted into digital signal)
		1	Selects ANI.	Operational amplifier output (to be converted into digital signal)
			Does not select ANI.	Operational amplifier output (not to be converted into digital signal)
	Output mode	–	–	Setting prohibited

Caution When using an operational amplifier, the AMPn+, AMPn-, and AMPnOUT pins are used as I/O pins for the operational amplifier, and therefore cannot be used as analog input pins. However, the operational amplifier's output signal can be used as an analog input.

Remark n = 0 to 2

Table 12-5. Setting Functions of ANI6/P26 and ANI11/P83 pins

ADPC Register	PM2, PM8 Register	ADS Register	ANI6/P26, ANI11/P83 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 into digital signal)
		Does not select ANI.	Analog input (not to be converted into digital signal)
	Output mode	–	Setting prohibited

Remark 78K0/KB2-A : ANI11/P83 only
78K0/KC2-A : ANI6/P26, ANI11/P83

Table 12-6. Setting Functions of ANI15/AV_{REFM}/P27 pin (78K0/KC2-A only)

ADPC Register	PM2 Register	ADREF bit	ADS Register	ANI15/AV_{REFM}/P27 pin
Digital I/O selection	Input mode	0	–	Digital input
		1	–	Setting prohibited
	Output mode	0	–	Digital output
		1	–	Setting prohibited
Analog input selection	Input mode	0	Selects ANI.	Analog input (to be converted into digital signal)
			Does not select ANI.	Analog input (not to be converted into digital signal)
		1	–	Negative reference voltage for the A/D converter input
	Output mode	–	–	Setting prohibited

12.4 A/D Converter Operations

12.4.1 Basic operations of A/D converter

<1> Specify the A/D conversion time by using bits 5 to 1 (FR2 to FR0, LV1, and LV0) of ADM, and the operating mode by using bit 6 (ADSCM).

<2> Specify the source of the reference voltage for the A/D converter and whether to enable operation of the A/D converter voltage booster by using bits 7 and 1 (ADREF and VRGV) of the analog reference voltage control register (ADVRC).

<3> Set bit 0 (ADCE) of the A/D converter mode register (ADM) to 1 to start the operation of the A/D voltage comparator.

<4> Set channels for A/D conversion to analog input by using the A/D port configuration register (ADPC) and set to input mode by using port mode registers (PM2, PM8).

<5> Select one channel for A/D conversion using the analog input channel specification register (ADS).

<6> Use the A/D converter mode register 1 (ADM1) to set the trigger mode.

<7> If software trigger mode was set as the trigger mode in <6>, A/D conversion is started by setting bit 7 of ADM (ADCS) to 1.

If timer trigger mode was set as the trigger mode in <6>, A/D conversion is started when the timer trigger signal is detected.

<8> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<9> 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.

<10> Bit 11 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.

<11> The voltage difference between the series resistor string voltage tap and sampled voltage is compared by the A/D voltage comparator. If the analog input is greater than (1/2) AV_{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.

<12> Next, bit 10 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 11, as described below.

- Bit 11 = 1: (3/4) AV_{REF}
- Bit 11 = 0: (1/4) AV_{REF}

The voltage tap and sampled voltage are compared and bit 10 of SAR is manipulated as follows.

- Analog input voltage ≥ Voltage tap: Bit 10 = 1
- Analog input voltage < Voltage tap: Bit 10 = 0

<13> Comparison is continued in this way up to bit 0 of SAR.

<14> Upon completion of the comparison of 12 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.

<15> If single conversion mode was set as the operating mode in <1>, ADCS is automatically cleared after one A/D conversion and the A/D converter enters the wait status.

If successive conversion mode was set as the operating mode in <1>, steps <8> to <14> are executed repeatedly.. To stop the A/D converter, clear ADCS to 0.

To restart A/D conversion from the status of ADCE = 1, start from <7>. To start A/D conversion again when ADCE = 0, set ADCE to 1, wait for 1 μs or longer, and start <7>. To change a channel of A/D conversion, start from <5>.

Cautions 1. Make sure the period of <3> to <7> is 1 μs or more.

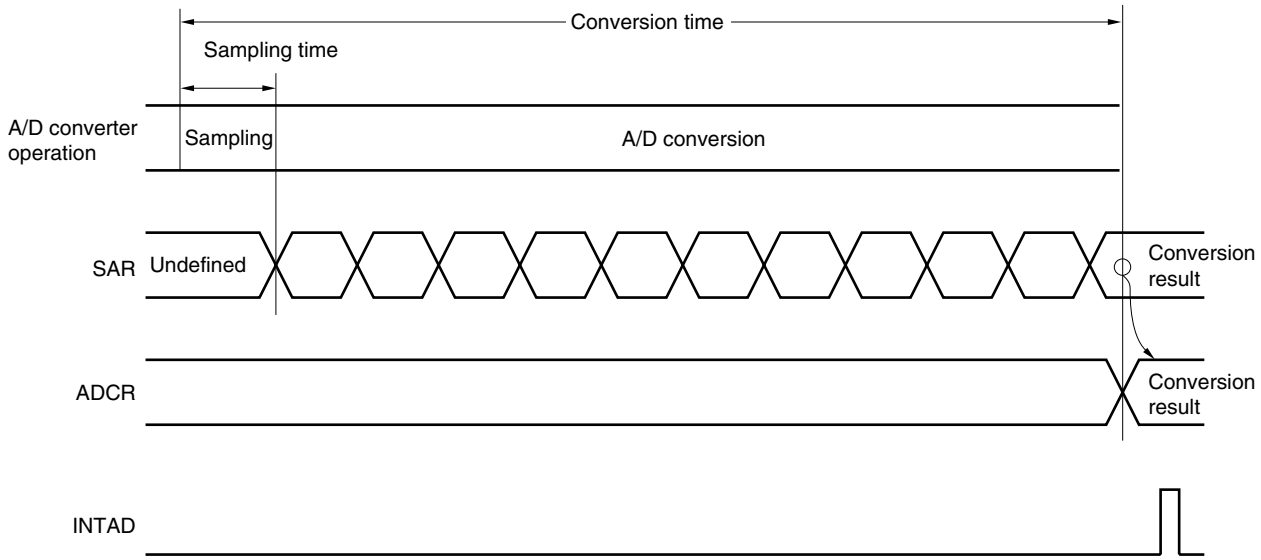
2. When using the analog input source, start up the operational amplifier before specifying the A/D conversion settings (see CHAPTER 13 OPERATIONAL AMPLIFIER for details). Also, avoid changing the settings of the operational amplifier during A/D conversion.

3. When using the A/D converter in normal mode 2 (LV1 = 0, LV0 = 1) or low-voltage mode (LV1 = 1, LV0 = 0), enable A/D conversion (by setting ADCE to 1) after starting up the A/D converter voltage booster (by setting VRGV to 1) and waiting for the reference voltage stabilization time (10 μs) to elapse. Note that it takes the A/D voltage comparator 1 μs to stabilize after being enabled. The validity of the data converted first can therefore be improved by enabling A/D conversion (by setting ADCS to 1) at least 1 μs after enabling the A/D voltage comparator. If A/D conversion is enabled without waiting for at least 1 μs, ignore the data converted first.

Remark Two types of A/D conversion result registers are available. Reset signal generation clears the A/D conversion result register to 0000H or 00H.

- ADCR (16 bits): Store 12-bit A/D conversion value
- ADCRH (8 bits): Store 8-bit A/D conversion value

Figure 12-13. Basic Operation of A/D Converter



12.4.2 Input voltage and conversion results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI6, ANI8 to ANI11, and ANI15) and the theoretical A/D conversion result (stored in the 12-bit A/D conversion result register (ADCR)) is shown by the following expression.

$$ADCR = INT\left(\frac{V_{AIN}}{AV_{REF}} \times 4096 + 0.5\right)$$

or

$$(ADCR - 0.5) \times \frac{AV_{REF}}{4096} \leq V_{AIN} < (ADCR + 0.5) \times \frac{AV_{REF}}{4096}$$

where, INT( ): Function which returns integer part of value in parentheses

V_{AIN}: Analog input voltage

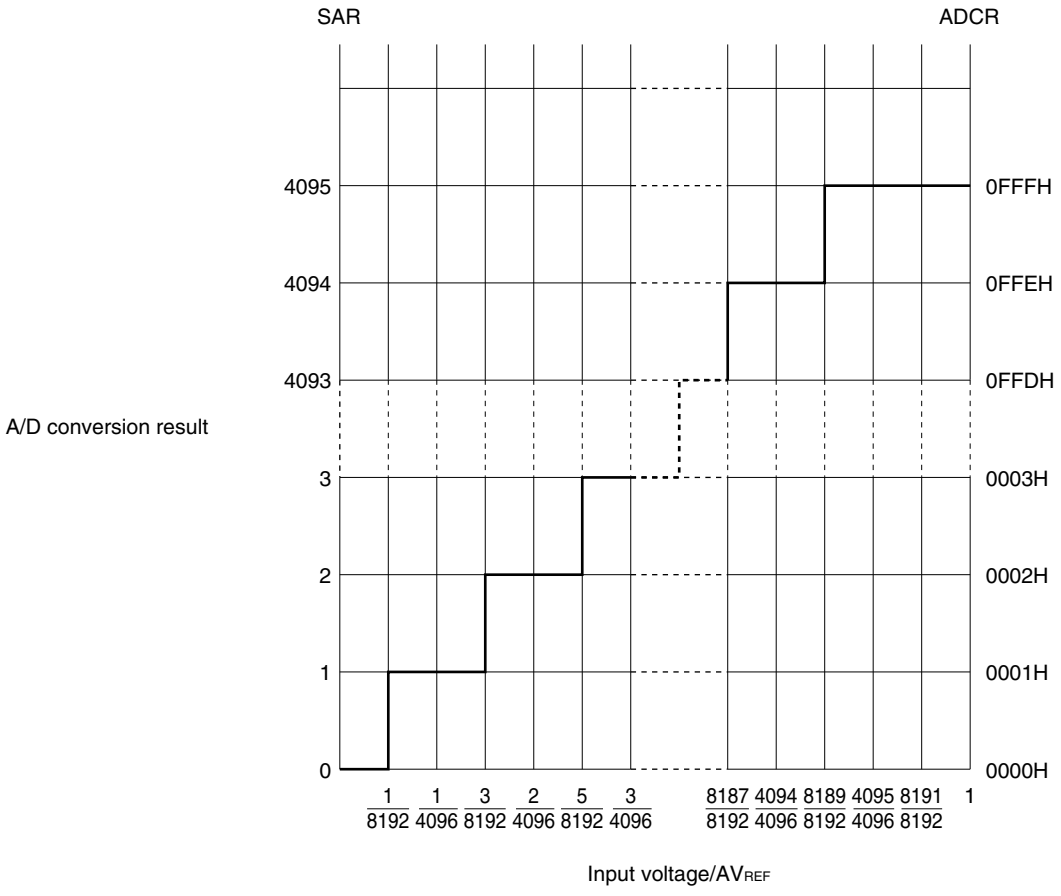
AV_{REF}: Conference voltage of A/D converter

ADCR: 12-bit A/D conversion result register (ADCR) value

Remark 78K0/KB2-A : ANI0 to ANI5, ANI8 to ANI11
78K0/KC2-A : ANI0 to ANI6, ANI8 to ANI11, ANI15

Figure 12-14 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 12-14. Relationship Between Analog Input Voltage and A/D Conversion Result



12.4.3 A/D converter operation mode

The operation mode of A/D converter has the following four functions.

- Software trigger mode (continuously convert mode)
- Software trigger mode (single convert mode)
- Timer trigger mode (continuously convert mode)
- Timer trigger mode (single convert mode)

(1) Software trigger mode (continuously convert mode)

<1> 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.

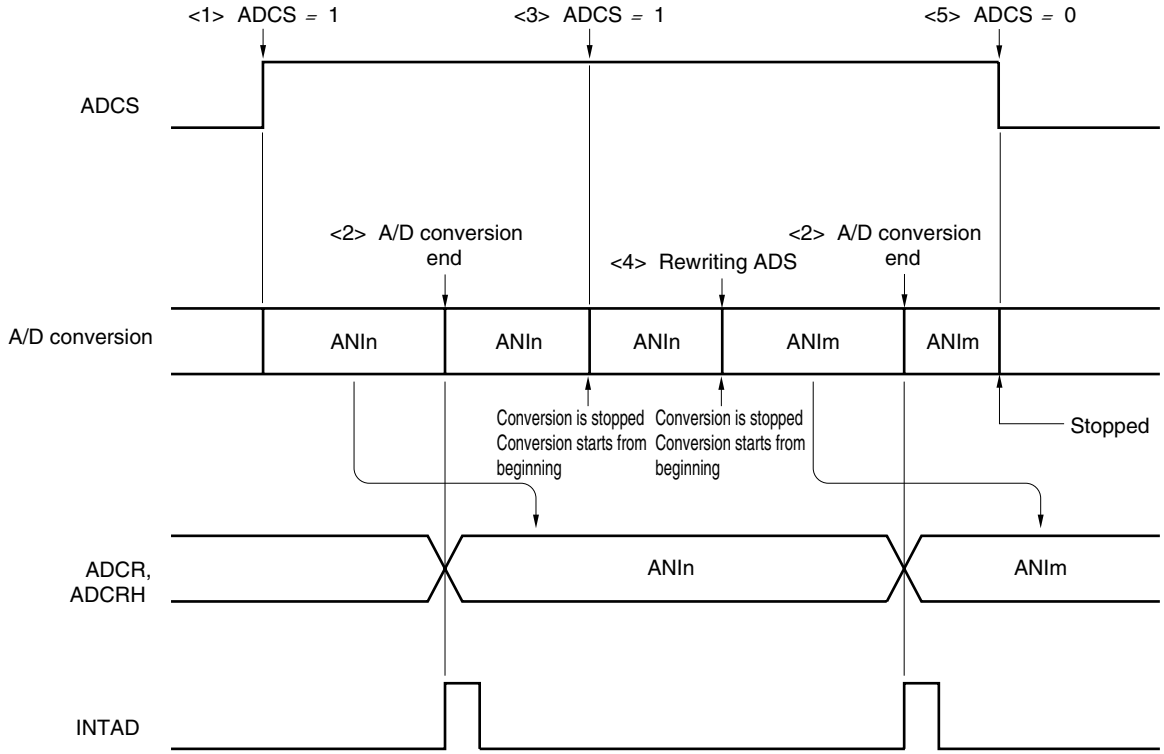
<2> When A/D conversion has been completed, the result of the A/D conversion is stored in the A/D conversion result register (ADCR, ADCRH), and an interrupt request signal (INTAD) is generated. When one A/D conversion has been completed, the next A/D conversion operation is immediately started. .

<3> If 1 is written to ADCS during A/D conversion, the A/D conversion operation under execution is stopped and restarted from the beginning. At this time, the conversion result immediately before is retained.

<4> If ADS is rewritten during A/D conversion, the A/D conversion operation under execution is stopped and restarted from the beginning. At this time, the conversion result immediately before is retained.

<5> 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 12-15. Software Trigger Mode (continuously convert mode)



Remark 78K0/KB2-A: n = 0 to 5, 8 to 11, m = 0 to 5, 8 to 11
78K0/KC2-A: n = 0 to 6, 8 to 11, 15, m = 0 to 6, 8 to 11, 15

(2) Software trigger mode (single convert mode)

<1> 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.

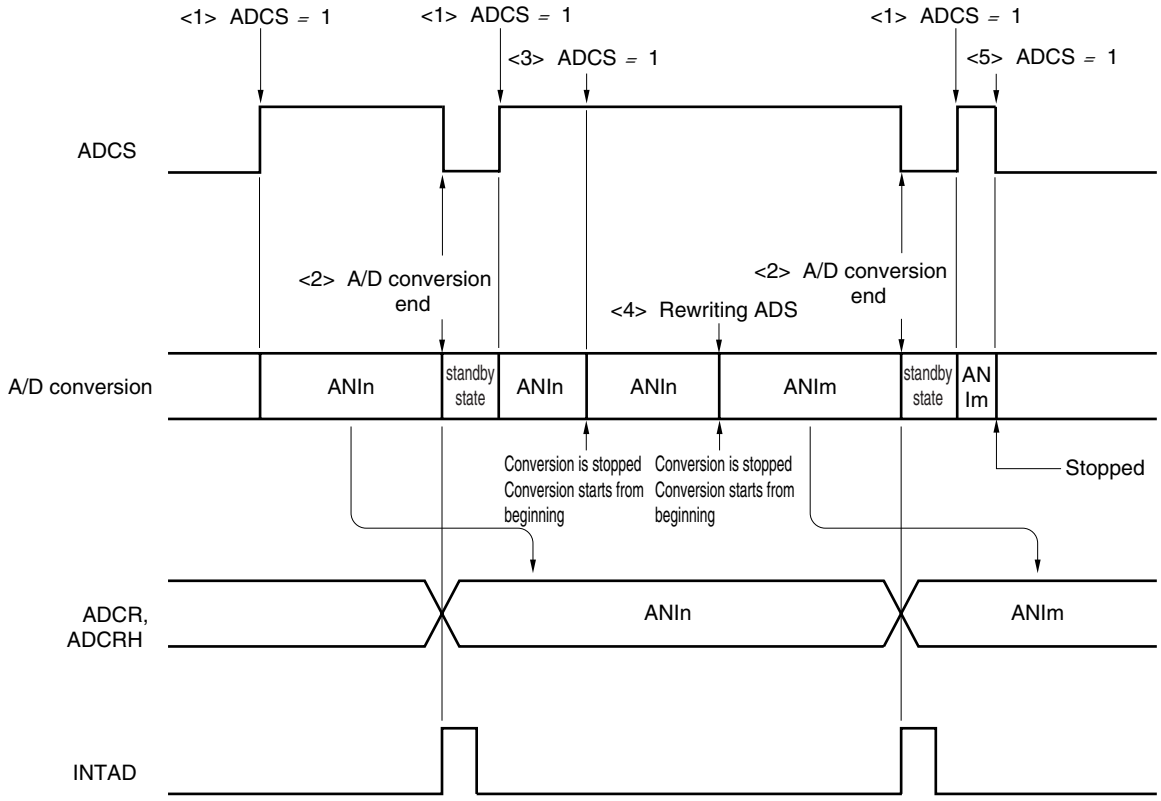
<2> When A/D conversion has been completed, the result of the A/D conversion is stored in the A/D conversion result register (ADCR, ADCRH), and an interrupt request signal (INTAD) is generated. When one A/D conversion has been completed, ADCS is automatically cleared after one A/D conversion and the A/D converter enters the wait status.

<3> If 1 is written to ADCS during A/D conversion, the A/D conversion operation under execution is stopped and restarted from the beginning. At this time, the conversion result immediately before is retained.

<4> If ADS is rewritten during A/D conversion, the A/D conversion operation under execution is stopped and restarted from the beginning. At this time, the conversion result immediately before is retained.

<5> 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 12-16. Software trigger mode (single convert mode)



Remark 78K0/KB2-A: n = 0 to 5, 8 to 11, m = 0 to 5, 8 to 11
78K0/KC2-A: n = 0 to 6, 8 to 11, 15, m = 0 to 6, 8 to 11, 15

(3) Timer trigger mode (continuously convert mode)

<1> Timer trigger mode is specified by setting bit 7 (ADTMD) of A/D converter mode register 1 (ADM1) to 1, after which the A/D converter waits for a timer trigger.

<2> When the timer trigger signal is detected, bit 7 (ADCS) of the A/D converter mode register (ADM) is automatically set to 1 and A/D conversion of the voltage applied to the analog input pin specified using the analog input channel specification register (ADS) starts.

<3> When A/D conversion has been completed, the result of the A/D conversion is stored in the A/D conversion result register (ADCR, ADCRH), 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. .

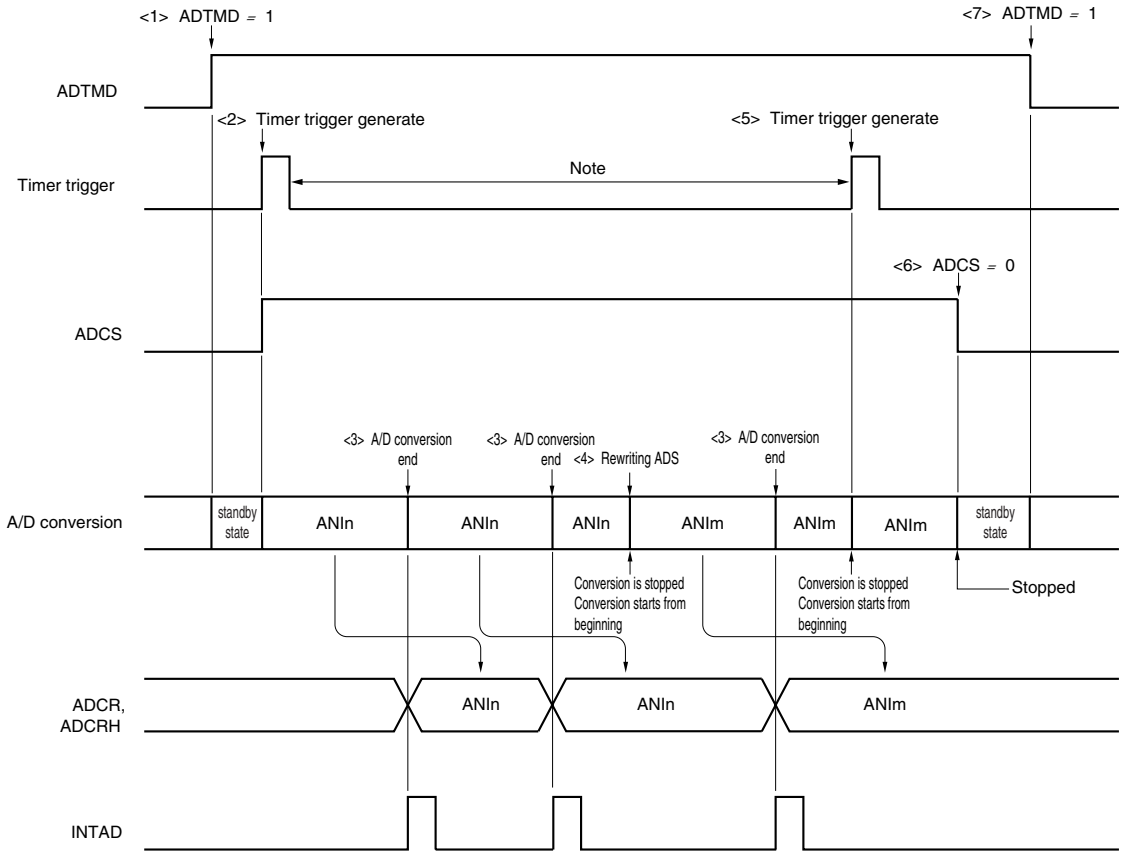
<4> If ADS is rewritten during A/D conversion, the A/D conversion operation under execution is stopped and restarted from the beginning. At this time, the conversion result immediately before is retained.

<5> If a timer trigger signal is generated during A/D conversion, the A/D conversion operation under execution is stopped and restarted from the beginning. At this time, the conversion result immediately before is retained.

<6> If 0 is written to ADCS during A/D conversion, A/D conversion immediately stops and the A/D converter waits for a timer trigger. At this time, the conversion result immediately before is retained.

<7> When 0 is written to ADTMD while A/D conversion operation is stopped (ADCS = 0), the software trigger mode is set and A/D conversion operation is not started, even if a timer trigger signal is generated.

Figure 12-17. Timer Trigger Mode (continuously convert mode)



Note Make sure that the timer trigger signal is generated at intervals equal to or longer than the A/D conversion time.

Remark 78K0/KB2-A: n = 0 to 5, 8 to 11, m = 0 to 5, 8 to 11
78K0/KC2-A: n = 0 to 6, 8 to 11, 15, m = 0 to 6, 8 to 11, 15

(4) Timer trigger mode (single convert mode)

<1> Timer trigger mode is specified by setting bit 7 (ADTMD) of A/D converter mode register 1 (ADM1) to 1, after which the A/D converter waits for a timer trigger.

<2> When the timer trigger signal is detected, bit 7 (ADCS) of the A/D converter mode register (ADM) is automatically set to 1 and A/D conversion of the voltage applied to the analog input pin specified using the analog input channel specification register (ADS) starts.

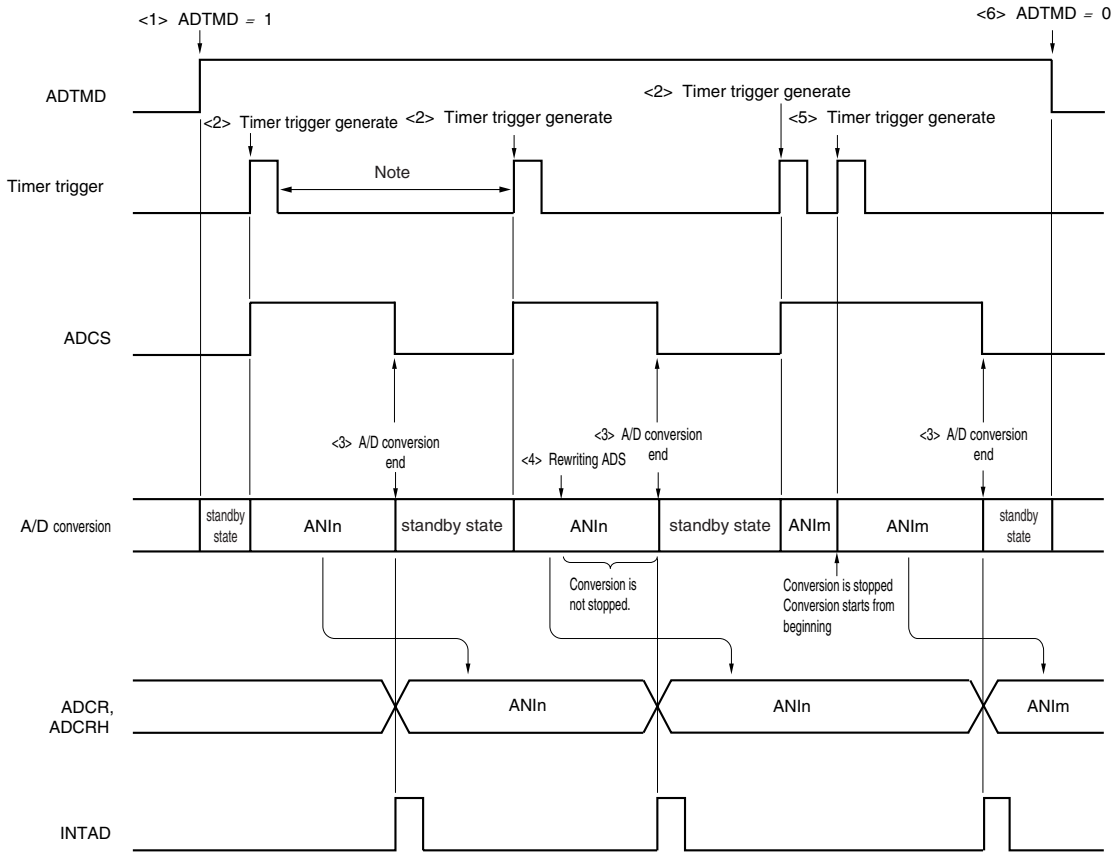
<3> When A/D conversion has been completed, the result of the A/D conversion is stored in the A/D conversion result register (ADCR, ADCRH), and an interrupt request signal (INTAD) is generated. When one A/D conversion has been completed, ADCS is automatically cleared after one A/D conversion and the A/D converter enters the wait status.

<4> If ADS is rewritten during A/D conversion, conversion continues until the end. The channel will be switched at the start of the next conversion.

<5> If a timer trigger signal is generated during A/D conversion, the A/D conversion operation under execution is stopped and restarted from the beginning. At this time, the conversion result immediately before is retained.

<6> When 0 is written to ADTMD while A/D conversion operation is stopped (ADCS = 0), the software trigger mode is set and A/D conversion operation is not started, even if a timer trigger signal is generated.

Figure 12-18. Timer Trigger Mode (single convert mode)



Note Make sure that the timer trigger signal is generated at intervals equal to or longer than the A/D conversion time.

Remark 78K0/KB2-A: n = 0 to 5, 8 to 11, m = 0 to 5, 8 to 11
78K0/KC2-A: n = 0 to 6, 8 to 11, 15, m = 0 to 6, 8 to 11, 15

The setting methods are described below.

<1> Select the conversion time by using bits 5 to 1 (FR2 to FR0, LV1, and LV0) of ADM, and select the operation mode by using bit 6 (ADSCM).

<2> Specify the source of the reference voltage for the A/D converter and whether to enable operation of the A/D converter voltage booster by using bits 7 and 1 (ADREF and VRGV) of the analog reference voltage control register (ADVRC).

<3> Set bit 0 (ADCE) of the A/D converter mode register (ADM) to 1.

<4> Set the channel to be used in the analog input mode by using bits 4 to 0 (ADPC4 to ADPC0) of the A/D port configuration register (ADPC), bits 7 to 0 (PM27 to PM20) of port mode register 2 (PM2), and bit 3 to 0 (PM83 to PM80) of port mode register 8 (PM8).

<5> Select a channel to be used by using bits 3 to 0 (ADS3 to ADS0) of the analog input channel specification register (ADS).

<6> Use the bits 0 and 7 (ADTRS, ADTMD) of the A/D converter mode register 1 (ADM1) to set the trigger mode.

<7> In software trigger mode:

→ Set bit 7 (ADCS) of ADM to 1 to start A/D conversion.

In timer trigger mode:

→ ADCS is automatically set to 1 and A/D conversion starts when the timer trigger signal is generated.

<8> When one A/D conversion has been completed, an interrupt request signal (INTAD) is generated.

<9> Transfer the A/D conversion data to the A/D conversion result register (ADCR, ADCRH).

<10> In successive conversion mode:

→ The next A/D conversion starts automatically.

In single conversion mode:

→ The A/D converter enters the wait status. To start another A/D conversion, go to step <7>.

<Change the channel>

<11> Select the channel using bits 3 to 0 (ADS3 to ADS0) of ADS[Note].

<12> When one A/D conversion has been completed, an interrupt request signal (INTAD) is generated.

<13> Transfer the A/D conversion data to the A/D conversion result register (ADCR, ADCRH).

<Complete A/D conversion>

<14> Clear ADCS to 0.

<14> In software trigger mode:

→ Clear ADCE to 0.

In timer trigger mode:

→ Clear ADCE and ADTMD to 0.

Note In timer trigger mode (single conversion mode), A/D conversion continues even if bits 3 to 0 of ADS are set during A/D conversion. The channel will be switched at the start of the next conversion. In all other modes, A/D conversion stops when bits 3 to 0 of ADS are set. The channel is then changed and A/D conversion starts again from the beginning.

Cautions 1. Make sure the period of <3> to <7> is 1 μs or more.

2. <3> may be done between <4> and <6>.

3. <3> can be omitted. However, ignore data of the first conversion after <7> in this case.

4. The period from <8> to <12> differs from the conversion time set using bits 5 to 1 (FR2 to FR0, LV1, LV0) of ADM. The period from <11> to <12> is the conversion time set using FR2 to FR0, LV1, and LV0.

5. When using the analog input source, start up the operational amplifier before specifying the A/D conversion settings (see CHAPTER 13 OPERATIONAL AMPLIFIER for details). Also, avoid changing the settings of the operational amplifier during A/D conversion

6. When using the A/D converter in normal mode 2 (LV1 = 0, LV0 = 1) or low-voltage mode (LV1 = 1, LV0 = 0), enable A/D conversion (by setting ADCE to 1) after starting up the A/D converter voltage booster (by setting VRGV to 1) and waiting for the reference voltage stabilization time (10 μs) to elapse. Note that it takes the A/D voltage comparator 1 μs to stabilize after being enabled. The validity of the data converted first can therefore be improved by enabling A/D conversion (by setting ADCS to 1) at least 1 μs after enabling the A/D voltage comparator. If A/D conversion is enabled without waiting for at least 1 μs, ignore the data converted first.

7. Make sure that the timer trigger signal is generated at intervals equal to or longer than the A/D conversion time.

12.5 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 12 bits.

$1LSB = 1/2^{12} = 1/4096$
$= 0.00091\%FSR$

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 12-19. Overall Error

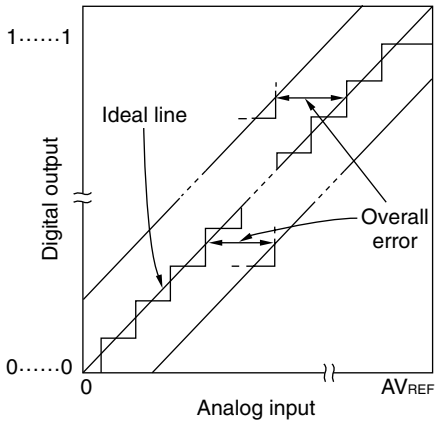
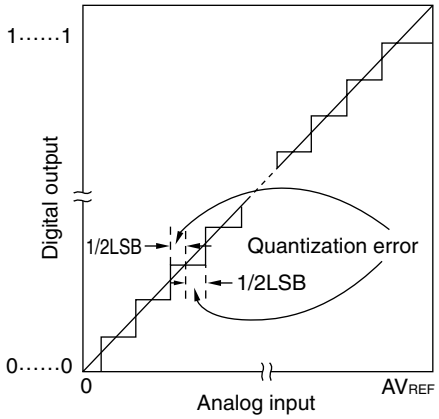


Figure 12-20. Quantization Error



(4) Zero-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (1/2LSB) 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 12-21. Zero-Scale Error

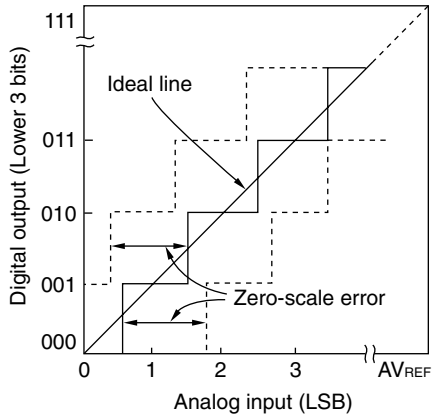


Figure 12-22. Full-Scale Error

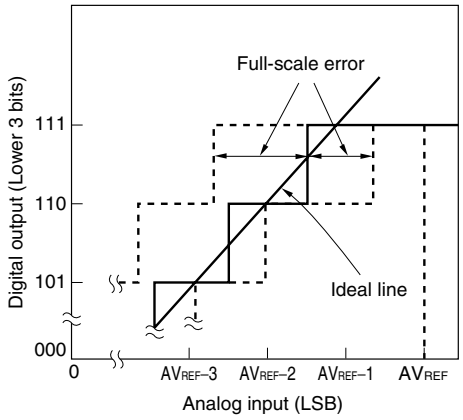


Figure 12-23. Integral Linearity Error

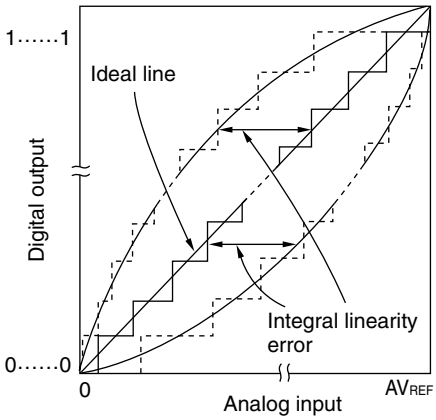
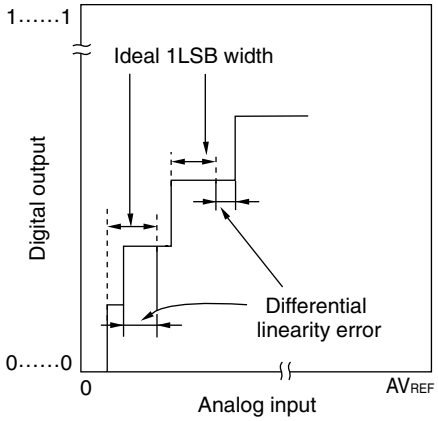


Figure 12-24. Differential Linearity Error



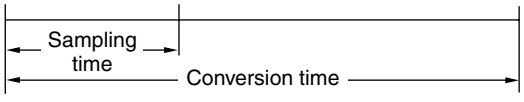
(8) Conversion time

This expresses the time from the start of sampling to when the digital output is obtained.

The sampling time is included in the conversion time in the characteristics table.

(9) Sampling time

This is the time the analog switch is turned on for the analog voltage to be sampled by the sample & hold circuit.



12.6 Cautions for A/D Converter

(1) Operating current in STOP mode

Shift to STOP mode after stopping the A/D converter (by setting bit 7 (ADCS) of the A/D converter mode register (ADM) to 0). The operating current can be reduced by setting bit 0 (ADCE) of the A/D converter mode register (ADM) to 0 at the same time.

When using the A/D converter in normal mode 2 (LV1 = 0, LV0 = 1) or low-voltage mode (LV1 = 1, LV0 = 0), be sure to clear bit 1 (VRGV) of the analog reference voltage control register (ADVRC) to 0 before shifting to STOP mode.

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 ANI6, ANI8 to ANI11, and ANI15

Observe the rated range of the ANI0 to ANI6, ANI8 to ANI11, and ANI15 input voltage. If a voltage of AV_{DD} 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 12-bit resolution, attention must be paid to noise input to the AV_{REF}, ANI0 to ANI6, ANI8 to ANI11, and ANI15 pins .

<1> Connect a capacitor with a low equivalent resistance and a good frequency response to the power supply.

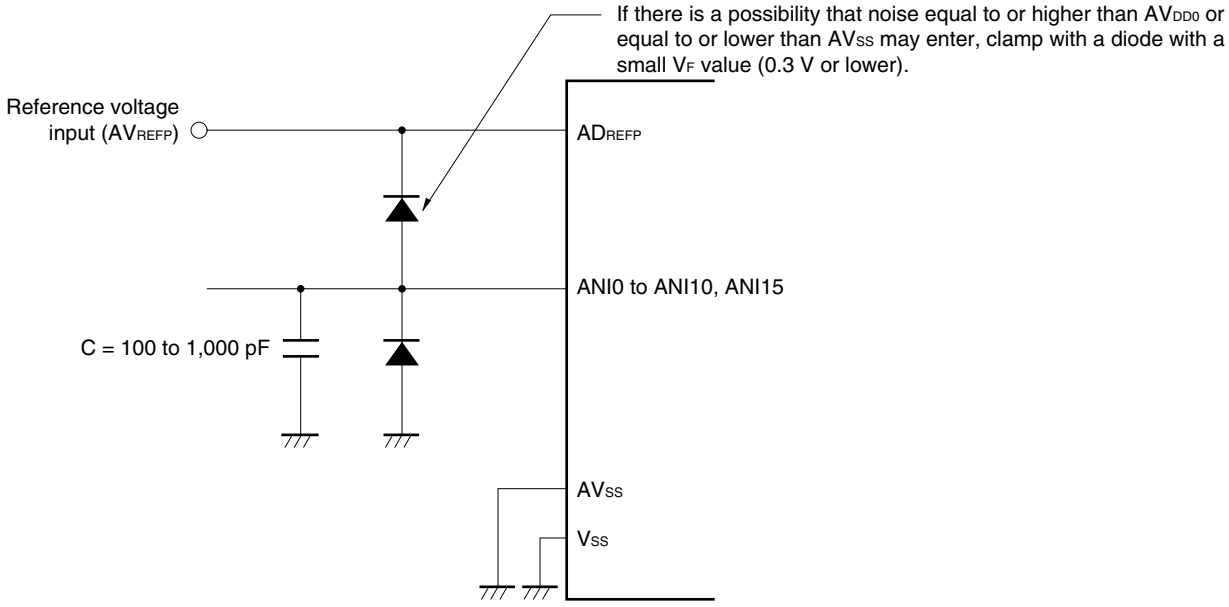
<2> The higher the output impedance of the analog input source, the greater the influence. To reduce the noise, connecting external C as shown in Figure 12-25 is recommended.

<3> Do not switch these pins with other pins during conversion.

<4> The accuracy is improved if the HALT mode is set immediately after the start of conversion.

Remark 78K0/KB2-A: ANI0 to ANI5, ANI8 to ANI11
78K0/KC2-A: ANI0 to ANI6, ANI8 to ANI11, ANI15

Figure 12-25. Analog Input Pin Connection



(5) ANI0 to ANI6, ANI8 to ANI11, ANI15

<1> The analog input pins (ANI0 to ANI6, ANI15) are also used as input port pins (P20 to P27).
The analog input pins (ANI8 to ANI11) are also used as input port pins (P80 to P83).
When A/D conversion is performed with any of ANI0 to ANI6, ANI8 to ANI11, and ANI15 selected, do not access P20 to P27 and P80 to P83 while conversion is in progress; otherwise the conversion resolution may be degraded. It is recommended to select pins used as P20 to P27 and P80 to P83 starting with the ANI0/P20 that is the furthest from AV_{DD}.

<2> If the pins adjacent to the pins currently used for A/D conversion used as a digital I/O pin, the expected value of the A/D conversion may not be obtained due to coupling noise. Make sure that a digital pulse is not input to or output from pins adjacent to the pin whose signal is being A/D converted.

<3> If any of the pins of port 2 or 8 is being used as a digital output port during A/D conversion, the expected A/D conversion value might not be able to be obtained due to coupling noise. Make sure, therefore, that a digital pulse is not output from any of the pins of port 2 or 8 during A/D conversion.

(6) Input impedance of ANI0 to ANI6, ANI8 to ANI11, and ANI15 pins

This A/D converter charges a sampling capacitor for sampling during sampling time.
Therefore, only a leakage current flows when sampling is not in progress, and a current that charges the capacitor flows during sampling. Consequently, the input impedance fluctuates depending on whether sampling is in progress, and on the other states.
To make sure that sampling is effective, however, it is recommended to keep the output impedance of the analog input source to within 1 kΩ, and to connect a capacitor of about 100 pF to the ANI0 to ANI6, ANI8 to ANI11, and ANI15 pins (see **Figure 12-25**).

Remark 78K0/KB2-A: ANI0 to ANI5, ANI8 to ANI11
78K0/KC2-A: ANI0 to ANI6, ANI8 to ANI11, ANI15

(7) AV_{REFP} pin[Note] input impedance

A series resistor string of several tens of kΩ is connected between the AV_{REFP} and AV_{REFM} (or 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_{REFM} (or AV_{SS}) pins, resulting in a large reference voltage of A/D converter (AV_{REF}) error.

Note As the AV_{REFM} and AV_{REFP} are not mounted on 78K0/KB2-A, replace AV_{REFP} with AV_{REF}, respectively.

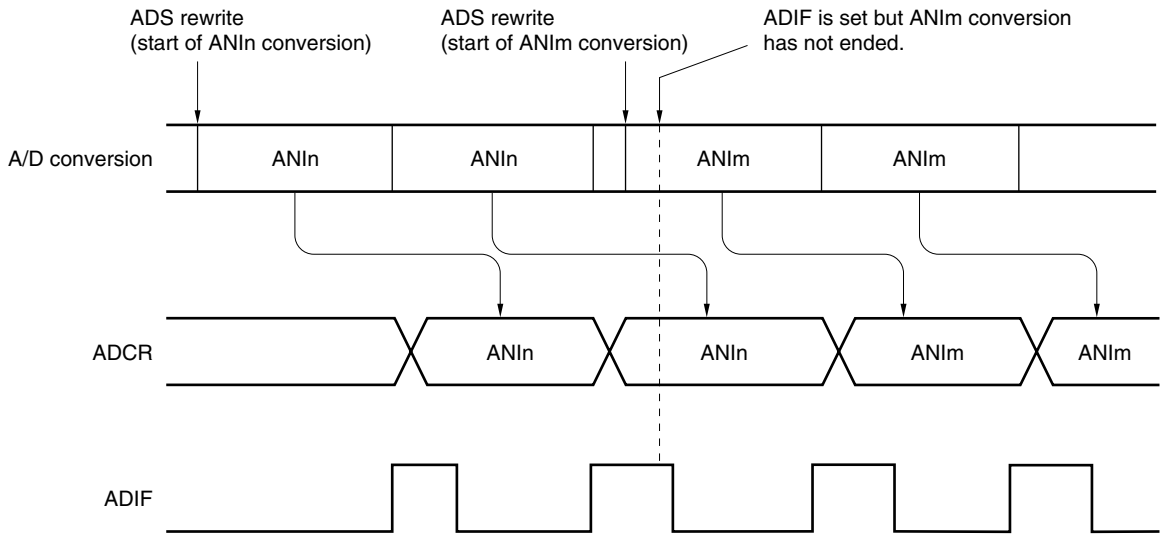
(8) Interrupt request flag (ADIF)

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 12-26. Timing of A/D Conversion End Interrupt Request Generation



Remark 78K0/KB2-A: n = 0 to 5, 8 to 11, m = 0 to 5, 8 to 11
78K0/KC2-A: n = 0 to 6, 8 to 11, 15, m = 0 to 6, 8 to 11, 15

(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 1 (ADM1), 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 ADM1, 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 12-27. Internal Equivalent Circuit of ANIn Pin

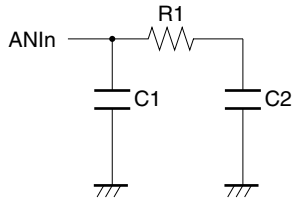


Table 12-7. Resistance and Capacitance Values of Equivalent Circuit (Reference Values)

AV_{DD}	R1	C1	C2
2.3 V ≤ AV_{DD} ≤ 5.5 V	11.5 kΩ	8.0 pF	8.0 pF

Remarks 1. The resistance and capacitance values shown in Table 12-7 are not guaranteed values.

2. 78K0/KB2-A: n = 0 to 5, 8 to 11
78K0/KC2-A: n = 0 to 6, 8 to 11, 15

CHAPTER 13 OPERATIONAL AMPLIFIER

13.1 Function of Operational Amplifier

Operational amplifier is mounted onto all 78K0/Kx2-A microcontrollers.

The operational amplifiers amplify the potential difference of the analog voltages input from two pins (the AMPn- pin and the AMPn+ pin) and output the amplified voltage from the AMPnOUT pin. The amplified voltage can be used as an analog input of the A/D converter, because the AMPnOUT pin is alternatively used with analog input pin of the A/D converter.

Remark n = 0-2.

13.2 Configuration of Operational Amplifier

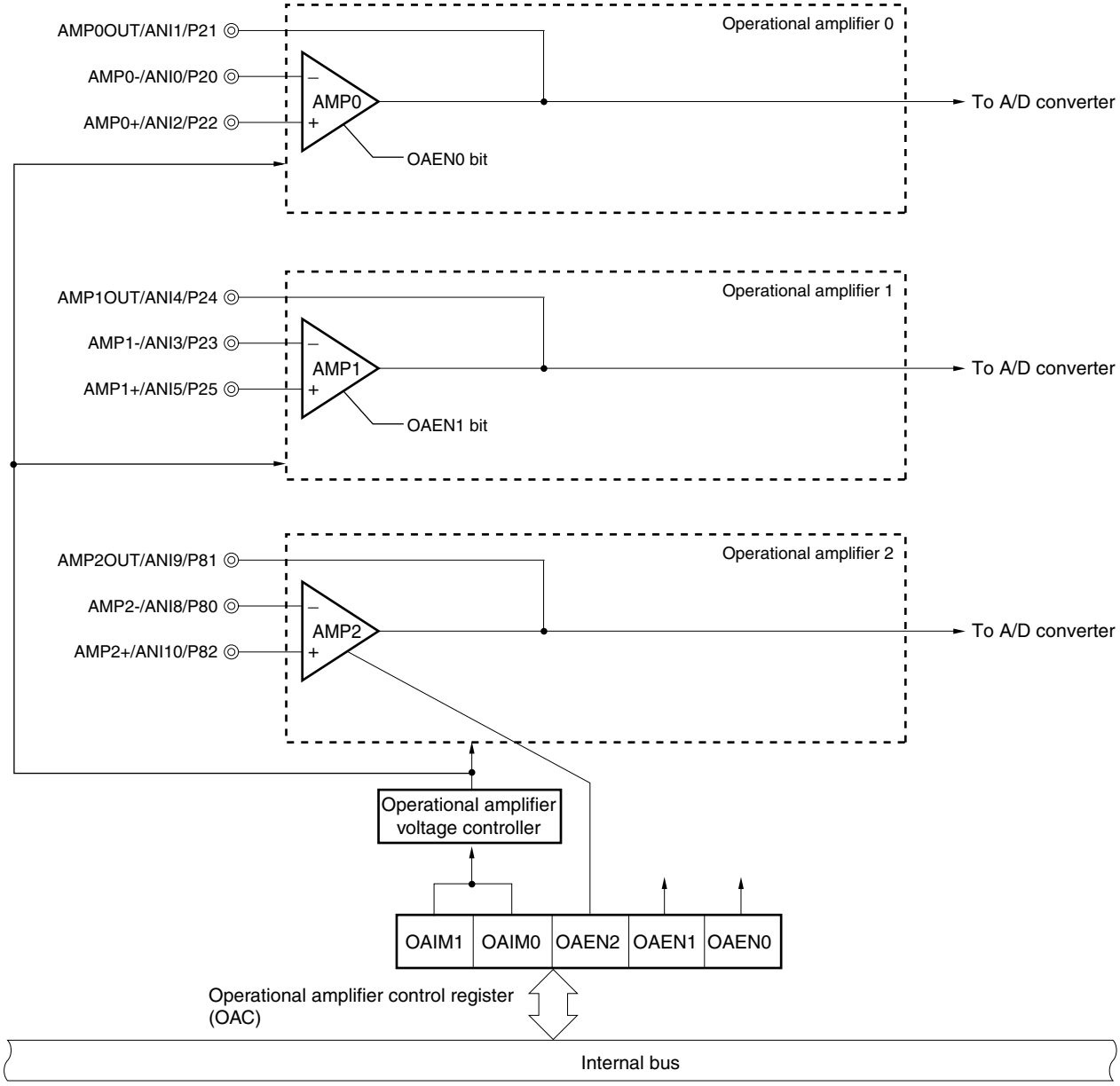
The operational amplifier consists of the following hardware.

Table 13-1. Configuration of Operational Amplifier

Item	Configuration
Operational amplifier input	AMPn- pin, AMPn+ pin
Operational amplifier output	AMPnOUT pin
Control registers	Operational amplifier control register (OAC) A/D configuration register (ADPC) Port mode register 2 and 8 (PM2, PM8)

Remark n = 0-2.

Figure 13-1. Block Diagram of Operational Amplifier



13.3 Registers Used in Operational Amplifier

The operational amplifier uses the following three registers.

- Operational amplifier control register (OAC)
- A/D port configuration register (ADPC)
- Port mode registers 2 and 8 (PM2, PM8)

(1) Operational amplifier control register (OAC)

AMP0 to AMP2 control the operations of operational amplifier.

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

Reset signal generation clears AMP0M to 00H.

Figure 13-2. Format of Operational Amplifier Control Register (OAC)

Address: FF3BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
OAC	OAIM1	OAIM0	0	0	0	OAEN2	OAEN1	OAEN0

OAIM1	OAIM0	Operational amplifier operation mope
0	0	Mode 2 (Slew rate : 0.4 V/μs (TYP.))
0	1	Mode 3 (Slew rate : 1.4 V/μs (TYP.))
1	0	Setting prohibited
1	1	Mode 1 (Slew rate : 0.2 V/μs (TYP.))

OAEN2	Operational amplifier 2 operation control
0	Stops operational amplifier 2 operation
1	Enables operational amplifier 2 operation

OAEN1	Operational amplifier 1 operation control
0	Stops operational amplifier 1 operation
1	Enables operational amplifier 1 operation

OAEN0	Operational amplifier 0 operation control
0	Stops operational amplifier 0 operation
1	Enables operational amplifier 0 operation

Cautions 1. Use the ADPC registers to select a pin used in the operational amplifier as an analog input.

2. When using as digital inputs the pins of port 2, 8, which are not used with the operational amplifier, when the operational amplifier is used, make sure that the input levels of digital input ports are fixed.

(2) A/D port configuration register 0 (ADPC)

ADPC switches the ANI0/AMP0-/P20 to ANI6/P26, ANI8/AMP2-/P80 to ANI11/P83, ANI15/AV_{REFM}/P27 pins to analog input of A/D converter or digital I/O of port. Set pins used in the operational amplifier to analog inputs.

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

Reset signal generation clears ADPC to 10H.

Remark 78K0/KB2-A: ANI0 to ANI5, ANI8 to ANI11
78K0/KC2-A: ANI0 to ANI6, ANI8 to ANI11, ANI15

Figure 13-3. Format of A/D Port Configuration Register (ADPC)

Address: FF2FH 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											
						ANI15 /AV_{REFM} /P27	ANI11 /P83	ANI10 /AMP2+ /P82	ANI9 /$AMP2_{OUT}$ /P81	ANI8 /AMP2- /P80	ANI6 /P26	ANI5 /AMP1+ /P25	ANI4 /$AMP1_{OUT}$ /P24	ANI3 /AMP1- /P23	ANI2 /AMP0+ /P22	ANI1 /$AMP0_{OUT}$ /P21	ANI0 /AMP0- /P20
	0	0	0	0	0	A	A	A	A	A	A	A	A	A	A	A	A
	0	0	0	0	1	A	A	A	A	A	A	A	A	A	A	A	D
	0	0	0	1	0	A	A	A	A	A	A	A	A	A	A	D	D
	0	0	0	1	1	A	A	A	A	A	A	A	A	A	D	D	D
	0	0	1	0	0	A	A	A	A	A	A	A	A	D	D	D	D
	0	0	1	0	1	A	A	A	A	A	A	A	D	D	D	D	D
Note→	0	0	1	1	0	A	A	A	A	A	A	D	D	D	D	D	D
	0	0	1	1	1	Setting prohibited											
	0	1	0	0	0	A	A	A	A	A	D	D	D	D	D	D	D
	0	1	0	0	1	A	A	A	A	D	D	D	D	D	D	D	D
	0	1	0	1	0	A	A	A	D	D	D	D	D	D	D	D	D
	0	1	0	1	1	A	A	D	D	D	D	D	D	D	D	D	D
Note→	0	1	1	1	1	A	D	D	D	D	D	D	D	D	D	D	D
	1	0	0	0	0	D	D	D	D	D	D	D	D	D	D	D	D
	Other than the above					Setting prohibited											

Note In the case of 78K0/KB2-A, setting is prohibited.

Caution Use the port mode register 2 or 8 (PM2, PM8) to select a pin used in the operational amplifier as an input input.

(3) Port mode registers 2 and 8 (PM2, PM8)

When using ANI0/AMP0-/P20 to ANI6/P26, ANI8/AMP2-/P80 to ANI11/P83, ANI15/AV_{REFM}/P27 pins for the analog input port, set PM20 to PM27 and PM80 to PM83 to 1. The output latches of P20 to P27 and P80 to P83 at this time may be 0 or 1.

If PM20 to PM27 and PM80 to PM83 are set to 0, they cannot be used as the analog input port.

PM2 and PM8 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets PM2, PM8 to FFH.

Caution If a pin is set as an analog input port, not the pin level but "0" is always read.

Remark 78K0/KB2-A : ANI0 to ANI5, ANI8 to ANI11
78K0/KC2-A : ANI0 to ANI6, ANI8 to ANI11, ANI15

Figure 13-4. Format of Port Mode Registers 2 and 8 (PM2, PM8)

(1) 78K0/KB2-A

Address: FF22H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM2	1	1	PM25	PM24	PM23	PM22	PM21	PM20

Address: FF28H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM8	1	1	1	1	PM83	PM82	PM81	PM80

(2) 78K0/KC2-A

Address: FF22H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20

Address: FF28H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM8	1	1	1	1	PM83	PM82	PM81	PM80

PMmn	Pmn pin I/O mode selection (mn = 20 to 27, 80 to 83)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

The functions of the ANI0/AMP0-/P20 to ANI6/P26, ANI8/AMP2-/P80 to ANI11/P83, ANI15/AV_{REFM}/P27 pins are determined according to the settings of ADPC register, ADS register, PM2 register, PM8 register, OAENn bit, ADREF bit.

Table 13-2. Setting Functions ofANI0/AMP0-/P20, ANI2/AMP0+/P22, ANI3/AMP1-/P23, ANI5/AMP1+/P25, ANI8/AMP2-/P80, ANI10/AMP2+/P82 Pins

ADPC Register	PM2, PM8 Register	OAENn bit	ADS Register	ANI0/AMP0-/P20, ANI2/AMP0+/P22, ANI3/AMP1-/P23, ANI5/AMP1+/P25, ANI8/AMP2-/P80, ANI10/AMP2+/P82 Pins
Digital I/O selection	Input mode	0	–	Digital input
		1	–	Setting prohibited
	Output mode	0	–	Digital output
		1	–	Setting prohibited
Analog input selection	Input mode	0	Selects ANI.	Analog input (to be converted into digital signal)
			Does not select ANI.	Analog input (not to be converted into digital signal)
		1	Selects ANI.	Setting prohibited
			Does not select ANI.	Operational amplifier input
	Output mode	–	–	Setting prohibited

Table 13-3. Setting Functions of ANI1/AMP0OUT/P21, ANI4/AMP1OUT/P24, ANI9/AMP2OUT/P81 Pins

ADPC Register	PM2, PM8 Register	OAENn bit	ADS Register	ANI1/AMP0OUT/P21, ANI4/AMP1OUT/P24, ANI9/AMP2OUT/P81 Pins
Digital I/O selection	Input mode	0	–	Digital input
		1	–	Setting prohibited
	Output mode	0	–	Digital output
		1	–	Setting prohibited
Analog input selection	Input mode	0	Selects ANI.	Analog input (to be converted into digital signal)
			Does not select ANI.	Analog input (not to be converted into digital signal)
		1	Selects ANI.	Operational amplifier output (to be converted into digital signal)
			Does not select ANI.	Operational amplifier output (not to be converted into digital signal)
	Output mode	–	–	Setting prohibited

Caution When using an operational amplifier, the AMPn+, AMPn-, and AMPnOUT pins are used as I/O pins for the operational amplifier, and therefore cannot be used as analog input pins. However, the operational amplifier's output signal can be used as an analog input.

Remark n = 0 to 2

Table 13-4. Setting Functions of ANI6/P26 and ANI11/P83 pins

ADPC Register	PM2, PM8 Register	ADS Register	ANI6/P26, ANI11/P83 pin
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input selection	Input mode	Selects ANI.	Analog input (to be converted into digital signal)
		Does not select ANI.	Analog input (not to be converted into digital signal)
	Output mode	–	Setting prohibited

Remark 78K0/KB2-A: ANI11/P83 only
78K0/KC2-A: ANI6/P26, ANI11/P83

Table 13-5. Setting Functions of ANI15/AV_{REFM}/P27 Pin (78K0/KC2-A only)

ADPC Register	PM2 Register	ADREF bit	ADS Register	ANI15/AV_{REFM}/P27 pin
Digital I/O selection	Input mode	0	–	Digital input
		1	–	Setting prohibited
	Output mode	0	–	Digital output
		1	–	Setting prohibited
Analog input selection	Input mode	0	Selects ANI.	Analog input (to be converted into digital signal)
			Does not select ANI.	Analog input (not to be converted into digital signal)
		1	–	Negative reference voltage for the A/D converter input
	Output mode	–	–	Setting prohibited

13.4 Operation of Operational Amplifier

The operational amplifiers amplify the potential difference of the analog voltages input from two pins (the AMPn- pin and the AMPn+ pin) and output the amplified voltage from the AMPnOUT pin.The amplification factor can be changed by inserting external components such as resistors.

The amplified voltage can be used as an analog input of the A/D converter, because the AMPnOUT pin is alternatively used with analog input pin of the A/D converter.

The procedure for starting operation in single amplifier mode is described below.

<1> Use the port mode register x (PMx) register to set the pins (AMPn-, AMPn+, AMPnOUT) to be used in single amplifier mode as input mode.
<2> Use the A/D port configuration register (ADPC) to set the pins (AMPn-, AMPn+, AMPnOUT) to be used in single amplifier mode to analog input.
<3> Specify the operating mode by using the OAIM1 and OAIM0 bits of the operational amplifier control register (OAC).
<4> Enable operation by setting the OAENn bit of the OAC register to 1.
<5> Program the software to wait for the turn-on time required for the operational amplifier to stabilize.

Caution To use as an input of the A/D converter a output of operational amplifier, enable operation in operational amplifier before selecting an analog input channel by using the ADS register.

Remark n = 0 to 2, x = 2, 8

CHAPTER 14 SERIAL INTERFACE UART6

14.1 Functions of Serial Interface UART6

Serial interface UART6 are mounted onto all 78K0/Kx2-A microcontrollers.

Serial interface UART6 has the following two modes.

(1) Operation stop mode

This mode is used when serial communication is not executed and can enable a reduction in the power consumption.

For details, see **14.4.1 Operation stop mode**.

(2) Asynchronous serial interface (UART) mode

This mode supports the LIN (Local Interconnect Network)-bus. The functions of this mode are outlined below.

For details, see **14.4.2 Asynchronous serial interface (UART) mode** and **14.4.3 Dedicated baud rate generator**.

- Maximum transfer rate: 625 kbps
- Two-pin configuration TxD6: Transmit data output pin
  RxD6: Receive data input pin
- Data length of communication data can be selected from 7 or 8 bits.
- Dedicated internal 8-bit baud rate generator allowing any baud rate to be set
- Transmission and reception can be performed independently (full duplex operation).
- MSB- or LSB-first communication selectable
- Inverted transmission operation
- Sync break field transmission from 13 to 20 bits
- More than 11 bits can be identified for sync break field reception (SBF reception flag provided).

Cautions 1. The TxD6 output inversion function inverts only the transmission side and not the reception side. To use this function, the reception side must be ready for reception of inverted data.

2. If clock supply to serial interface UART6 is not stopped (e.g., in the HALT mode), normal operation continues. If clock supply to serial interface UART6 is stopped (e.g., in the STOP mode), each register stops operating, and holds the value immediately before clock supply was stopped. The TxD6 pin also holds the value immediately before clock supply was stopped and outputs it. However, the operation is not guaranteed after clock supply is resumed. Therefore, reset the circuit so that POWER6 = 0, RXE6 = 0, and TXE6 = 0.

3. Set POWER6 = 1 and then set TXE6 = 1 (transmission) or RXE6 = 1 (reception) to start communication.

4. TXE6 and RXE6 are synchronized by the base clock (f_{XCLK6}) set by CKSR6. To enable transmission or reception again, set TXE6 or RXE6 to 1 at least two clocks of the base clock after TXE6 or RXE6 has been cleared to 0. If TXE6 or RXE6 is set within two clocks of the base clock, the transmission circuit or reception circuit may not be initialized.

5. Set transmit data to TXB6 at least one base clock (f_{XCLK6}) after setting TXE6 = 1.

6. If data is continuously transmitted, the communication timing from the stop bit to the next start bit is extended two operating clocks of the macro. However, this does not affect the result of communication because the reception side initializes the timing when it has detected a start bit. Do not use the continuous transmission function if the interface is used in LIN communication operation.

Remark LIN stands for Local Interconnect Network and is a low-speed (1 to 20 kbps) serial communication protocol intended to aid the cost reduction of an automotive network.

LIN communication is single-master communication, and up to 15 slaves can be connected to one master.

The LIN slaves are used to control the switches, actuators, and sensors, and these are connected to the LIN master via the LIN network.

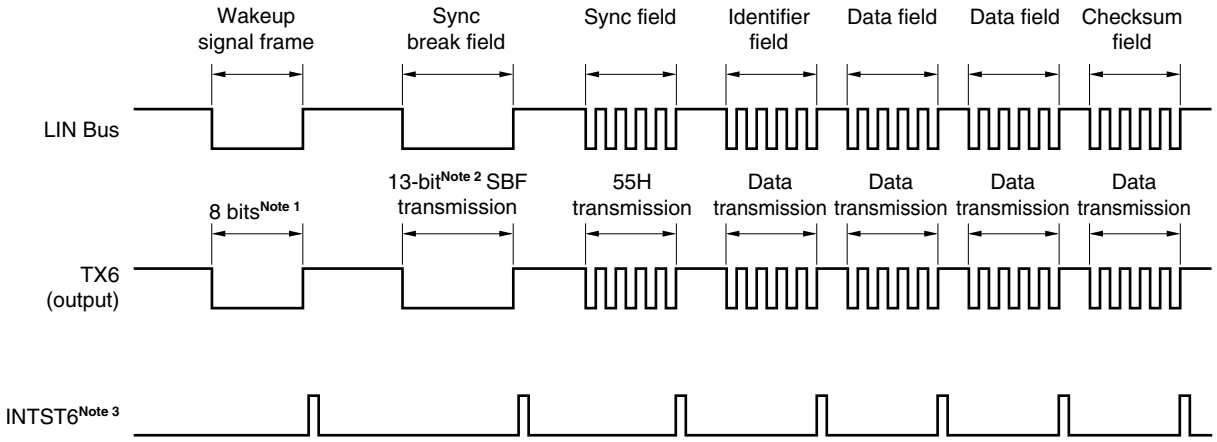
Normally, the LIN master is connected to a network such as CAN (Controller Area Network).

In addition, the LIN bus uses a single-wire method and is connected to the nodes via a transceiver that complies with ISO9141.

In the LIN protocol, the master transmits a frame with baud rate information and the slave receives it and corrects the baud rate error. Therefore, communication is possible when the baud rate error in the slave is ±15% or less.

Figures 14-1 and 14-2 outline the transmission and reception operations of LIN.

Figure 14-1. LIN Transmission Operation



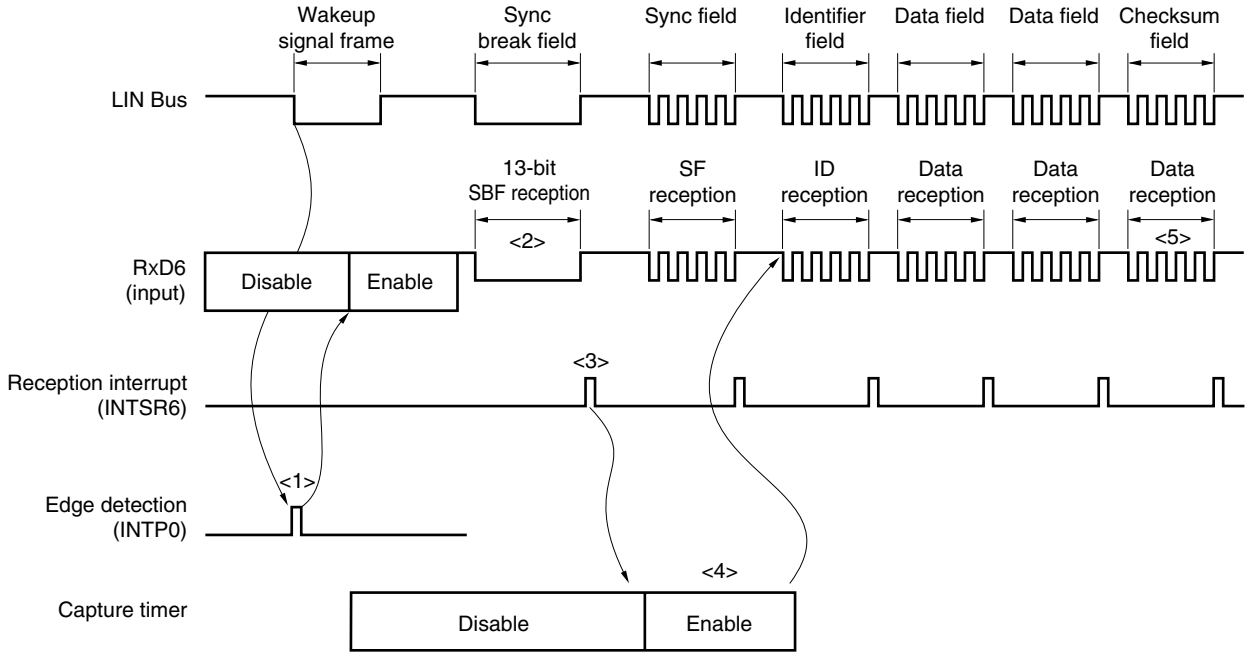
Notes 1. The wakeup signal frame is substituted by 80H transmission in the 8-bit mode.

2. The sync break field is output by hardware. The output width is the bit length set by bits 4 to 2 (SBL62 to SBL60) of asynchronous serial interface control register 6 (ASICL6) (see **14.4.2 (2) (h) SBF transmission**).

3. INTST6 is output on completion of each transmission. It is also output when SBF is transmitted.

Remark The interval between each field is controlled by software.

Figure 14-2. LIN Reception Operation



Reception processing is as follows.

<1> The wakeup signal is detected at the edge of the pin, and enables UART6 and sets the SBF reception mode.

<2> Reception continues until the STOP bit is detected. When an SBF with low-level data of 11 bits or more has been detected, it is assumed that SBF reception has been completed correctly, and an interrupt signal is output. If an SBF with low-level data of less than 11 bits has been detected, it is assumed that an SBF reception error has occurred. The interrupt signal is not output and the SBF reception mode is restored.

<3> If SBF reception has been completed correctly, an interrupt signal is output. Start 16-bit timer/event counter 00 by the SBF reception end interrupt servicing and measure the bit interval (pulse width) of the sync field (see **6.4.8 Pulse width measurement operation**). Detection of errors OVE6, PE6, and FE6 is suppressed, and error detection processing of UART communication and data transfer of the shift register and RXB6 is not performed. The shift register holds the reset value FFH.

<4> Calculate the baud rate error from the bit interval of the sync field, disable UART6 after SF reception, and then re-set baud rate generator control register 6 (BRGC6).

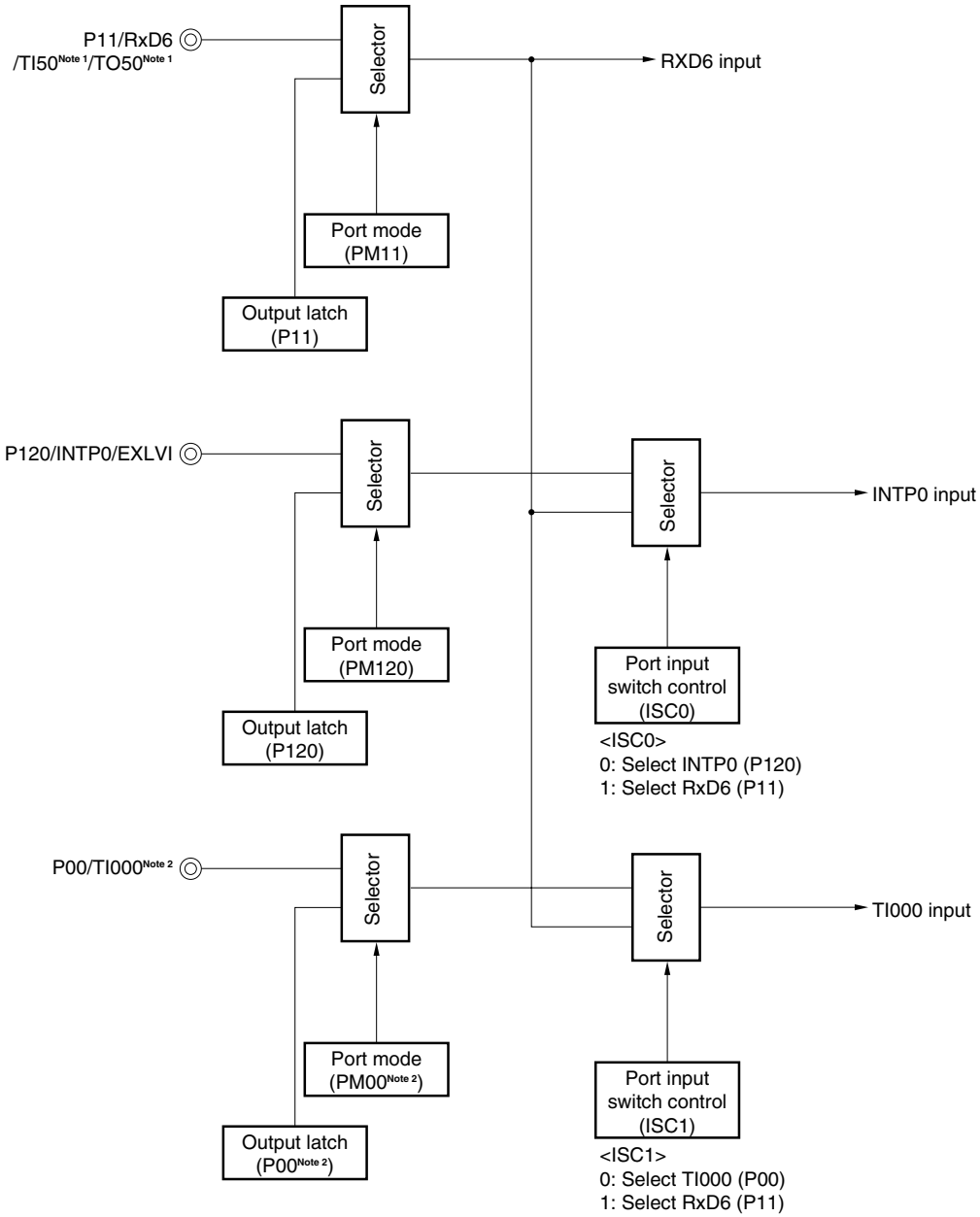
<5> Distinguish the checksum field by software. Also perform processing by software to initialize UART6 after reception of the checksum field and to set the SBF reception mode again.

Figure 14-3 shows the port configuration for LIN reception operation.

The wakeup signal transmitted from the LIN master is received by detecting the edge of the external interrupt (INTP0). The length of the sync field transmitted from the LIN master can be measured using the external event capture operation of 16-bit timer/event counter 00, and the baud rate error can be calculated.

The input source of the reception port input (RxD6) can be input to the external interrupt (INTP0) and 16-bit timer/event counter 00 by port input switch control (ISC0/ISC1), without connecting RxD6 and INTP0/TI000 externally.

Figure 14-3. Port Configuration for LIN Reception Operation



Notes 1. 78K0/KB2-A only.

2. 78K0/KB2-A : TI000/TOH0/INTP7/P12, output latch (P12), port mode (PM12)
78K0/KC2-A : TI000/P00, output latch (P00), port mode (PM00)

Remark ISC0, ISC1: Bits 0 and 1 of the input switch control register (ISC) (see **Figure 14-11**)

The peripheral functions used in the LIN communication operation are shown below.

<Peripheral functions used>

- External interrupt (INTP0); wakeup signal detection
  Use: Detects the wakeup signal edges and detects start of communication.
- 16-bit timer/event counter 00 (TI000); baud rate error detection
  Use: Detects the baud rate error (measures the TI000 input edge interval in the capture mode) by detecting the sync field (SF) length and divides it by the number of bits.
- Serial interface UART6

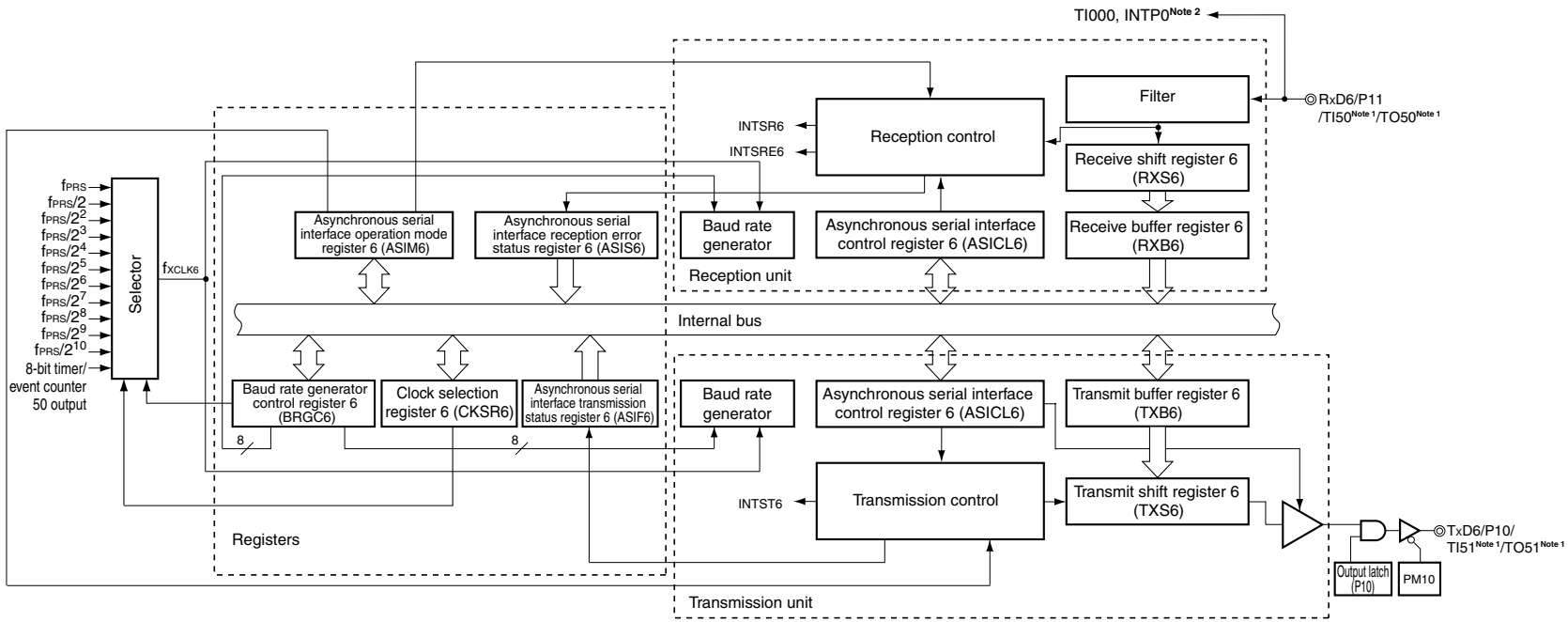
14.2 Configuration of Serial Interface UART6

Serial interface UART6 includes the following hardware.

Table 14-1. Configuration of Serial Interface UART6

Item	Configuration
Registers	Receive buffer register 6 (RXB6) Receive shift register 6 (RXS6) Transmit buffer register 6 (TXB6) Transmit shift register 6 (TXS6)
Control registers	Asynchronous serial interface operation mode register 6 (ASIM6) Asynchronous serial interface reception error status register 6 (ASIS6) Asynchronous serial interface transmission status register 6 (ASIF6) Clock selection register 6 (CKSR6) Baud rate generator control register 6 (BRGC6) Asynchronous serial interface control register 6 (ASICL6) Input switch control register (ISC) Port mode register 1 (PM1) Port register 1 (P1)

Figure 14-4. Block Diagram of Serial Interface UART6



Notes 1. 78K0/KB2-A only.

2. Selectable with input switch control register (ISC).

(1) Receive buffer register 6 (RXB6)

This 8-bit register stores parallel data converted by receive shift register 6 (RXS6).

Each time 1 byte of data has been received, new receive data is transferred to this register from RXS6. If the data length is set to 7 bits, data is transferred as follows.

- In LSB-first reception, the receive data is transferred to bits 0 to 6 of RXB6 and the MSB of RXB6 is always 0.
- In MSB-first reception, the receive data is transferred to bits 1 to 7 of RXB6 and the LSB of RXB6 is always 0.

If an overrun error (OVE6) occurs, the receive data is not transferred to RXB6.

RXB6 can be read by an 8-bit memory manipulation instruction. No data can be written to this register.

Reset signal generation sets this register to FFH.

(2) Receive shift register 6 (RXS6)

This register converts the serial data input to the RxD6 pin into parallel data.

RXS6 cannot be directly manipulated by a program.

(3) Transmit buffer register 6 (TXB6)

This buffer register is used to set transmit data. Transmission is started when data is written to TXB6.

This register can be read or written by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Cautions 1. Do not write data to TXB6 when bit 1 (TXBF6) of asynchronous serial interface transmission status register 6 (ASIF6) is 1.

2. Do not refresh (write the same value to) TXB6 by software during a communication operation (when bits 7 and 6 (POWER6, TXE6) of asynchronous serial interface operation mode register 6 (ASIM6) are 1 or when bits 7 and 5 (POWER6, RXE6) of ASIM6 are 1).

3. Set transmit data to TXB6 at least one base clock (f_{XCLK6}) after setting TXE6 = 1.

(4) Transmit shift register 6 (TXS6)

This register transmits the data transferred from TXB6 from the TxD6 pin as serial data. Data is transferred from TXB6 immediately after TXB6 is written for the first transmission, or immediately before INTST6 occurs after one frame was transmitted for continuous transmission. Data is transferred from TXB6 and transmitted from the TxD6 pin at the falling edge of the base clock.

TXS6 cannot be directly manipulated by a program.

14.3 Registers Controlling Serial Interface UART6

Serial interface UART6 is controlled by the following nine registers.

- Asynchronous serial interface operation mode register 6 (ASIM6)
- Asynchronous serial interface reception error status register 6 (ASIS6)
- Asynchronous serial interface transmission status register 6 (ASIF6)
- Clock selection register 6 (CKSR6)
- Baud rate generator control register 6 (BRGC6)
- Asynchronous serial interface control register 6 (ASICL6)
- Input switch control register (ISC)
- Port mode register 1 (PM1)
- Port register 1 (P1)

(1) Asynchronous serial interface operation mode register 6 (ASIM6)

This 8-bit register controls the serial communication operations of serial interface UART6.

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

Reset signal generation sets this register to 01H.

Remark ASIM6 can be refreshed (the same value is written) by software during a communication operation (when bits 7 and 6 (POWER6, TXE6) of ASIM6 = 1 or bits 7 and 5 (POWER6, RXE6) of ASIM6 = 1).

Figure 14-5. Format of Asynchronous Serial Interface Operation Mode Register 6 (ASIM6) (1/2)

Address: FF50H After reset: 01H R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
ASIM6	POWER6	TXE6	RXE6	PS61	PS60	CL6	SL6	ISRM6

POWER6	Enables/disables operation of internal operation clock
0[Note 1]	Disables operation of the internal operation clock (fixes the clock to low level) and asynchronously resets the internal circuit[Note 2].
1	Enables operation of the internal operation clock

TXE6	Enables/disables transmission
0	Disables transmission (synchronously resets the transmission circuit).
1	Enables transmission

RXE6	Enables/disables reception
0	Disables reception (synchronously resets the reception circuit).
1	Enables reception

Notes 1. If POWER6 = 0 is set while transmitting data, the output of the TxD6 pin will be fixed to high level (if TXDLV6 = 0). Furthermore, the input from the RxD6 pin will be fixed to high level.

2. Asynchronous serial interface reception error status register 6 (ASIS6), asynchronous serial interface transmission status register 6 (ASIF6), bit 7 (SBRF6) and bit 6 (SBRT6) of asynchronous serial interface control register 6 (ASICL6), and receive buffer register 6 (RXB6) are reset.

Figure 14-5. Format of Asynchronous Serial Interface Operation Mode Register 6 (ASIM6) (2/2)

PS61	PS60	Transmission operation	Reception operation
0	0	Does not output parity bit.	Reception without parity
0	1	Outputs 0 parity.	Reception as 0 parity[Note]
1	0	Outputs odd parity.	Judges as odd parity.
1	1	Outputs even parity.	Judges as even parity.

CL6	Specifies character length of transmit/receive data
0	Character length of data = 7 bits
1	Character length of data = 8 bits

SL6	Specifies number of stop bits of transmit data
0	Number of stop bits = 1
1	Number of stop bits = 2

ISRM6	Enables/disables occurrence of reception completion interrupt in case of error
0	“INTSRE6” occurs in case of error (at this time, INTSR6 does not occur).
1	“INTSR6” occurs in case of error (at this time, INTSRE6 does not occur).

Note If “reception as 0 parity” is selected, the parity is not judged. Therefore, bit 2 (PE6) of asynchronous serial interface reception error status register 6 (ASIS6) is not set and the error interrupt does not occur.

Cautions 1. To start the transmission, set POWER6 to 1 and then set TXE6 to 1. To stop the transmission, clear TXE6 to 0, and then clear POWER6 to 0.

2. To start the reception, set POWER6 to 1 and then set RXE6 to 1. To stop the reception, clear RXE6 to 0, and then clear POWER6 to 0.

3. Set POWER6 to 1 and then set RXE6 to 1 while a high level is input to the RxD6 pin. If POWER6 is set to 1 and RXE6 is set to 1 while a low level is input, reception is started.

4. TXE6 and RXE6 are synchronized by the base clock (f_{XCLK6}) set by CKSR6. To enable transmission or reception again, set TXE6 or RXE6 to 1 at least two clocks of the base clock after TXE6 or RXE6 has been cleared to 0. If TXE6 or RXE6 is set within two clocks of the base clock, the transmission circuit or reception circuit may not be initialized.

5. Set transmit data to TXB6 at least one base clock (f_{XCLK6}) after setting TXE6 = 1.

6. Clear the TXE6 and RXE6 bits to 0 before rewriting the PS61, PS60, and CL6 bits.

7. Fix the PS61 and PS60 bits to 0 when used in LIN communication operation.

8. Clear TXE6 to 0 before rewriting the SL6 bit. Reception is always performed with “the number of stop bits = 1”, and therefore, is not affected by the set value of the SL6 bit.

9. Make sure that RXE6 = 0 when rewriting the ISRM6 bit.

(2) Asynchronous serial interface reception error status register 6 (ASIS6)

This register indicates an error status on completion of reception by serial interface UART6. It includes three error flag bits (PE6, FE6, OVE6).

This register is read-only by an 8-bit memory manipulation instruction.

Reset signal generation, or clearing bit 7 (POWER6) or bit 5 (RXE6) of ASIM6 to 0 clears this register to 00H. 00H is read when this register is read. If a reception error occurs, read ASIS6 and then read receive buffer register 6 (RXB6) to clear the error flag.

Figure 14-6. Format of Asynchronous Serial Interface Reception Error Status Register 6 (ASIS6)

Address: FF53H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS6	0	0	0	0	0	PE6	FE6	OVE6

PE6	Status flag indicating parity error
0	If POWER6 = 0 or RXE6 = 0, or if ASIS6 register is read
1	If the parity of transmit data does not match the parity bit on completion of reception

FE6	Status flag indicating framing error
0	If POWER6 = 0 or RXE6 = 0, or if ASIS6 register is read
1	If the stop bit is not detected on completion of reception

OVE6	Status flag indicating overrun error
0	If POWER6 = 0 or RXE6 = 0, or if ASIS6 register is read
1	If receive data is set to the RXB6 register and the next reception operation is completed before the data is read.

Cautions 1. The operation of the PE6 bit differs depending on the set values of the PS61 and PS60 bits of asynchronous serial interface operation mode register 6 (ASIM6).

2. For the stop bit of the receive data, only the first stop bit is checked regardless of the number of stop bits.

3. If an overrun error occurs, the next receive data is not written to receive buffer register 6 (RXB6) but discarded.

4. If data is read from ASIS6, a wait cycle is generated. Do not read data from ASIS6 when the peripheral hardware clock (f_{PRS}) is stopped. For details, see CHAPTER 30 CAUTIONS FOR WAIT.

(3) Asynchronous serial interface transmission status register 6 (ASIF6)

This register indicates the status of transmission by serial interface UART6. It includes two status flag bits (TXBF6 and TXSF6).

Transmission can be continued without disruption even during an interrupt period, by writing the next data to the TXB6 register after data has been transferred from the TXB6 register to the TXS6 register.

This register is read-only by an 8-bit memory manipulation instruction.

Reset signal generation, or clearing bit 7 (POWER6) or bit 6 (TXE6) of ASIM6 to 0 clears this register to 00H.

Figure 14-7. Format of Asynchronous Serial Interface Transmission Status Register 6 (ASIF6)

Address: FF55H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIF6	0	0	0	0	0	0	TXBF6	TXSF6

TXBF6	Transmit buffer data flag
0	If POWER6 = 0 or TXE6 = 0, or if data is transferred to transmit shift register 6 (TXS6)
1	If data is written to transmit buffer register 6 (TXB6) (if data exists in TXB6)

TXSF6	Transmit shift register data flag
0	If POWER6 = 0 or TXE6 = 0, or if the next data is not transferred from transmit buffer register 6 (TXB6) after completion of transfer
1	If data is transferred from transmit buffer register 6 (TXB6) (if data transmission is in progress)

Cautions 1. To transmit data continuously, write the first transmit data (first byte) to the TXB6 register. Be sure to check that the TXBF6 flag is "0". If so, write the next transmit data (second byte) to the TXB6 register. If data is written to the TXB6 register while the TXBF6 flag is "1", the transmit data cannot be guaranteed.

2. To initialize the transmission unit upon completion of continuous transmission, be sure to check that the TXSF6 flag is "0" after generation of the transmission completion interrupt, and then execute initialization. If initialization is executed while the TXSF6 flag is "1", the transmit data cannot be guaranteed.

(4) Clock selection register 6 (CKSR6)

This register selects the base clock of serial interface UART6.

CKSR6 can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

Remark CKSR6 can be refreshed (the same value is written) by software during a communication operation (when bits 7 and 6 (POWER6, TXE6) of ASIM6 = 1 or bits 7 and 5 (POWER6, RXE6) of ASIM6 = 1).

Figure 14-8. Format of Clock Selection Register 6 (CKSR6)

Address: FF56H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CKSR6	0	0	0	0	TPS63	TPS62	TPS61	TPS60

TPS63	TPS62	TPS61	TPS60	Base clock (f_{XCLK6}) selection[Note 1]				
					f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	0	0	f_{PRS}[Note 2]	2 MHz	5 MHz	10 MHz	20 MHz[Note 3]
0	0	0	1	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz
0	0	1	0	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	$f_{PRS}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	$f_{PRS}/2^4$	125 kHz	312.5 kHz	625 kHz	1.25 MHz
0	1	0	1	$f_{PRS}/2^5$	62.5 kHz	156.25 kHz	312.5 kHz	625 kHz
0	1	1	0	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	156.25 kHz	312.5 kHz
0	1	1	1	$f_{PRS}/2^7$	15.625 kHz	39.06 kHz	78.13 kHz	156.25 kHz
1	0	0	0	$f_{PRS}/2^8$	7.813 kHz	19.53 kHz	39.06 kHz	78.13 kHz
1	0	0	1	$f_{PRS}/2^9$	3.906 kHz	9.77 kHz	19.53 kHz	39.06 kHz
1	0	1	0	$f_{PRS}/2^{10}$	1.953 kHz	4.88 kHz	9.77 kHz	19.53 kHz
1	0	1	1	TM50 output[Note 4]				
Other than above				Setting prohibited				

Notes 1. The frequency that can be used for the peripheral hardware clock (f_{PRS}) differs depending on the power supply voltage.

Supply Voltage	Use frequency range of peripheral hardware clock (f_{PRS})
$2.7\ V \le V_{DD} \le 5.5\ V$	$f_{PRS} \le 20$ MHz
$1.8\ V \le V_{DD} < 2.7\ V$	$f_{PRS} \le 5$ MHz

(The values shown in the table above are those when $f_{PRS} = f_{XH}$ (XSEL = 1).)

2. If the peripheral hardware clock (f_{PRS}) operates on the internal high-speed oscillation clock (f_{RH}) (XSEL = 0), when $1.8\ V \le V_{DD} < 2.7\ V$, the setting of TPS63 = TPS62 = TPS61 = TPS60 = 0 (base clock: f_{PRS}) is prohibited.

3. This is settable only if $4.0\ V \le V_{DD} \le 5.5\ V$.

4. Note the following points when selecting the TM50 output as the base clock.

- Mode in which the count clock is cleared and started upon a match of TM50 and CR50 (TMC506 = 0)

  Start the operation of 8-bit timer/event counter 50 first and then enable the timer F/F inversion operation (TMC501 = 1).
- PWM mode (TMC506 = 1)

  Start the operation of 8-bit timer/event counter 50 first and then set the count clock to make the duty = 50%.

It is not necessary to enable (TOE50 = 1) TO50 output in any mode.

Caution Make sure POWER6 = 0 when rewriting TPS63 to TPS60.

Remarks 1. f_{PRS}: Peripheral hardware clock frequency

2. TMC506: Bit 6 of 8-bit timer mode control register 50 (TMC50)

TMC501: Bit 1 of TMC50

(5) Baud rate generator control register 6 (BRGC6)

This register sets the division value of the 8-bit counter of serial interface UART6.

BRGC6 can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Remark BRGC6 can be refreshed (the same value is written) by software during a communication operation (when bits 7 and 6 (POWER6, TXE6) of ASIM6 = 1 or bits 7 and 5 (POWER6, RXE6) of ASIM6 = 1).

Figure 14-9. Format of Baud Rate Generator Control Register 6 (BRGC6)

Address: FF57H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
BRGC6	MDL67	MDL66	MDL65	MDL64	MDL63	MDL62	MDL61	MDL60

MDL67	MDL66	MDL65	MDL64	MDL63	MDL62	MDL61	MDL60	k	Output clock selection of 8-bit counter
0	0	0	0	0	0	×	×	×	Setting prohibited
0	0	0	0	0	1	0	0	4	$f_{XCLK6}/4$
0	0	0	0	0	1	0	1	5	$f_{XCLK6}/5$
0	0	0	0	0	1	1	0	6	$f_{XCLK6}/6$
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
1	1	1	1	1	1	0	0	252	$f_{XCLK6}/252$
1	1	1	1	1	1	0	1	253	$f_{XCLK6}/253$
1	1	1	1	1	1	1	0	254	$f_{XCLK6}/254$
1	1	1	1	1	1	1	1	255	$f_{XCLK6}/255$

Cautions 1. Make sure that bit 6 (TXE6) and bit 5 (RXE6) of the ASIM6 register = 0 when rewriting the MDL67 to MDL60 bits.

2. The baud rate is the output clock of the 8-bit counter divided by 2.

Remarks 1. f_{XCLK6}: Frequency of base clock selected by the TPS63 to TPS60 bits of CKSR6 register

2. k: Value set by MDL67 to MDL60 bits (k = 4, 5, 6, ..., 255)

3. ×: Don't care

(6) Asynchronous serial interface control register 6 (ASICL6)

This register controls the serial communication operations of serial interface UART6.

ASICL6 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 16H.

Caution ASICL6 can be refreshed (the same value is written) by software during a communication operation (when bits 7 and 6 (POWER6, TXE6) of ASIM6 = 1 or bits 7 and 5 (POWER6, RXE6) of ASIM6 = 1). However, do not set both SBRT6 and SBTT6 to 1 by a refresh operation during SBF reception (SBRT6 = 1) or SBF transmission (until INTST6 occurs since SBTT6 has been set (1)), because it may re-trigger SBF reception or SBF transmission.

Figure 14-10. Format of Asynchronous Serial Interface Control Register 6 (ASICL6) (1/2)

Address: FF58H After reset: 16H R/W[Note]

Symbol	<7>	<6>	5	4	3	2	1	0
ASICL6	SBRF6	SBRT6	SBTT6	SBL62	SBL61	SBL60	DIR6	TXDLV6

SBRF6	SBF reception status flag
0	If POWER6 = 0 and RXE6 = 0 or if SBF reception has been completed correctly
1	SBF reception in progress

SBRT6	SBF reception trigger
0	–
1	SBF reception trigger

SBTT6	SBF transmission trigger
0	–
1	SBF transmission trigger

Note Bit 7 is read-only.

Figure 14-10. Format of Asynchronous Serial Interface Control Register 6 (ASICL6) (2/2)

SBL62	SBL61	SBL60	SBF transmission output width control
1	0	1	SBF is output with 13-bit length.
1	1	0	SBF is output with 14-bit length.
1	1	1	SBF is output with 15-bit length.
0	0	0	SBF is output with 16-bit length.
0	0	1	SBF is output with 17-bit length.
0	1	0	SBF is output with 18-bit length.
0	1	1	SBF is output with 19-bit length.
1	0	0	SBF is output with 20-bit length.

DIR6	First-bit specification
0	MSB
1	LSB

TXDLV6	Enables/disables inverting TxD6 output
0	Normal output of TxD6
1	Inverted output of TxD6

Cautions 1. In the case of an SBF reception error, the mode returns to the SBF reception mode. The status of the SBRF6 flag is held (1).

2. Before setting the SBRT6 bit, make sure that bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 = 1. After setting the SBRT6 bit to 1, do not clear it to 0 before SBF reception is completed (before an interrupt request signal is generated).

3. The read value of the SBRT6 bit is always 0. SBRT6 is automatically cleared to 0 after SBF reception has been correctly completed.

4. Before setting the SBTT6 bit to 1, make sure that bit 7 (POWER6) and bit 6 (TXE6) of ASIM6 = 1. After setting the SBTT6 bit to 1, do not clear it to 0 before SBF transmission is completed (before an interrupt request signal is generated).

5. The read value of the SBTT6 bit is always 0. SBTT6 is automatically cleared to 0 at the end of SBF transmission.

6. Do not set the SBRT6 bit to 1 during reception, and do not set the SBTT6 bit to 1 during transmission.

7. Before rewriting the DIR6 and TXDLV6 bits, clear the TXE6 and RXE6 bits to 0.

8. When the TXDLV6 bit is set to 1 (inverted TxD6 output), the TxD6/P10/TI51[Note]/TO51[Note] pin cannot be used as a general-purpose port, regardless of the settings of POWER6 and TXE6. When using the TxD6/P10/TI51[Note]/TO51[Note] pin as a general-purpose port, clear the TXDLV6 bit to 0 (normal TxD6 output).

Note 78K0/KB2-A only.

(7) Input switch control register (ISC)

The input switch control register (ISC) is used to receive a status signal transmitted from the master during LIN (Local Interconnect Network) reception.

The signal input from the P11/RxD6/TI50[Note]/TO50[Note] pin is selected as the input source of INTP0 and TI000 when ISC0 and ISC1 are set to 1.

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

Reset signal generation sets this register to 00H.

Note 78K0/KB2-A only.

Figure 14-11. Format of Input Switch Control Register (ISC)

Address: FF4FH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	0	0	0	0	0	ISC2[Note]	ISC1	ISC0

ISC1	TI000 input source selection
0	TI000 (P12 : 78K0/KB2-A, P00 : 78K0/KC2-A
1	RxD6 (P11)

ISC0	INTP0 input source selection
0	INTP0 (P120)
1	RxD6 (P11)

Note The assignment of pins used in serial interface CSI10 is set by ISC2. For details, see **CHAPTER 15 SERIAL INTERFACE CSI10**.

(8) Port mode register 1 (PM1)

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

When using the P10/TxD6/TI51[Note]/TO51[Note] pin for serial interface data output, clear PM10 to 0 and set the output latch of P10 to 1.

When using the P11/RxD6/TI50[Note]/TO50[Note] pin for serial interface data input, set PM11 to 1. The output latch of P11 at this time may be 0 or 1.

PM1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Note 78K0/KB2-A only.

Figure 14-12. Format of Port Mode Register 1 (PM1)

Address: FF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	1	1	1	1	PM13	PM12	PM11	PM10

PM1n	P1n pin I/O mode selection (n = 0 to 3)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

14.4 Operation of Serial Interface UART6

Serial interface UART6 has the following two modes.

- Operation stop mode
- Asynchronous serial interface (UART) mode

14.4.1 Operation stop mode

In this mode, serial communication cannot be executed; therefore, the power consumption can be reduced. In addition, the pins can be used as ordinary port (or 8-bit timer 50, 51[Not]) pins in this mode. To set the operation stop mode, clear bits 7, 6, and 5 (POWER6, TXE6, and RXE6) of ASIM6 to 0.

Note 78K0/KB2-A only.

(1) Register used

The operation stop mode is set by asynchronous serial interface operation mode register 6 (ASIM6).

ASIM6 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 01H.

Address: FF50H After reset: 01H R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
ASIM6	POWER6	TXE6	RXE6	PS61	PS60	CL6	SL6	ISRM6

POWER6	Enables/disables operation of internal operation clock
0[Note 1]	Disables operation of the internal operation clock (fixes the clock to low level) and asynchronously resets the internal circuit[Note 2].

TXE6	Enables/disables transmission
0	Disables transmission operation (synchronously resets the transmission circuit).

RXE6	Enables/disables reception
0	Disables reception (synchronously resets the reception circuit).

Notes 1. If POWER6 = 0 is set while transmitting data, the output of the TxD6 pin will be fixed to high level (if TXDLV6 = 0). Furthermore, the input from the RxD6 pin will be fixed to high level.

2. Asynchronous serial interface reception error status register 6 (ASIS6), asynchronous serial interface transmission status register 6 (ASIF6), bit 7 (SBRF6) and bit 6 (SBRT6) of asynchronous serial interface control register 6 (ASICL6), and receive buffer register 6 (RXB6) are reset.

Caution Clear POWER6 to 0 after clearing TXE6 and RXE6 to 0 to stop the operation. To start the communication, set POWER6 to 1, and then set TXE6 or RXE6 to 1.

Remark To use the RxD6/P11 and TxD6/P10 pins as general-purpose port pins, see **CHAPTER 4 PORT FUNCTIONS**.

14.4.2 Asynchronous serial interface (UART) mode

In this mode, data of 1 byte is transmitted/received following a start bit, and a full-duplex operation can be performed.

A dedicated UART baud rate generator is incorporated, so that communication can be executed at a wide range of baud rates.

(1) Registers used

- Asynchronous serial interface operation mode register 6 (ASIM6)
- Asynchronous serial interface reception error status register 6 (ASIS6)
- Asynchronous serial interface transmission status register 6 (ASIF6)
- Clock selection register 6 (CKSR6)
- Baud rate generator control register 6 (BRGC6)
- Asynchronous serial interface control register 6 (ASICL6)
- Input switch control register (ISC)
- Port mode register 1 (PM1)
- Port register 1 (P1)

The basic procedure of setting an operation in the UART mode is as follows.

<1> Set the CKSR6 register (see **Figure 14-8**).
<2> Set the BRGC6 register (see **Figure 14-9**).
<3> Set bits 0 to 4 (ISRM6, SL6, CL6, PS60, PS61) of the ASIM6 register (see **Figure 14-5**).
<4> Set bits 0 and 1 (TXDLV6, DIR6) of the ASICL6 register (see **Figure 14-10**).
<5> Set bit 7 (POWER6) of the ASIM6 register to 1.
<6> Set bit 6 (TXE6) of the ASIM6 register to 1. → Transmission is enabled.
Set bit 5 (RXE6) of the ASIM6 register to 1. → Reception is enabled.
<7> Write data to transmit buffer register 6 (TXB6). → Data transmission is started.

Caution Take relationship with the other party of communication when setting the port mode register and port register.

The relationship between the register settings and pins is shown below.

Table 14-2. Relationship Between Register Settings and Pins

POWER6	TXE6	RXE6	PM10	P10	PM11	P11	UART6 Operation	Pin Function	
								TxD6/P10 /TI51[Note 2]/TO51[Note 2]	RxD6/P12 /TI50[Note 2]/TO50[Note 2]
0	0	0	×[Note 1]	×[Note 1]	×[Note 1]	×[Note 1]	Stop	P10 /TI51[Note 2]/TO51[Note 2]	P11 /TI50[Note 2]/TO50[Note 2]
1	0	1	×[Note 1]	×[Note 1]	1	×	Reception	P10 /TI51[Note 2]/TO51[Note 2]	RxD6
	1	0	0	1	×[Note 1]	×[Note 1]	Transmission	TxD6	P11 /TI50[Note2]TO50[Note 2]
	1	1	0	1	1	×	Transmission/ reception	TxD6	RxD6

Notes 1. Can be set as port function or 8-bit timer 50 and 51[Note 2].
2. 78K0/KB2-A only.

Remark ×: don't care
POWER6: Bit 7 of asynchronous serial interface operation mode register 6 (ASIM6)
TXE6: Bit 6 of ASIM6
RXE6: Bit 5 of ASIM6
PM1×: Port mode register
P1×: Port output latch

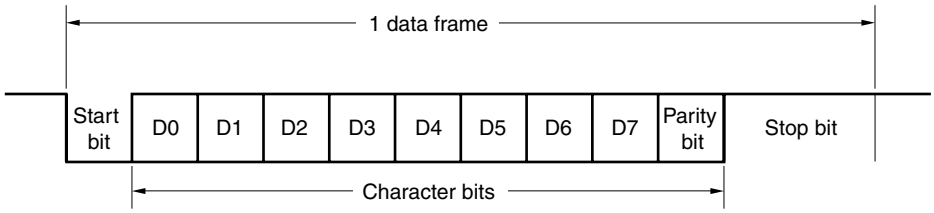
(2) Communication operation

(a) Format and waveform example of normal transmit/receive data

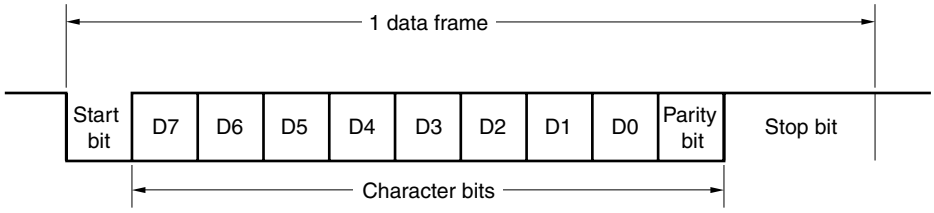
Figures 14-13 and 14-14 shows the format and waveform example of the normal transmit/receive data.

Figure 14-13. Format of Normal UART Transmit/Receive Data

1. LSB-first transmission/reception



2. MSB-first transmission/reception



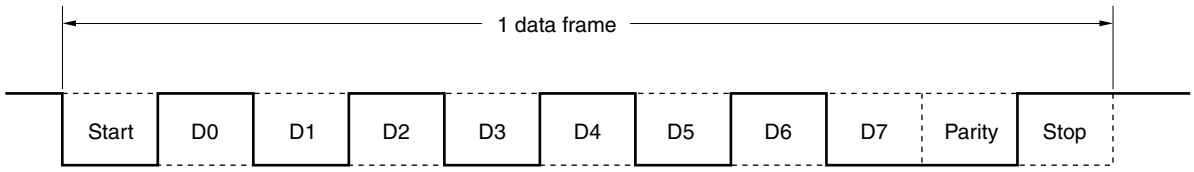
One data frame consists of the following bits.

- Start bit ... 1 bit
- Character bits ... 7 or 8 bits
- Parity bit ... Even parity, odd parity, 0 parity, or no parity
- Stop bit ... 1 or 2 bits

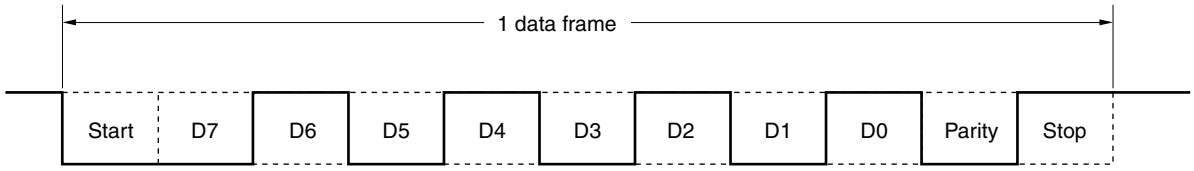
The character bit length, parity, and stop bit length in one data frame are specified by asynchronous serial interface operation mode register 6 (ASIM6).
Whether data is communicated with the LSB or MSB first is specified by bit 1 (DIR6) of asynchronous serial interface control register 6 (ASICL6).
Whether the TxD6 pin outputs normal or inverted data is specified by bit 0 (TXDLV6) of ASICL6.

Figure 14-14. Example of Normal UART Transmit/Receive Data Waveform

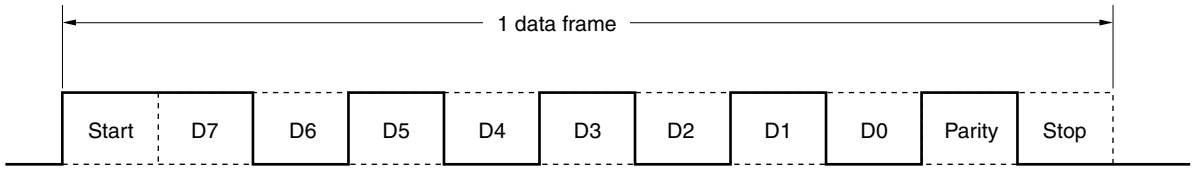
1. Data length: 8 bits, LSB first, Parity: Even parity, Stop bit: 1 bit, Communication data: 55H



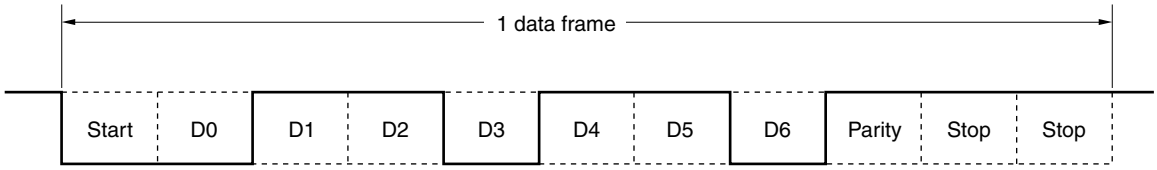
2. Data length: 8 bits, MSB first, Parity: Even parity, Stop bit: 1 bit, Communication data: 55H



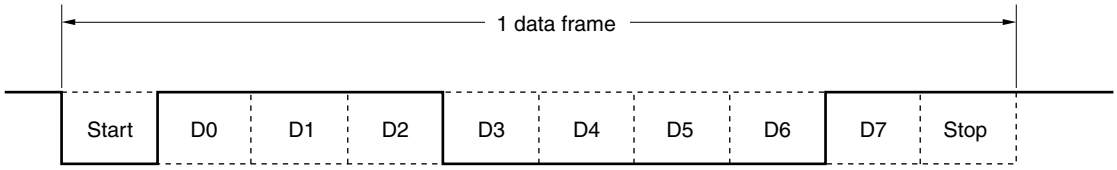
3. Data length: 8 bits, MSB first, Parity: Even parity, Stop bit: 1 bit, Communication data: 55H, TxD6 pin inverted output



4. Data length: 7 bits, LSB first, Parity: Odd parity, Stop bit: 2 bits, Communication data: 36H



5. Data length: 8 bits, LSB first, Parity: None, Stop bit: 1 bit, Communication data: 87H



(b) Parity types and operation

The parity bit is used to detect a bit error in communication data. Usually, the same type of parity bit is used on both the transmission and reception sides. With even parity and odd parity, a 1-bit (odd number) error can be detected. With zero parity and no parity, an error cannot be detected.

Caution Fix the PS61 and PS60 bits to 0 when the device is used in LIN communication operation.

(i) Even parity

- Transmission

  Transmit data, including the parity bit, is controlled so that the number of bits that are “1” is even.
  The value of the parity bit is as follows.

  If transmit data has an odd number of bits that are “1”: 1
  If transmit data has an even number of bits that are “1”: 0

- Reception

  The number of bits that are “1” in the receive data, including the parity bit, is counted. If it is odd, a parity error occurs.

(ii) Odd parity

- Transmission

  Unlike even parity, transmit data, including the parity bit, is controlled so that the number of bits that are “1” is odd.

  If transmit data has an odd number of bits that are “1”: 0
  If transmit data has an even number of bits that are “1”: 1

- Reception

  The number of bits that are “1” in the receive data, including the parity bit, is counted. If it is even, a parity error occurs.

(iii) 0 parity

The parity bit is cleared to 0 when data is transmitted, regardless of the transmit data.
The parity bit is not detected when the data is received. Therefore, a parity error does not occur regardless of whether the parity bit is “0” or “1”.

(iv) No parity

No parity bit is appended to the transmit data.
Reception is performed assuming that there is no parity bit when data is received. Because there is no parity bit, a parity error does not occur.

(c) Normal transmission

When bit 7 (POWER6) of asynchronous serial interface operation mode register 6 (ASIM6) is set to 1 and bit 6 (TXE6) of ASIM6 is then set to 1, transmission is enabled. Transmission can be started by writing transmit data to transmit buffer register 6 (TXB6). The start bit, parity bit, and stop bit are automatically appended to the data.

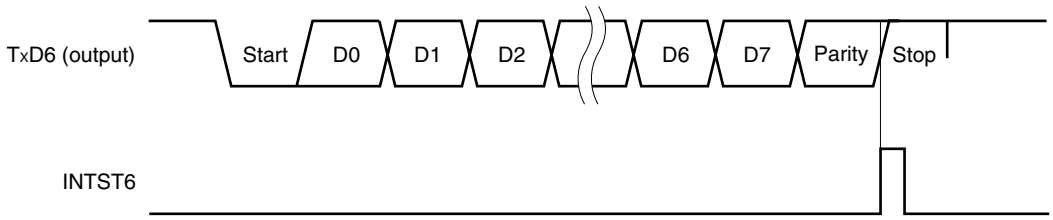
When transmission is started, the data in TXB6 is transferred to transmit shift register 6 (TXS6). After that, the transmit data is sequentially output from TXS6 to the TxD6 pin. When transmission is completed, the parity and stop bits set by ASIM6 are appended and a transmission completion interrupt request (INTST6) is generated.

Transmission is stopped until the data to be transmitted next is written to TXB6.

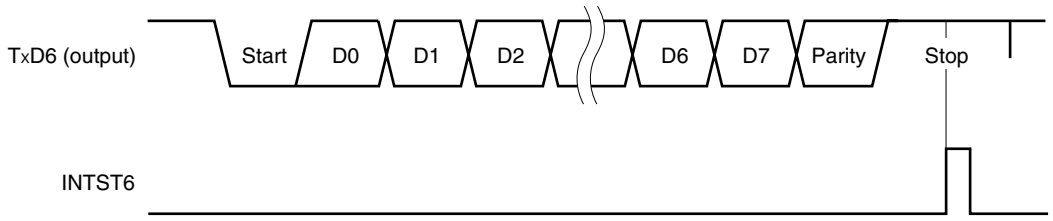
Figure 14-15 shows the timing of the transmission completion interrupt request (INTST6). This interrupt occurs as soon as the last stop bit has been output.

Figure 14-15. Normal Transmission Completion Interrupt Request Timing

1. Stop bit length: 1



2. Stop bit length: 2



(d) Continuous transmission

The next transmit data can be written to transmit buffer register 6 (TXB6) as soon as transmit shift register 6 (TXS6) has started its shift operation. Consequently, even while the INTST6 interrupt is being serviced after transmission of one data frame, data can be continuously transmitted and an efficient communication rate can be realized. In addition, the TXB6 register can be efficiently written twice (2 bytes) without having to wait for the transmission time of one data frame, by reading bit 0 (TXSF6) of asynchronous serial interface transmission status register 6 (ASIF6) when the transmission completion interrupt has occurred.

To transmit data continuously, be sure to reference the ASIF6 register to check the transmission status and whether the TXB6 register can be written, and then write the data.

Cautions 1. The TXBF6 and TXSF6 flags of the ASIF6 register change from "10" to "11", and to "01" during continuous transmission. To check the status, therefore, do not use a combination of the TXBF6 and TXSF6 flags for judgment. Read only the TXBF6 flag when executing continuous transmission.

2. When the device is use in LIN communication operation, the continuous transmission function cannot be used. Make sure that asynchronous serial interface transmission status register 6 (ASIF6) is 00H before writing transmit data to transmit buffer register 6 (TXB6).

TXBF6	Writing to TXB6 Register
0	Writing enabled
1	Writing disabled

Caution To transmit data continuously, write the first transmit data (first byte) to the TXB6 register. Be sure to check that the TXBF6 flag is "0". If so, write the next transmit data (second byte) to the TXB6 register. If data is written to the TXB6 register while the TXBF6 flag is "1", the transmit data cannot be guaranteed.

The communication status can be checked using the TXSF6 flag.

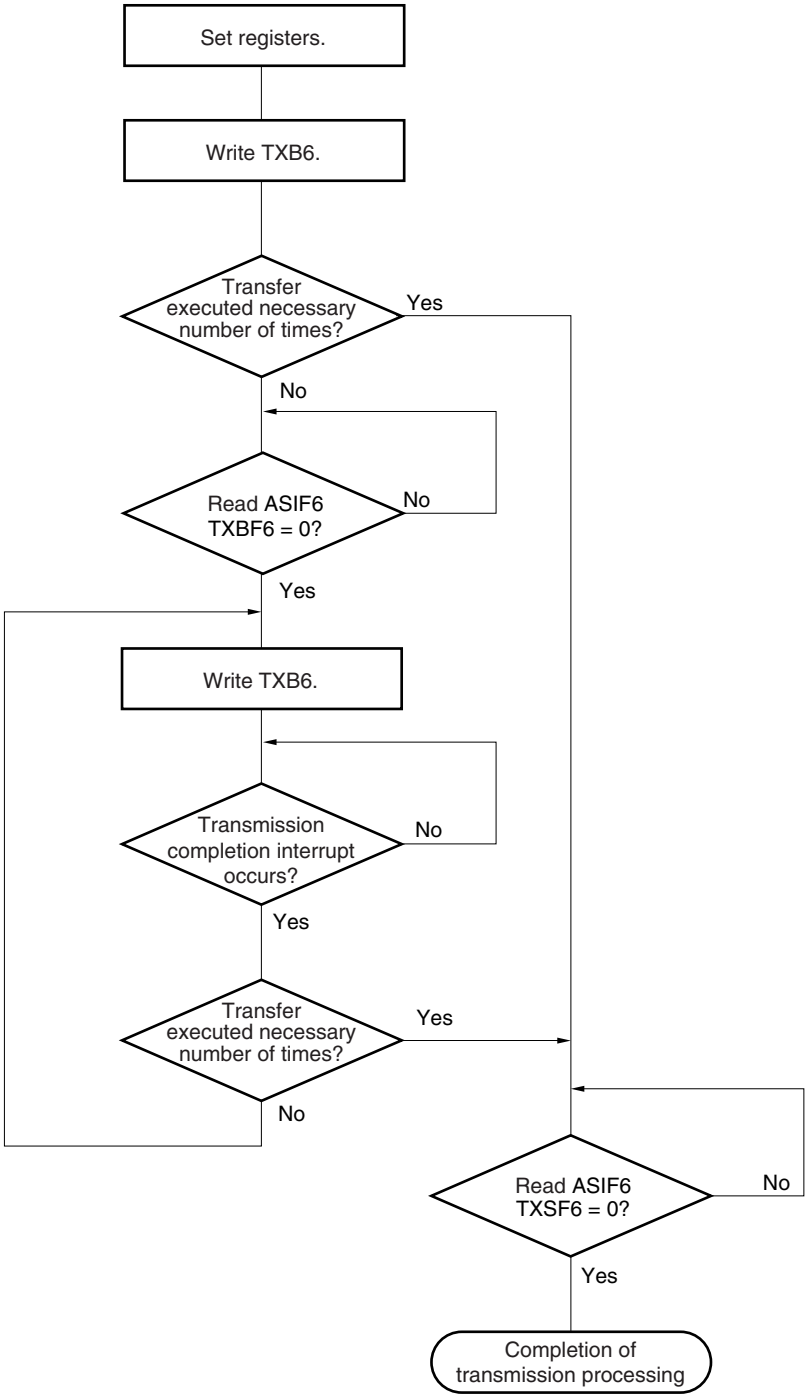
TXSF6	Transmission Status
0	Transmission is completed.
1	Transmission is in progress.

Cautions 1. To initialize the transmission unit upon completion of continuous transmission, be sure to check that the TXSF6 flag is "0" after generation of the transmission completion interrupt, and then execute initialization. If initialization is executed while the TXSF6 flag is "1", the transmit data cannot be guaranteed.

2. During continuous transmission, the next transmission may complete before execution of INTST6 interrupt servicing after transmission of one data frame. As a countermeasure, detection can be performed by developing a program that can count the number of transmit data and by referencing the TXSF6 flag.

Figure 14-16 shows an example of the continuous transmission processing flow.

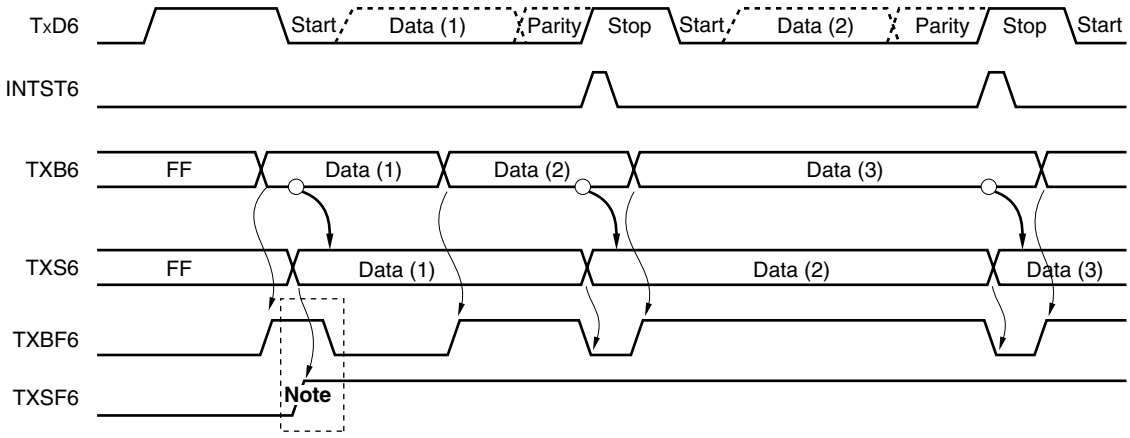
Figure 14-16. Example of Continuous Transmission Processing Flow



Remark TXB6: Transmit buffer register 6
ASIF6: Asynchronous serial interface transmission status register 6
TXBF6: Bit 1 of ASIF6 (transmit buffer data flag)
TXSF6: Bit 0 of ASIF6 (transmit shift register data flag)

Figure 14-17 shows the timing of starting continuous transmission, and Figure 14-18 shows the timing of ending continuous transmission.

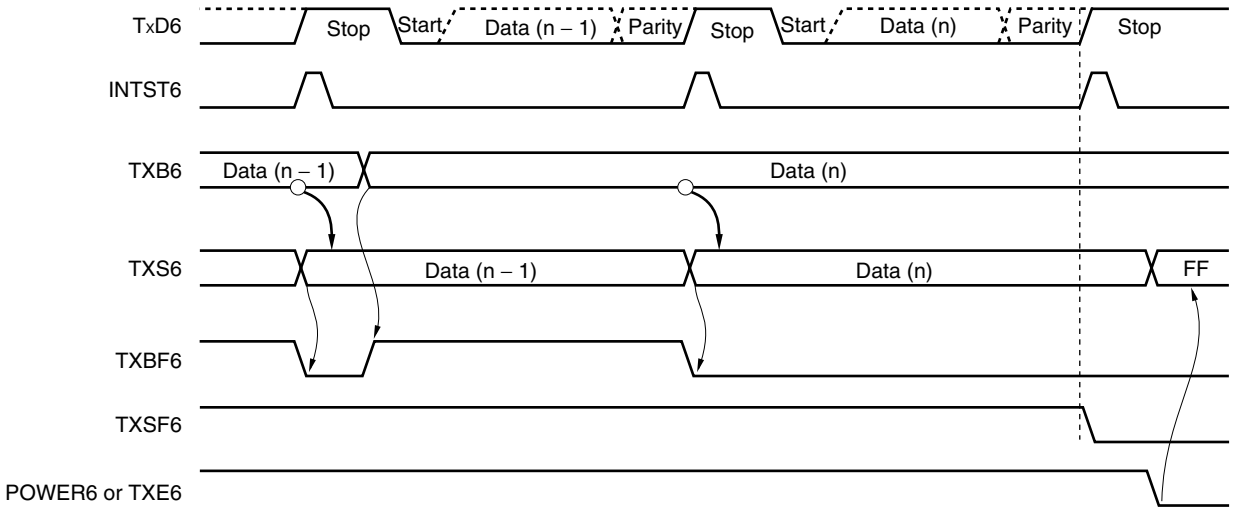
Figure 14-17. Timing of Starting Continuous Transmission



Note When ASIF6 is read, there is a period in which TXBF6 and TXSF6 = 1, 1. Therefore, judge whether writing is enabled using only the TXBF6 bit.

Remark TxD6: TxD6 pin (output)
INTST6: Interrupt request signal
TXB6: Transmit buffer register 6
TXS6: Transmit shift register 6
ASIF6: Asynchronous serial interface transmission status register 6
TXBF6: Bit 1 of ASIF6
TXSF6: Bit 0 of ASIF6

Figure 14-18. Timing of Ending Continuous Transmission



Remark TxD6: TxD6 pin (output)
INTST6: Interrupt request signal
TXB6: Transmit buffer register 6
TXS6: Transmit shift register 6
ASIF6: Asynchronous serial interface transmission status register 6
TXBF6: Bit 1 of ASIF6
TXSF6: Bit 0 of ASIF6
POWER6: Bit 7 of asynchronous serial interface operation mode register (ASIM6)
TXE6: Bit 6 of asynchronous serial interface operation mode register (ASIM6)

(e) Normal reception

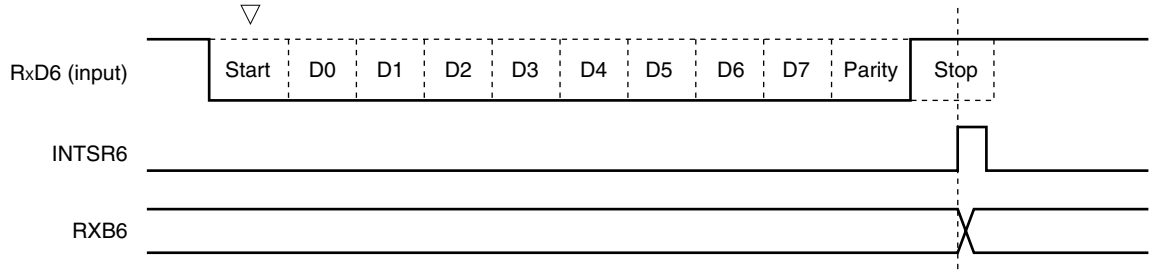
Reception is enabled and the RxD6 pin input is sampled when bit 7 (POWER6) of asynchronous serial interface operation mode register 6 (ASIM6) is set to 1 and then bit 5 (RXE6) of ASIM6 is set to 1.

The 8-bit counter of the baud rate generator starts counting when the falling edge of the RxD6 pin input is detected. When the set value of baud rate generator control register 6 (BRGC6) has been counted, the RxD6 pin input is sampled again (▽ in Figure 14-19). If the RxD6 pin is low level at this time, it is recognized as a start bit.

When the start bit is detected, reception is started, and serial data is sequentially stored in the receive shift register (RXS6) at the set baud rate. When the stop bit has been received, the reception completion interrupt (INTSR6) is generated and the data of RXS6 is written to receive buffer register 6 (RXB6). If an overrun error (OVE6) occurs, however, the receive data is not written to RXB6.

Even if a parity error (PE6) occurs while reception is in progress, reception continues to the reception position of the stop bit, and a reception error interrupt (INTSR6/INTSRE6) is generated on completion of reception.

Figure 14-19. Reception Completion Interrupt Request Timing



Cautions 1. If a reception error occurs, read ASIS6 and then RXB6 to clear the error flag. Otherwise, an overrun error will occur when the next data is received, and the reception error status will persist.

2. Reception is always performed with the "number of stop bits = 1". The second stop bit is ignored.

3. Be sure to read asynchronous serial interface reception error status register 6 (ASIS6) before reading RXB6.

(f) Reception error

Three types of errors may occur during reception: a parity error, framing error, or overrun error. If the error flag of asynchronous serial interface reception error status register 6 (ASIS6) is set as a result of data reception, a reception error interrupt request (INTSR6/INTSRE6) is generated.

Which error has occurred during reception can be identified by reading the contents of ASIS6 in the reception error interrupt (INTSR6/INTSRE6) servicing (see **Figure 14-6**).

The contents of ASIS6 are cleared to 0 when ASIS6 is read.

Table 14-3. Cause of Reception Error

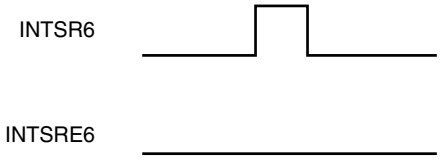
Reception Error	Cause
Parity error	The parity specified for transmission does not match the parity of the receive data.
Framing error	Stop bit is not detected.
Overrun error	Reception of the next data is completed before data is read from receive buffer register 6 (RXB6).

The reception error interrupt can be separated into reception completion interrupt (INTSR6) and error interrupt (INTSRE6) by clearing bit 0 (ISRM6) of asynchronous serial interface operation mode register 6 (ASIM6) to 0.

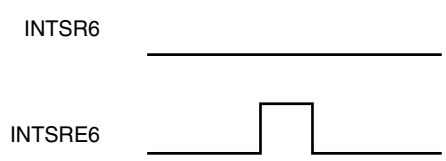
Figure 14-20. Reception Error Interrupt

1. If ISRM6 is cleared to 0 (reception completion interrupt (INTSR6) and error interrupt (INTSRE6) are separated)

(a) No error during reception

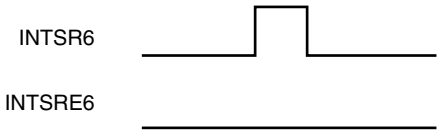


(b) Error during reception

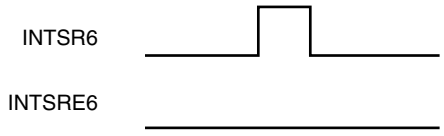


2. If ISRM6 is set to 1 (error interrupt is included in INTSR6)

(a) No error during reception



(b) Error during reception



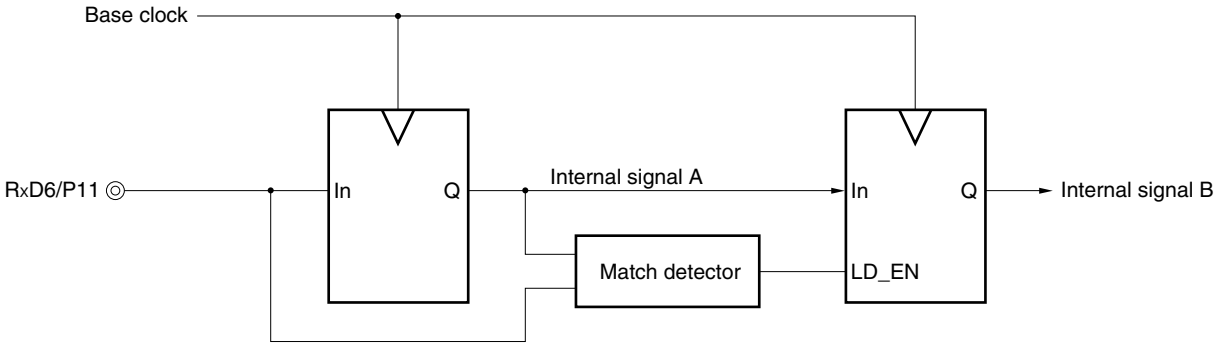
(g) Noise filter of receive data

The RxD6 signal is sampled with the base clock output by the prescaler block.

If two sampled values are the same, the output of the match detector changes, and the data is sampled as input data.

Because the circuit is configured as shown in Figure 14-21, the internal processing of the reception operation is delayed by two clocks from the external signal status.

Figure 14-21. Noise Filter Circuit



(h) SBF transmission

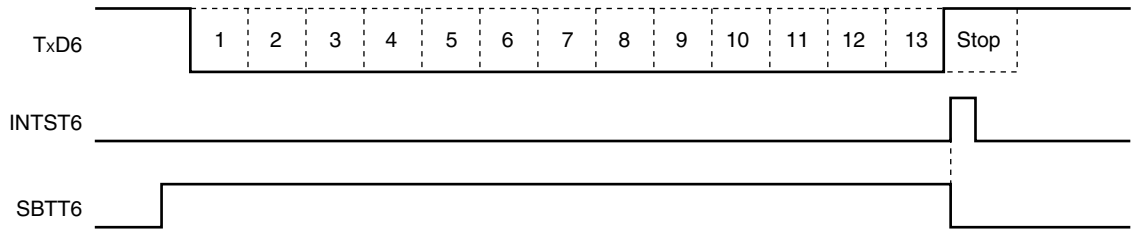
When the device is use in LIN communication operation, the SBF (Synchronous Break Field) transmission control function is used for transmission. For the transmission operation of LIN, see **Figure 14-1 LIN Transmission Operation**.

When bit 7 (POWER6) of asynchronous serial interface mode register 6 (ASIM6) is set to 1, the TxD6 pin outputs high level. Next, when bit 6 (TXE6) of ASIM6 is set to 1, the transmission enabled status is entered, and SBF transmission is started by setting bit 5 (SBTT6) of asynchronous serial interface control register 6 (ASICL6) to 1.

Thereafter, a low level of bits 13 to 20 (set by bits 4 to 2 (SBL62 to SBL60) of ASICL6) is output. Following the end of SBF transmission, the transmission completion interrupt request (INTST6) is generated and SBTT6 is automatically cleared. Thereafter, the normal transmission mode is restored.

Transmission is suspended until the data to be transmitted next is written to transmit buffer register 6 (TXB6), or until SBTT6 is set to 1.

Figure 14-22. SBF Transmission



Remark TxD6: TxD6 pin (output)
INTST6: Transmission completion interrupt request
SBTT6: Bit 5 of asynchronous serial interface control register 6 (ASICL6)

(i) SBF reception

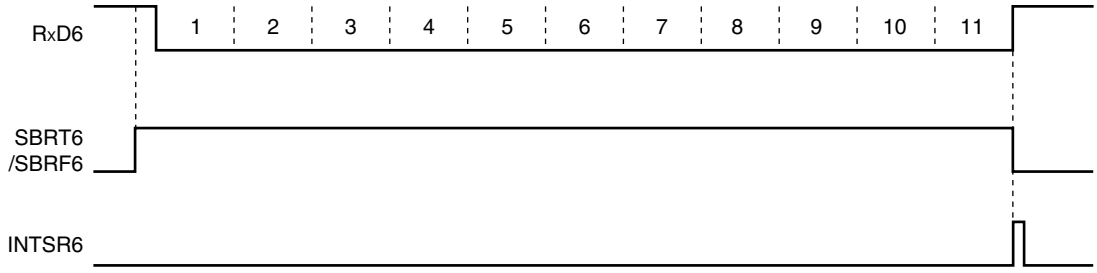
When the device is used in LIN communication operation, the SBF (Synchronous Break Field) reception control function is used for reception. For the reception operation of LIN, see **Figure 14-2 LIN Reception Operation**.

Reception is enabled when bit 7 (POWER6) of asynchronous serial interface operation mode register 6 (ASIM6) is set to 1 and then bit 5 (RXE6) of ASIM6 is set to 1. SBF reception is enabled when bit 6 (SBRT6) of asynchronous serial interface control register 6 (ASICL6) is set to 1. In the SBF reception enabled status, the RxD6 pin is sampled and the start bit is detected in the same manner as the normal reception enable status.

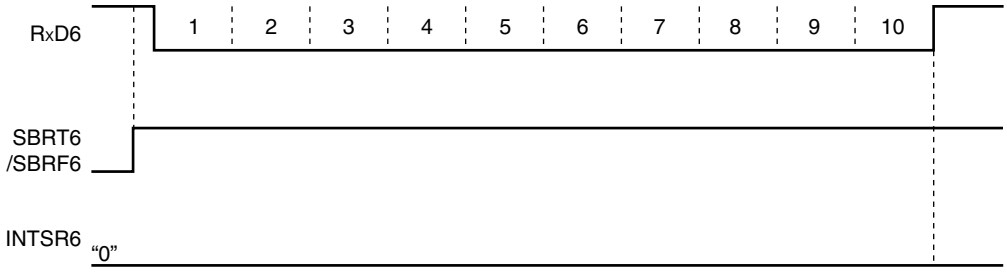
When the start bit has been detected, reception is started, and serial data is sequentially stored in the receive shift register 6 (RXS6) at the set baud rate. When the stop bit is received and if the width of SBF is 11 bits or more, a reception completion interrupt request (INTSR6) is generated as normal processing. At this time, the SBRF6 and SBRT6 bits are automatically cleared, and SBF reception ends. Detection of errors, such as OVE6, PE6, and FE6 (bits 0 to 2 of asynchronous serial interface reception error status register 6 (ASIS6)) is suppressed, and error detection processing of UART communication is not performed. In addition, data transfer between receive shift register 6 (RXS6) and receive buffer register 6 (RXB6) is not performed, and the reset value of FFH is retained. If the width of SBF is 10 bits or less, an interrupt does not occur as error processing after the stop bit has been received, and the SBF reception mode is restored. In this case, the SBRF6 and SBRT6 bits are not cleared.

Figure 14-23. SBF Reception

1. Normal SBF reception (stop bit is detected with a width of more than 10.5 bits)



2. SBF reception error (stop bit is detected with a width of 10.5 bits or less)



Remark RxD6: RxD6 pin (input)
SBRT6: Bit 6 of asynchronous serial interface control register 6 (ASICL6)
SBRF6: Bit 7 of ASICL6
INTSR6: Reception completion interrupt request

14.4.3 Dedicated baud rate generator

The dedicated baud rate generator consists of a source clock selector and an 8-bit programmable counter, and generates a serial clock for transmission/reception of UART6.

Separate 8-bit counters are provided for transmission and reception.

(1) Configuration of baud rate generator

- Base clock

  The clock selected by bits 3 to 0 (TPS63 to TPS60) of clock selection register 6 (CKSR6) is supplied to each module when bit 7 (POWER6) of asynchronous serial interface operation mode register 6 (ASIM6) is 1. This clock is called the base clock and its frequency is called f_{XCLK6}. The base clock is fixed to low level when POWER6 = 0.

- Transmission counter

  This counter stops operation, cleared to 0, when bit 7 (POWER6) or bit 6 (TXE6) of asynchronous serial interface operation mode register 6 (ASIM6) is 0.

  It starts counting when POWER6 = 1 and TXE6 = 1.

  The counter is cleared to 0 when the first data transmitted is written to transmit buffer register 6 (TXB6).

  If data are continuously transmitted, the counter is cleared to 0 again when one frame of data has been completely transmitted. If there is no data to be transmitted next, the counter is not cleared to 0 and continues counting until POWER6 or TXE6 is cleared to 0.

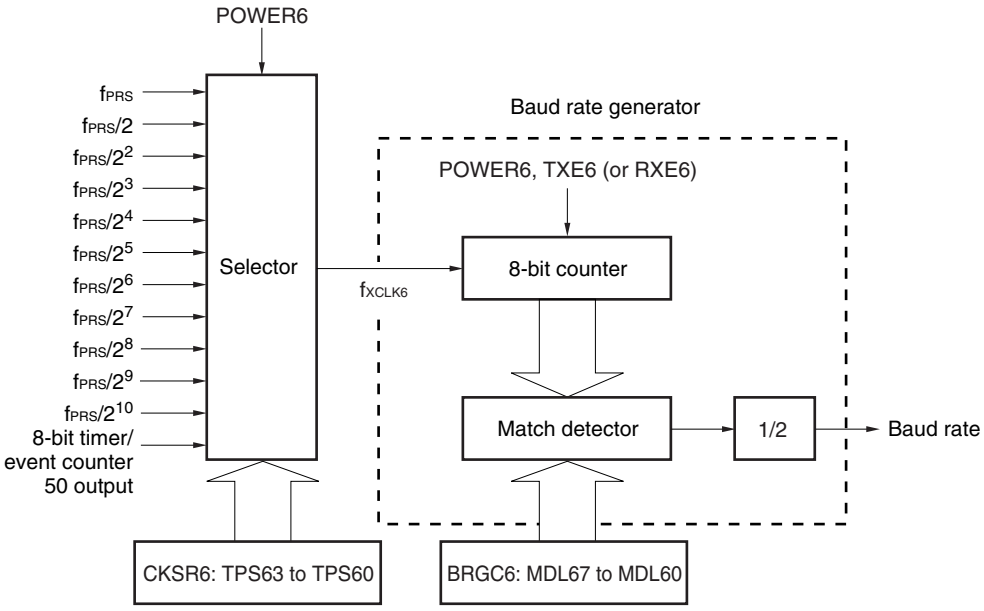
- Reception counter

  This counter stops operation, cleared to 0, when bit 7 (POWER6) or bit 5 (RXE6) of asynchronous serial interface operation mode register 6 (ASIM6) is 0.

  It starts counting when the start bit has been detected.

  The counter stops operation after one frame has been received, until the next start bit is detected.

Figure 14-24. Configuration of Baud Rate Generator



Remark POWER6: Bit 7 of asynchronous serial interface operation mode register 6 (ASIM6)
TXE6: Bit 6 of ASIM6
RXE6: Bit 5 of ASIM6
CKSR6: Clock selection register 6
BRGC6: Baud rate generator control register 6

(2) Generation of serial clock

A serial clock to be generated can be specified by using clock selection register 6 (CKSR6) and baud rate generator control register 6 (BRGC6).

The clock to be input to the 8-bit counter can be set by bits 3 to 0 (TPS63 to TPS60) of CKSR6 and the division value ($f_{XCLK6}/4$ to $f_{XCLK6}/255$) of the 8-bit counter can be set by bits 7 to 0 (MDL67 to MDL60) of BRGC6.

14.4.4 Calculation of baud rate

(1) Baud rate calculation expression

The baud rate can be calculated by the following expression.

- Baud rate = $\frac{f_{XCLK6}}{2 \times k}$ [bps]

f_{XCLK6}: Frequency of base clock selected by TPS63 to TPS60 bits of CKSR6 register
k: Value set by MDL67 to MDL60 bits of BRGC6 register (k = 4, 5, 6, ..., 255)

Table 14-4. Set Value of TPS63 to TPS60

TPS63	TPS62	TPS61	TPS60	Base Clock (f_{XCLK6}) Selection[Note 1]				
					f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	0	0	f_{PRS}[Note 2]	2 MHz	5 MHz	10 MHz	20 MHz[Note 3]
0	0	0	1	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz
0	0	1	0	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	$f_{PRS}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	$f_{PRS}/2^4$	125 kHz	312.5 kHz	625 kHz	1.25 MHz
0	1	0	1	$f_{PRS}/2^5$	62.5 kHz	156.25 kHz	312.5 kHz	625 kHz
0	1	1	0	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	156.25 kHz	312.5 kHz
0	1	1	1	$f_{PRS}/2^7$	15.625 kHz	39.06 kHz	78.13 kHz	156.25 kHz
1	0	0	0	$f_{PRS}/2^8$	7.813 kHz	19.53 kHz	39.06 kHz	78.13 kHz
1	0	0	1	$f_{PRS}/2^9$	3.906 kHz	9.77 kHz	19.53 kHz	39.06 kHz
1	0	1	0	$f_{PRS}/2^{10}$	1.953 kHz	4.88 kHz	9.77 kHz	19.53 kHz
1	0	1	1	TM50 output[Note 4]				
Other than above				Setting prohibited				

Notes 1. The frequency that can be used for the peripheral hardware clock (f_{PRS}) differs depending on the power supply voltage.

Supply Voltage	Use frequency range of peripheral hardware clock (f_{PRS})
$2.7\ V \le V_{DD} \le 5.5\ V$	$f_{PRS} \le 20$ MHz
$1.8\ V \le V_{DD} < 2.7\ V$	$f_{PRS} \le 5$ MHz

(The values shown in the table above are those when $f_{PRS} = f_{XH}$ (XSEL = 1).)

2. If the peripheral hardware clock (f_{PRS}) operates on the internal high-speed oscillation clock (f_{RH}) (XSEL = 0), when $1.8\ V \le V_{DD} < 2.7\ V$, the setting of TPS63 = TPS62 = TPS61 = TPS60 = 0 (base clock: f_{PRS}) is prohibited.

3. This is settable only if $4.0\ V \le V_{DD} \le 5.5\ V$.

4. Note the following points when selecting the TM50 output as the base clock.

- Mode in which the count clock is cleared and started upon a match of TM50 and CR50 (TMC506 = 0)
  Start the operation of 8-bit timer/event counter 50 first and then enable the timer F/F inversion operation (TMC501 = 1).
- PWM mode (TMC506 = 1)
  Start the operation of 8-bit timer/event counter 50 first and then set the count clock to make the duty = 50%.

It is not necessary to enable (TOE50 = 1) TO50 output in any mode.

(2) Error of baud rate

The baud rate error can be calculated by the following expression.

- $\text{Error (\%)} = \left(\frac{\text{Actual baud rate (baud rate with error)}}{\text{Desired baud rate (correct baud rate)}} - 1 \right) \times 100 \text{ [\%]}$

Cautions 1. Keep the baud rate error during transmission to within the permissible error range at the reception destination.

2. Make sure that the baud rate error during reception satisfies the range shown in (4) Permissible baud rate range during reception.

Example: Frequency of base clock = 10 MHz = 10,000,000 Hz
Set value of MDL67 to MDL60 bits of BRGC6 register = 00100001B (k = 33)
Target baud rate = 153600 bps

Baud rate = 10 M / (2 × 33)
= 10000000 / (2 × 33) = 151,515 [bps]

Error = (151515/153600 − 1) × 100
= −1.357 [%]

(3) Example of setting baud rate

Table 14-5. Set Data of Baud Rate Generator

Baud Rate [bps]	f_{PRS} = 2.0 MHz				f_{PRS} = 5.0 MHz				f_{PRS} = 10.0 MHz				f_{PRS} = 20.0 MHz			
	TPS63-TPS60	k	Calculated Value	ERR [%]	TPS63-TPS60	k	Calculated Value	ERR [%]	TPS63-TPS60	k	Calculated Value	ERR [%]	TPS63-TPS60	k	Calculated Value	ERR [%]
300	8H	13	301	0.16	7H	65	301	0.16	8H	65	301	0.16	9H	65	301	0.16
600	7H	13	601	0.16	6H	65	601	0.16	7H	65	601	0.16	8H	65	601	0.16
1200	6H	13	1202	0.16	5H	65	1202	0.16	6H	65	1202	0.16	7H	65	1202	0.16
2400	5H	13	2404	0.16	4H	65	2404	0.16	5H	65	2404	0.16	6H	65	2404	0.16
4800	4H	13	4808	0.16	3H	65	4808	0.16	4H	65	4808	0.16	5H	65	4808	0.16
9600	3H	13	9615	0.16	2H	65	9615	0.16	3H	65	9615	0.16	4H	65	9615	0.16
19200	2H	13	19231	0.16	1H	65	19231	0.16	2H	65	19231	0.16	3H	65	19231	0.16
24000	1H	21	23810	−0.79	3H	13	24038	0.16	4H	13	24038	0.16	5H	13	24038	0.16
31250	1H	16	31250	0	4H	5	31250	0	5H	5	31250	0	6H	5	31250	0
38400	1H	13	38462	0.16	0H	65	38462	0.16	1H	65	38462	0.16	2H	65	38462	0.16
48000	0H	21	47619	−0.79	2H	13	48077	0.16	3H	13	48077	0.16	4H	13	48077	0.16
76800	0H	13	76923	0.16	0H	33	75758	−1.36	0H	65	76923	0.16	1H	65	76923	0.16
115200	0H	9	111111	−3.55	1H	11	113636	−1.36	0H	43	116279	0.94	0H	87	114943	−0.22
153600	−	−	−	−	1H	8	156250	1.73	0H	33	151515	−1.36	1H	33	151515	−1.36
312500	−	−	−	−	0H	8	312500	0	1H	8	312500	0	2H	8	312500	0
625000	−	−	−	−	0H	4	625000	0	1H	4	625000	0	2H	4	625000	0

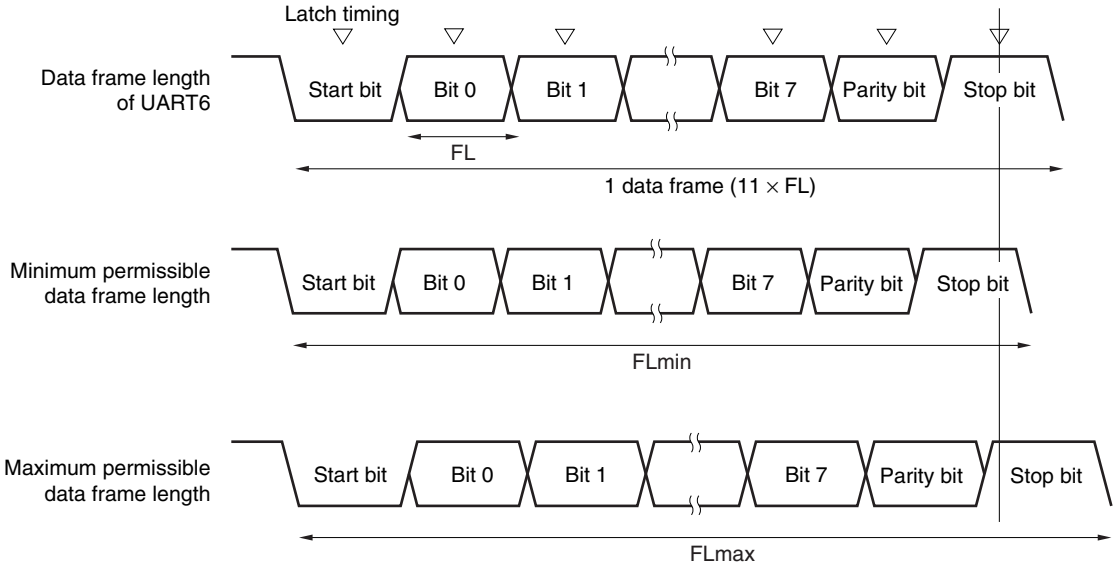
Remark TPS63 to TPS60: Bits 3 to 0 of clock selection register 6 (CKSR6) (setting of base clock (f_{XCLK6}))
k: Value set by MDL67 to MDL60 bits of baud rate generator control register 6 (BRGC6) (k = 4, 5, 6, ..., 255)
f_{PRS}: Peripheral hardware clock frequency
ERR: Baud rate error

(4) Permissible baud rate range during reception

The permissible error from the baud rate at the transmission destination during reception is shown below.

Caution Make sure that the baud rate error during reception is within the permissible error range, by using the calculation expression shown below.

Figure 14-25. Permissible Baud Rate Range During Reception



As shown in Figure 14-25, the latch timing of the receive data is determined by the counter set by baud rate generator control register 6 (BRGC6) after the start bit has been detected. If the last data (stop bit) meets this latch timing, the data can be correctly received.

Assuming that 11-bit data is received, the theoretical values can be calculated as follows.

$FL = (Brate)^{-1}$

Brate: Baud rate of UART6
k: Set value of BRGC6
FL: 1-bit data length
Margin of latch timing: 2 clocks

$$\text{Minimum permissible data frame length: FLmin} = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$$

Therefore, the maximum receivable baud rate at the transmission destination is as follows.

$$BRmax = (FLmin/11)^{-1} = \frac{22k}{21k+2} \text{ Brate}$$

Similarly, the maximum permissible data frame length can be calculated as follows.

$$\frac{10}{11} \times FLmax = 11 \times FL - \frac{k+2}{2 \times k} \times FL = \frac{21k-2}{2 \times k} FL$$

$$FLmax = \frac{21k-2}{20k} FL \times 11$$

Therefore, the minimum receivable baud rate at the transmission destination is as follows.

$$BRmin = (FLmax/11)^{-1} = \frac{20k}{21k-2} \text{ Brate}$$

The permissible baud rate error between UART6 and the transmission destination can be calculated from the above minimum and maximum baud rate expressions, as follows.

Table 14-6. Maximum/Minimum Permissible Baud Rate Error

Division Ratio (k)	Maximum Permissible Baud Rate Error	Minimum Permissible Baud Rate Error
4	+2.33%	–2.44%
8	+3.53%	–3.61%
20	+4.26%	–4.31%
50	+4.56%	–4.58%
100	+4.66%	–4.67%
255	+4.72%	–4.73%

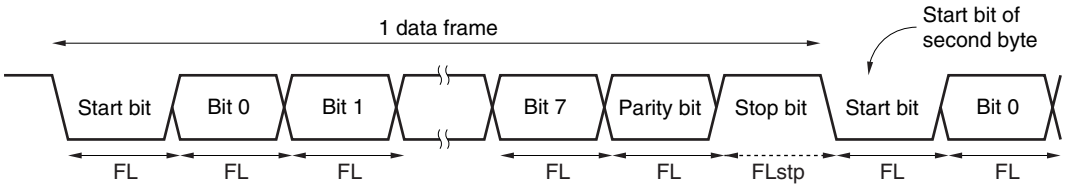
Remarks 1. The permissible error of reception depends on the number of bits in one frame, input clock frequency, and division ratio (k). The higher the input clock frequency and the higher the division ratio (k), the higher the permissible error.

2. k: Set value of BRGC6

(5) Data frame length during continuous transmission

When data is continuously transmitted, the data frame length from a stop bit to the next start bit is extended by two clocks of base clock from the normal value. However, the result of communication is not affected because the timing is initialized on the reception side when the start bit is detected.

Figure 14-26. Data Frame Length During Continuous Transmission



Where the 1-bit data length is FL, the stop bit length is FLstp, and base clock frequency is f_{XCLK6}, the following expression is satisfied.

$$FLstp = FL + 2/f_{XCLK6}$$

Therefore, the data frame length during continuous transmission is:

$$\text{Data frame length} = 11 \times FL + 2/f_{XCLK6}$$

CHAPTER 15 SERIAL INTERFACE CSI10

15.1 Functions of Serial Interface CSI10

Serial interface CSI10 is mounted onto all 78K0/Kx2-A microcontrollers.

Serial interface CSI10 has the following two modes.

(1) Operation stop mode

This mode is used when serial communication is not performed and can enable a reduction in the power consumption.

For details, see **15.4.1 Operation stop mode**.

(2) 3-wire serial I/O mode (MSB/LSB-first selectable)

This mode is used to communicate 8-bit data using three lines: a serial clock line ($\overline{\text{SCK10}}$) and two serial data lines (SI10 and SO10).

The processing time of data communication can be shortened in the 3-wire serial I/O mode because transmission and reception can be simultaneously executed.

In addition, whether 8-bit data is communicated with the MSB or LSB first can be specified, so this interface can be connected to any device.

The 3-wire serial I/O mode is used for connecting peripheral ICs and display controllers with a clocked serial interface.

For details, see **15.4.2 3-wire serial I/O mode**.

Caution The pins to which the serial clock ($\overline{\text{SCK10}}$) and serial data input (SI10) are assigned can be changed by setting bit 2 (ISC2) of the input switch control register. The transmission and reception capabilities differ according to the assigned pins, as shown in the table below..

ISC2	Pin to Which Serial Clock ($\overline{\text{SCK10}}$) Is Assigned	Pin to Which Serial Data Input (SI10) Is Assigned	Transmit/receive function
0	P60/SCLA0	P61/SDAA0	Slave transmission/reception only
1	P31/INTP5/OCD1A	P32/INTP4/OCD1B	Master transmission/reception, Slave transmission/reception

15.2 Configuration of Serial Interface CSI10

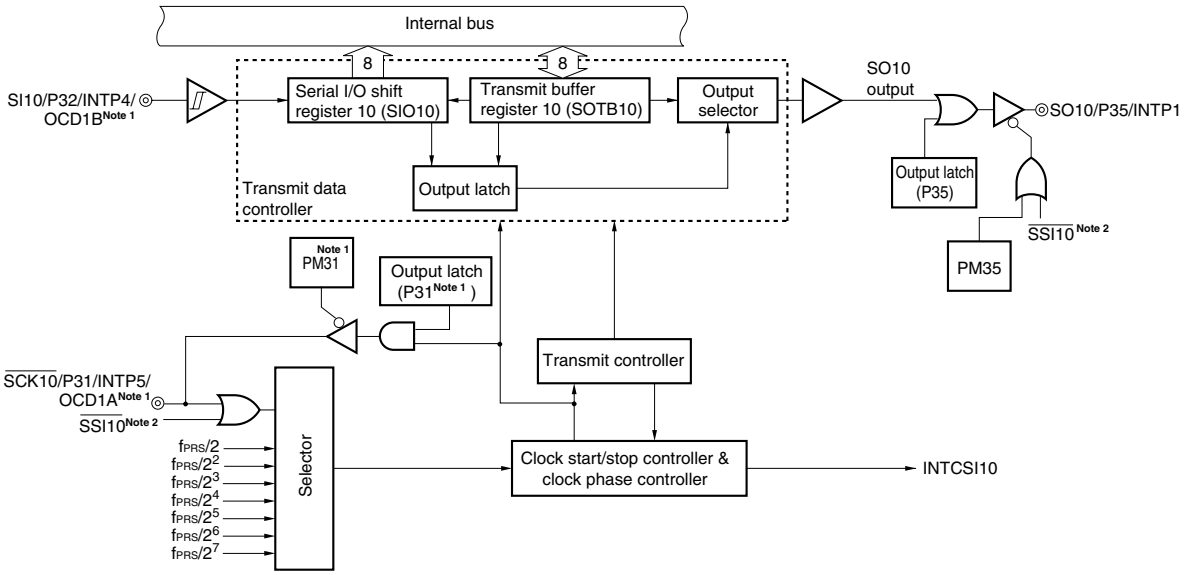
Serial interface CSI10 includes the following hardware.

Table 15-1. Configuration of Serial Interface CSI10

Item	Configuration
Controller	Transmit controller Clock start/stop controller & clock phase controller
Registers	Transmit buffer register 10 (SOTB10) Serial I/O shift register 10 (SIO10)
Control registers	Serial operation mode register 10 (CSIM10) Serial clock selection register 10 (CSIC10) Input switch control register (ISC) Port mode register 3, 4[Note], 6 (PM3, PM4[Note], PM6) Port register 3, 4[Note], 6 (P3, P4[Note], P6)

Note 78K0/KC2-A only.

Figure 15-1. Block Diagram of Serial Interface CSI10



Notes 1. These items differ as follows according to the setting of bit 2 (ISC2) of the input switch control register (ISC):

ISC2	Pin to Which Serial Clock ($\overline{SCK10}$) Is Assigned	Pin to Which Serial Data Input (SI10) Is Assigned	Port mode register (SCK10)	Output latch (SCK10)	Transmit/receive function
0	P60/SCLA0	P61/SDAA0	PM60 (Fixed to 1)	P60	Slave transmission/reception only
1	P31/INTP5/OCD1A	P32/INTP4/OCD1B	PM31	P31	Master transmission/reception, Slave transmission/reception

2. 78K0/KC2-A only

(1) Transmit buffer register 10 (SOTB10)

This register sets the transmit data.

Transmission/reception is started by writing data to SOTB10 when bit 7 (CSIE10) and bit 6 (TRMD10) of serial operation mode register 10 (CSIM10) is 1.

The data written to SOTB10 is converted from parallel data into serial data by serial I/O shift register 10, and output to the serial output pin (SO10).

SOTB10 can be written or read by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Cautions 1. Do not access SOTB10 when CSOT10 = 1 (during serial communication).

2. In the case of 78K0/KC2-A, in the slave mode, transmission/reception is started when data is written to SOTB11 with a low level input to the $\overline{\text{SSI10}}$ pin. For details on the transmission/reception operation, refer to 15.4.2 (2) Communication operation.

(2) Serial I/O shift register 10 (SIO10)

This is an 8-bit register that converts data from parallel data into serial data and vice versa.

This register can be read by an 8-bit memory manipulation instruction.

Reception is started by reading data from SIO10 if bit 6 (TRMD10) of serial operation mode register 10 (CSIM10) is 0.

During reception, the data is read from the serial input pin (SI10) to SIO10.

Reset signal generation clears this register to 00H.

Cautions 1. Do not access SIO10 when CSOT10 = 1 (during serial communication).

2. In the case of 78K0/KC2-A, in the slave mode, reception is started when data read from SIO10 with a low level input to the $\overline{\text{SSI10}}$ pin. For details on the reception operation, refer to 15.4.2 (2) Communication operation.

15.3 Registers Controlling Serial Interface CSI10

Serial interface CSI10 is controlled by the following five registers.

- Serial operation mode register 10 (CSIM10)
- Serial clock selection register 10 (CSIC10)
- Input switch control register (ISC)
- Port mode register 3, 4[Note], 6 (PM3, PM4[Note], PM6)
- Port register 3, 4[Note], 6 (P3, P4[Note], P6)

Note 78K0/KC2-A only.

(1) Serial operation mode register 10 (CSIM10)

CSIM10 is used to select the operation mode and enable or disable operation.

CSIM10 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 15-2. Format of Serial Operation Mode Register 10 (CSIM10)

Address: FF80H After reset: 00H R/W[Note 1]

Symbol	<7>	6	5	4	3	2	1	0
CSIM10	CSIE10	TRMD10	SSE10[Note 2]	DIR10	0	0	0	CSOT10

CSIE10	Operation control in 3-wire serial I/O mode
0	Disables operation[Note 3] and asynchronously resets the internal circuit[Note 4].
1	Enables operation

TRMD10[Note 5]	Transmit/receive mode control
0[Note 6]	Receive mode (transmission disabled).
1	Transmit/receive mode

SSE10[Notes 7, 8]	$\overline{\text{SSI10}}$ pin use selection
0	$\overline{\text{SSI10}}$ pin is not used
1	$\overline{\text{SSI10}}$ pin is used

DIR10[Note 9]	First bit specification
0	MSB
1	LSB

CSOT10	Communication status flag
0	Communication is stopped.
1	Communication is in progress.

Notes 1. Bit 0 is a read-only bit.

2. 78K0/KC2-A only.

3. To use P60/$\overline{\text{SCK10}}$/SCLA0, P31/INTP5/OCD1A/$\overline{\text{SCK10}}$, and P35/SO10/INTP1 as general-purpose ports, set CSIM10 in the default status (00H).

4. Bit 0 (CSOT10) of CSIM10 and serial I/O shift register 10 (SIO10) are reset.

5. Do not rewrite TRMD10 when CSOT10 = 1 (during serial communication).

6. The SO10 output (refer to **Figure 15-1**) is fixed to the low level when TRMD10 is 0. Reception is started when data is read from SIO10.

7. Do not rewrite SSE10 when CSOT10 = 1 (during serial communication).

8. Before setting this bit to 1, fix the $\overline{\text{SSI10}}$ pin input level to 0 or 1.

9. Do not rewrite DIR10 when CSOT10 = 1 (during serial communication).

(2) Serial clock selection register 10 (CSIC10)

This register specifies the timing of the data transmission/reception and sets the serial clock.

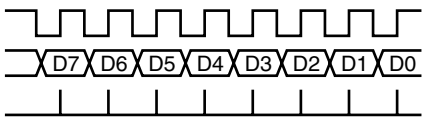
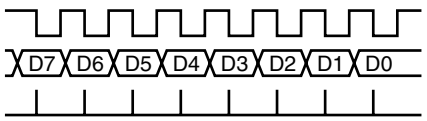
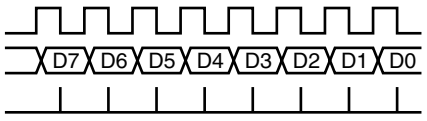
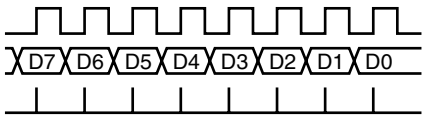
CSIC10 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 15-3. Format of Serial Clock Selection Register 10 (CSIC10)

Address: FF81H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIC10	0	0	0	CKP10	DAP10	CKS102	CKS101	CKS100

CKP10	DAP10	Specification of data transmission/reception timing	Type
0	0		1
0	1		2
1	0		3
1	1		4

CKS102	CKS101	CKS100	CSI10 serial clock selection[Notes 1, 2]					Mode
				f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz	
0	0	0	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz	Setting prohibited	Master mode[Note 4]
0	0	1	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz	
0	1	0	$f_{PRS}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz	
0	1	1	$f_{PRS}/2^4$	125 kHz	312.5 kHz	625 kHz	1.25 MHz	
1	0	0	$f_{PRS}/2^5$	62.5 kHz	156.25 kHz	312.5 kHz	625 kHz	
1	0	1	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	156.25 kHz	312.5 kHz	
1	1	0	$f_{PRS}/2^7$	15.63 kHz	39.06 kHz	78.13 kHz	156.25 kHz	
1	1	1	External clock input from $\overline{SCK10}$[Note 3]					Slave mode

Notes 1. The frequency that can be used for the peripheral hardware clock (f_{PRS}) differs depending on the power supply voltage.

Supply Voltage	Use frequency range of peripheral hardware clock (f_{PRS})
$2.7\ V \le V_{DD} \le 5.5\ V$	$f_{PRS} \le 20$ MHz
$1.8\ V \le V_{DD} < 2.7\ V$	$f_{PRS} \le 5$ MHz

(The values shown in the table above are those when $f_{PRS} = f_{XH}$ (XSEL = 1).)

2. Set the serial clock to satisfy the following conditions.

Supply Voltage	Use frequency range of serial clock
$4.0\ V \le V_{DD} \le 5.5\ V$	Serial clock ≤ 6.25 MHz
$2.7\ V \le V_{DD} < 4.0\ V$	Serial clock ≤ 4 MHz
$1.8\ V \le V_{DD} < 2.7\ V$	Serial clock ≤ 2 MHz

3. Do not start communication operation with the external clock from SCK10 when the internal high-speed oscillation clock and high-speed system clock are stopped while the CPU operates with the subsystem clock, or when in the STOP mode.

4. When clear the bit 2 (ISC2) of the Input switch control register (ISC) to 0, to select the master mode is prohibited.

.

Cautions 1. Do not write to CSIC10 while CSIE10 = 1 (operation enabled).

2. To use P60/$\overline{\text{SCK10}}$/SCLA0, P31/INTP5/OCD1A/$\overline{\text{SCK10}}$, or P35/SO10/INTP1 as general-purpose ports, set CSIC10 in the default status (00H).

3. The phase type of the data clock is type 1 after reset.

Remark f_{PRS}: Peripheral hardware clock oscillation frequency

(3) Input switch control register (ISC)

The input switch control register (ISC) is used to switch the pins to which the serial clock (SCK10) and serial data input (SI10) of serial interface $\overline{\text{CSI10}}$ are assigned.

By setting ISC2 to 1, $\overline{\text{SCK10}}$ and SI10 are assigned to the P31/INTP5/OCD1A and P32/INTP4/OCD1B pins, respectively. Note that the transmission and reception capabilities (master, slave) of CSI10 differ according to the assigned pins.This register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 15-4. Format of Input Switch Control Register (ISC)

Address: FF4FH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	0	0	0	0	0	ISC2	ISC1 Note	ISC0 Note

ISC2	Pin to Which Serial Clock ($\overline{\text{SCK10}}$) Is Assigned	Pin to Which Serial Data Input (SI10) Is Assigned	Transmit/receive function
0	P60/SCLA0	P61/SDAA0	Slave transmission/reception only
1	P31/INTP5/OCD1A	P32/INTP4/OCD1B	Master transmission/reception, Slave transmission/reception

Note ISC0 and ISC1 are setting about serial interface UART6, refer to **CHAPTER 14 SERIAL INTERFACE UART6**.

(4) Port mode registers 3, 4[Note 1], 6 (PM3, PM4[Note 1], PM6)

This register sets port 3, 4, and 6 input/output in 1-bit units.

When using P31/INTP5/OCD1A/$\overline{\text{SCK10}}$ as the clock output pin of the serial interface, clear PM31 to 0, and set the output latches of P31 to 1.

When using P35/SO10/INTP1 as the data output pin of the serial interface, clear PM35 and the output latches of P35 to 0.

When using $\overline{\text{SCK10}}$ as the clock input pin of the serial interface, SI10 as the data input pin, and P42/$\overline{\text{SSI10}}$/INTP9/PCL[Note 1] as the chip select input of the serial interface, set PM60[Note 2], PM61[Note 2], and PM42[Note 1] to 1. At this time, the output latches of P60[Note 3], P61[Note 3], and P42[Note 1] may be 0 or 1.

PM3, PM4[Note 1], and PM6 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Notes 1. 78K0/KC2-A only.

2. If bit 2 (ISC2) of the input switch control register is set to “1”, replace PM60 and PM61 with PM31 and PM32.

3. If bit 2 (ISC2) of the input switch control register is set to “1”, replace P60 and P61 with P31 and P32.

Figure 15-5. Format of Port Mode Register 3 (PM3)

Address: FF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	1	PM35	PM34	PM33	PM32	PM31	1

PM3n	P3n pin I/O mode selection (n = 1 to 5)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Remark The figure shown above presents the format of port mode register 3 of 78K0/KB2-A. For the format of port mode register 3 of 78K0/KC2-A, see **(1) Port mode registers (PMxx)** in **4.3 Registers Controlling Port Function**.

Figure 15-6. Format of Port Mode Register 4 (PM4) (78K0/KC2-A)

Address: FF24H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM4	1	1	1	1	1	PM42	PM41	PM40

PM4n	P4n pin I/O mode selection (n = 0 to 2)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Figure 15-7. Format of Port Mode Register 6 (PM6)

Address: FF26H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	1	1	1	1	1	1	PM61	PM60

PM6n	P6n pin I/O mode selection (n = 0, 1)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

15.4 Operation of Serial Interface CSI10

Serial interface CSI10 can be used in the following two modes.

- Operation stop mode
- 3-wire serial I/O mode

15.4.1 Operation stop mode

Serial communication is not executed in this mode. Therefore, the power consumption can be reduced.

In addition, the P60/$\overline{\text{SCK10}}$/SCLA0, P31/INTP5/OCD1A/$\overline{\text{SCK10}}$, P61/SI10/SDAA0, P32/INTP4/OCD1B/SI10, or P35/SO10/INTP1 pins can be used as ordinary I/O port pins in this mode.

(1) Register used

The operation stop mode is set by serial operation mode register 10 (CSIM10).

To set the operation stop mode, clear bit 7 (CSIE10) of CSIM10 to 0.

(a) Serial operation mode register 10 (CSIM10)

CSIM10 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears CSIM10 to 00H.

Address: FF80H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
CSIM10	CSIE10	TRMD10	SSE10	DIR10	0	0	0	CSOT10

CSIE10	Operation control in 3-wire serial I/O mode
0	Disables operation[Note 1] and asynchronously resets the internal circuit[Note 2].

Notes 1. To use P60/$\overline{\text{SCK10}}$/SCLA0, P31/INTP5/OCD1A/$\overline{\text{SCK10}}$, P35/SO10/INTP1, or P42/$\overline{\text{SSI10}}$/INTP9/PCL[Note 3] as general-purpose ports, set CSIM10 in the default status (00H).

2. Bit 0 (CSOT10) of CSIM10 and serial I/O shift register 10 (SIO10) are reset.

3. 78K0/KC2-A only.

15.4.2 3-wire serial I/O mode

The 3-wire serial I/O mode is used for connecting peripheral ICs and display controllers with a clocked serial interface.

In this mode, communication is executed by using three lines: the serial clock ($\overline{SCK10}$), serial output (SO10), and serial input (SI10) lines.

(1) Registers used

- Serial operation mode register 10 (CSIM10)
- Serial clock selection register 10 (CSIC10)
- Input switch control register (ISC)
- Port mode register 3, 4[Note], 6 (PM3, PM4[Note], PM6)
- Port register 3, 4[Note], 6 (P3, P4[Note], P6)

The basic procedure of setting an operation in the 3-wire serial I/O mode is as follows.

<1> Set the bit 2 (ISC2) of the input switch register (ISC).
<2> Set the CSIC10 register (see **Figures 15-3**).
<3> Set bits 4 to 6 (DIR10, SSE10[Note], TRMD10) of the CSIM10 register (see **Figures 15-2**).
<4> Set bit 7 (CSIE10) of the CSIM10 register to 1. → Transmission/reception is enabled.
<5> Write data to transmit buffer register 10 (SOTB10). → Data transmission/reception is started.
Read data from serial I/O shift register 10 (SIO10). → Data reception is started.

Note 78K0/KC2-A only.

Caution Take relationship with the other party of communication when setting the port mode register and port register.

The relationship between the register settings and pins is shown below.

Table 15-2. Relationship Between Register Settings and Pins

CSIE10	TRMD10	SSE10	PM61 (PM32)	P61 (P32)	PM35	P35	PM60 (PM31)	P60 (P31)	PM42 Note 1	P42 Note 1	CSI10 Operation	Pin Function			
												SI10/ SDAA0/ P61 (SI10/ INTP4/ P32)	SO10/ INTP1/ P35	$\overline{\text{SCK10}}$/ SCLA0/ P60 ($\overline{\text{SCK10}}$/ INTP5/ P31)	$\overline{\text{SSI10}}$/ INTP9/ PCL/ P42 Note 1
0	0	×	× Note 2	× Note 2	× Note 2	× Note 2	× Note 2	× Note 2	× Note 2	× Note 2	Stop	SDAA0/ P61 (INTP4/ P32)	INTP1/ P35 Note 3	SCLA0/ P60 Note 4 (INTP5/ P31) Note 4	INTP9/ PCL/P42
1	0	0	1	×	× Note 2	× Note 2	1	×	× Note 2	× Note 2	Slave reception Note 5	SI10	INTP1/ P35 Note 3	$\overline{\text{SCK10}}$ (Input) Note 5	INTP9/ PCL/P42
		1							1	×					$\overline{\text{SSI11}}$
1	1	0	× Note 2	× Note 2	0	0	1	×	× Note 2	× Note 2	Slave transmission Note 5	P61 (P32)	SO10	$\overline{\text{SCK10}}$ (Input) Note 5	INTP9/ PCL/P42
		1							1	×					$\overline{\text{SSI11}}$
1	1	0	1	×	0	0	1	×	× Note 2	× Note 2	Slave transmission Note 5	SI10	SO10	$\overline{\text{SCK10}}$ (Input) Note 5	INTP9/ PCL/P42
		1							1	×					$\overline{\text{SSI11}}$
1	0	0	1	×	× Note 2	× Note 2	0	1	× Note 2	× Note 2	Master reception Note 6	SI10	INTP1/ P35 Note 3	$\overline{\text{SCK10}}$ (Output)	INTP9/ PCL/P42
1	1	0	× Note 2	× Note 2	0	0	0	1	× Note 2	× Note 2	Master transmission Note 6	P32	SO10	$\overline{\text{SCK10}}$ (Output)	INTP9/ PCL/P42
1	1	0	1	×	0	0	0	1	× Note 2	× Note 2	Master transmission/ reception Note 6	SI10	SO10	$\overline{\text{SCK10}}$ (Output)	INTP9/ PCL/P42

(**Note** and **Remark** are listed on next page.)

Notes 1. 78K0/KC2-A only.

2. Can be set as port function.

3. To use P35/SO10/INTP1 as general-purpose port, set the serial clock selection register 10 (CSIC10) in the default status (00H).

4. To use P60/$\overline{\text{SCK10}}$/SCLA0 or P31/INTP5/OCD1A/$\overline{\text{SCK10}}$ as port pins, clear CKP10 to 0.

5. To use the slave mode, set CKS102, CKS101, and CKS100 to 1, 1, 1.

6. To use the master mode, set bit 2 (ISC2) of the input switch control register (ISC) to 1.

Remark ×: don't care

CSIE10: Bit 7 of serial operation mode register 10 (CSIM10)

TRMD10: Bit 6 of CSIM10

CKP10: Bit 4 of serial clock selection register 10 (CSIC10)

CKS102, CKS101, CKS100: Bits 2 to 0 of CSIC10

PMx×: Port mode register

Px×: Port output latch

The functions of pins whose names are in parentheses can be used by setting bit 2 (ISC2) of the input switch control register (ISC) to 1..

(2) Communication operation

In the 3-wire serial I/O mode, data is transmitted or received in 8-bit units. Each bit of the data is transmitted or received in synchronization with the serial clock.

Data can be transmitted or received if bit 6 (TRMD10) of serial operation mode register 10 (CSIM10) is 1. Transmission/reception is started when a value is written to transmit buffer register 10 (SOTB10). In addition, data can be received when bit 6 (TRMD10) of serial operation mode register 10 (CSIM10) is 0.

Reception is started when data is read from serial I/O shift register 10 (SIO10).

In the case of 78K0/KC3-A, communication is performed as follows if bit 5 (SSE10) of CSIM10 is 1 when serial interface CSI11 is in the slave mode.

<1> Low level input to the $\overline{\text{SSI10}}$ pin
→ Transmission/reception is started when SOTB10 is written, or reception is started when SIO10 is read.

<2> High level input to the $\overline{\text{SSI10}}$ pin
→ Transmission/reception or reception is held, therefore, even if SOTB10 is written or SIO10 is read, transmission/reception or reception will not be started.

<3> Data is written to SOTB10 or data is read from SIO10 while a high level is input to the $\overline{\text{SSI10}}$ pin, then a low level is input to the $\overline{\text{SSI10}}$ pin
→ Transmission/reception or reception is started.

<4> A high level is input to the $\overline{\text{SSI10}}$ pin during transmission/reception or reception
→ Transmission/reception or reception is suspended.

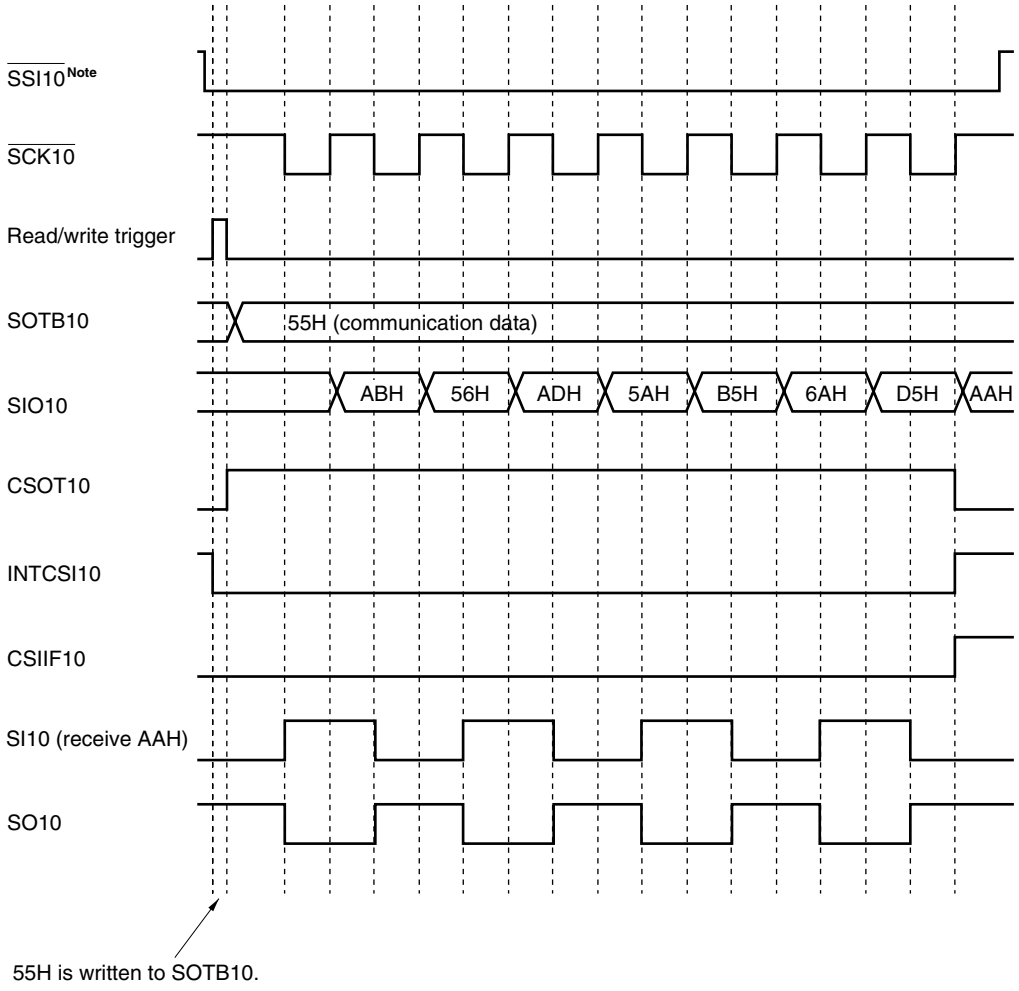
After communication has been started, bit 0 (CSOT10) of CSIM10 is set to 1. When communication of 8-bit data has been completed, a communication completion interrupt request flag (CSIIF10) is set, and CSOT10 is cleared to 0. Then the next communication is enabled.

Cautions 1. Do not access the control register and data register when CSOT10 = 1 (during serial communication).

2. In the case of 78K0/KC2-A, Wait for the duration of at least one clock before the clock operation is started to change the level of the $\overline{\text{SSI10}}$ pin in the slave mode; otherwise, malfunctioning may occur.

Figure 15-8. Timing in 3-Wire Serial I/O Mode (1/2)

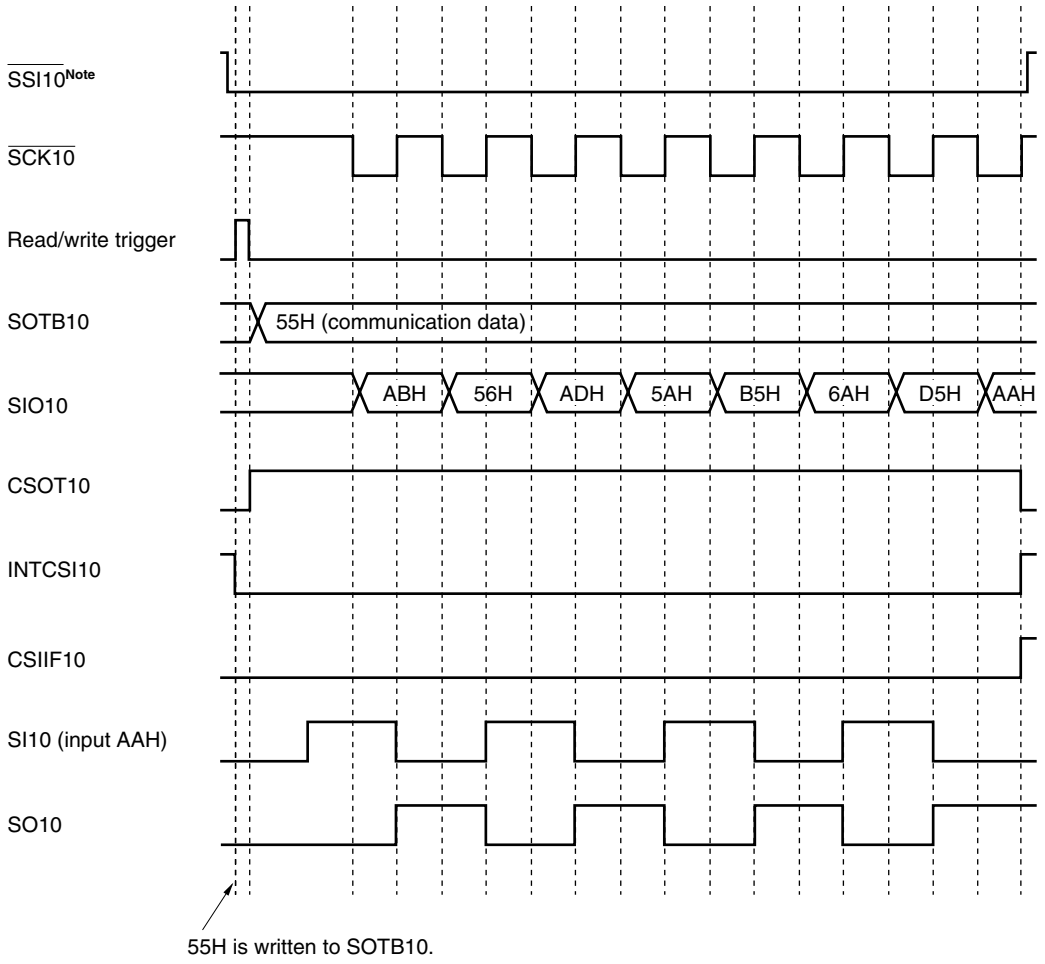
(a) Transmission/reception timing (Type 1: TRMD10 = 1, DIR10 = 0, CKP10 = 0, DAP10 = 0, SSE10 = 1[Note]))



Note The SSE10 flag and $\overline{\text{SSI10}}$ pins are only mounted on 78K0/KC2-A only. These pins are used in the slave mode.

Figure 15-8. Timing in 3-Wire Serial I/O Mode (2/2)

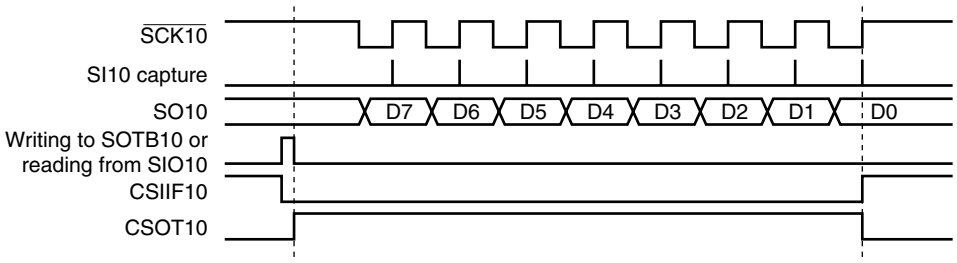
(b) Transmission/reception timing (Type 2: TRMD10 = 1, DIR10 = 0, CKP10 = 0, DAP10 = 1, SSE10 = 1[Note])



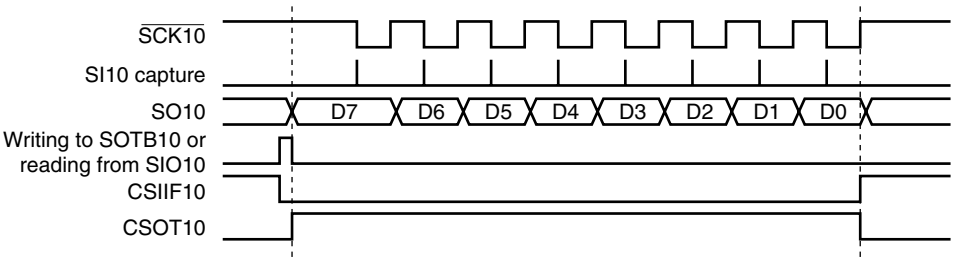
Note The SSE10 flag and $\overline{\text{SSI10}}$ pins are only mounted on 78K0/KC2-A only. These pins are used in the slave mode.

Figure 15-9. Timing of Clock/Data Phase

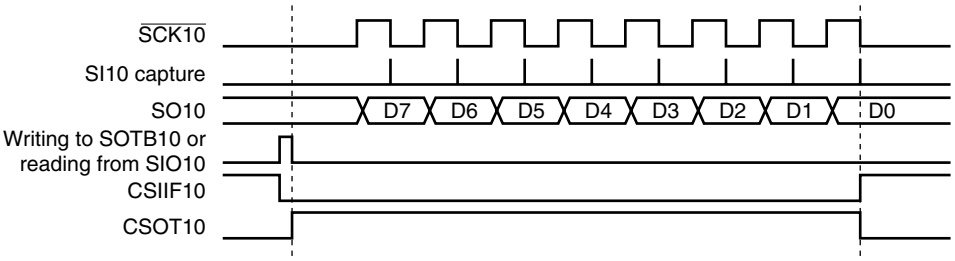
(a) Type 1: CKP10 = 0, DAP10 = 0, DIR10 = 0



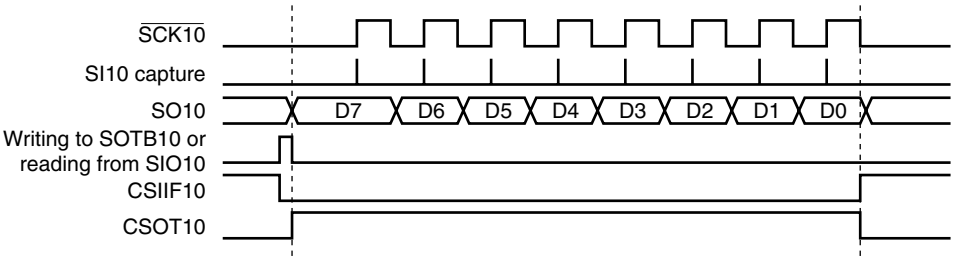
(b) Type 2: CKP10 = 0, DAP10 = 1, DIR10 = 0



(c) Type 3: CKP10 = 1, DAP10 = 0, DIR10 = 0



(d) Type 4: CKP10 = 1, DAP10 = 1, DIR10 = 0



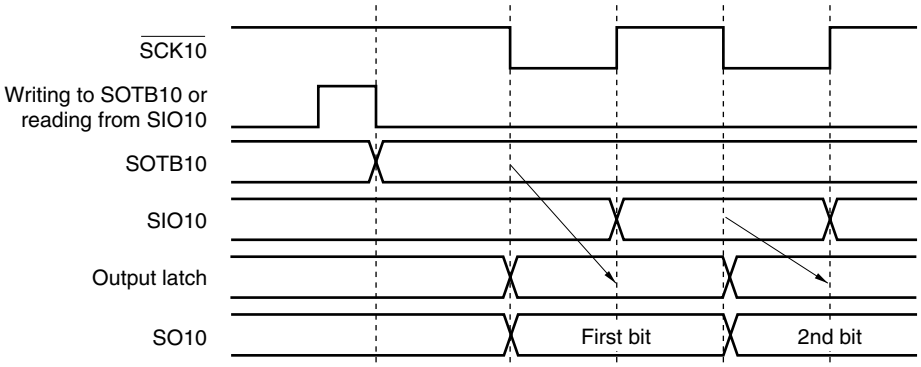
Remark The above figure illustrates a communication operation where data is transmitted with the MSB first.

(3) Timing of output to SO10 pin (first bit)

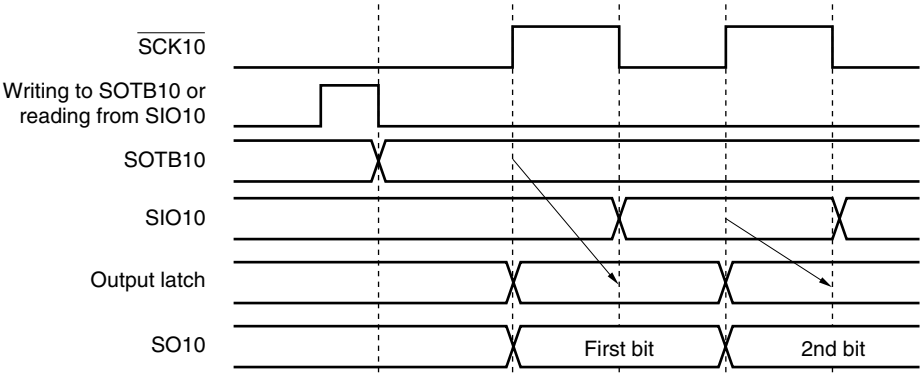
When communication is started, the value of transmit buffer register 10 (SOTB10) is output from the SO10 pin. The output operation of the first bit at this time is described below.

Figure 15-10. Output Operation of First Bit (1/2)

(a) Type 1: CKP10 = 0, DAP10 = 0



(b) Type 3: CKP10 = 1, DAP10 = 0

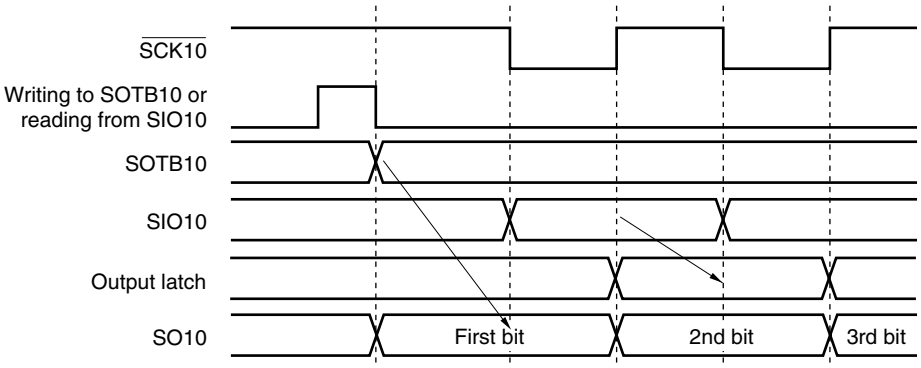


The first bit is directly latched by the SOTB10 register to the output latch at the falling (or rising) edge of $\overline{SCK10}$, and output from the SO10 pin via an output selector. Then, the value of the SOTB10 register is transferred to the SIO10 register at the next rising (or falling) edge of $\overline{SCK10}$, and shifted one bit. At the same time, the first bit of the receive data is stored in the SIO10 register via the SI10 pin.

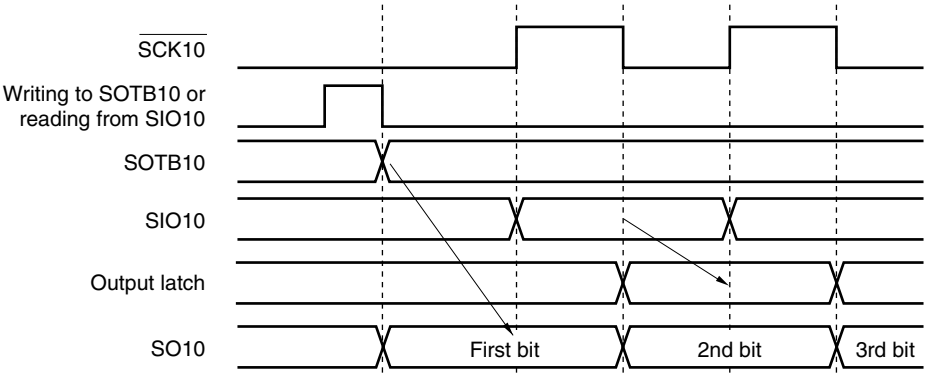
The second and subsequent bits are latched by the SIO10 register to the output latch at the next falling (or rising) edge of $\overline{SCK10}$, and the data is output from the SO10 pin.

Figure 15-10. Output Operation of First Bit (2/2)

(c) Type 2: CKP10 = 0, DAP10 = 1



(d) Type 4: CKP10 = 1, DAP10 = 1



The first bit is directly latched by the SOTB10 register at the falling edge of the write signal of the SOTB10 register or the read signal of the SIO10 register, and output from the SO10 pin via an output selector. Then, the value of the SOTB10 register is transferred to the SIO10 register at the next falling (or rising) edge of $\overline{\text{SCK10}}$, and shifted one bit. At the same time, the first bit of the receive data is stored in the SIO10 register via the SI10 pin.

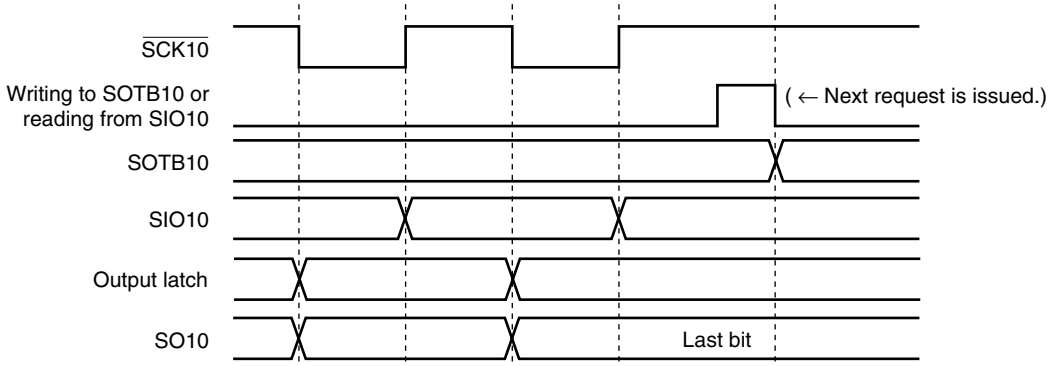
The second and subsequent bits are latched by the SIO10 register to the output latch at the next rising (or falling) edge of $\overline{\text{SCK10}}$, and the data is output from the SO10 pin.

(4) Output value of SO10 pin (last bit)

After communication has been completed, the SO10 pin holds the output value of the last bit.

Figure 15-11. Output Value of SO10 Pin (Last Bit) (1/2)

(a) Type 1: CKP10 = 0, DAP10 = 0



(b) Type 3: CKP10 = 1, DAP10 = 0

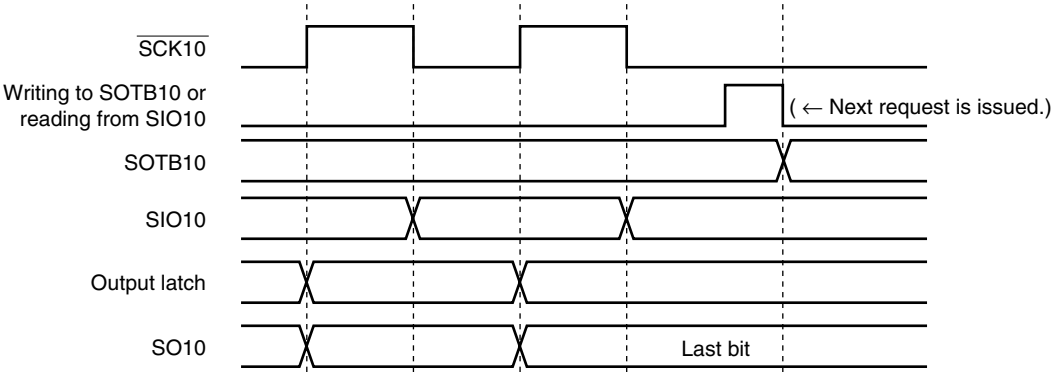
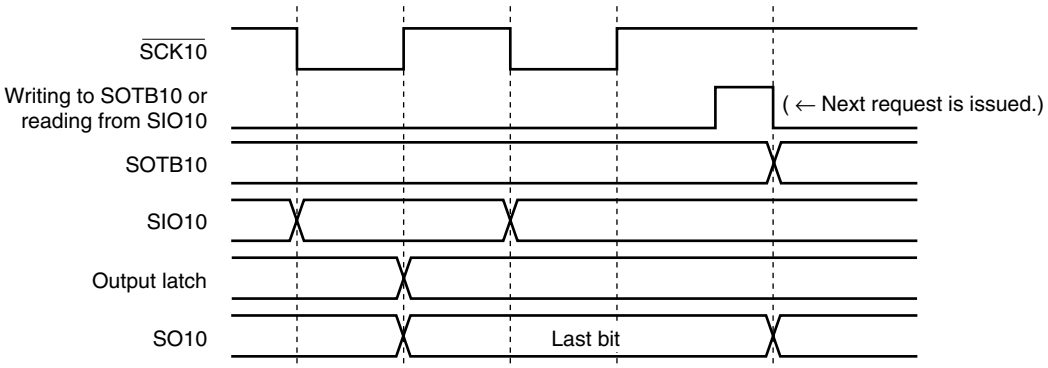
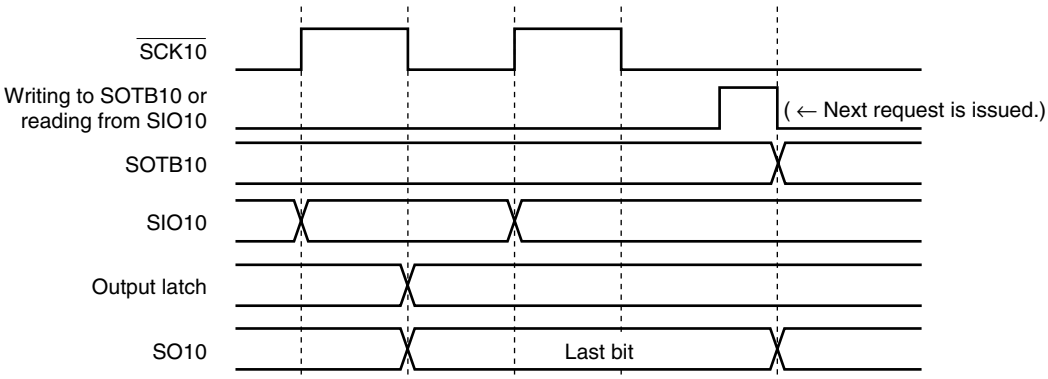


Figure 15-11. Output Value of SO10 Pin (Last Bit) (2/2)

(c) Type 2: CKP10 = 0, DAP10 = 1



(d) Type 4: CKP10 = 1, DAP10 = 1



(5) SO10 output (see Figure 15-1)

The status of the SO10 output is as follows by setting CSIE10, TRMD10, DAP10, and DIR10.

Table 15-3. SO10 Output Status

CSIE10	TRMD10	DAP10	DIR10	SO10 Output[Note 1]
CSIE10 = 0[Note 2]	TRMD10 = 0[Notes 2, 3]	–	–	Outputs low level[Note 2]
	TRMD10 = 1	DAP10 = 0	–	Outputs low leve
		DAP10 = 1	DIR 10 = 0	Value of bit 7 of SOTB10
			DIR 10 = 1	Value of bit 0 of SOTB10
CSIE10 = 1	TRMD10 = 0[Note 3]	–	–	Outputs low leve
	TRMD10 = 1	–	–	Transmit data[Note 4]

Notes 1. The actual output of the SO10/INTP1/P35 pin is determined according to PM35 and P35, as well as the SO10 output.

2. Status after reset.

3. To use P35/SO10/INTP1 as general-purpose port, set the serial clock selection register 10 (CSIC10) in the default status (00H).

4. After transmission has been completed, the SO1n pin holds the output value of the last bit of transmission data.

Caution If a value is written to CSIE10, TRMD10, DAP10, and DIR10, the output value of SO10 changes.

CHAPTER 16 SERIAL INTERFACE IICA

16.1 Functions of Serial Interface IICA

Serial interface IICA is mounted onto all 78K0/Kx2-A microcontrollers.
Serial interface IICA has the following three modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed. It can therefore be used to reduce power consumption.

(2) I^2C bus mode (multimaster supported)

This mode is used for 8-bit data transfers with several devices via two lines: a serial clock (SCLA0) line and a serial data bus (SDAA0) 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 SCLA0 and SDAA0 pins are used for open drain outputs, IICA requires pull-up resistors for the serial clock line and the serial data bus line.

(3) Wakeup mode

The STOP mode can be released by generating an interrupt request signal (INTIICA0) when an extension code from the master device or a local address has been received while in STOP mode. This can be set by using the WUP bit of IICA control register 1 (IICACTL1).

Figure 16-1 shows a block diagram of serial interface IICA.

Figure 16-1. Block Diagram of Serial Interface IICA

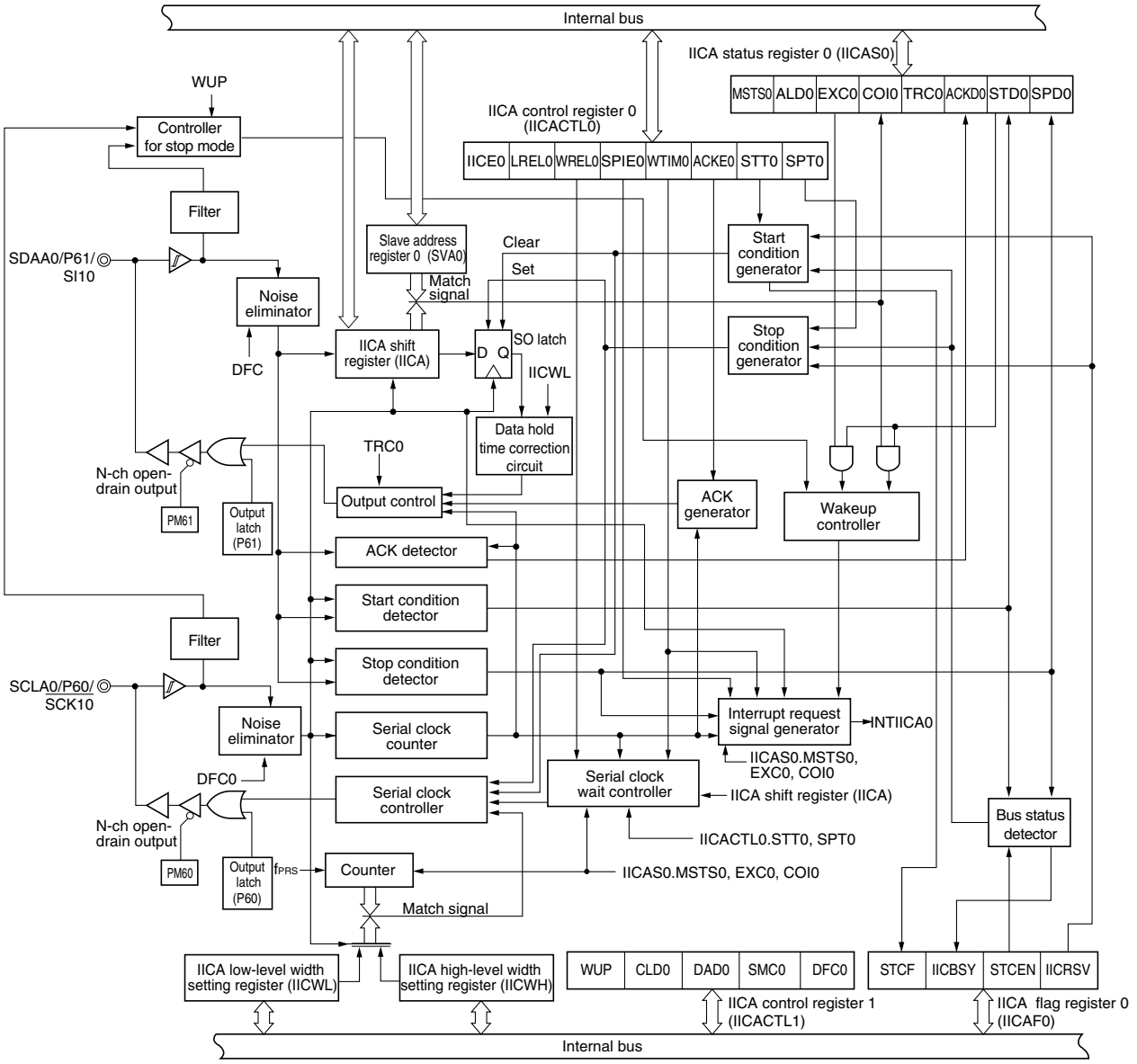
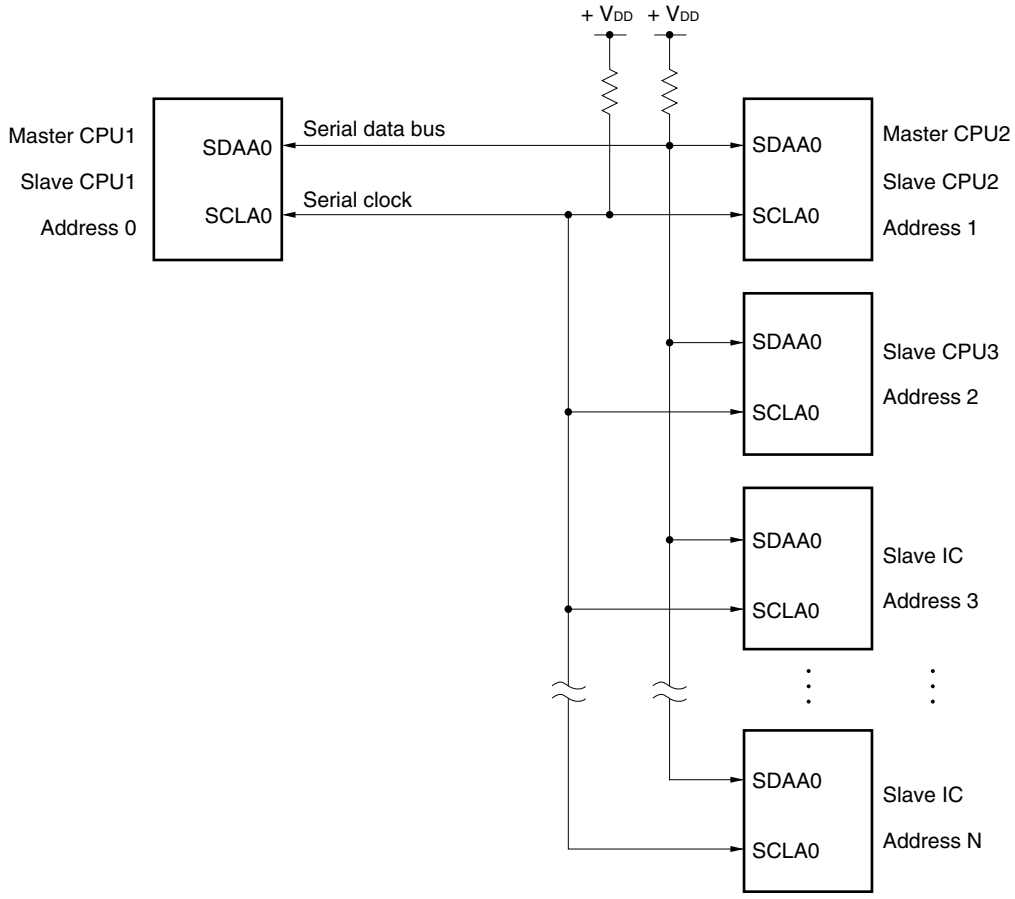


Figure 16-2 shows a serial bus configuration example.

Figure 16-2. Serial Bus Configuration Example Using I^2C Bus



16.2 Configuration of Serial Interface IICA

Serial interface IICA includes the following hardware.

Table 16-1. Configuration of Serial Interface IICA

Item	Configuration
Registers	IICA shift register (IICA) Slave address register (SVA0)
Control registers	IICA control register 0 (IICACTL0) IICA status register 0 (IICAS0) IICA flag register 0 (IICAF0) IICA control register 1 (IICACTL1) IICA low-level width setting register (IICWL) IICA high-level width setting register (IICWH) Port mode register 6 (PM6) Port register 6 (P6)

(1) IICA shift register (IICA)

IICA is used to convert 8-bit serial data to 8-bit parallel data and vice versa in synchronization with the serial clock. IICA can be used for both transmission and reception.

The actual transmit and receive operations can be controlled by writing and reading operations to IICA.

Cancel the wait state and start data transfer by writing data to IICA during the wait period.

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

Reset signal generation clears IICA to 00H.

Figure 16-3. Format of IICA Shift Register (IICA)

Address: FFA6H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IICA								

Cautions 1. Do not write data to IICA during data transfer.

2. Write or read IICA only during the wait period. Accessing IICA in a communication state other than during the wait period is prohibited. When the device serves as the master, however, IICA 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 SVA to 00H.

Figure 16-4. Format of Slave Address Register 0 (SVA0)

Address: FFA7H 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 SDAA0 pin's output level.

(4) Wakeup controller

This circuit generates an interrupt request (INTIICA0) when the address received by this register matches the address value set to the slave address register 0 (SVA0) or when an extension code is received.

(5) Serial clock counter

This counter counts the serial clocks that are output or input during transmit/receive operations and is used to verify that 8-bit data was transmitted or received.

(6) Interrupt request signal generator

This circuit controls the generation of interrupt request signals (INTIICA0).

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 IICA control register 0 (IICACTL0)
SPIE0 bit: Bit 4 of IICA control register 0 (IICACTL0)

(7) Serial clock controller

In master mode, this circuit generates the clock output via the SCLA0 pin from a sampling clock.

(8) Serial clock wait controller

This circuit controls the wait timing.

(9) ACK generator, stop condition detector, start condition detector, and ACK detector

These circuits generate and detect each status.

(10) Data hold time correction circuit

This circuit generates the hold time for data corresponding to the falling edge of the serial clock.

(11) Start condition generator

This circuit generates a start condition when the 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.

(12) Stop condition generator

This circuit generates a stop condition when the SPT0 bit is set to 1.

(13) Bus status detector

This circuit detects whether or not the bus is released by detecting start conditions and stop conditions.

However, as the bus status cannot be detected immediately following operation, the initial status is set by the STCEN bit.

Remark STT0 bit: Bit 1 of IICA control register 0 (IICACTL0)
SPT0 bit: Bit 0 of IICA control register 0 (IICACTL0)
IICRSV bit: Bit 0 of IICA flag register 0(IICAF0)
IICBSY bit: Bit 6 of IICA flag register 0(IICAF0)
STCF bit: Bit 7 of IICA flag register 0(IICAF0)
STCEN bit: Bit 1 of IICA flag register 0(IICAF0)

16.3 Registers Controlling Serial Interface IICA

Serial interface IICA is controlled by the following eight registers.

- IICA control register 0 (IICACTL0)
- IICA status register 0 (IICAS0)
- IICA flag register 0 (IICAF0)
- IICA control register 1 (IICACTL1)
- IICA low-level width setting register (IICWL)
- IICA high-level width setting register (IICWH)
- Port mode register 6 (PM6)
- Port register 6 (P6)

(1) IICA control register 0 (IICACTL0)

This register is used to enable/stop I^2C operations, set wait timing, and set other I^2C operations.

IICACTL0 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 16-5. Format of IICA Control Register 0 (IICACTL0) (1/4)

Address: FFA8H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICACTL0	IICE0	LREL0	WREL0	SPIE0	WTIM0	ACKE0	STT0	SPT0

IICE0	I^2C operation enable
0	Stop operation. Reset the IICA status register 0 (IICAS0)[Note 1]. Stop internal operation.
1	Enable operation.
Be sure to set this bit (1) while the SCLA0 and SDAA0 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 SCLA0 and SDAA0 lines are set to high impedance. The following flags of IICA control register 0 (IICACTL0) and IICA status register 0 (IICAS0) 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 SDAA0 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 IICAS0 status register, the STCF and IICBSY bits of the IICAF0 register, and the CLD0 and DAD0 bits of the IICACTL1 register are reset.

2. The signal of this bit is invalid while IICE0 is 0.

Caution If the operation of I^2C is enabled (IICE0 = 1) when the SCLA0 line is high level, the SDAA0 line is low level, a start condition will be inadvertently detected immediately. In this case, set (1) the LREL0 bit by using a 1-bit memory manipulation instruction immediately after enabling operation of I^2C (IICE0 = 1).

Figure 16-5. Format of IICA Control Register 0 (IICACTL0) (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 SDAA0 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. The signal of this bit is invalid while IICE0 is 0. Set this bit during that period.

2. The set value is invalid during address transfer and if the code is not an extension code.
When the device serves as a slave and the addresses match, an acknowledgment is generated regardless of the set value.

Figure 16-5. Format of IICA Control Register 0 (IICACTL0) (3/4)

STT0[Note]	Start condition trigger
0	Do not generate a start condition.
1	When bus is released (in stop state): Generate a start condition (for starting as master). When the SCLA0 line is high level, the SDAA0 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, SCLA0 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 STT0 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 The signal of this bit is invalid while IICE0 is 0.

Remarks 1. Bit 1 (STT0) becomes 0 when it is read after data setting.

2. IICRSV: Bit 0 of IICA flag register (IICAF0)
STCF: Bit 7 of IICA flag register (IICAF0)

Figure 16-5. Format of IICA Control Register 0 (IICACTL0) (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 SDAA0 line goes to low level, either set the SCLA0 line to high level or wait until it goes to high level. Next, after the rated amount of time has elapsed, the SDAA0 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 the bit 3 (TRC0) of the IIC status register 0 (IICAS0) is set to 1, WREL0 is set to 1 during the ninth clock and wait is canceled, after which TRC0 is cleared and the SDAA0 line is set to high impedance.

Remark Bit 0 (SPT0) becomes 0 when it is read after data setting.

(2) IICA status register 0 (IICAS0)

This register indicates the status of I^2C.

IICAS0 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.

Caution Reading the IICA control register 1 (IICACTL1) while WUP = 1 is prohibited. When WUP is changed from 1 to 0 (wakeup operation is stopped), regardless of the INTIICA0 interrupt request, the change in status is not reflected until the next start condition or stop condition is detected. To use the wakeup mode, therefore, enable (SPIE0 = 1) the interrupt generated by detecting a stop condition and read the IICAS0 register after the interrupt has been detected.

Figure 16-6. Format of IICA Status Register 0 (IICAS0) (1/3)

Address: FFABH After reset: 00H R

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICAS0	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 IICAS0 is read[Note] • When IICE0 changes from 1 to 0 (operation stop) • Reset	• When the arbitration result is a “loss”.

Note This register is also cleared when a 1-bit memory manipulation instruction is executed for bits other than IICAS0. Therefore, when using the ALD0 bit, read the data of this bit before the data of the other bits.

Remark LREL0: Bit 6 of IICA control register 0 (IICACTL0)
IICE0: Bit 7 of IICA control register 0 (IICACTL0)

Figure 16-6. Format of IICA Status Register 0 (IICAS0) (2/3)

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).

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 SDAA0 line is set for high impedance.
1	Transmit status. The value in the SO0 latch is enabled for output to the SDAA0 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)	<Master> • When a start condition is generated • When 0 is output to the LSB (transfer direction specification bit) of the first byte <Slave> • When 1 is input to the LSB (transfer direction specification bit) of the first byte

Note When bit 3 (TRC0) of the IICA status register 0 (IICAS0) is set to 1, bit 5 (WREL0) of the IICA control register 0 (IICACTL0) is set to 1 during the ninth clock and wait is canceled, TRC0 is cleared and the SDAA0 line is set to high impedance.

Remark LREL0: Bit 6 of IICA control register 0 (IICACTL0)
IICE0: Bit 7 of IICA control register 0 (IICACTL0)

Figure 16-6. Format of IICA Status Register 0 (IICAS0) (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 SDAA0 line is set to low level at the rising edge of SCLA0'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 IICA control register 0 (IICACTL0)
IICE0: Bit 7 of IICA control register 0 (IICACTL0)

(3) IICA flag register 0 (IICAF0)

This register sets the operation mode of I^2C and indicates the status of the I^2C bus.

IICAF0 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 IICA control register 0 (IICACTL0) = 0). When operation is enabled, the IICAF0 register can be read.

Reset signal generation clears this register to 00H.

Figure 16-7. Format of IICA Flag Register 0 (IICAF0)

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

Symbol	<7>	<6>	5	4	3	2	<1>	<0>
IICAF0	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)
• Cleared by instruction • Detection of start condition • Reset	• Set by instruction

IICRSV	Communication reservation function disable bit
0	Enable communication reservation
1	Disable communication reservation

Condition for clearing (IICRSV = 0)	Condition for setting (IICRSV = 1)
• Cleared by instruction • Reset	• Set by instruction

Note Bits 6 and 7 are read-only.

Cautions 1. Write to 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 IICA control register 0 (IICACTL0)
IICE0: Bit 7 of IICA control register 0 (IICACTL0)

(4) IICA control register 1 (IICACTL1)

This register is used to set the operation mode of I^2C and detect the statuses of the SCLA0 and SDAA0 pins.

IICACTL1 can be set by a 1-bit or 8-bit memory manipulation instruction. However, the CLD0 and DAD0 bits are read-only.

Set the IICACTL1 register, except the WUP bit, while operation of I^2C is disabled (bit 7 (IICE0) of IICA control register 0 (IICACTL0) is 0).

Reset signal generation clears this register to 00H.

Figure 16-8. Format of IICA Control Register 1 (IICACTL1) (1/2)

Address: FFA9H After reset: 00H R/W[Note 1]

Symbol	7	6	<5>	<4>	<3>	<2>	1	0
IICACTL1	WUP	0	CLD0	DAD0	SMC0	DFC0	0	0

WUP	Control of address match wakeup
0	Stops operation of address match wakeup function in STOP mode.
1	Enables operation of address match wakeup function in STOP mode.

Clear (0) WUP after the address has matched or an extension code has been received. The subsequent communication can be entered by clearing (0) WUP. (The wait must be released and transmit data must be written after WUP has been cleared (0).)

The interrupt timing when the address has matched or when an extension code has been received, while WUP = 1, is identical to the interrupt timing when WUP = 0. (A delay of the difference of sampling by the clock will occur.) Furthermore, when WUP = 1, a stop condition interrupt is not generated even if the SPIE0 bit is set to 1. When WUP = 0 is set by a source other than an interrupt from serial interface IICA, operation as the master device cannot be performed until the subsequent start condition or stop condition is detected. Do not output a start condition by setting (1) the STT0, without waiting for the detection of the subsequent start condition or stop condition.

Condition for clearing (WUP = 0)	Condition for setting (WUP = 1)
• Cleared by instruction (after address match or extension code reception)	• Set by instruction (when MSTS0, EXC0, and COI0 are “0”, and STD0 also “0” (communication not entered))[Note 2]

CLD0	Detection of SCLA0 pin level (valid only when IICE0 = 1)
0	The SCLA0 pin was detected at low level.
1	The SCLA0 pin was detected at high level.

Condition for clearing (CLD0 = 0)	Condition for setting (CLD0 = 1)
• When the SCLA0 pin is at low level • When IICE0 = 0 (operation stop) • Reset	• When the SCLA0 pin is at high level

Figure 16-8. Format of IICA Control Register 1 (IICACTL1) (2/2)

DAD0	Detection of SDAA0 pin level (valid only when IICE0 = 1)
0	The SDAA0 pin was detected at low level.
1	The SDAA0 pin was detected at high level.

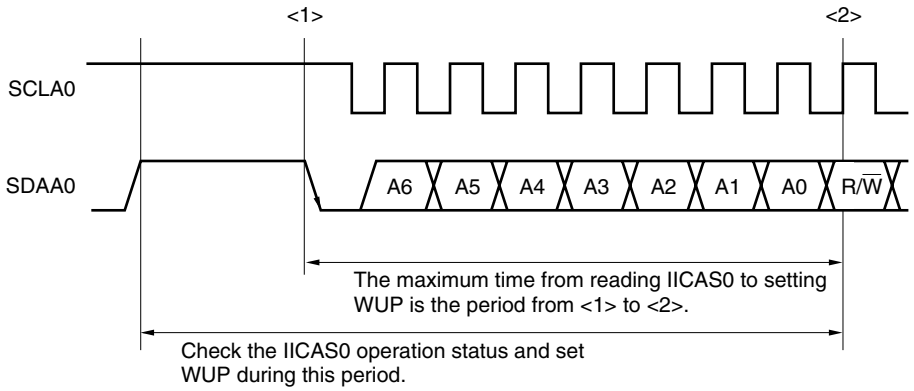
Condition for clearing (DAD0 = 0)	Condition for setting (DAD0 = 1)
• When the SDAA0 pin is at low level • When IICE0 = 0 (operation stop) • Reset	• When the SDAA0 pin is at high level

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 the DFC0 bit being set (1) or cleared (0). The digital filter is used for noise elimination in fast mode.	

Notes 1. Bits 4 and 5 are read-only.

2. The status of IICAS0 must be checked and WUP must be set during the period shown below.



Remark IICE0: Bit 7 of IICA control register 0 (IICACTL0)

(5) IICA low-level width setting register (IICWL)

This register is used to set the low-level width (t_{LOW}) of the SCLA0 pin signal and data hold time ($t_{HD:DAT}$) that is output by serial interface IICA being in master mode. Data hold time is determined by higher 6-bit of IICWL.

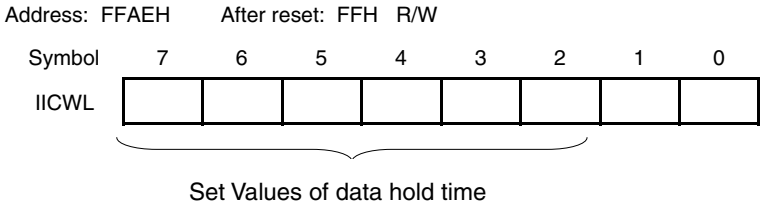
This register can be set by an 8-bit memory manipulation instruction.

Set this register while operation of I^2C is disabled (bit 7 (IICE0) of the IICA control register 0 (IICACTL0) is 0).

Reset signal generation sets this register to FFH.

See the **16.4.2 Setting transfer clock by using IICWL and IICWH registers** for how to set the IICWL.

Figure 16-9. Format of IICA Low-Level Width Setting Register (IICWL)



(6) IICA high-level width setting register (IICWH)

This register is used to set the high-level width (t_{HIGH}) of the SCLA0 pin signal that is output by serial interface IICA being in master mode.

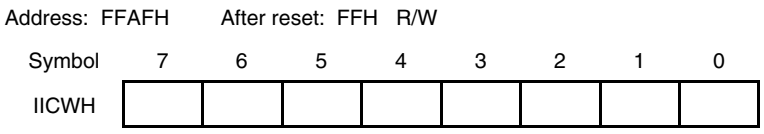
This register can be set by an 8-bit memory manipulation instruction.

Set IICWH while bit 7 (IICE) of IICA control register 0 (IICACTL0) is 0).

Reset signal generation sets this register to FFH.

See the **16.4.2 Setting transfer clock by using IICWL and IICWH registers** for how to set the IICWL.

Figure 16-10. Format of IICA High-Level Width Setting Register (IICWH)



(7) Port mode register 6 (PM6)

This register sets the input/output of port 6 in 1-bit units.

When using the P60/SCLA0/$\overline{\text{SCK10}}$ pin as clock I/O and the P61/SDAA0/SI10 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 IICA control register 0 (IICACTL0)) to 1 before setting the output mode because the P60/SCLA0/$\overline{\text{SCK10}}$ and P61/SDAA0/SI10 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 16-11. Format of Port Mode Register 6 (PM6)

Address: FF26H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	1	1	1	1	1	1	PM61	PM60

PM6n	P6n pin I/O mode selection (n = 0, 1)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

16.4 I²C Bus Mode Functions

16.4.1 Pin configuration

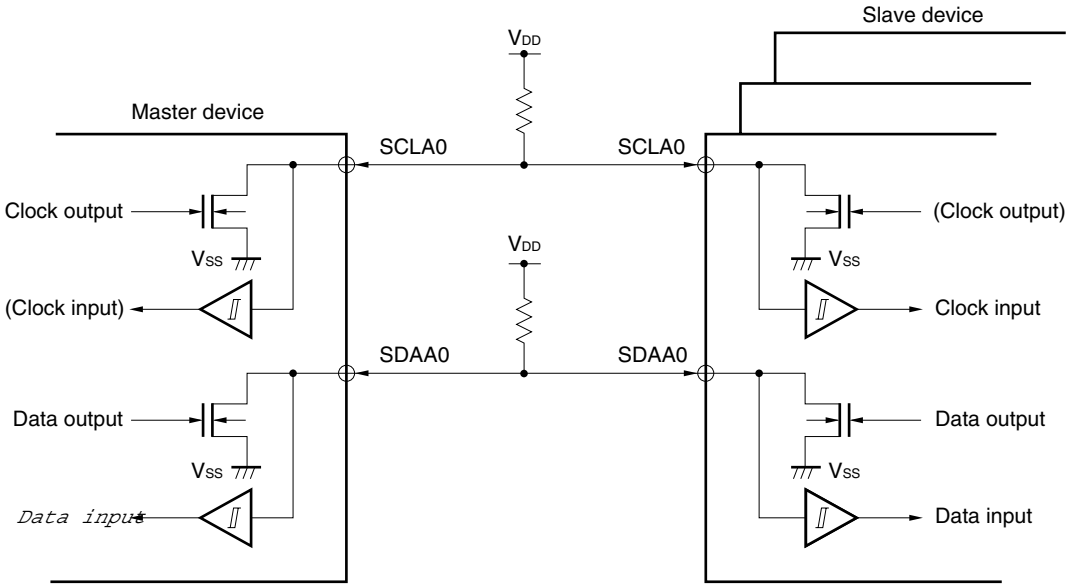
The serial clock pin (SCLA0) and serial data bus pin (SDAA0) are configured as follows.

(1) SCLA0 .... 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) SDAA0 .... 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 16-12. Pin Configuration Diagram



16.4.2 Setting transfer clock by using IICWL and IICWH registers

(1) Setting transfer clock on master side

$$\text{Transfer clock} = \frac{f_{PRS}}{IICWL + IICWH + f_{PRS}(t_R + t_F)}$$

At this time, the optimal setting values of IICWL and IICWH are as follows.
(The fractional parts of all setting values are rounded up.)

- When the fast mode

$$IICWL = \frac{0.52}{\text{Transfer clock}} \times f_{PRS}$$

$$IICWH = \left(\frac{0.48}{\text{Transfer clock}} - t_R - t_F\right) \times f_{PRS}$$

- When the normal mode

$$IICWL = \frac{0.47}{\text{Transfer clock}} \times f_{PRS}$$

$$IICWH = \left(\frac{0.53}{\text{Transfer clock}} - t_R - t_F\right) \times f_{PRS}$$

Remark The data hold time is calculated as follows, based on the setting of the IICWL register:

$$\text{Data hold time} = \frac{\text{Higher 6-bit of IICWL}}{f_{PRS}}$$

Example: If the transfer clock is 400 [kHz] (fast mode), f_{PRS} is 20 [MHz], and IICWL is 26 (higher 6 bits = 6), then:

Caution Make sure that the data hold time does not exceed 0.9 μs in fast mode and 3.45 μs in standard mode.

(2) Setting IICWL and IICWH on slave side

(The fractional parts of all setting values are truncated.)

- When the fast mode

$$IICWL = 1.3\ \mu s \times f_{PRS}$$
$$IICWH = (1.2\ \mu s - t_R - t_F) \times f_{PRS}$$

- When the normal mode

$$IICWL = 4.7\ \mu s \times f_{PRS}$$
$$IICWH = (5.3\ \mu s - t_R - t_F) \times f_{PRS}$$

Caution Note the minimum f_{PRS} operation frequency when setting the transfer clock. The minimum f_{PRS} operation frequency for serial interface IICA is determined according to the mode.

Fast mode: f_{PRS} = 3.5 MHz (min.)

Normal mode: f_{PRS} = 1 MHz (min.)

Remark IICWL : IICA low-level width setting register

IICWH : IICA high-level width setting register

t_F: SDAA0 and SCLA0 signal falling times (refer to **CHAPTER 28 ELECTRIC SPECIFICTION**)

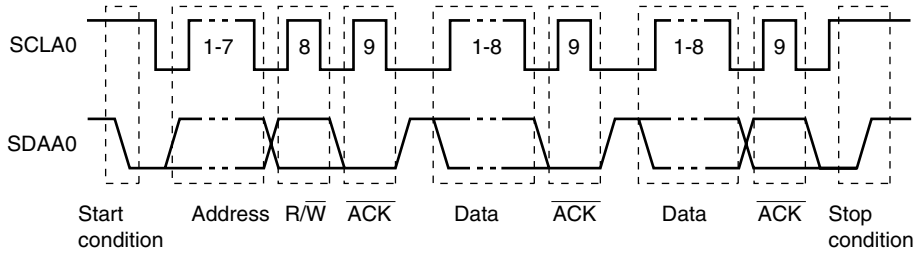
t_R: SDAA0 and SCLA0 signal rising times (refer to **CHAPTER 28 ELECTRIC SPECIFICTION**)

f_{PRS}: Peripheral hardware clock frequency

16.5 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 16-13 shows the transfer timing for the "start condition", "address", "data", and "stop condition" output via the I²C bus's serial data bus.

Figure 16-13. I²C Bus Serial Data Transfer Timing



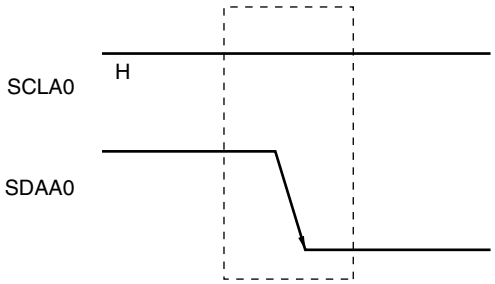
The master device generates the start condition, slave address, and stop condition.
The acknowledge ($\overline{ACK}$) can be generated by either the master or slave device (normally, it is output by the device that receives 8-bit data).

The serial clock (SCLA0) is continuously output by the master device. However, in the slave device, the SCLA0's low level period can be extended and a wait can be inserted.

16.5.1 Start conditions

A start condition is met when the SCLA0 pin is at high level and the SDAA0 pin changes from high level to low level. The start conditions for the SCLA0 pin and SDAA0 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 16-14. Start Conditions



A start condition is output when bit 1 (STT0) of IICA control register 0 (IICACTL0) is set (1) after a stop condition has been detected (SPD0: Bit 0 of the IICA status register 0 (IICAS0) = 1). When a start condition is detected, bit 1 (STD0) of IICAS0 is set (1).

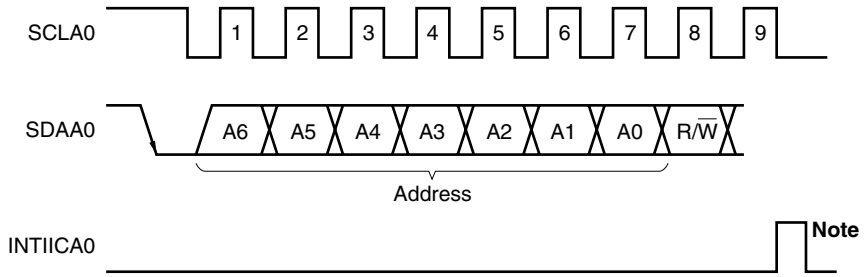
16.5.2 Addresses

The address is defined by the 7 bits of data that follow the start condition.

An address is a 7-bit data segment that is output in order to select one of the slave devices that are connected to the master device via the bus lines. Therefore, each slave device connected via the bus lines must have a unique address.

The slave devices include hardware that detects the start condition and checks whether or not the 7-bit address data matches the data values stored in the slave address register 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 16-15. Address



Note INTIICA0 is not issued if data other than a local address or extension code is received during slave device operation.

Addresses are output when a total of 8 bits consisting of the slave address and the transfer direction described in **16.5.3 Transfer direction specification** are written to the IICA shift register (IICA). The received addresses are written to IICA.

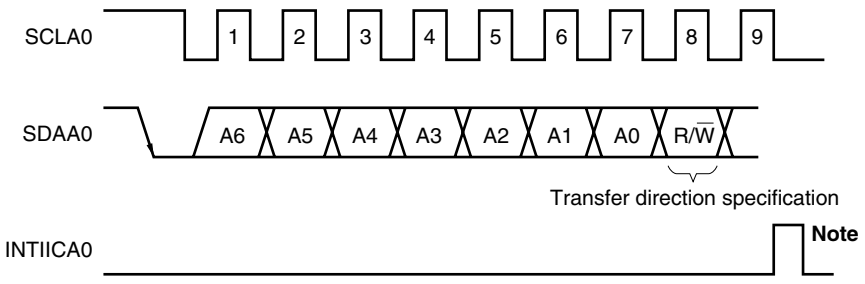
The slave address is assigned to the higher 7 bits of IICA.

16.5.3 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 16-16. Transfer Direction Specification



Note INTIICA0 is not issued if data other than a local address or extension code is received during slave device operation.

16.5.4 Acknowledge ($\overline{\text{ACK}}$)

$\overline{\text{ACK}}$ is used to check the status of serial data at the transmission and reception sides.

The reception side returns $\overline{\text{ACK}}$ each time it has received 8-bit data.

The transmission side usually receives $\overline{\text{ACK}}$ after transmitting 8-bit data. When $\overline{\text{ACK}}$ is returned from the reception side, it is assumed that reception has been correctly performed and processing is continued. Whether $\overline{\text{ACK}}$ has been detected can be checked by using bit 2 (ACKD0) of the IICA status register 0 (IICAS0).

When the master receives the last data item, it does not return $\overline{\text{ACK}}$ and instead generates a stop condition. If a slave does not return $\overline{\text{ACK}}$ after receiving data, the master outputs a stop condition or restart condition and stops transmission. If $\overline{\text{ACK}}$ 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.

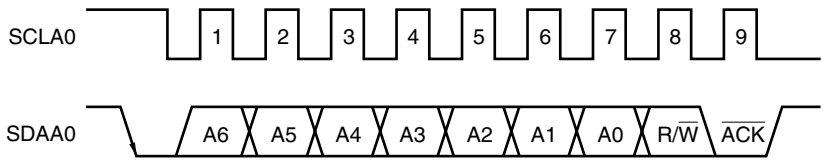
To generate $\overline{\text{ACK}}$, the reception side makes the SDAA0 line low at the ninth clock (indicating normal reception).

Automatic generation of $\overline{\text{ACK}}$ is enabled by setting bit 2 (ACKE0) of IICA control register 0 (IICACTL0) to 1. Bit 3 (TRC0) of the IICAS0 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 $\overline{\text{ACK}}$ 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 16-17. $\overline{\text{ACK}}$



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.

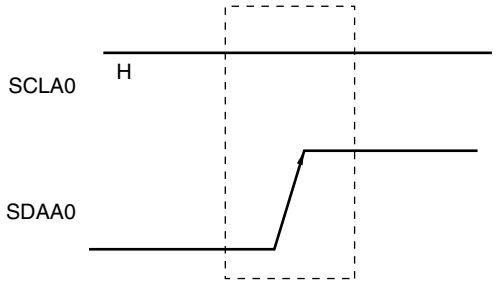
- When 8-clock wait state is selected (bit 3 (WTIM0) of IICACTL0 register = 0):
  By setting ACKE0 to 1 before releasing the wait state, $\overline{\text{ACK}}$ is generated at the falling edge of the eighth clock of the SCLA0 pin.
- When 9-clock wait state is selected (bit 3 (WTIM0) of IICACTL0 register = 1):
  $\overline{\text{ACK}}$ is generated by setting ACKE0 to 1 in advance.

16.5.5 Stop condition

When the SCLA0 pin is at high level, changing the SDAA0 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 16-18. Stop Condition



A stop condition is generated when bit 0 (SPT0) of IICA control register 0 (IICACTL0) is set to 1. When the stop condition is detected, bit 0 (SPD0) of the IICA status register 0 (IICAS0) is set to 1 and INTIICA0 is generated when bit 4 (SPIE0) of IICACTL0 is set to 1.

16.5.6 Wait

The wait is used to notify the communication partner that a device (master or slave) is preparing to transmit or receive data (in a wait state).

Setting the SCLA0 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 16-19. 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)

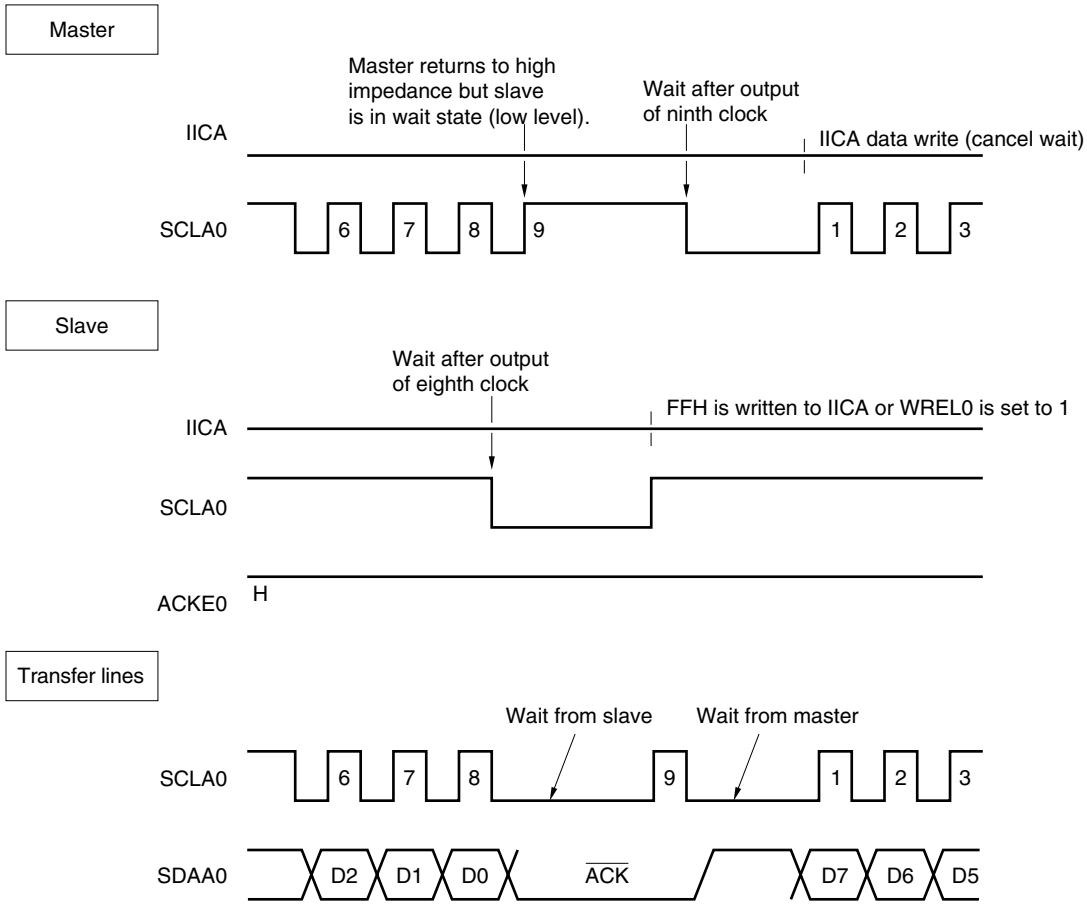
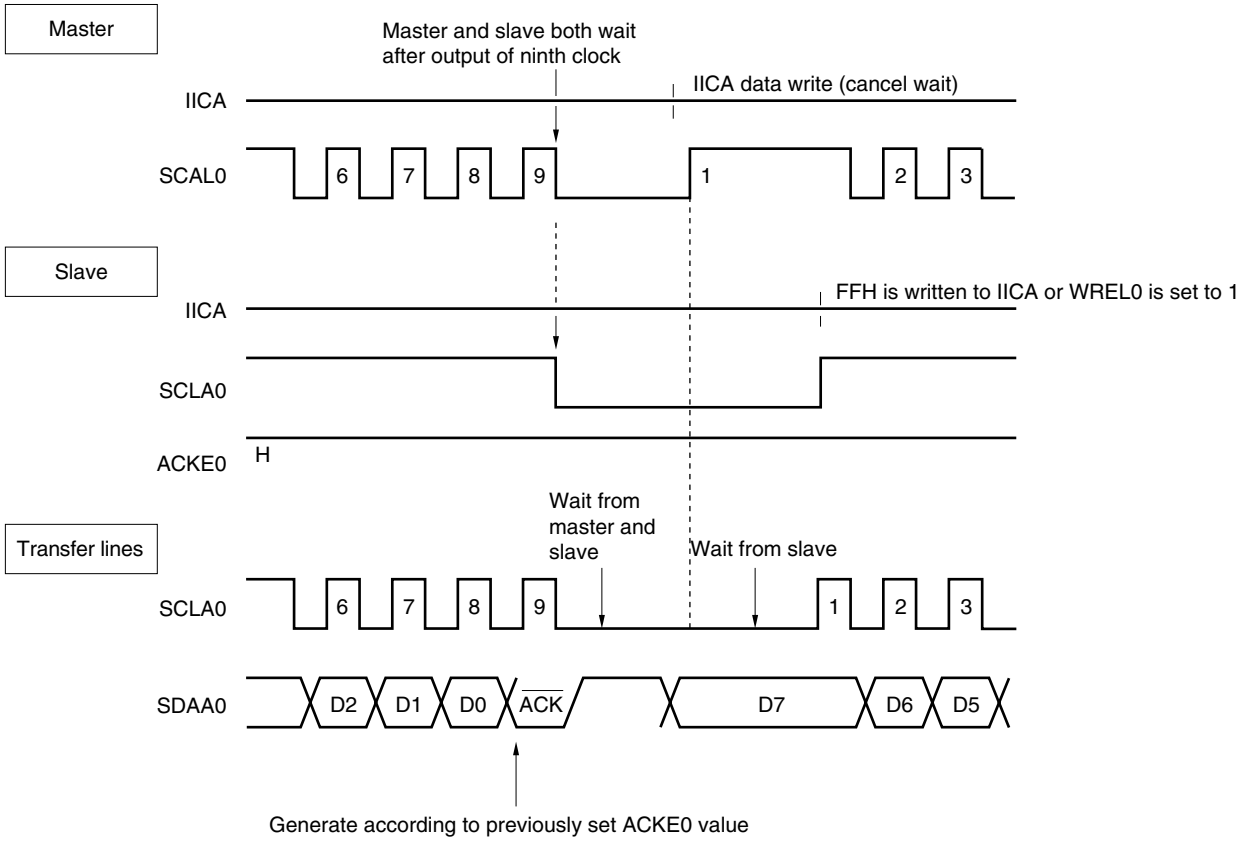


Figure 16-19. Wait (2/2)

(2) When master and slave devices both have a nine-clock wait (master transmits, slave receives, and ACKE0 = 1)



Remark ACKE0: Bit 2 of IICA control register 0 (IICACTL0)
WREL0: Bit 5 of IICA control register 0 (IICACTL0)

A wait may be automatically generated depending on the setting of bit 3 (WTIM0) of IICA control register 0 (IICACTL0).

Normally, the receiving side cancels the wait state when bit 5 (WREL0) of IICACTL0 is set to 1 or when FFH is written to the IICA shift register (IICA), and the transmitting side cancels the wait state when data is written to IICA.

The master device can also cancel the wait state via either of the following methods.

- By setting bit 1 (STT0) of IICACTL0 to 1
- By setting bit 0 (SPT0) of IICACTL0 to 1

16.5.7 Canceling wait

The I^2C usually cancels a wait state by the following processing.

- Writing data to IICA shift register (IICA)
- Setting bit 5 (WREL0) of IICA control register 0 (IICACTL0) (canceling wait)
- Setting bit 1 (STT0) of IICACTL0 register (generating start condition)[Note]
- Setting bit 0 (SPT0) of IICACTL0 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 IICA.

To receive data after canceling a wait state, or to complete data transmission, set bit 5 (WREL0) of IICA control register 0 (IICACTL0) to 1.

To generate a restart condition after canceling a wait state, set bit 1 (STT0) of IICACTL0 to 1.

To generate a stop condition after canceling a wait state, set bit 0 (SPT0) of IICACTL0 to 1.

Execute the canceling processing only once for one wait state.

If, for example, data is written to IICA after canceling a wait state by setting WREL0 to 1, an incorrect value may be output to SDAA0 because the timing for changing the SDAA0 line conflicts with the timing for writing IICA.

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 IICACTL0, so that the wait state can be canceled.

Caution If a processing to cancel a wait state is executed when WUP (bit 7 of IICA control register 1 (IICACTL1)) = 1, the wait state will not be canceled.

16.5.8 Interrupt request (INTIICA0) generation timing and wait control

The setting of bit 3 (WTIM0) of IICA control register 0 (IICACTL0) determines the timing by which INTIICA0 is generated and the corresponding wait control, as shown in Table 16-2.

Table 16-2. INTIICA0 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 INTIICA0 signal and wait period occurs at the falling edge of the ninth clock only when there is a match with the address set to the slave address register 0 (SVA0).
At this point, $\overline{ACK}$ is generated regardless of the value set to IICACTL0's bit 2 (ACKE0). For a slave device that has received an extension code, INTIICA0 occurs at the falling edge of the eighth clock.
However, if the address does not match after restart, INTIICA0 is generated at the falling edge of the 9th clock, but wait does not occur.

2. If the received address does not match the contents of the slave address register 0 (SVA0) and extension code is not received, neither INTIICA0 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 IICA shift register (IICA)
- Setting bit 5 (WREL0) of IICA control register 0 (IICACTL0) (canceling wait)
- Setting bit 1 (STT0) of IICACTL0 register (generating start condition)Note
- Setting bit 0 (SPT0) of IICACTL0 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

INTIICA0 is generated when a stop condition is detected (only when SPIE0 = 1).

16.5.9 Address match detection method

In I^2C bus mode, the master device can select a particular slave device by transmitting the corresponding slave address.

Address match can be detected automatically by hardware. An interrupt request (INTIICA0) occurs when a local address has been set to the 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.

16.5.10 Error detection

In I^2C bus mode, the status of the serial data bus (SDAA0) during data transmission is captured by the IICA shift register (IICA) of the transmitting device, so the IICA data prior to transmission can be compared with the transmitted IICA data to enable detection of transmission errors. A transmission error is judged as having occurred when the compared data values do not match.

16.5.11 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 (INTIICA0) is issued at the falling edge of the eighth clock. The local address stored in the 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 INTIICA0 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 IICA status register 0 (IICAS0)
COI0: Bit 4 of IICA status register 0 (IICAS0)

(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 IICA control register 0 (IICACTL0) to 1 to set the standby mode for the next communication operation.

Table 16-3. Bit Definitions of Major Extension Codes

Slave Address	R/W Bit	Description
0 0 0 0 0 0 0	0	General call address
1 1 1 1 0 x x	0	10-bit slave address specification (during address authentication)
1 1 1 1 0 x x	1	10-bit slave address specification (after address match, when read command is issued)

Remark See the I^2C bus specifications issued by NXP Semiconductors for details of extension codes other than those described above.

16.5.12 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 the IICA status register 0 (IICAS0) is set (1) via the timing by which the arbitration loss occurred, and the SCLA0 and SDAA0 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 **16.5.8 Interrupt request (INTIICA0) generation timing and wait control**.

Remark STD0: Bit 1 of IICA status register 0 (IICAS0)
STT0: Bit 1 of IICA control register 0 (IICACTL0)

Figure 16-20. Arbitration Timing Example

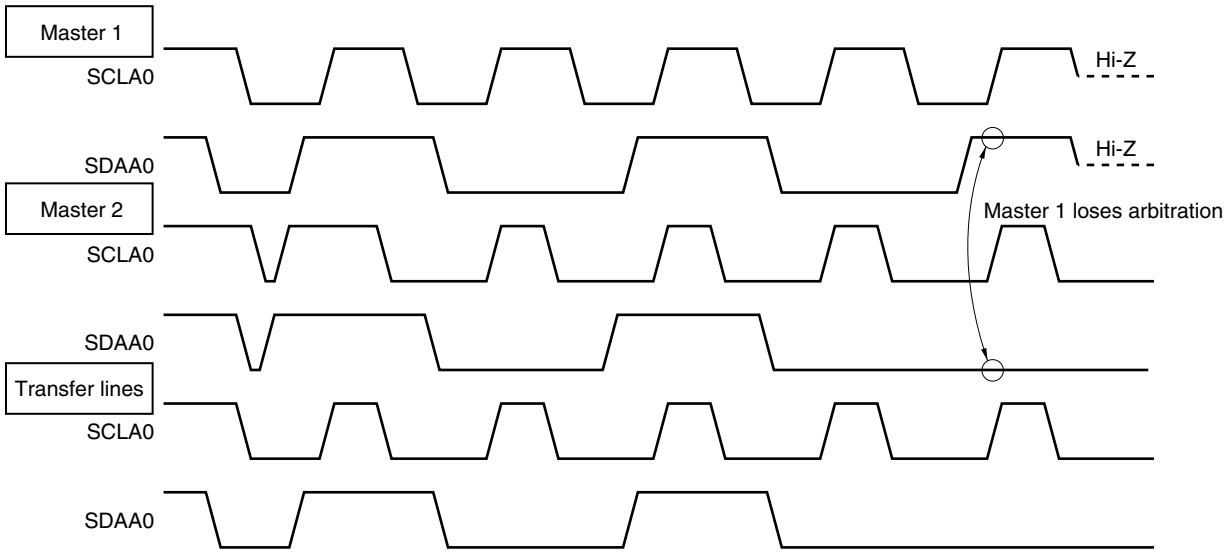


Table 16-4. Status During Arbitration and Interrupt Request Generation Timing

Status During Arbitration	Interrupt Request Generation Timing
During address transmission	At falling edge of eighth or ninth clock following byte transfer[Note 1]
Read/write data after address transmission	
During extension code transmission	
Read/write data after extension code transmission	
During data transmission	
During $\overline{\text{ACK}}$ transfer period after data transmission	
When restart condition is detected during data transfer	
When stop condition is detected during data transfer	When stop condition is generated (when 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 SCLA0 is at low level while attempting to generate a restart condition	

Notes 1. When WTIM0 (bit 3 of IICA control register 0 (IICACTL0)) = 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 IICA control register 0 (IICACTL0)

16.5.13 Wakeup function

The I^2C bus slave function is a function that generates an interrupt request signal (INTIICA0) when a local address and extension code have been received.

This function makes processing more efficient by preventing unnecessary INTIICA0 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 IICA control register 0 (IICACTL0) is set regardless of the wakeup function, and this determines whether interrupt requests are enabled or disabled.

To use the wakeup function in the STOP mode, set WUP to 1. Addresses can be received regardless of the operation clock. An interrupt request signal (INTIICA0) is also generated when a local address and extension code have been received. Operation returns to normal operation by using an instruction to clear (0) the WUP bit after this interrupt has been generated.

Figure 16-21 shows the flow for setting WUP and Figure 16-22 shows the flow for clear WUP = 0 upon an address match.

Figure 16-21. Flow When Setting WUP = 1

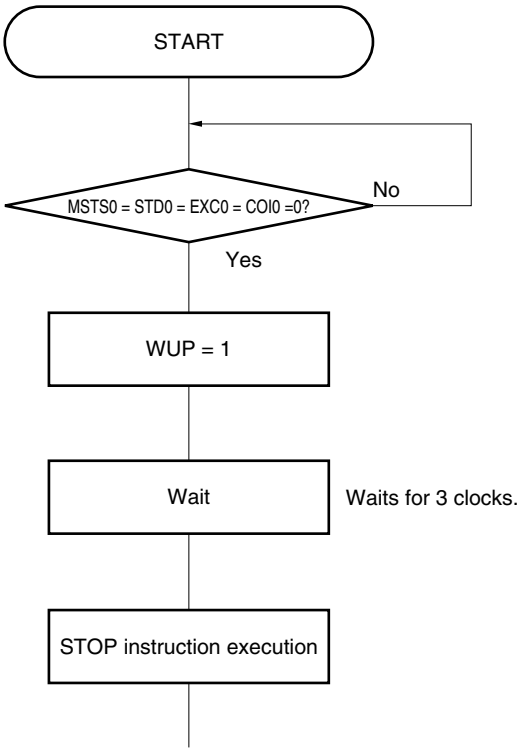
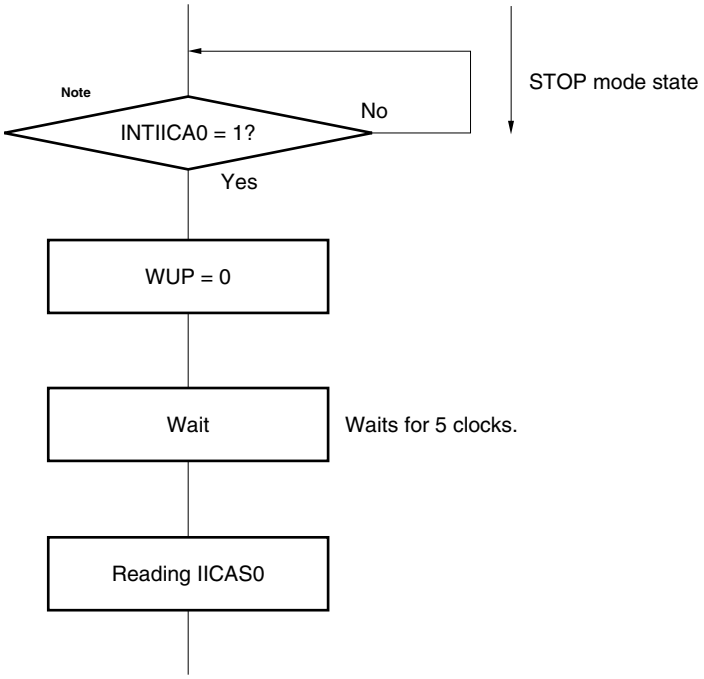


Figure 16-22. Flow When Setting WUP = 0 upon Address Match (Including Extension Code Reception)



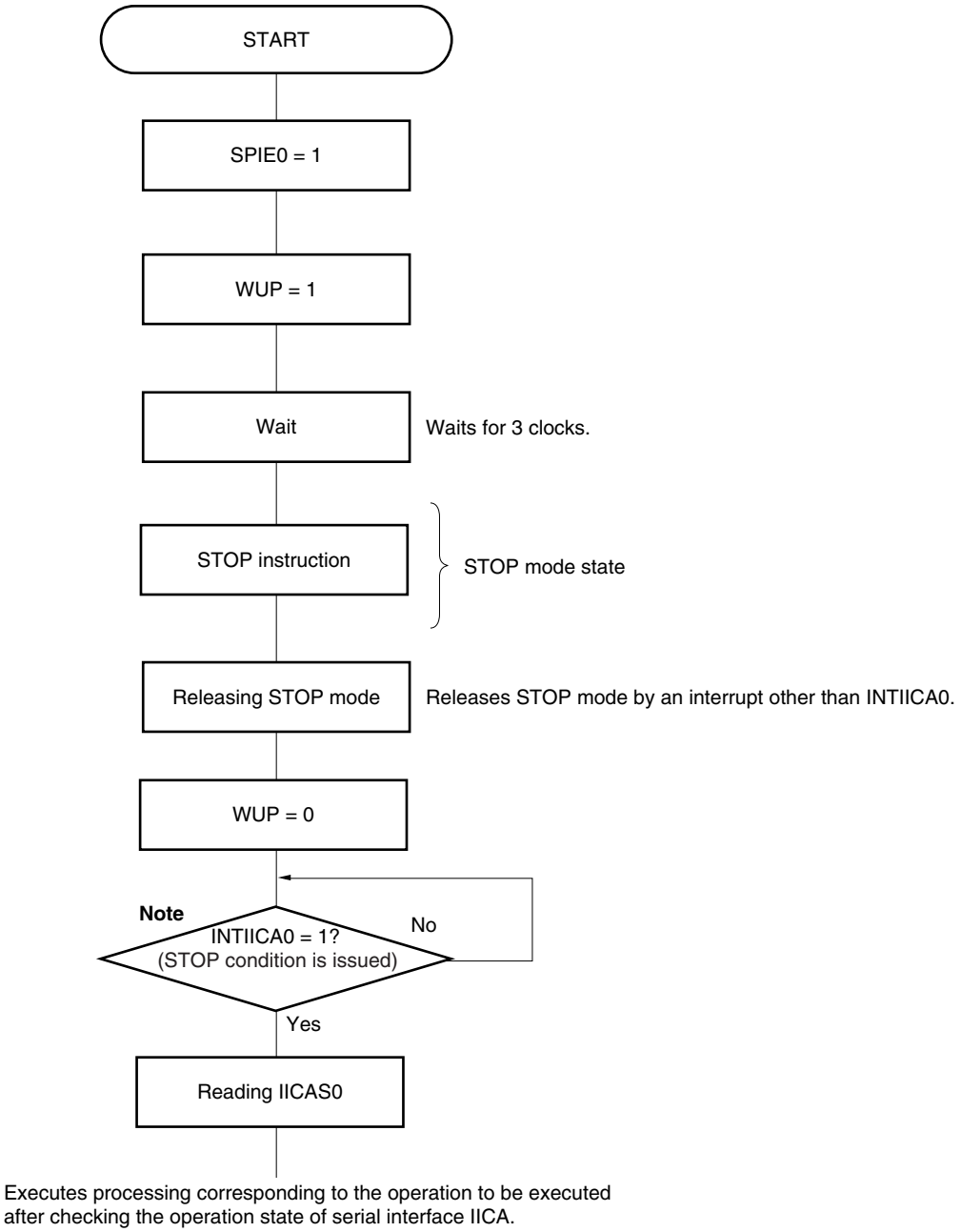
Executes processing corresponding to the operation to be executed after checking the operation state of serial interface IICA.

Note Perform the processing after “INTIICA0 = 1” also when an INTIICA0 vector interrupt occurs.

Use the following flows to perform the processing to release the STOP mode other than by an interrupt request (INTIICA0) generated from serial interface IICA.

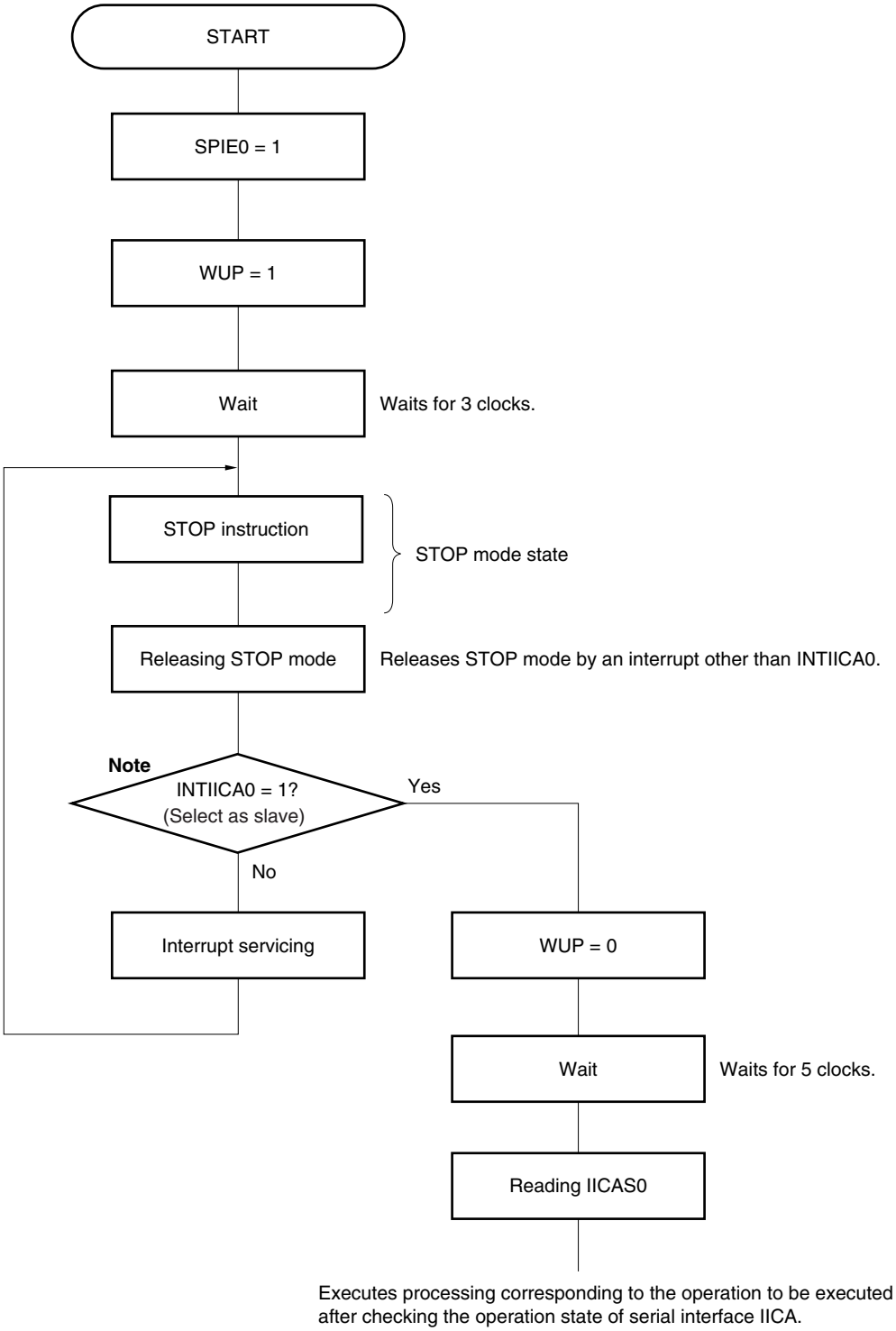
- Master device operation: Flow shown in Figure 16-23
- Slave device operation: Same as the flow in Figure 16-22 or Figure 16-24

Figure 16-23. To Specify IICA as Master After STOP Mode Is Released by Other Than INTIICA0



Note Even if IICA is specified as a slave, INTIICA0 is still set to 1. To specify IICA as a master, execute the above processing after the slave processing is complete.

Figure 16-24. To Specify IICA as Master After STOP Mode Is Released by Other Than INTIICA (If IICA Does Not Have to Operate as Master)



Note INTIICA0 is set to 1 even if a STOP condition is generated.

16.5.14 Communication reservation

(1) When communication reservation function is enabled (bit 0 (IICRSV) of IICA flag register 0 (IICAF0) = 0)

To start master device communications when not currently using a bus, a communication reservation can be made to enable transmission of a start condition when the bus is released. There are two modes under which the bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled ($\overline{ACK}$ is not returned and the bus was released by setting bit 6 (LREL0) of IICA control register 0 (IICACTL0) to 1 and saving communication).

If bit 1 (STT0) of IICACTL0 is set to 1 while the bus is not used (after a stop condition is detected), a start condition is automatically generated and wait state is set.

If an address is written to the IICA shift register (IICA) after bit 4 (SPIE0) of IICACTL0 was set to 1, and it was detected by generation of an interrupt request signal (INTIICA0) that the bus was released (detection of the stop condition), then the device automatically starts communication as the master. Data written to IICA 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 the IICA status register 0 (IICAS0)) after STT0 is set to 1 and the wait time elapses.

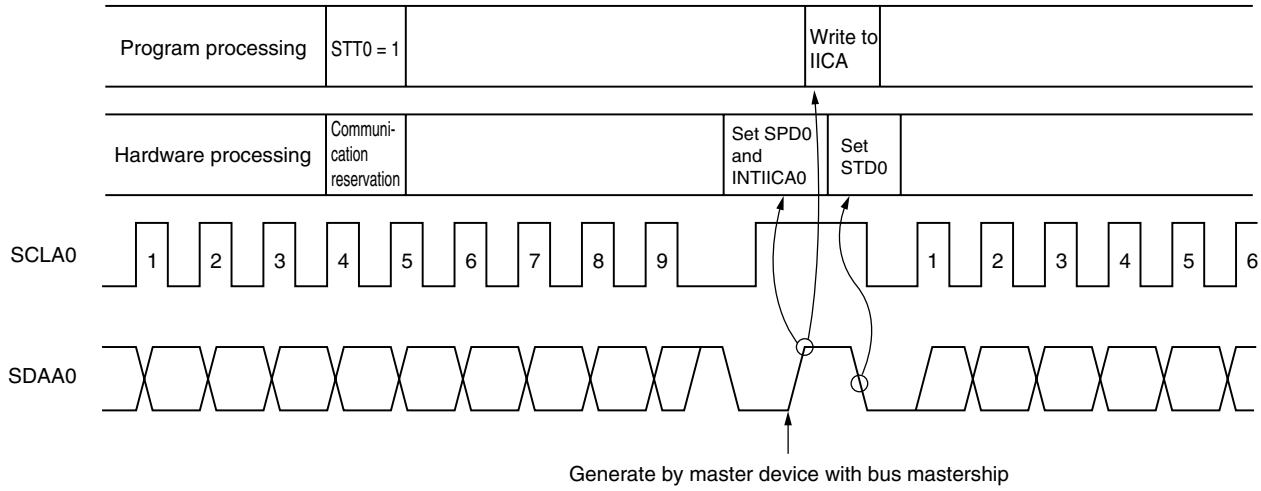
Use software to secure the wait time calculated by the following expression.

Wait time from setting STT0 = 1 to checking the MSTS0 flag: (IICWL setting value + IICWH setting value + 4) + $t_F \times 2 \times f_{PRS}$ [clocks]

Remark IICWL: IICA low-level width setting register
IICWH: IICA high-level width setting register
t_F: SDAA0 and SCLA0 signal falling times (refer to **CHAPTER 28 ELECTRIC SPECIFITION**)
f_{PRS}: Peripheral hardware clock frequency

Figure 16-25 shows the communication reservation timing.

Figure 16-25. Communication Reservation Timing



Remark IICA: IICA shift register
STT0: Bit 1 of IICA control register 0 (IICACTL0)
STD0: Bit 1 of IICA status register 0 (IICAS0)
SPD0: Bit 0 of IICA status register 0 (IICAS0)

Communication reservations are accepted via the timing shown in Figure 16-26. After bit 1 (STD0) of the IICA status register 0 (IICAS0) is set to 1, a communication reservation can be made by setting bit 1 (STT0) of IICA control register 0 (IICCATL0) to 1 before a stop condition is detected.

Figure 16-26. Timing for Accepting Communication Reservations

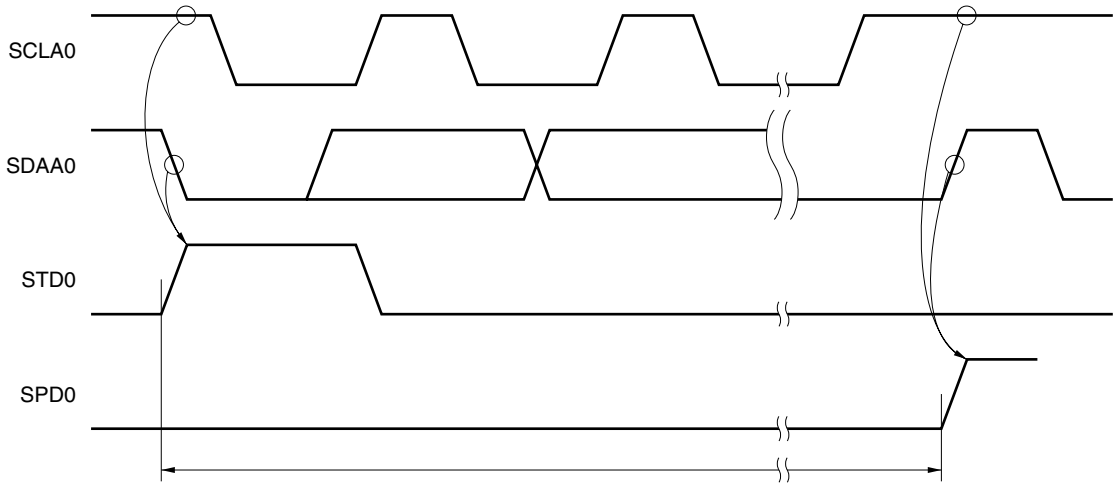
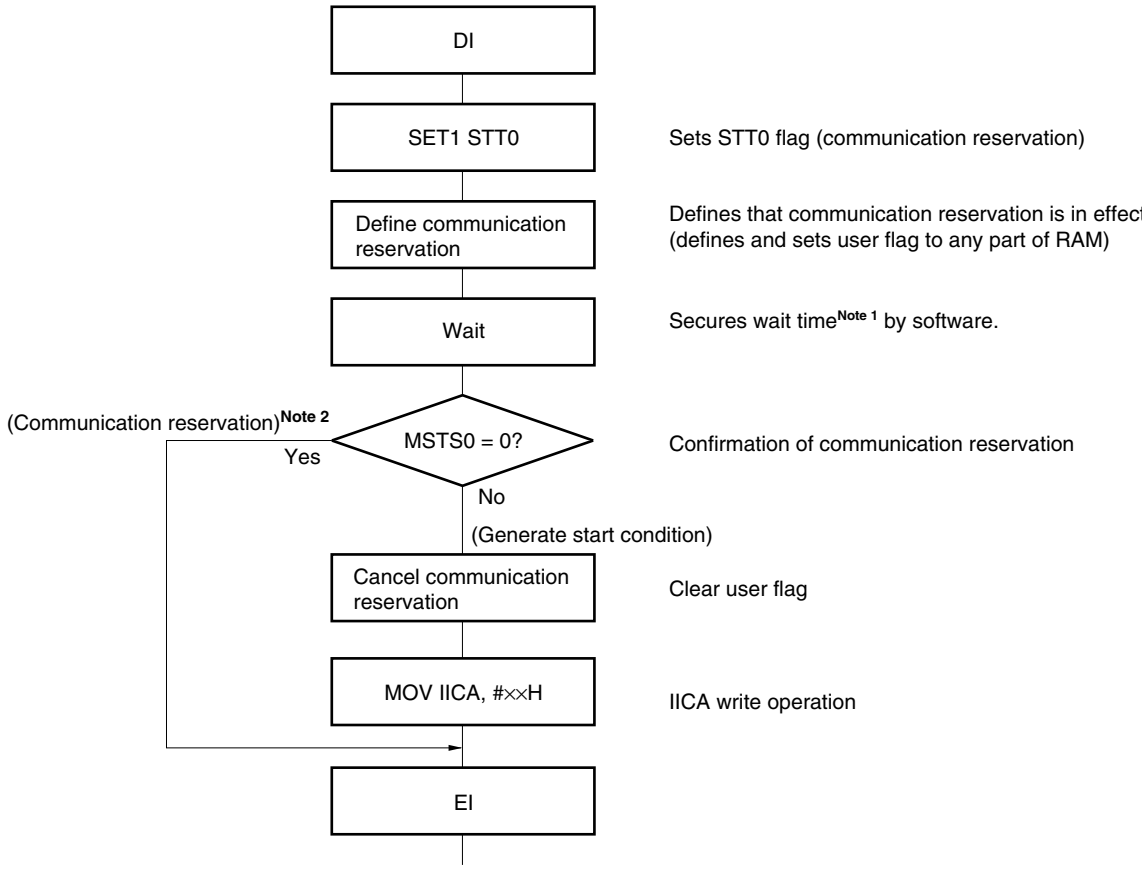


Figure 16-27 shows the communication reservation protocol.

Figure 16-27. Communication Reservation Protocol



Notes 1. The wait time is calculated as follows.
(IICWL setting value + IICWH setting value + 4) + $t_F \times 2 \times f_{PRS}$ [clocks]

2. The communication reservation operation executes a write to the IICA shift register (IICA) when a stop condition interrupt request occurs.

Remark STT0: Bit 1 of IICA control register 0 (IICACTL0)
MSTS0: Bit 7 of IICA status register 0 (IICAS0)
IICA: IICA shift register
IICWL: IICA low-level width setting register
IICWH: IICA high-level width setting register
t_F: SDAA0 and SCLA0 signal falling times (refer to **CHAPTER 28 ELECTRIC SPECIFITION**)
f_{PRS}: Peripheral hardware clock frequency

(2) When communication reservation function is disabled (bit 0 (IICRSV) of IICA flag register 0 (IICAF0) = 1)

When bit 1 (STT0) of IICA control register 0 (IICACTL0) 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 ($\overline{\text{ACK}}$ is not returned and the bus was released by setting bit 6 (LREL0) of IICACTL0 to 1 and saving communication)

To confirm whether the start condition was generated or request was rejected, check STCF (bit 7 of IICAF0). It takes up to 5 clocks until STCF is set to 1 after setting STT0 = 1. Therefore, secure the time by software.

16.5.15 Cautions

(1) When STCEN (bit 1 of IICA flag register 0 (IICAF0)) = 0

Immediately after I^2C operation is enabled (IICE0 = 1), the bus communication status (IICBSY (bit 6 of IICAF0) = 1) is recognized regardless of the actual bus status. When changing from a mode in which no stop condition has been detected to a master device communication mode, first generate a stop condition to release the bus, then perform master device communication.

When using multiple masters, it is not possible to perform master device communication when the bus has not been released (when a stop condition has not been detected).

Use the following sequence for generating a stop condition.

<1> Set IICA control register 1 (IICACTL1).
<2> Set bit 7 (IICE0) of IICA control register 0 (IICACTL0) to 1.
<3> Set bit 0 (SPT0) of IICACTL0 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 IICA control register 0 (IICACTL0)) = 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 SDAA0 pin is low and the SCLA0 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 IICACTL0 to 0 to disable generation of an interrupt request signal (INTIICA0) when the stop condition is detected.
<2> Set bit 7 (IICE0) of IICACTL0 to 1 to enable the operation of I^2C.
<3> Wait for detection of the start condition.
<4> Set bit 6 (LREL0) of IICCTL0 to 1 before $\overline{ACK}$ is returned (4 to 80 clocks after setting IICE0 to 1), to forcibly disable detection.

(4) Setting STT0 and SPT0 (bits 1 and 0 of IICACTL0) again after they are set and before they are cleared to 0 is prohibited.

(5) When transmission is reserved, set SPIE0 (bit 4 of IICACTL0) to 1 so that an interrupt request is generated when the stop condition is detected. Transfer is started when communication data is written to IICA after the interrupt request is generated. Unless the interrupt is generated when the stop condition is detected, the device stops in the wait state because the interrupt request is not generated when communication is started. However, it is not necessary to set SPIE0 to 1 when MSTS0 (bit 7 of IICAS0) is detected by software.

16.5.16 Communication operations

The following shows three operation procedures with the flowchart.

(1) Master operation in single master system

The flowchart when using the 78K0R/IE3 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/IE3 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/IE3 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

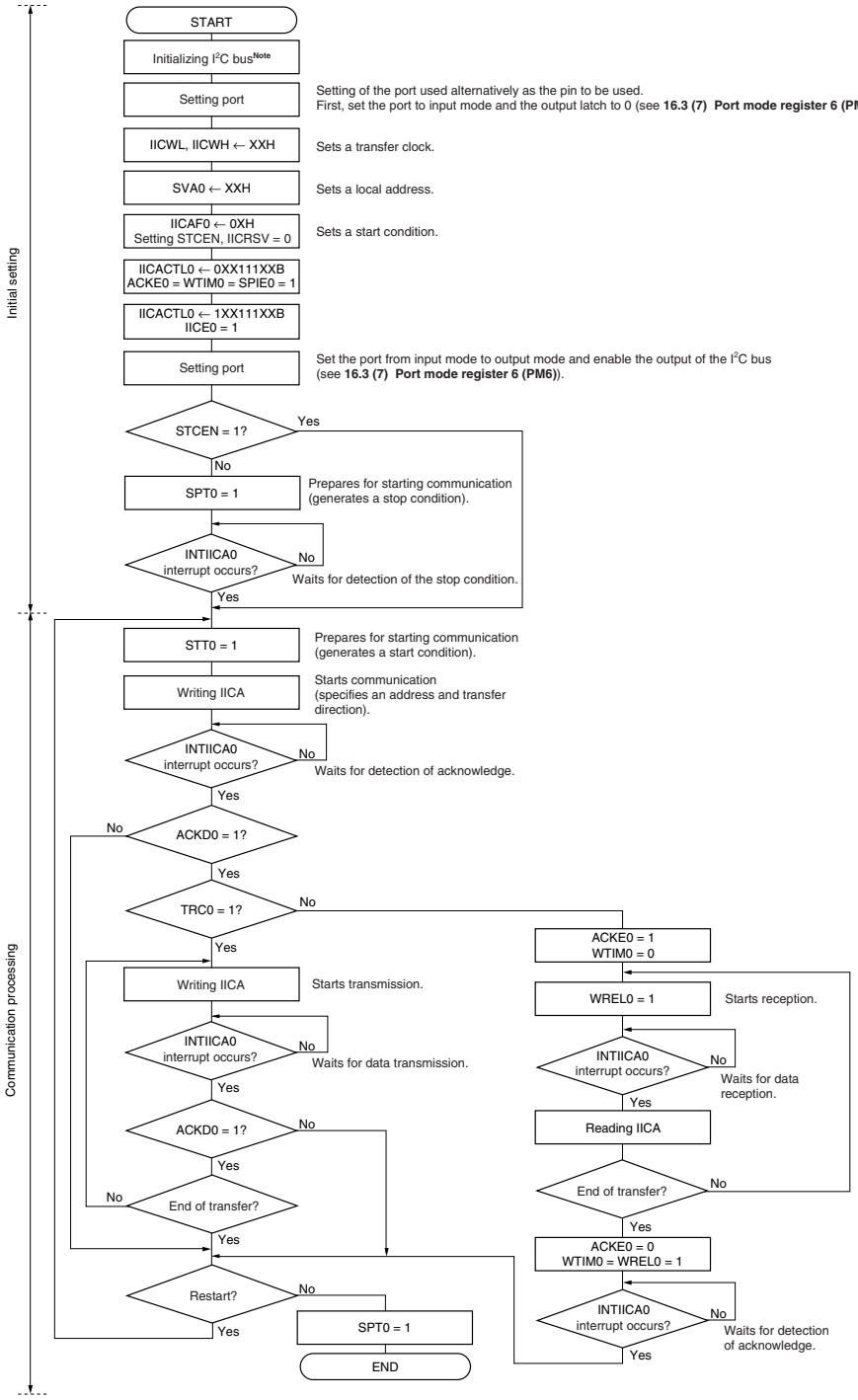
An example of when the 78K0R/IE3 is used as the I^2C bus slave is shown below.

When used as the slave, operation is started by an interrupt. Execute the initial settings at startup, then wait for the INTIICA0 interrupt occurrence (communication waiting). When an INTIICA0 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 16-28. Master Operation in Single-Master System

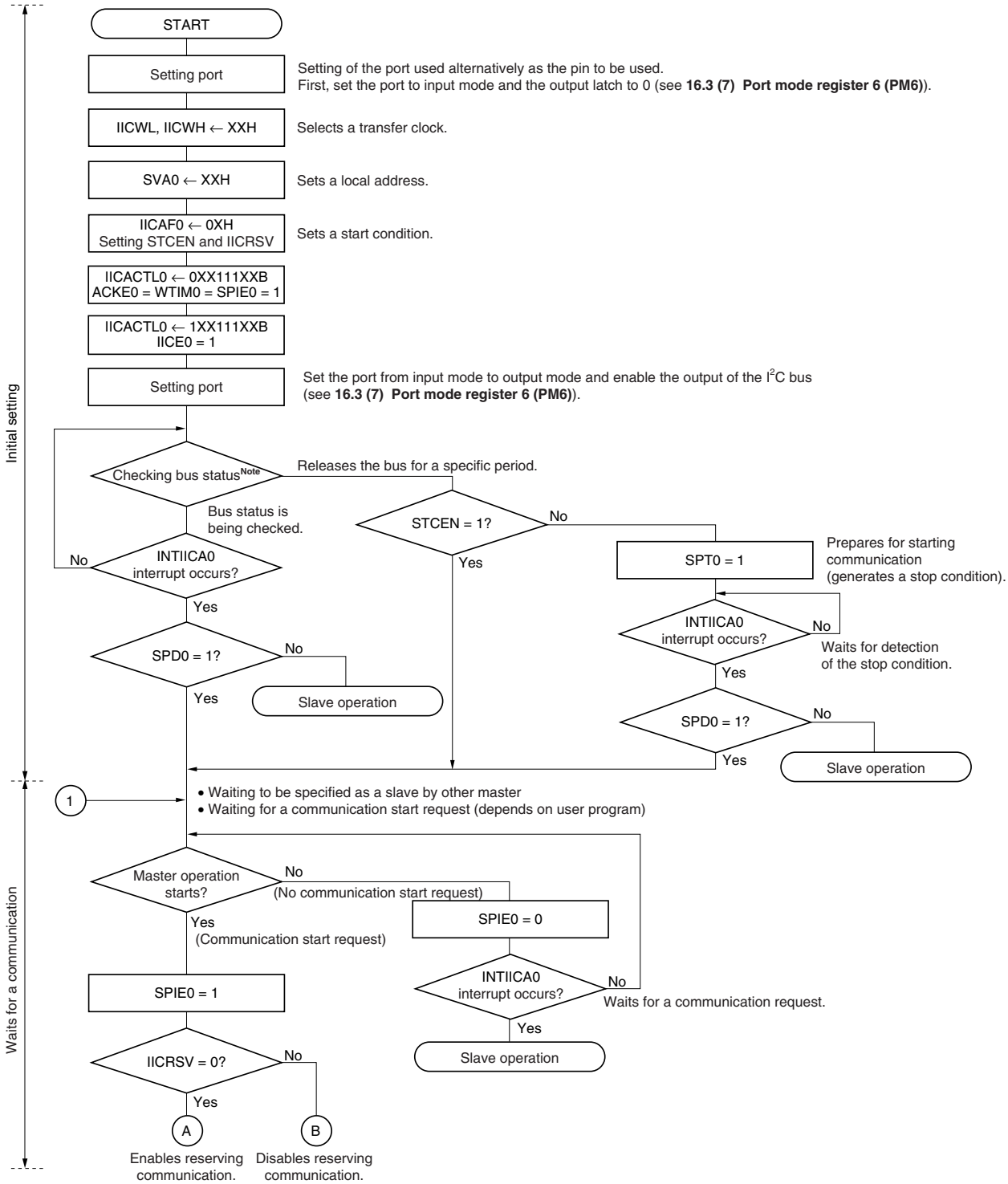


Note Release (SCLA0 and SDAA0 pins = high level) the I^2C bus in conformance with the specifications of the product that is communicating. If EEPROM is outputting a low level to the SDAA0 pin, for example, set the SCLA0 pin in the output port mode, and output a clock pulse from the output port until the SDAA0 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.

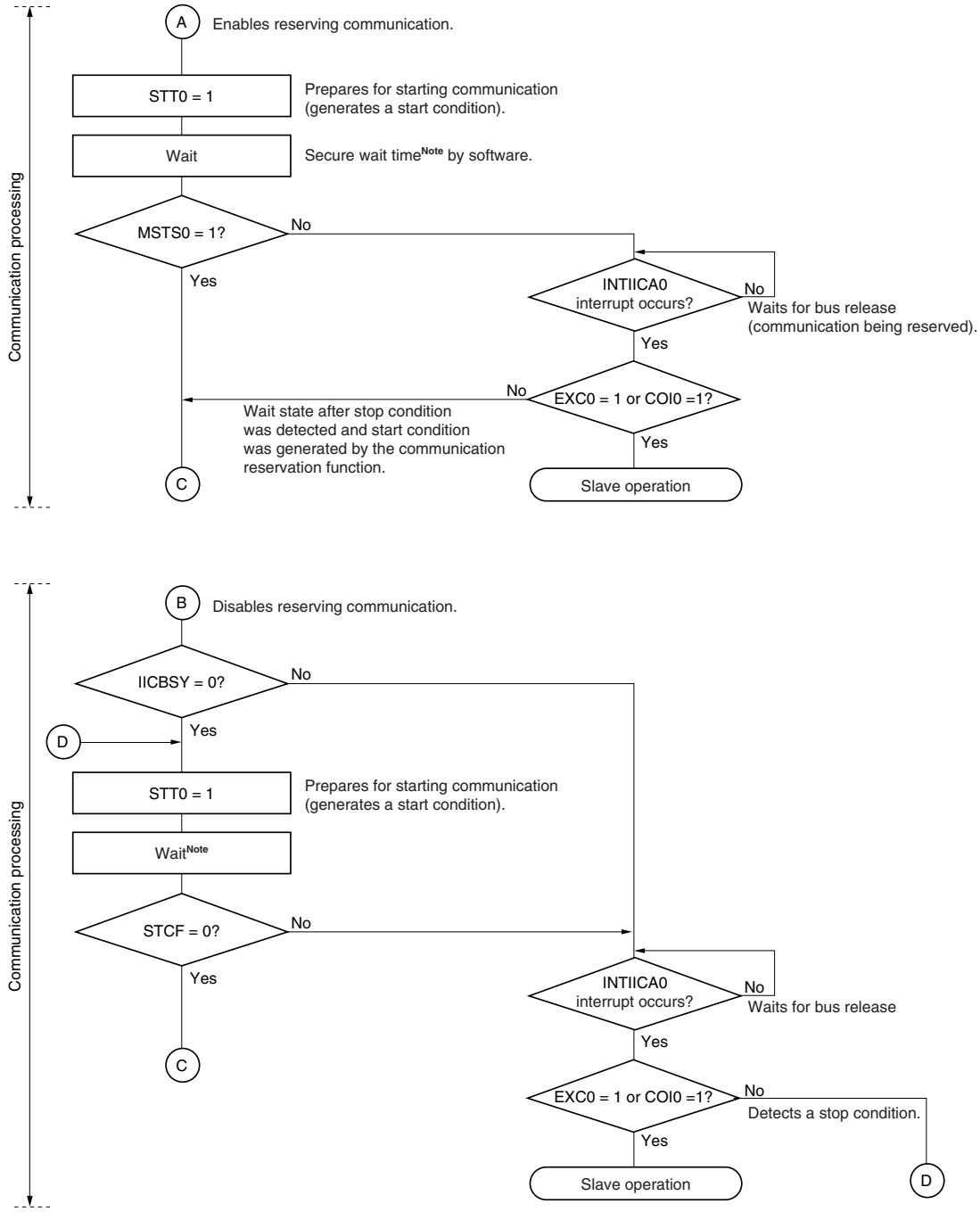
(2) Master operation in multi-master system

Figure 16-29. 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 SDAA0 pin is constantly at low level, decide whether to release the I^2C bus (SCLA0 and SDAA0 pins = high level) in conformance with the specifications of the product that is communicating.

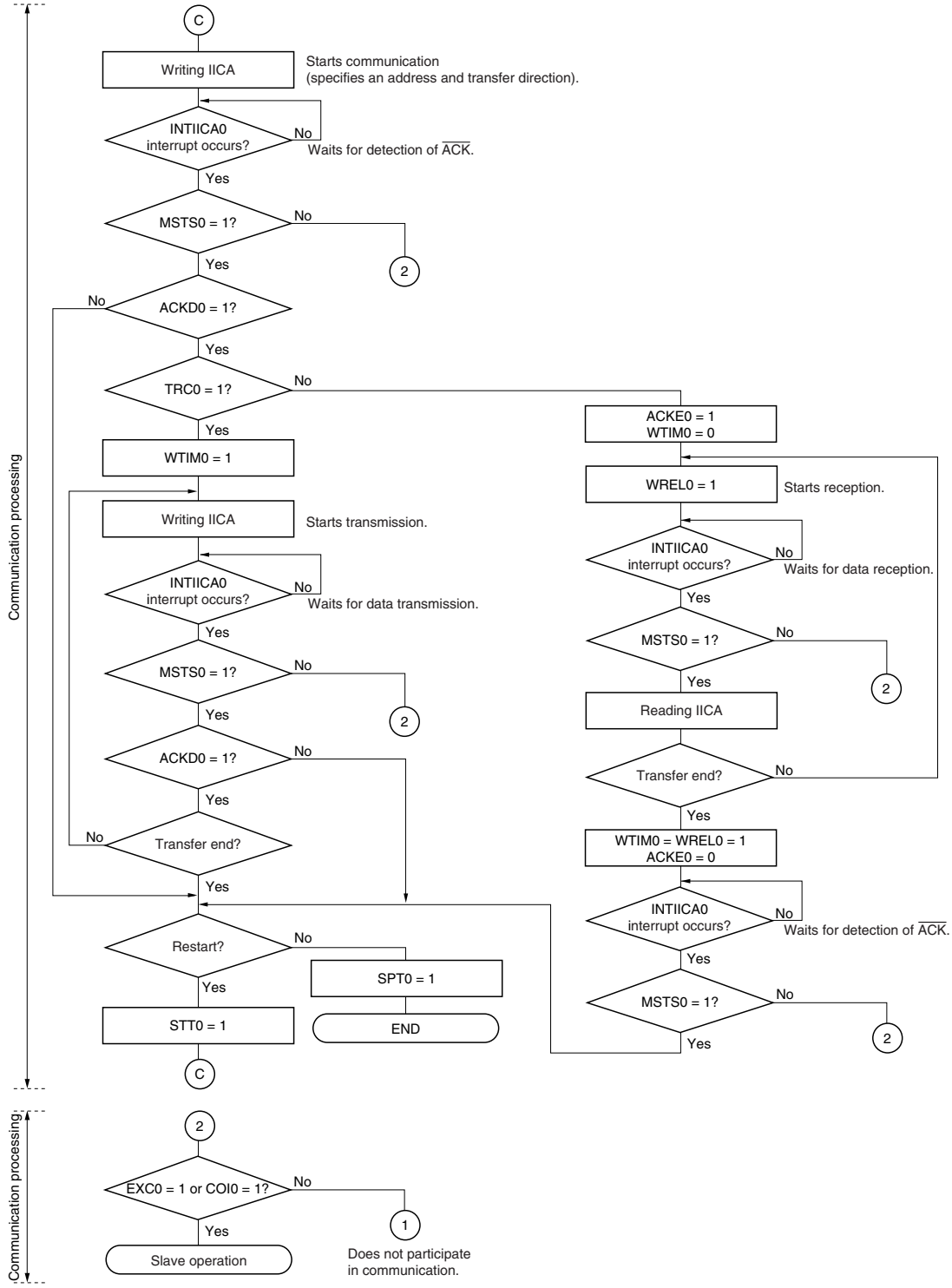
Figure 16-29. Master Operation in Multi-Master System (2/3)



Note The wait time is calculated as follows.
(IICWL setting value + IICWH setting value + 4) + $t_F \times 2 \times f_{PRS}$ [clocks]

Remark IICWL: IICA low-level width setting register
IICWH: IICA high-level width setting register
t_F: SDAA0 and SCLA0 signal falling times (see **CHAPTER 28 ELECTRICAL SPECIFICATIONS**)
f_{PRS}: CPU/peripheral hardware clock frequency

Figure 16-29. Master Operation in Multi-Master System (3/3)



Remarks 1. Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

2. To use the device as a master in a multi-master system, read the MSTS0 bit each time interrupt INTIICA0 has occurred to check the arbitration result.

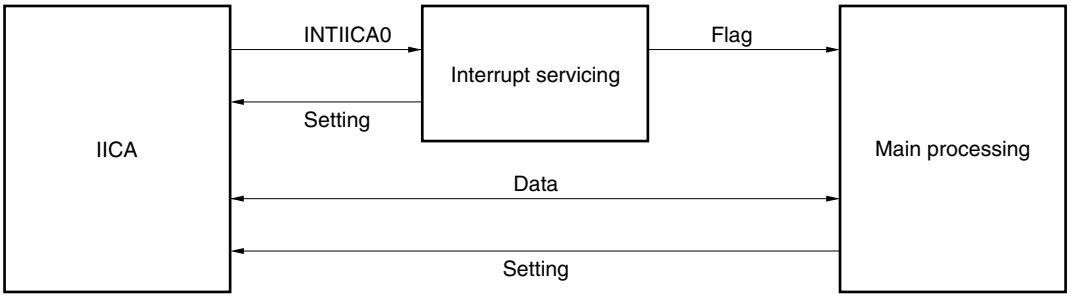
3. To use the device as a slave in a multi-master system, check the status by using the IICAS0 and IICAF0 registers each time interrupt INTIICA0 has occurred, and determine the processing to be performed next.

(3) Slave operation

The processing procedure of the slave operation is as follows.

Basically, the slave operation is event-driven. Therefore, processing by the INTIICA0 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 INTIICA0 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 INTIICA0.

<1> Communication mode flag

This flag indicates the following two communication statuses.

- Clear mode: Status in which data communication is not performed
- Communication mode: Status in which data communication is performed (from valid address detection to stop condition detection, no detection of $\overline{\text{ACK}}$ from master, address mismatch)

<2> Ready flag

This flag indicates that data communication is enabled. Its function is the same as the INTIICA0 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.

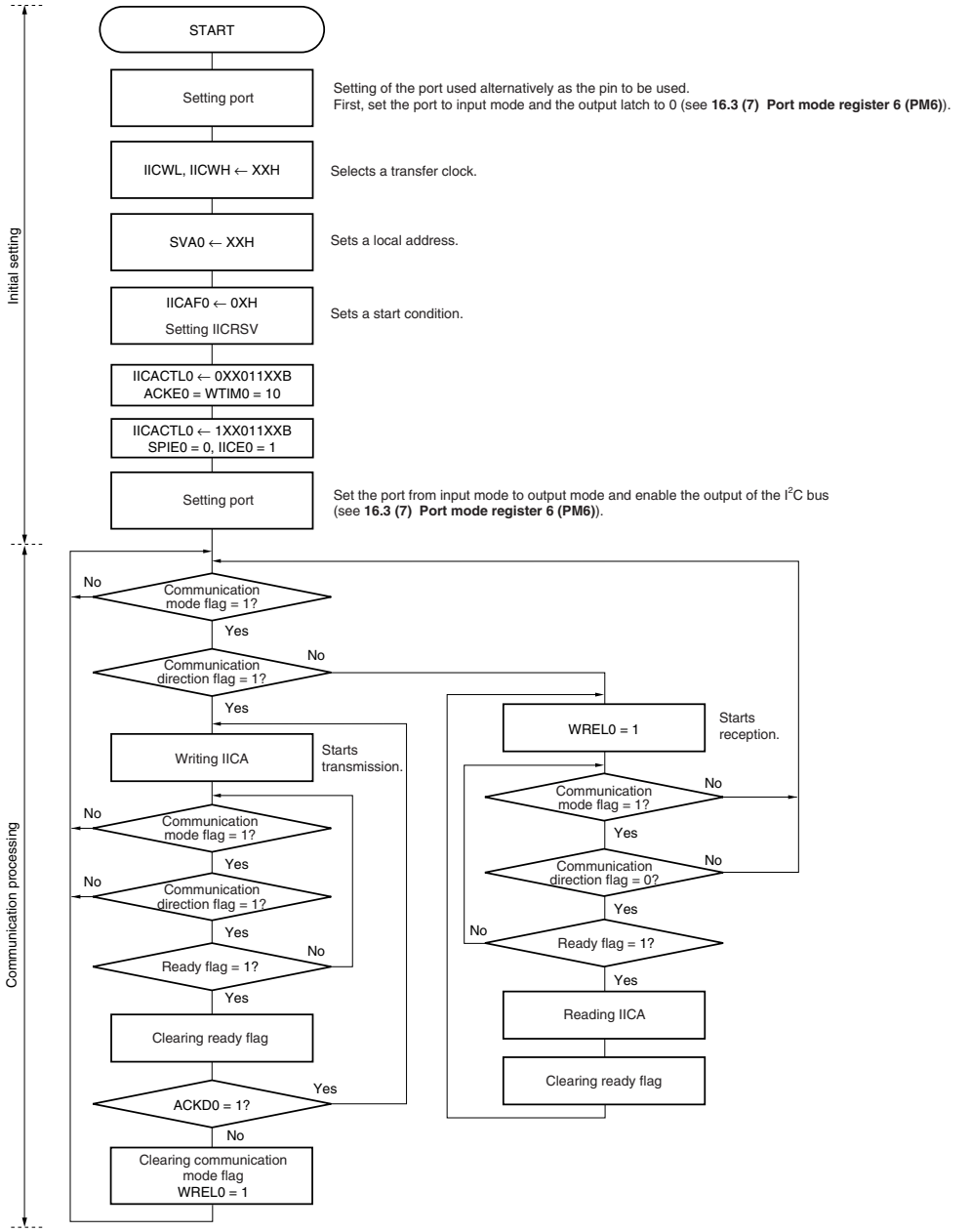
The main processing of the slave operation is explained next.

Start serial interface IICA and wait until communication is enabled. When communication is enabled, execute communication by using the communication mode flag and ready flag (processing of the stop condition and start condition is performed by an interrupt. Here, check the status by using the flags).

The transmission operation is repeated until the master no longer returns $\overline{ACK}$. If $\overline{ACK}$ is not returned from the master, communication is completed.

For reception, the necessary amount of data is received. When communication is completed, $\overline{ACK}$ is not returned as the next data. After that, the master generates a stop condition or restart condition. Exit from the communication status occurs in this way.

Figure 16-30. Slave Operation Flowchart (1)



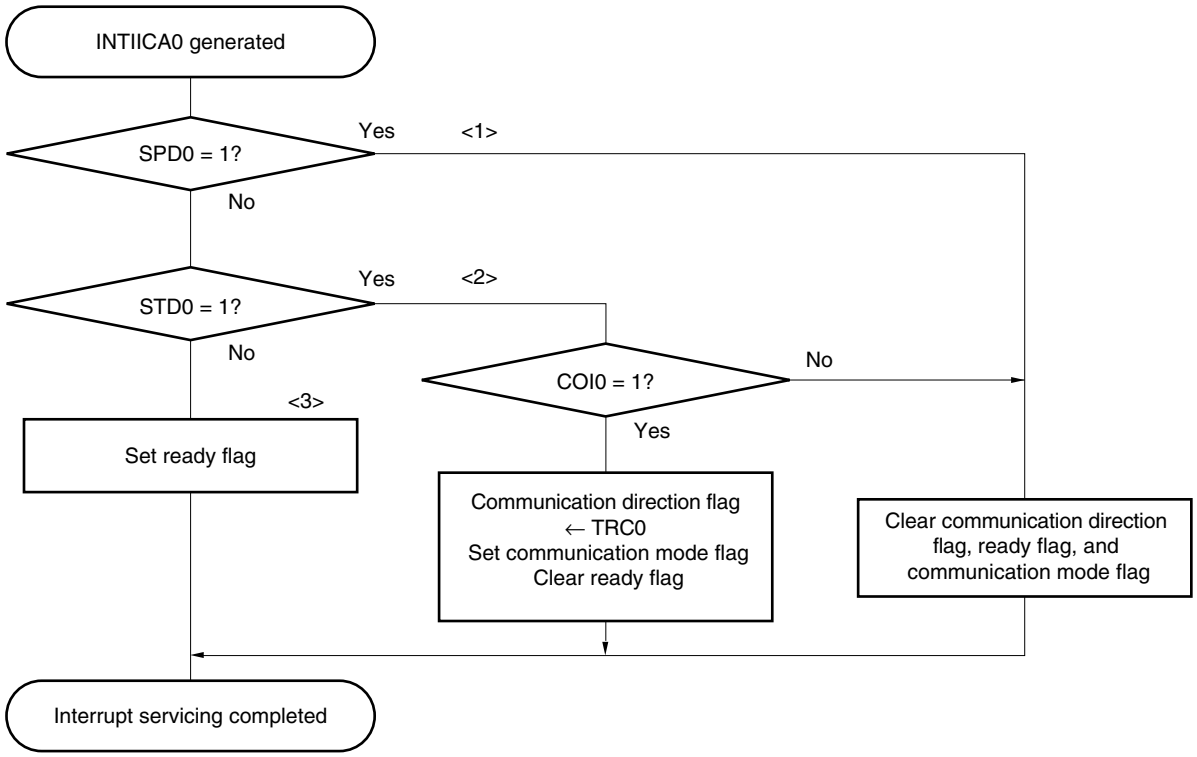
Remark Conform to the specifications of the product that is in communication, regarding the transmission and reception formats.

An example of the processing procedure of the slave with the INTIICA0 interrupt is explained below (processing is performed assuming that no extension code is used). The INTIICA0 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 16-31 Slave Operation Flowchart (2).

Figure 16-31. Slave Operation Flowchart (2)



16.5.17 Timing of I²C interrupt request (INTIICA0) occurrence

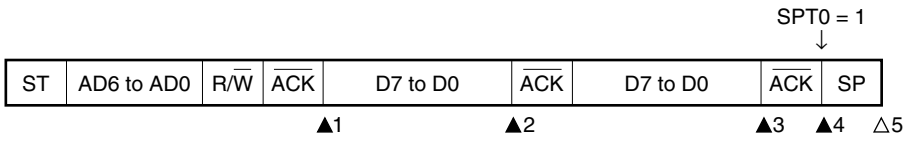
The timing of transmitting or receiving data and generation of interrupt request signal INTIICA0, and the value of the IICAS0 register when the INTIICA0 signal is generated are shown below.

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

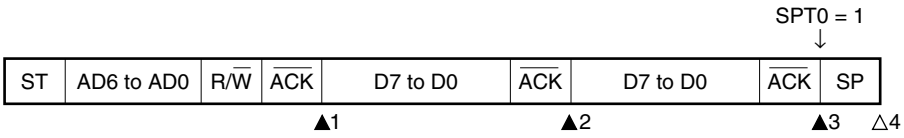


▲1: IICAS0 = 1000×110B
▲2: IICAS0 = 1000×000B
▲3: IICAS0 = 1000×000B (Sets WTIM0 to 1)[Note]
▲4: IICAS0 = 1000××00B (Sets SPT0 to 1)[Note]
△5: IICAS0 = 00000001B

Note To generate a stop condition, set WTIM0 to 1 and change the timing for generating the INTIICA0 interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

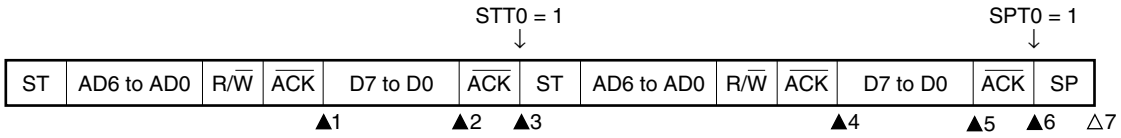


▲1: IICAS0 = 1000×110B
▲2: IICAS0 = 1000×100B
▲3: IICAS0 = 1000××00B (Sets SPT0 to 1)
△4: IICAS0 = 00000001B

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: IICAS0 = 1000×110B
▲2: IICAS0 = 1000×000B (Sets WTIM0 to 1)[Note 1]
▲3: IICAS0 = 1000××00B (Clears WTIM0 to 0[Note 2], sets STT0 to 1)
▲4: IICAS0 = 1000×110B
▲5: IICAS0 = 1000×000B (Sets WTIM0 to 1)[Note 3]
▲6: IICAS0 = 1000××00B (Sets SPT0 to 1)
△7: IICAS0 = 00000001B

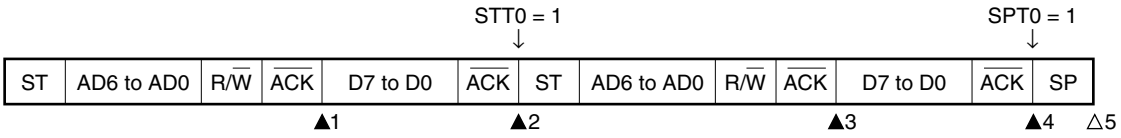
Notes 1. To generate a start condition, set WTIM0 to 1 and change the timing for generating the INTIICA0 interrupt request signal.

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 INTIICA0 interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

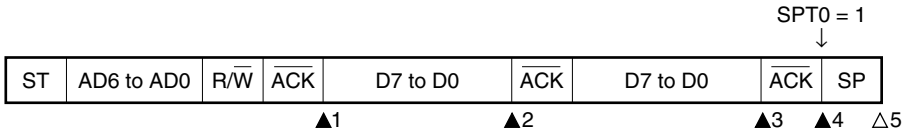


▲1: IICAS0 = 1000×110B
▲2: IICAS0 = 1000××00B (Sets STT0 to 1)
▲3: IICAS0 = 1000×110B
▲4: IICAS0 = 1000××00B (Sets SPT0 to 1)
△5: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(c) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

(i) When WTIM0 = 0

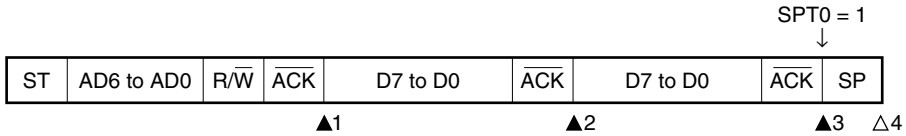


▲1: IICAS0 = 1010×110B
▲2: IICAS0 = 1010×000B
▲3: IICAS0 = 1010×000B (Sets WTIM0 to 1)[Note]
▲4: IICAS0 = 1010××00B (Sets SPT0 to 1)
△5: IICAS0 = 00000001B

Note To generate a stop condition, set WTIM0 to 1 and change the timing for generating the INTIICA0 interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



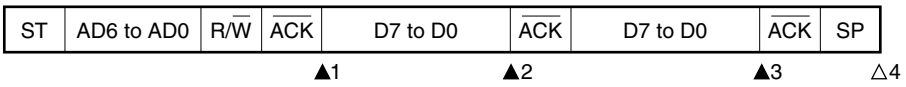
▲1: IICAS0 = 1010×110B
▲2: IICAS0 = 1010×100B
▲3: IICAS0 = 1010××00B (Sets SPT0 to 1)
△4: IICAS0 = 00001001B

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

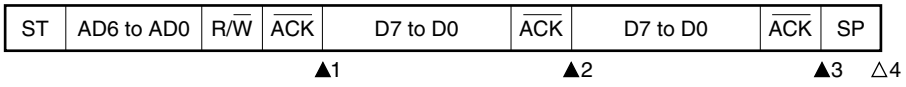
(i) When WTIM0 = 0



▲1: IICAS0 = 0001×110B
▲2: IICAS0 = 0001×000B
▲3: IICAS0 = 0001×000B
△4: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

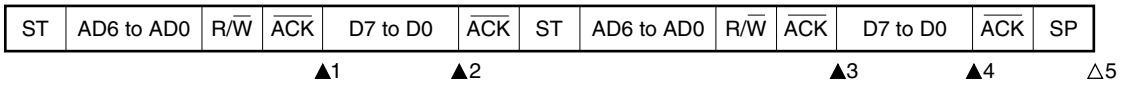


▲1: IICAS0 = 0001×110B
▲2: IICAS0 = 0001×100B
▲3: IICAS0 = 0001××00B
△4: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

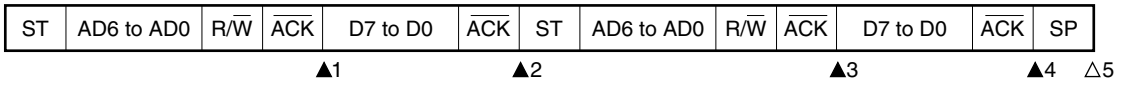
(i) When WTIM0 = 0 (after restart, matches with SVA0)



▲1: IICAS0 = 0001×110B
▲2: IICAS0 = 0001×000B
▲3: IICAS0 = 0001×110B
▲4: IICAS0 = 0001×000B
△5: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, matches with SVA0)

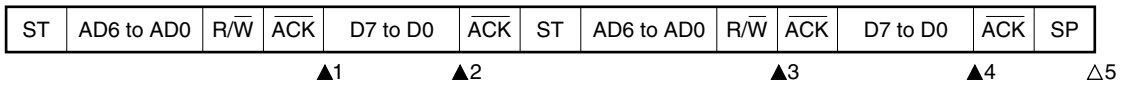


▲1: IICAS0 = 0001×110B
▲2: IICAS0 = 0001××00B
▲3: IICAS0 = 0001×110B
▲4: IICAS0 = 0001××00B
△5: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

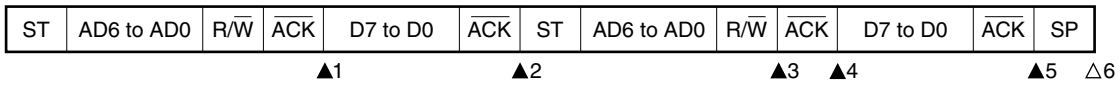
(i) When WTIM0 = 0 (after restart, does not match address (= extension code))



▲1: IICAS0 = 0001×110B
▲2: IICAS0 = 0001×000B
▲3: IICAS0 = 0010×010B
▲4: IICAS0 = 0010×000B
△5: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, does not match address (= extension code))

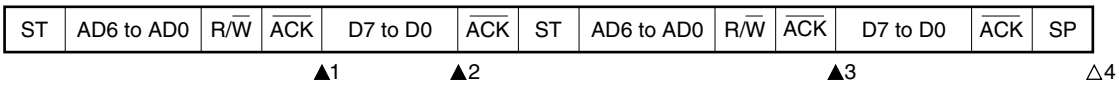


▲1: IICAS0 = 0001×110B
▲2: IICAS0 = 0001××00B
▲3: IICAS0 = 0010×010B
▲4: IICAS0 = 0010×110B
▲5: IICAS0 = 0010××00B
△6: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

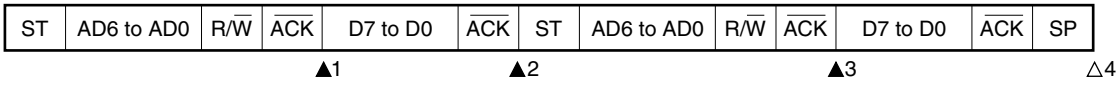
(i) When WTIM0 = 0 (after restart, does not match address (= not extension code))



▲1: IICAS0 = 0001×110B
▲2: IICAS0 = 0001×000B
▲3: IICAS0 = 00000110B
△4: IICAS0 = 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: IICAS0 = 0001×110B
▲2: IICAS0 = 0001××00B
▲3: IICAS0 = 00000110B
△4: IICAS0 = 00000001B

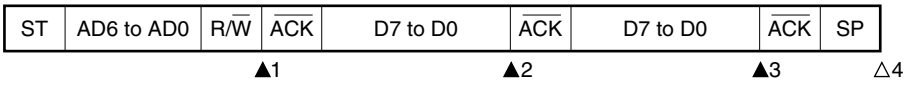
Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(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

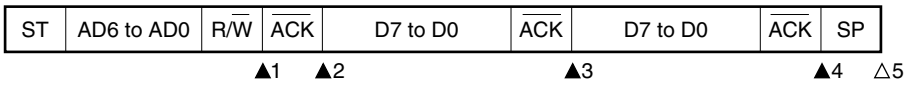
(i) When WTIM0 = 0



▲1: IICAS0 = 0010×010B
▲2: IICAS0 = 0010×000B
▲3: IICAS0 = 0010×000B
△4: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



▲1: IICAS0 = 0010×010B
▲2: IICAS0 = 0010×110B
▲3: IICAS0 = 0010×100B
▲4: IICAS0 = 0010××00B
△5: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

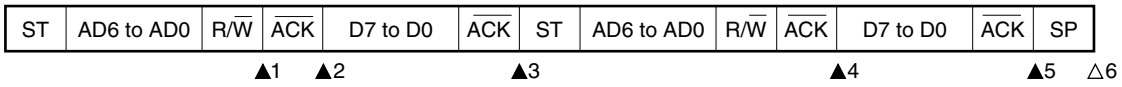
(b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIM0 = 0 (after restart, matches SVA0)

▲1: IICAS0 = 0010×010B
▲2: IICAS0 = 0010×000B
▲3: IICAS0 = 0001×110B
▲4: IICAS0 = 0001×000B
△5: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, matches SVA0)

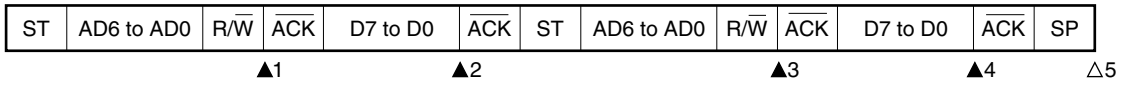


▲1: IICAS0 = 0010×010B
▲2: IICAS0 = 0010×110B
▲3: IICAS0 = 0010××00B
▲4: IICAS0 = 0001×110B
▲5: IICAS0 = 0001××00B
△6: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

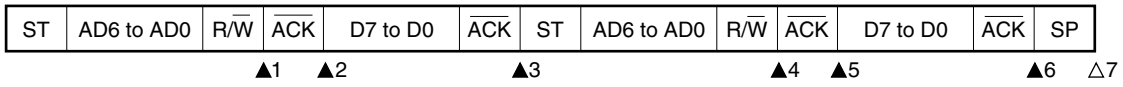
(i) When WTIM0 = 0 (after restart, extension code reception)



▲1: IICAS0 = 0010×010B
▲2: IICAS0 = 0010×000B
▲3: IICAS0 = 0010×010B
▲4: IICAS0 = 0010×000B
△5: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, extension code reception)

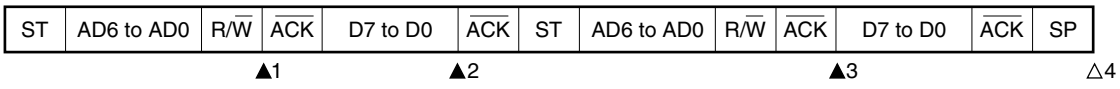


▲1: IICAS0 = 0010×010B
▲2: IICAS0 = 0010×110B
▲3: IICAS0 = 0010××00B
▲4: IICAS0 = 0010×010B
▲5: IICAS0 = 0010×110B
▲6: IICAS0 = 0010××00B
△7: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

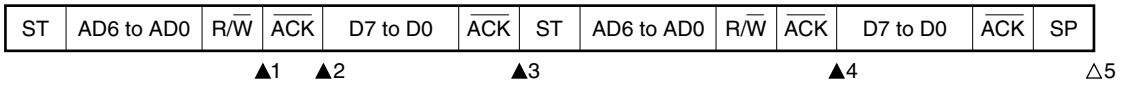
(i) When WTIM0 = 0 (after restart, does not match address (= not extension code))



▲1: IICAS0 = 00100010B
▲2: IICAS0 = 00100000B
▲3: IICAS0 = 00000110B
△4: IICAS0 = 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))

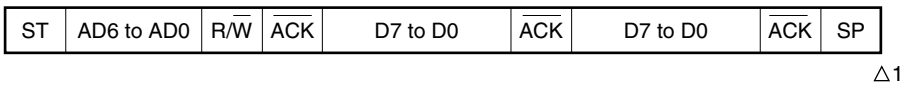


▲1: IICAS0 = 00100010B
▲2: IICAS0 = 00100110B
▲3: IICAS0 = 00100×00B
▲4: IICAS0 = 00000110B
△5: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(4) Operation without communication

(a) Start ~ Code ~ Data ~ Data ~ Stop



△1: IICAS0 = 00000001B

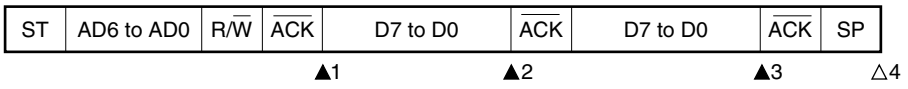
Remark △: Generated only when SPIE0 = 1

(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 INTIICA0 has occurred to check the arbitration result.

(a) When arbitration loss occurs during transmission of slave address data

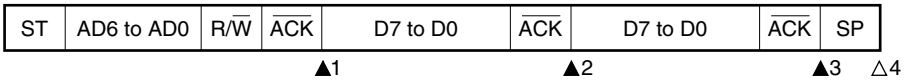
(i) When WTIM0 = 0



▲1: IICAS0 = 0101×110B
▲2: IICAS0 = 0001×000B
▲3: IICAS0 = 0001×000B
△4: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

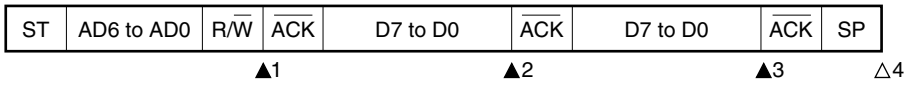


▲1: IICAS0 = 0101×110B
▲2: IICAS0 = 0001×100B
▲3: IICAS0 = 0001××00B
△4: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(b) When arbitration loss occurs during transmission of extension code

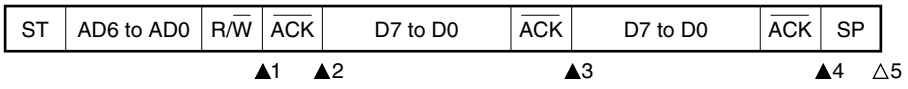
(i) When WTIM0 = 0



▲1: IICAS0 = 0110×010B
▲2: IICAS0 = 0010×000B
▲3: IICAS0 = 0010×000B
△4: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



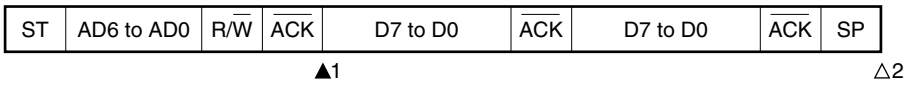
▲1: IICAS0 = 0110×010B
▲2: IICAS0 = 0010×110B
▲3: IICAS0 = 0010×100B
▲4: IICAS0 = 0010××00B
△5: IICAS0 = 00000001B

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 INTIICA0 has occurred to check the arbitration result.

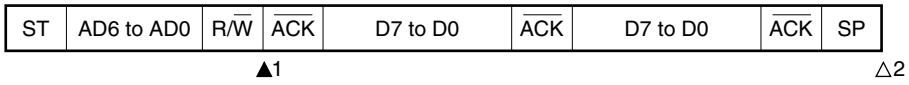
(a) When arbitration loss occurs during transmission of slave address data (when WTIM0 = 1)



▲1: IICAS0 = 01000110B
△2: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1

(b) When arbitration loss occurs during transmission of extension code

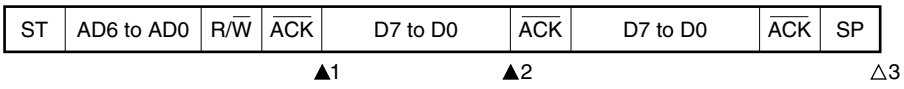


▲1: IICAS0 = 0110×010B
Sets LREL0 = 1 by software
△2: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(c) When arbitration loss occurs during transmission of data

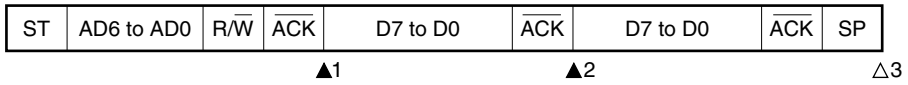
(i) When WTIM0 = 0



▲1: IICAS0 = 10001110B
▲2: IICAS0 = 01000000B
△3: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1

(ii) When WTIM0 = 1

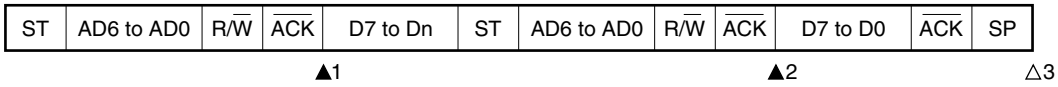


▲1: IICAS0 = 10001110B
▲2: IICAS0 = 01000100B
△3: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1

(d) When loss occurs due to restart condition during data transfer

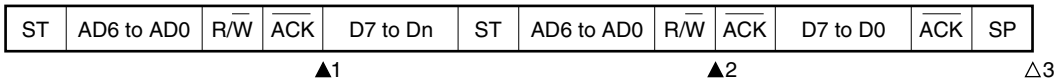
(i) Not extension code (Example: unmatches with SVA0)



▲1: IICAS0 = 1000×110B
▲2: IICAS0 = 01000110B
△3: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care
n = 6 to 0

(ii) Extension code



▲1: IICAS0 = 1000×110B

▲2: IICAS0 = 01100010B

Sets LREL0 = 1 by software

△3: IICAS0 = 00000001B

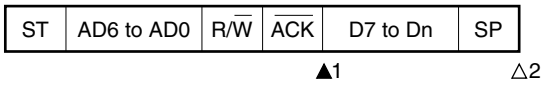
Remark ▲: Always generated

△: Generated only when SPIE0 = 1

×: Don't care

n = 6 to 0

(e) When loss occurs due to stop condition during data transfer



▲1: IICAS0 = 10000110B

△2: IICAS0 = 01000001B

Remark ▲: Always generated

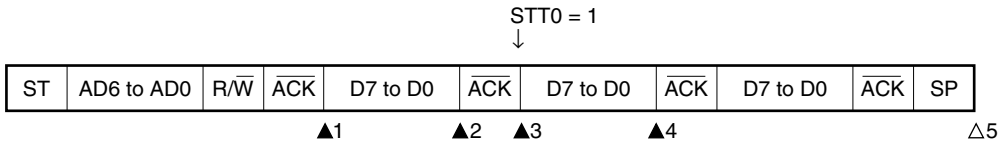
△: Generated only when SPIE0 = 1

×: Don't care

n = 6 to 0

(f) When arbitration loss occurs due to low-level data when attempting to generate a restart condition

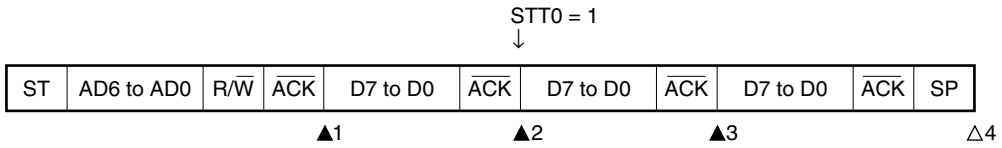
(i) When WTIM0 = 0



▲1: IICAS0 = 1000×110B
▲2: IICAS0 = 1000×000B (Sets WTIM0 to 1)
▲3: IICAS0 = 1000×100B (Clears WTIM0 to 0)
▲4: IICAS0 = 01000000B
△5: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

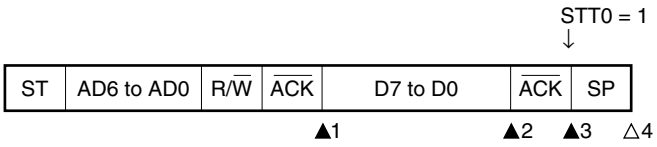


▲1: IICAS0 = 1000×110B
▲2: IICAS0 = 1000×100B (Sets STT0 to 1)
▲3: IICAS0 = 01000100B
△4: IICAS0 = 00000001B

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

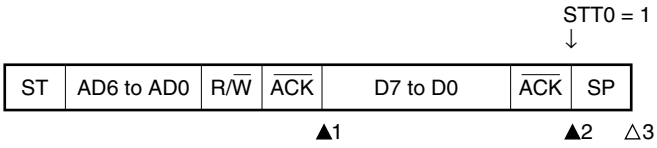
(i) When WTIM0 = 0



▲1: IICAS0 = 1000×110B
▲2: IICAS0 = 1000×000B (Sets WTIM0 to 1)
▲3: IICAS0 = 1000××00B (Sets STT0 to 1)
△4: IICAS0 = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

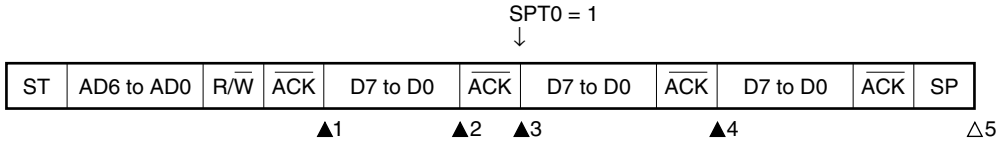


▲1: IICAS0 = 1000×110B
▲2: IICAS0 = 1000××00B (Sets STT0 to 1)
△3: IICAS0 = 01000001B

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

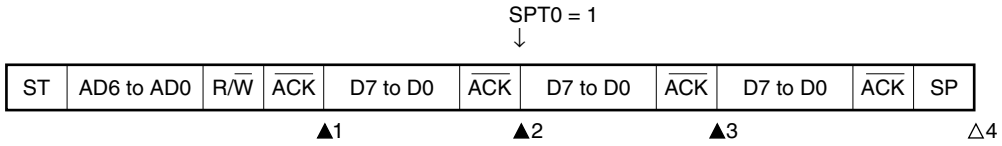
(i) When WTIM0 = 0



▲1: IICAS0 = 1000×110B
▲2: IICAS0 = 1000×000B (Sets WTIM0 to 1)
▲3: IICAS0 = 1000×100B (Clears WTIM0 to 0)
▲4: IICAS0 = 01000100B
△5: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



▲1: IICAS0 = 1000×110B
▲2: IICAS0 = 1000×100B (Sets SPT0 to 1)
▲3: IICAS0 = 01000100B
△4: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

16.6 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 the IICA status register 0 (IICAS0)), which specifies the data transfer direction, and then starts serial communication with the slave device.

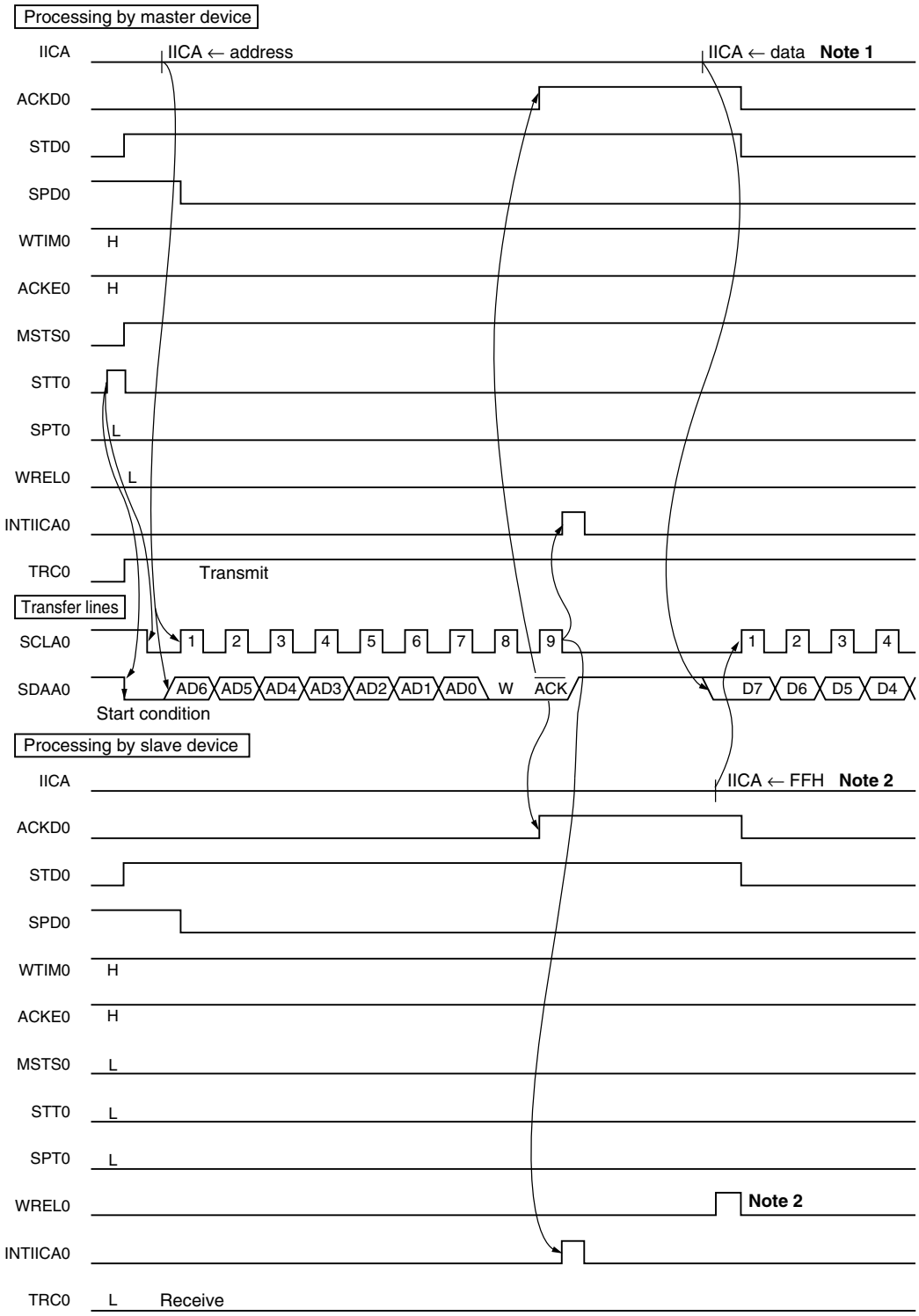
Figures 16-32 and 16-33 show timing charts of the data communication.

The IICA shift register (IICA)'s shift operation is synchronized with the falling edge of the serial clock (SCLA0). The transmit data is transferred to the SO latch and is output (MSB first) via the SDAA0 pin.

Data input via the SDAA0 pin is captured into IICA at the rising edge of SCLA0.

Figure 16-32. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (1/3)

(1) Start condition ~ address

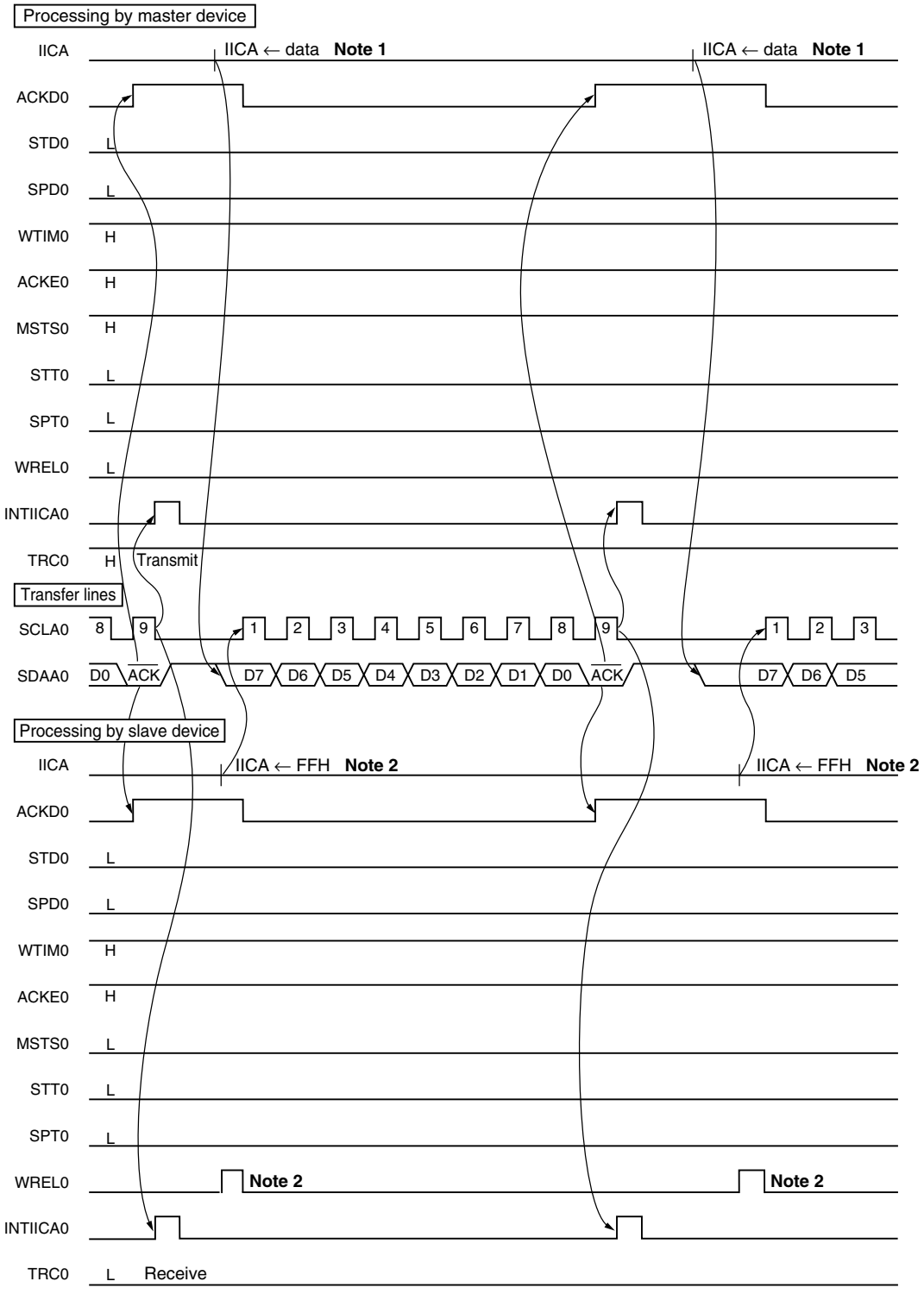


Notes 1. Write data to IICA, not setting WREL0, in order to cancel a wait state during master transmission.

2. To cancel slave wait, write “FFH” to IICA or set WREL0.

Figure 16-32. Example of Master to Slave Communication
(When 9-Clock Wait Is Selected for Both Master and Slave) (2/3)

(2) Data

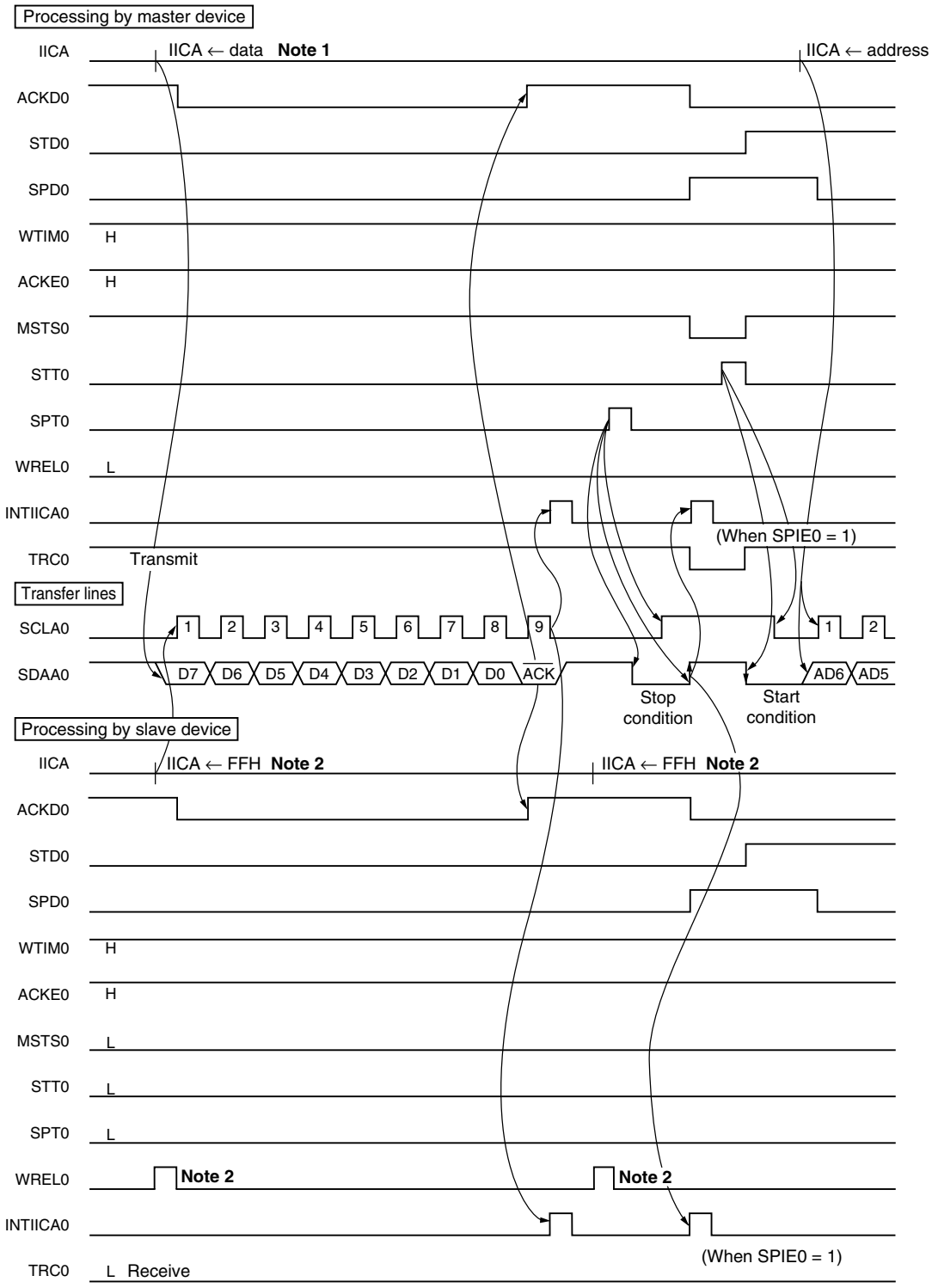


Notes 1. Write data to IICA, not setting WREL0, in order to cancel a wait state during master transmission.

2. To cancel slave wait, write “FFH” to IICA or set WREL0.

Figure 16-32. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (3/3)

(3) Stop condition

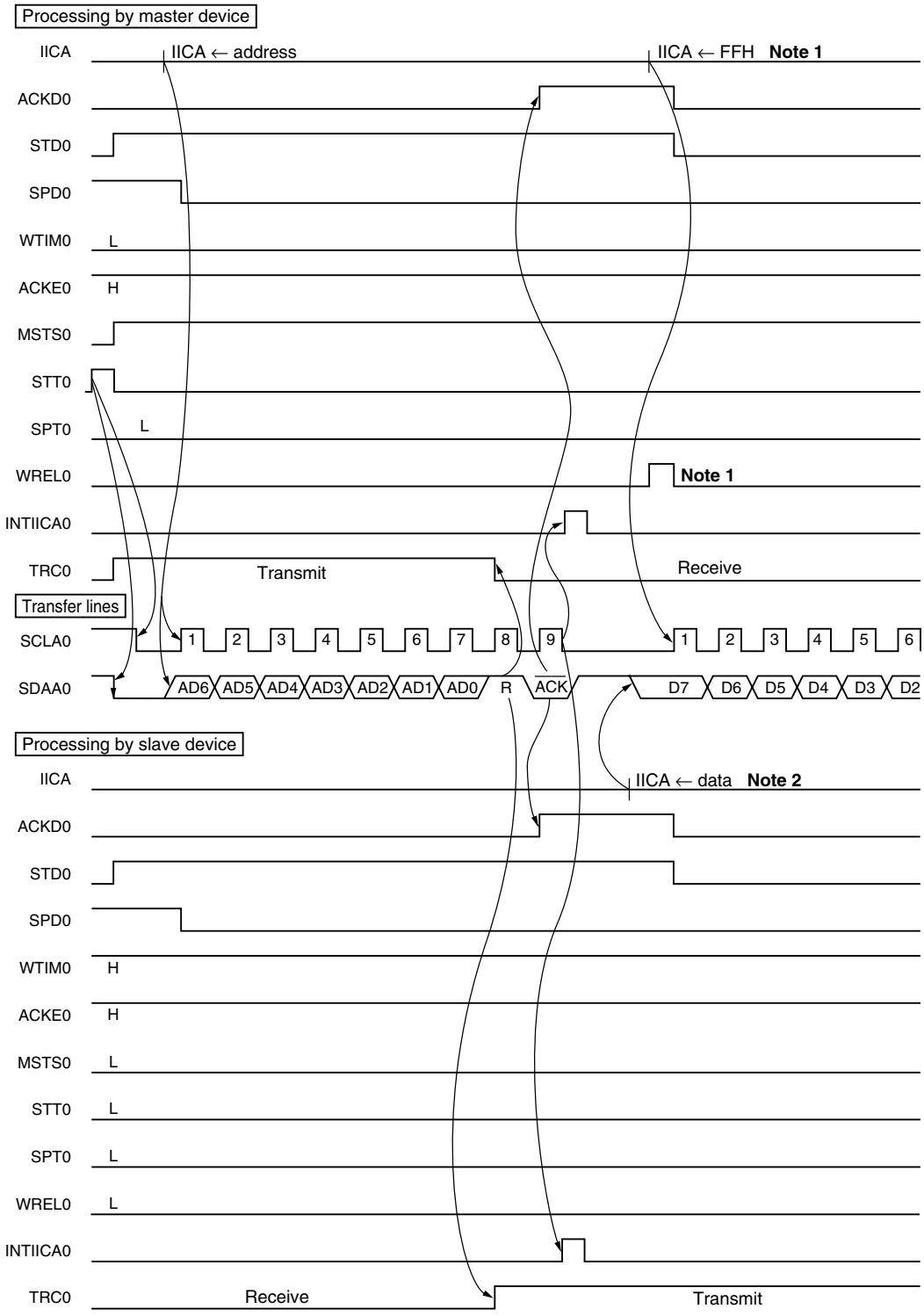


Notes 1. Write data to IICA, not setting WREL0, in order to cancel a wait state during master transmission.

2. To cancel slave wait, write “FFH” to IICA or set WREL0.

Figure 16-33. Example of Slave to Master Communication
(When 8-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (1/3)

(1) Start condition ~ address

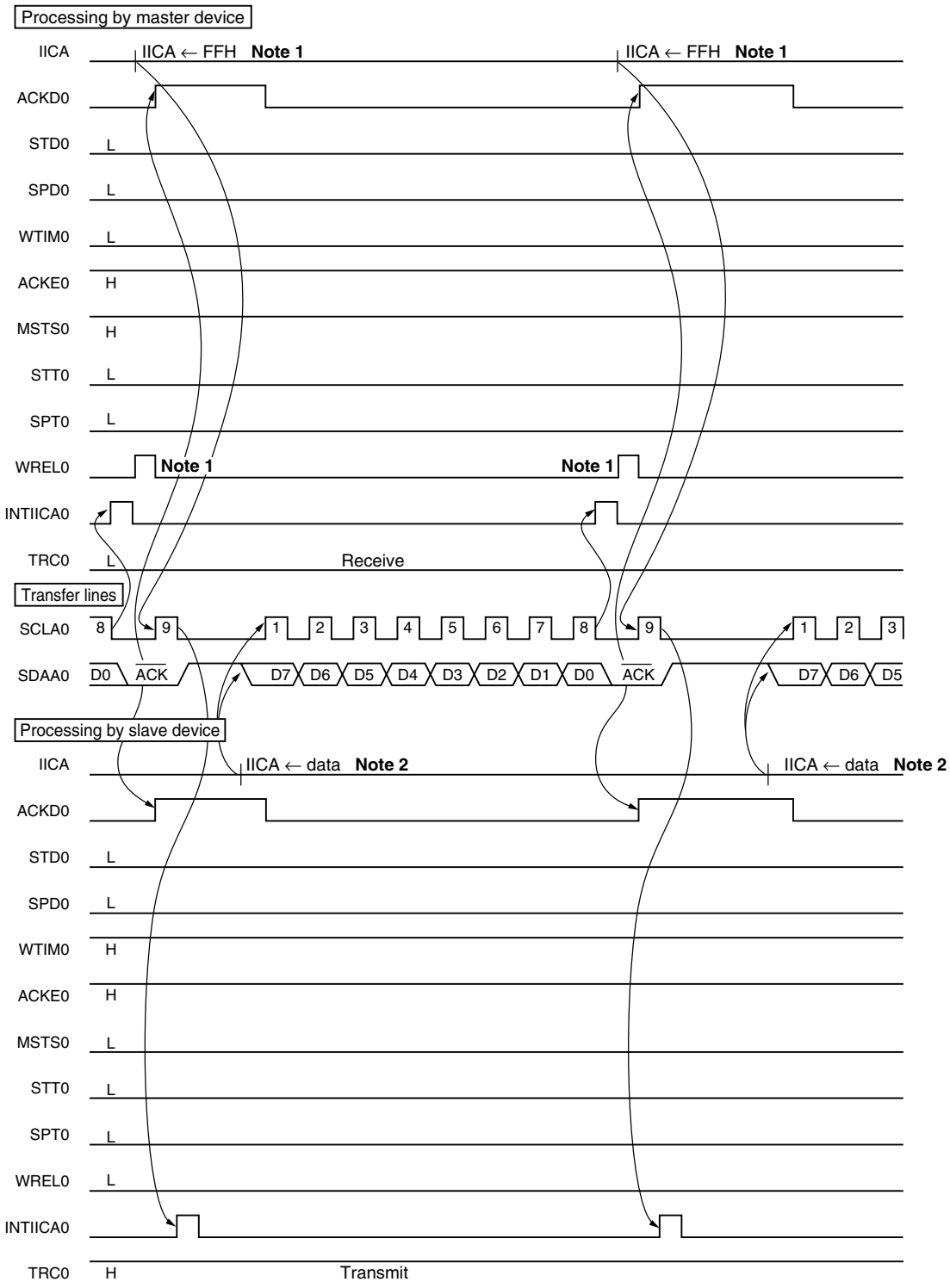


Notes 1. To cancel master wait, write “FFH” to IICA or set WREL0.

2. Write data to IICA, not setting WREL0, in order to cancel a wait state during slave transmission.

Figure 16-33. Example of Slave to Master Communication
(When 8-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (2/3)

(2) Data

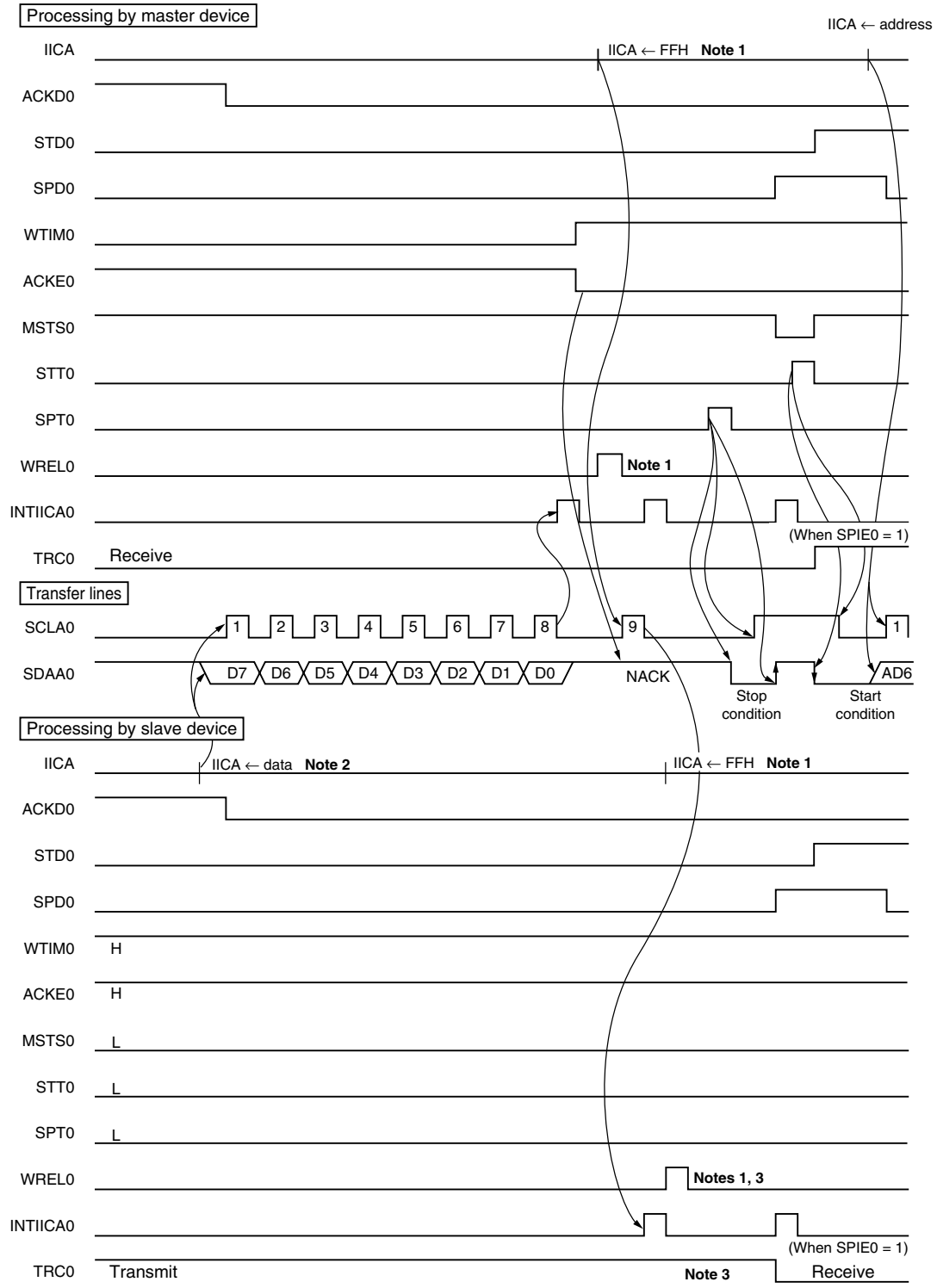


Notes 1. To cancel master wait, write “FFH” to IICA or set WREL0.

2. Write data to IICA, not setting WREL0, in order to cancel a wait state during slave transmission.

Figure 16-33. Example of Slave to Master Communication
(When 8-Clock and 9-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (3/3)

(3) Stop condition



Notes 1. To cancel wait, write “FFH” to IICA or set WREL0.

2. Write data to IICA, not setting WREL0, in order to cancel a wait state during slave transmission.

3. If a wait state during slave transmission is canceled by setting WREL0, TRC0 will be cleared.

CHAPTER 17 MULTIPLIER/DIVIDER

17.1 Functions of Multiplier/Divider

The multiplier/divider is mounted onto all 78K0/Kx2-A microcontrollers.
The multiplier/divider has the following functions.

- 16 bits × 16 bits = 32 bits (multiplication)
- 32 bits ÷ 16 bits = 32 bits, 16-bit remainder (division)

17.2 Configuration of Multiplier/Divider

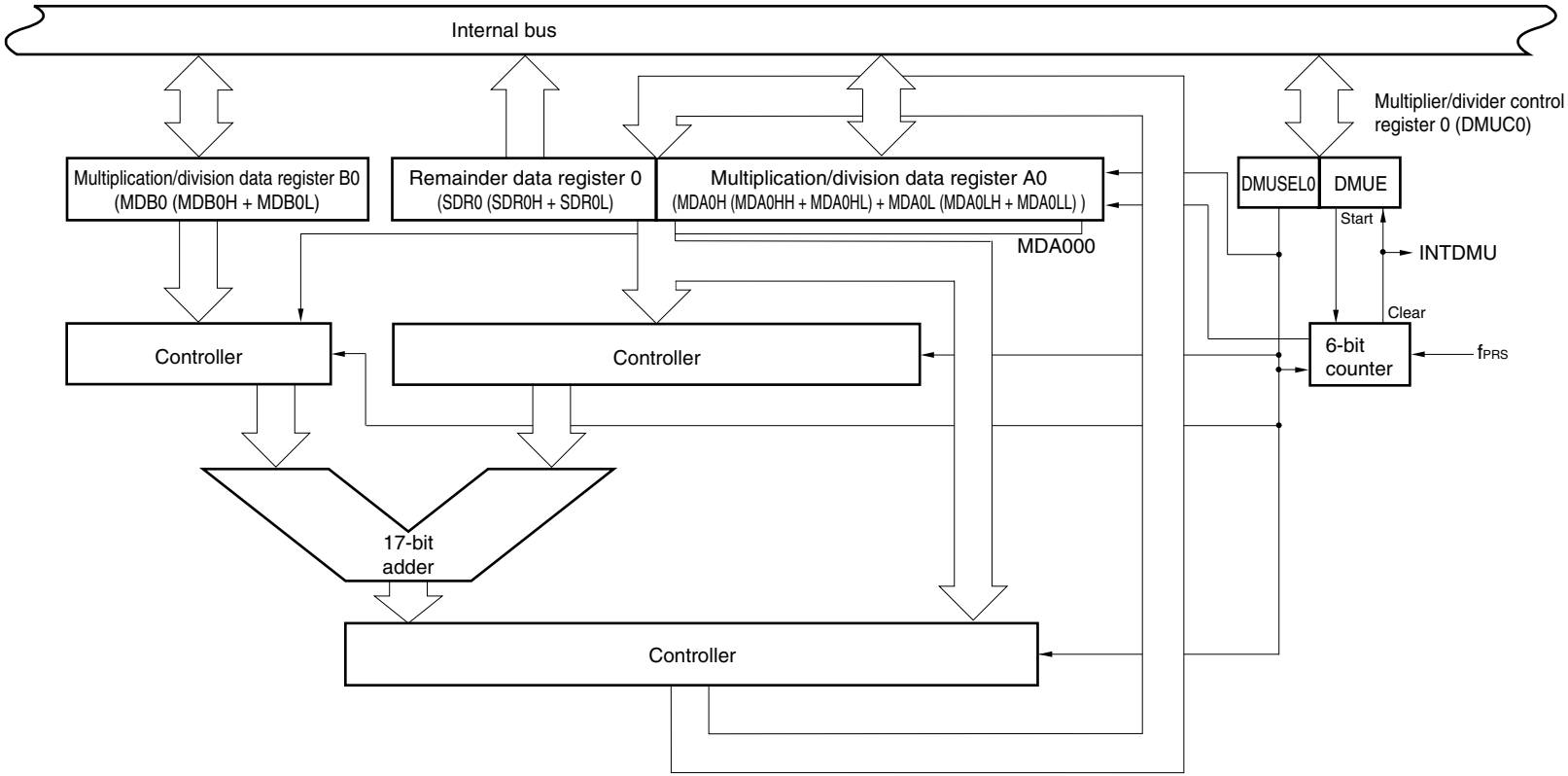
The multiplier/divider includes the following hardware.

Table 17-1. Configuration of Multiplier/Divider

Item	Configuration
Registers	Remainder data register 0 (SDR0) Multiplication/division data registers A0 (MDA0H, MDA0L) Multiplication/division data registers B0 (MDB0)
Control register	Multiplier/divider control register 0 (DMUC0)

Figure 17-1 shows the block diagram of the multiplier/divider.

Figure 17-1. Block Diagram of Multiplier/Divider



(1) Remainder data register 0 (SDR0)

SDR0 is a 16-bit register that stores a remainder. This register stores 0 in the multiplication mode and the remainder of an operation result in the division mode.

SDR0 can be read by an 8-bit or 16-bit memory manipulation instruction.

Reset signal generation clears SDR0 to 0000H.

Figure 17-2. Format of Remainder Data Register 0 (SDR0)

Address: FF60H, FF61H After reset: 0000H R

Symbol	FF61H (SDR0H)								FF60H (SDR0L)							
SDR0	SDR 015	SDR 014	SDR 013	SDR 012	SDR 011	SDR 010	SDR 009	SDR 008	SDR 007	SDR 006	SDR 005	SDR 004	SDR 003	SDR 002	SDR 001	SDR 000

Cautions 1. The value read from SDR0 during operation processing (while bit 7 (DMUE) of multiplier/divider control register 0 (DMUC0) is 1) is not guaranteed.

2. SDR0 is reset when the operation is started (when DMUE is set to 1).

(2) Multiplication/division data register A0 (MDA0H, MDA0L)

MDA0 is a 32-bit register that sets a 16-bit multiplier A in the multiplication mode and a 32-bit dividend in the division mode, and stores the 32-bit result of the operation (higher 16 bits: MDA0H, lower 16 bits: MDA0L).

Figure 17-3. Format of Multiplication/Division Data Register A0 (MDA0H, MDA0L)

Address: FF62H, FF63H, FF64H, FF65H After reset: 0000H, 0000H R/W

Symbol	FF65H (MDA0HH)								FF64H (MDA0HL)							
MDA0H	MDA 031	MDA 030	MDA 029	MDA 028	MDA 027	MDA 026	MDA 025	MDA 024	MDA 023	MDA 022	MDA 021	MDA 020	MDA 019	MDA 018	MDA 017	MDA 016

Symbol	FF63H (MDA0LH)								FF62H (MDA0LL)							
MDA0L	MDA 015	MDA 014	MDA 013	MDA 012	MDA 011	MDA 010	MDA 009	MDA 008	MDA 007	MDA 006	MDA 005	MDA 004	MDA 003	MDA 002	MDA 001	MDA 000

Cautions 1. MDA0H is cleared to 0 when an operation is started in the multiplication mode (when multiplier/divider control register 0 (DMUC0) is set to 81H).

2. Do not change the value of MDA0 during operation processing (while bit 7 (DMUE) of multiplier/divider control register 0 (DMUC0) is 1). Even in this case, the operation is executed, but the result is undefined.

3. The value read from MDA0 during operation processing (while DMUE is 1) is not guaranteed.

The functions of MDA0 when an operation is executed are shown in the table below.

Table 17-2. Functions of MDA0 During Operation Execution

DMUSEL0	Operation Mode	Setting	Operation Result
0	Division mode	Dividend	Division result (quotient)
1	Multiplication mode	Higher 16 bits: 0, Lower 16 bits: Multiplier A	Multiplication result (product)

Remark DMUSEL0: Bit 0 of multiplier/divider control register 0 (DMUC0)

The register configuration differs between when multiplication is executed and when division is executed, as follows.

- Register configuration during multiplication

  <Multiplier A> <Multiplier B> <Product>

  MDA0 (bits 15 to 0) × MDB0 (bits 15 to 0) = MDA0 (bits 31 to 0)

- Register configuration during division

  <Dividend> <Divisor> <Quotient> <Remainder>

  MDA0 (bits 31 to 0) ÷ MDB0 (bits 15 to 0) = MDA0 (bits 31 to 0) … SDR0 (bits 15 to 0)

MDA0 fetches the calculation result as soon as the clock is input, when bit 7 (DMUE) of multiplier/divider control register 0 (DMUC0) is set to 1.
MDA0H and MDA0L can be set by an 8-bit or 16-bit memory manipulation instruction.
Reset signal generation clears MDA0H and MDA0L to 0000H.

(3) Multiplication/division data register B0 (MDB0)

MDB0 is a register that stores a 16-bit multiplier B in the multiplication mode and a 16-bit divisor in the division mode.
MDB0 can be set by an 8-bit or 16-bit memory manipulation instruction.
Reset signal generation clears MDB0 to 0000H.

Figure 17-4. Format of Multiplication/Division Data Register B0 (MDB0)

Address: FF66H, FF67H After reset: 0000H R/W

Symbol	FF67H (MDB0H)								FF66H (MDB0L)							
MDB0	MDB 015	MDB 014	MDB 013	MDB 012	MDB 011	MDB 010	MDB 009	MDB 008	MDB 007	MDB 006	MDB 005	MDB 004	MDB 003	MDB 002	MDB 001	MDB 000

Cautions 1. Do not change the value of MDB0 during operation processing (while bit 7 (DMUE) of multiplier/divider control register 0 (DMUC0) is 1). Even in this case, the operation is executed, but the result is undefined.

2. Do not clear MDB0 to 0000H in the division mode. If set, undefined operation results are stored in MDA0 and SDR0.

17.3 Register Controlling Multiplier/Divider

The multiplier/divider is controlled by multiplier/divider control register 0 (DMUC0).

(1) Multiplier/divider control register 0 (DMUC0)

DMUC0 is an 8-bit register that controls the operation of the multiplier/divider.

DMUC0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears DMUC0 to 00H.

Figure 17-5. Format of Multiplier/Divider Control Register 0 (DMUC0)

Address: FF68H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
DMUC0	DMUE	0	0	0	0	0	0	DMUSEL0

DMUE[Note]	Operation start/stop
0	Stops operation
1	Starts operation

DMUSEL0	Operation mode (multiplication/division) selection
0	Division mode
1	Multiplication mode

Note When DMUE is set to 1, the operation is started. DMUE is automatically cleared to 0 after the operation is complete.

Cautions 1. If DMUE is cleared to 0 during operation processing (when DMUE is 1), the operation result is not guaranteed. If the operation is completed while the clearing instruction is being executed, the operation result is guaranteed, provided that the interrupt flag is set.

2. Do not change the value of DMUSEL0 during operation processing (while DMUE is 1). If it is changed, undefined operation results are stored in multiplication/division data register A0 (MDA0) and remainder data register 0 (SDR0).

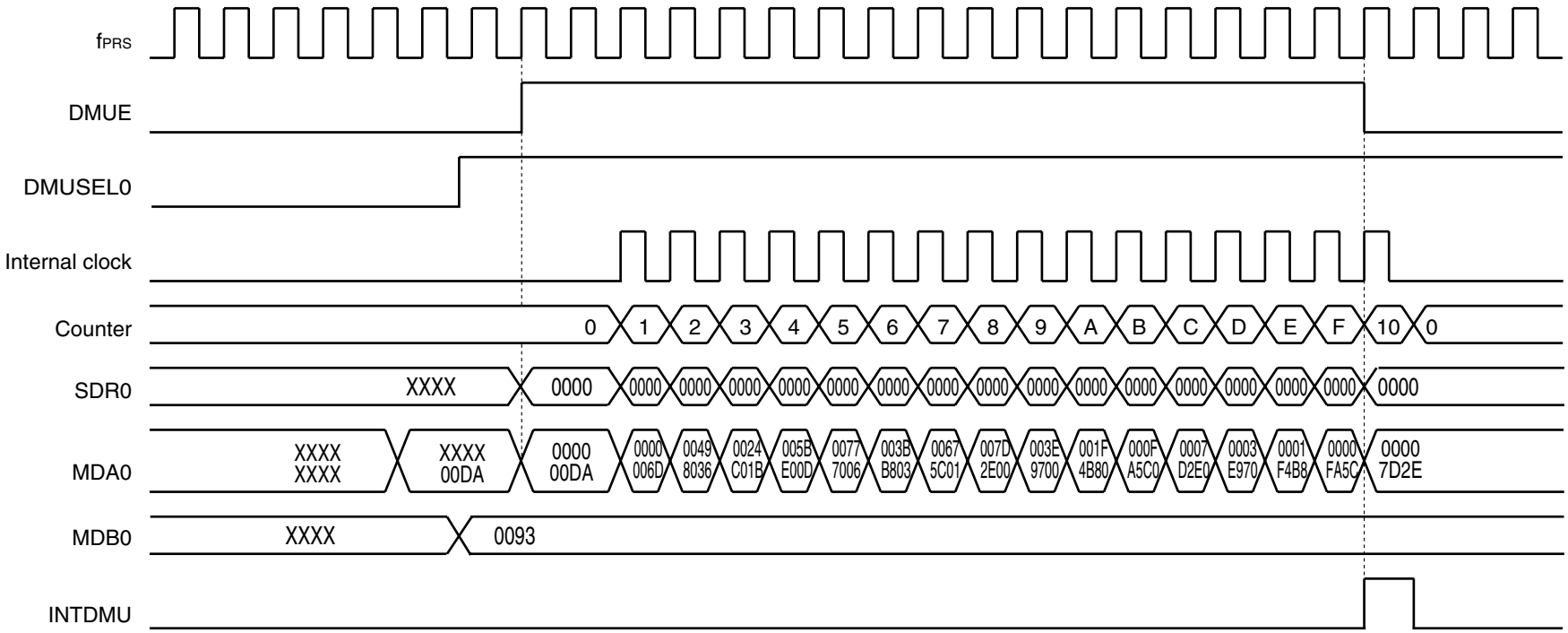
3. If DMUE is cleared to 0 during operation processing (while DMUE is 1), the operation processing is stopped. To execute the operation again, set multiplication/division data register A0 (MDA0), multiplication/division data register B0 (MDB0), and multiplier/divider control register 0 (DMUC0), and start the operation (by setting DMUE to 1).

17.4 Operations of Multiplier/Divider

17.4.1 Multiplication operation

- Initial setting
  1. Set operation data to multiplication/division data register A0L (MDA0L) and multiplication/division data register B0 (MDB0).
  2. Set bits 0 (DMUSEL0) and 7 (DMUE) of multiplier/divider control register 0 (DMUC0) to 1. Operation will start.
- During operation
  3. The operation will be completed when 16 peripheral hardware clocks (f_{PRS}) have been issued after the start of the operation (intermediate data is stored in the MDA0L and MDA0H registers during operation, and therefore the read values of these registers are not guaranteed).
- End of operation
  4. The operation result data is stored in the MDA0L and MDA0H registers.
  5. DMUE is cleared to 0 (end of operation).
  6. After the operation, an interrupt request signal (INTDMU) is generated.
- Next operation
  7. To execute multiplication next, start from the initial setting in **17.4.1 Multiplication operation**.
  8. To execute division next, start from the initial setting in **17.4.2 Division operation**.

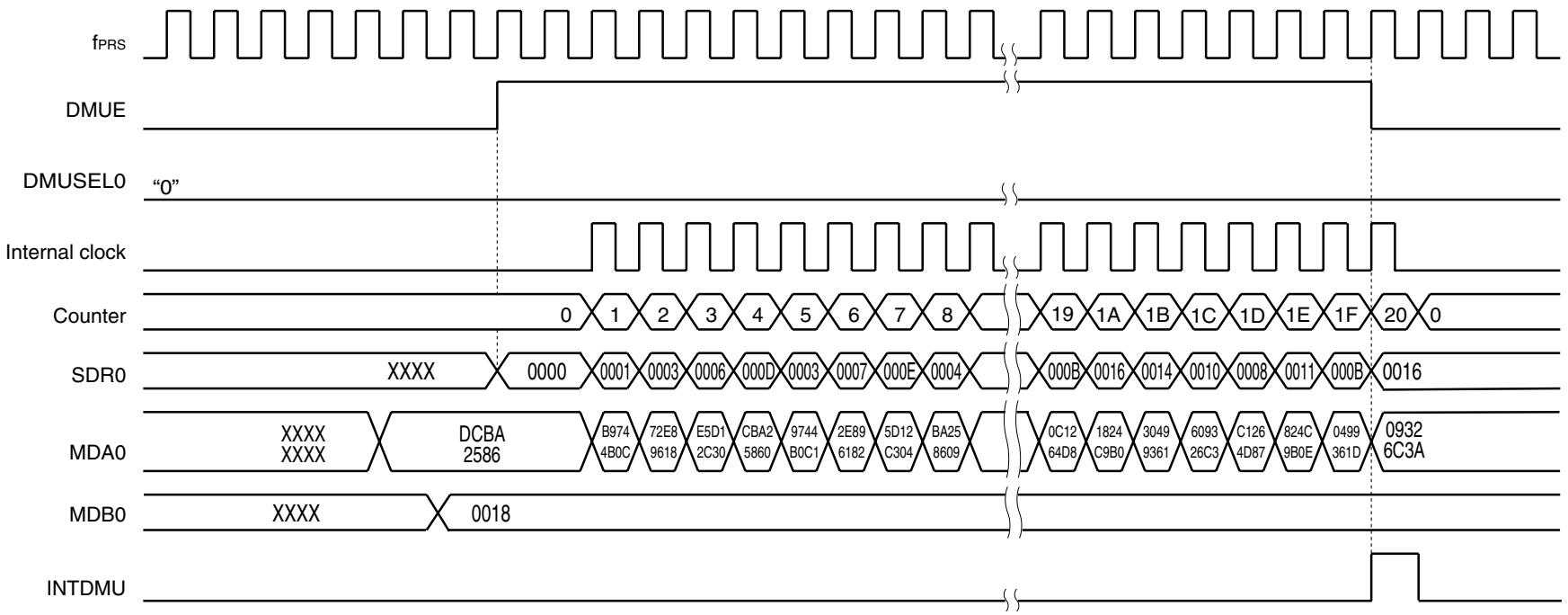
Figure 17-6. Timing Chart of Multiplication Operation (00DAH × 0093H)



17.4.2 Division operation

- Initial setting
  1. Set operation data to multiplication/division data register A0 (MDA0L and MDA0H) and multiplication/division data register B0 (MDB0).
  2. Set bits 0 (DMUSEL0) and 7 (DMUE) of multiplier/divider control register 0 (DMUC0) to 0 and 1, respectively. Operation will start.
- During operation
  3. The operation will be completed when 32 peripheral hardware clocks (f_{PRS}) have been issued after the start of the operation (intermediate data is stored in the MDA0L and MDA0H registers and remainder data register 0 (SDR0) during operation, and therefore the read values of these registers are not guaranteed).
- End of operation
  4. The result data is stored in the MDA0L, MDA0H, and SDR0 registers.
  5. DMUE is cleared to 0 (end of operation).
  6. After the operation, an interrupt request signal (INTDMU) is generated.
- Next operation
  7. To execute multiplication next, start from the initial setting in **17.4.1 Multiplication operation**.
  8. To execute division next, start from the initial setting in **17.4.2 Division operation**.

Figure 17-7. Timing Chart of Division Operation (DCBA2586H ÷ 0018H)



CHAPTER 18 INTERRUPT FUNCTIONS

18.1 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 a high interrupt priority group and a low interrupt priority group by setting the priority specification flag registers (PR0L, PR0H, PR1L, PR1H).

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 18-1**.

A standby release signal is generated and STOP and HALT modes are released.

External interrupt requests and internal interrupt requests are provided as maskable interrupts.

(2) Software interrupt

This is a vectored interrupt generated by executing the BRK instruction. It is acknowledged even when interrupts are disabled. The software interrupt does not undergo interrupt priority control.

18.2 Interrupt Sources and Configuration

The interrupt sources consist of maskable interrupts and software interrupts. In addition, they also have up to four reset sources (see **Table 18-1**).

Table 18-1. Interrupt Source List (1/2)

Interrupt Type	Internal/ External	Basic Configuration Type[Note 1]	Default Priority[Note 2]	Interrupt Source		Vector Table Address	78K0/ KB2-A	78K0/ KC2-A
				Name	Trigger			
Maskable	Internal	(A)	0	INTLVI	Low-voltage detection[Note 3]	0004H	√	√
	External	(B)	1	INTP0	Pin input edge detection	0006H	√	√
			2	INTP1		0008H	√	√
			3	INTP2		000AH	–	√
			4	INTP3		000CH	–	√
			5	INTP4		000EH	√	√
			6	INTP5		0010H	√	√
	Internal	(A)	7	INTSRE6	UART6 reception error generation	0012H	√	√
			8	INTSR6	End of UART6 reception	0014H	√	√
			9	INTST6	End of UART6 transmission	0016H	√	√
			10	INTCSI10	End of CSI10 communication	0018H	√	√
			11	INTTMH1	Match between TMH1 and CMP01 (when compare register is specified)	001AH	√	√
			12	INTTMH0	Match between TMH0 and CMP00 (when compare register is specified)	001CH	√	√
			13	INTTM50	Match between TM50 and CR50 (when compare register is specified)	001EH	√	√
			14	INTTM000	Match between TM00 and CR000 (when compare register is specified), TI010 pin valid edge detection (when capture register is specified)	0020H	√	√
			15	INTTM010	Match between TM00 and CR010 (when compare register is specified), TI000 pin valid edge detection (when capture register is specified)	0022H	√	√
			16	INTAD	End of A/D conversion	0024H	√	√
			17	INTIICA0	End of IICA communication	0026H	√	√
			18	INTRTCI	Real-time counter interval signal	0028H	–	√
			19	INTTM51 [Note 4]	Match between TM51 and CR51 (when compare register is specified)	002AH	√	√

Notes 1. Basic configuration types (A) to (D) correspond to (A) to (D) in **Figure 18-1**.

2. The default priority determines the sequence of processing vectored interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 26 indicates the lowest priority.

3. When bit 1 (LVIMD) of the low-voltage detection register (LVIM) is cleared to 0.

4. When 8-bit timer/event counter 51 is used in the carrier generator mode, an interrupt is generated upon the timing when the INTTM5H1 signal is generated (see **Figure 8-13 Transfer Timing**).

Table 18-1. Interrupt Source List (2/2)

Interrupt Type	Internal/ External	Basic Configuration Type[Note 1]	Default Priority[Note 2]	Interrupt Source		Vector Table Address	78K0/ KB2-A	78K0/ KC2-A
				Name	Trigger			
Maskable	External	(C)	20	INTKR	Key interrupt detection	002CH	–	√
	Internal	(A)	21	INTRTC	Constant-period signal of real-time counter/Match detection of alarm	002EH	–	√
	External	(B)	22	INTP6	Pin input edge detection	0030H	√	√
			23	INTP7		0032H	√	√
	Internal	(A)	24	INTDMU	End of multiply/divide operation	0034H	√	√
	External	(B)	25	INTP8	Pin input edge detection	0038H	–	√
			26	INTP9		003AH	–	√
Software	–	(D)	–	BRK	BRK instruction execution	003EH	√	√
Reset	–	–	–	RESET	Reset input	0000H	√	√
				POC	Power-on clear			
				LVI	Low-voltage detection[Note 3]			
				WDT	WDT overflow			

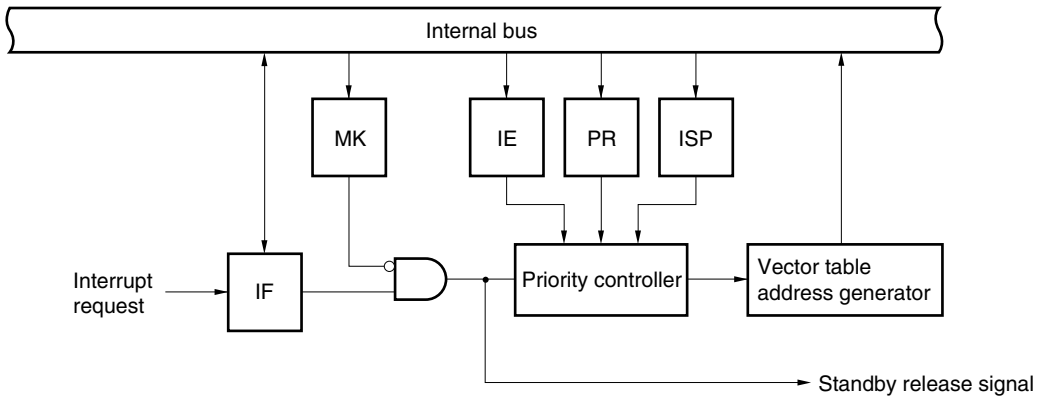
Notes 1. Basic configuration types (A) to (D) correspond to (A) to (D) in **Figure 18-1**.

2. The default priority determines the sequence of processing vectored interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 26 indicates the lowest priority.

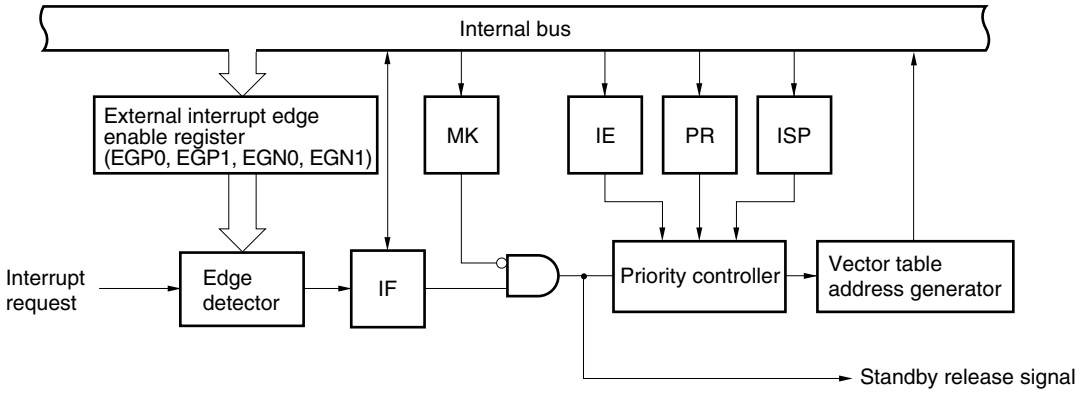
3. When bit 1 (LVIMD) of the low-voltage detection register (LVIM) is set to 1.

Figure 18-1. Basic Configuration of Interrupt Function (1/2)

(A) Internal maskable interrupt



(B) External maskable interrupt (INTPn)

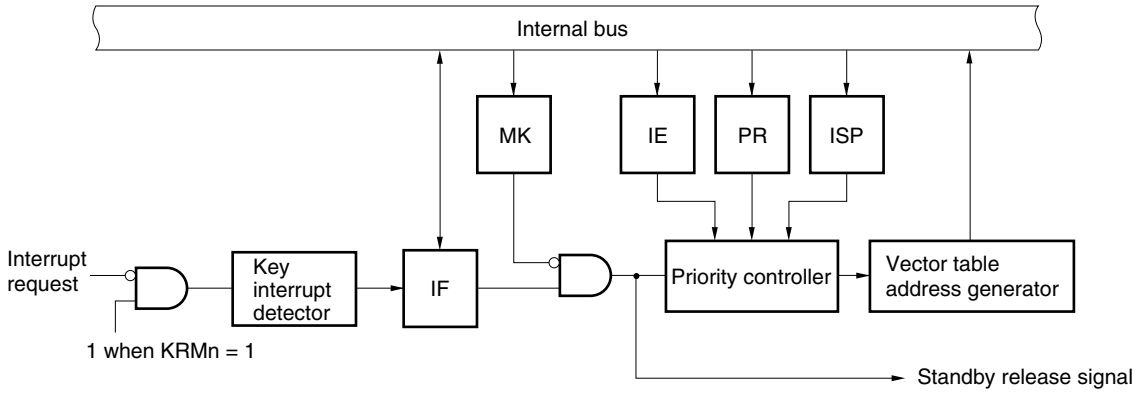


Remark n = 0, 1, 4 to 7: 78K0/KB2-A
n = 0 to 9: 78K0/KC2-A

IF: Interrupt request flag
IE: Interrupt enable flag
ISP: In-service priority flag
MK: Interrupt mask flag
PR: Priority specification flag

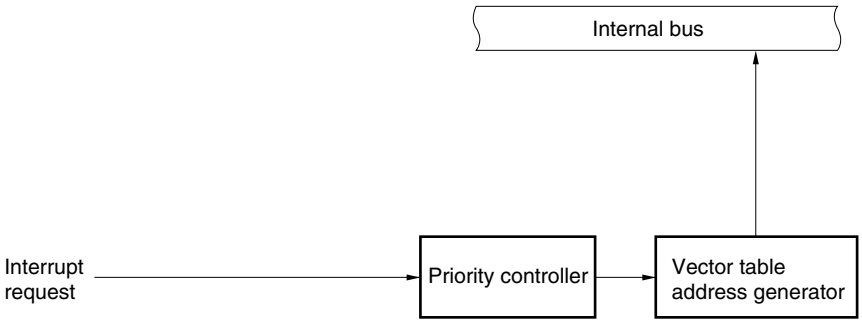
Figure 18-1. Basic Configuration of Interrupt Function (2/2)

(C) External maskable interrupt (INTKR)



Remark n = 0 to 5: 78K0/KC2-A

(D) Software interrupt



IF: Interrupt request flag
IE: Interrupt enable flag
ISP: In-service priority flag
MK: Interrupt mask flag
PR: Priority specification flag
KRM: Key return mode register

18.3 Registers Controlling Interrupt Functions

The following 6 types of registers are used to control the interrupt functions.

- Interrupt request flag register (IF0L, IF0H, IF1L, IF1H)
- Interrupt mask flag register (MK0L, MK0H, MK1L, MK1H)
- Priority specification flag register (PR0L, PR0H, PR1L, PR1H)
- External interrupt rising edge enable register (EGP0, EGP1)
- External interrupt falling edge enable register (EGN0, EGN1)
- Program status word (PSW)

Table 18-2 shows a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 18-2. Flags Corresponding to Interrupt Request Sources (1/2)

78K0/ KB2-A	78K0/ KC2-A	Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
				Register		Register		Register
√	√	INTLVI	LVIIF	IF0L	LVIMK	MK0L	LVIPR	PR0L
√	√	INTP0	PIF0		PMK0		PPR0	
√	√	INTP1	PIF1		PMK1		PPR1	
–	√	INTP2	PIF2		PMK2		PPR2	
–	√	INTP3	PIF3		PMK3		PPR3	
√	√	INTP4	PIF4		PMK4		PPR4	
√	√	INTP5	PIF5		PMK5		PPR5	
√	√	INTSRE6	SREIF6		SREMK6		SREPR6	
√	√	INTSR6	SRIF6	IF0H	SRMK6	MK0H	SRPR6	PR0H
√	√	INTST6	STIF6		STMK6		STPR6	
√	√	INTCSI10	CSIIF10		CSIMK10		CSIPR10	
√	√	INTTMH1	TMIFH1		TMMKH1		TMPRH1	
√	√	INTTMH0	TMIFH0		TMMKH0		TMPRH0	
√	√	INTTM50	TMIF50		TMMK50		TMPR50	
√	√	INTTM000	TMIF000		TMMK000		TMPR000	
√	√	INTTM010	TMIF010		TMMK010		TMPR010	

Table 18-2. Flags Corresponding to Interrupt Request Sources (2/2)

78K0/KB2-A	78K0/KC2-A	Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
				Register		Register		Register
√	√	INTAD	ADIF	IF1L	ADMK	MK1L	ADPR	PR1L
√	√	INTIICA0	IICAIF0		IICAMK0		IICAPR0	
–	√	INTRTCI	RTCIIF		RTCIMK		RTCIPR	
√	√	INTTM51[Note]	TMIF51		TMMK51		TMPR51	
–	√	INTKR	KRIF		KRMK		KRPR	
–	√	INTRTC	RTCIF		RTCMK		RTCPR	
√	√	INTP6	PIF6		PMK6		PPR6	
√	√	INTP7	PIF7		PMK7		PPR7	
√	√	INTDMU	DMUIF	IF1H	DMUMK	MK1H	DMUPR	PR1H
–	√	INTP8	PIF8		PMK8		PPR8	
–	√	INTP9	PIF9		PMK9		PPR9	

Note When 8-bit timer/event counter 51 is used in the carrier generator mode, an interrupt is generated upon the timing when the INTTM5H1 signal is generated (see **Figure 8-13 Transfer Timing**).

(1) Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H)

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, and IF1H are set by a 1-bit or 8-bit memory manipulation instruction. When IF0L and IF0H, and IF1L and IF1H are combined to form 16-bit registers IF0 and IF1, they are set by a 16-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Cautions 1. When operating a timer, serial interface, or A/D converter after standby release, operate it once after clearing the interrupt request flag. An interrupt request flag may be set by noise.

2. When manipulating a flag of the interrupt request flag register, use a 1-bit memory manipulation instruction (CLR1). When describing in C language, use a bit manipulation instruction such as "IF0L.0 = 0;" or "_asm("clr1 IF0L, 0");" because the compiled assembler must be a 1-bit memory manipulation instruction (CLR1).

If a program is described in C language using an 8-bit memory manipulation instruction such as "IF0L &= 0xfe;" and compiled, it becomes the assembler of three instructions.

```
mov a, IF0L
and a, #0FEH
mov IF0L, a
```

In this case, even if the request flag of another bit of the same interrupt request flag register (IF0L) is set to 1 at the timing between "mov a, IF0L" and "mov IF0L, a", the flag is cleared to 0 at "mov IF0L, a". Therefore, care must be exercised when using an 8-bit memory manipulation instruction in C language.

Figure 18-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H) (1/2)

<1> 78K0/KB2-A

Address: FFE0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	4	3	<2>	<1>	<0>
IF0L	SREIF6	PIF5	PIF4	0	0	PIF1	PIF0	LVIIF

Address: FFE1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0H	TMIF010	TMIF000	TMIF50	TMIFH0	TMIFH1	CSIIF10	STIF6	SRIF6

Address: FFE2H After reset: 00H R/W

Symbol	<7>	<6>	5	4	<3>	2	<1>	<0>
IF1L	PIF7	PIF6	0	0	TMIF51	0	IICAIF0	ADIF

Address: FFE3H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
IF1H	0	0	0	0	0	0	0	DMUIF

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Caution Be sure to clear bits 3 and 4 of IF0L and bits 2, 4, and 5 of IF1L, and bits 1 to 7 of IF1H to 0.

Figure 18-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H) (2/2)

<2> 78K0/KC2-A

Address: FFE0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	SREIF6	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	LVIIF

Address: FFE1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0H	TMIF010	TMIF000	TMIF50	TMIFH0	TMIFH1	CSIIF10	STIF6	SRIF6

Address: FFE2H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1L	PIF7	PIF6	RTCIF	KRIF	TMIF51	RTCIIF	IICAIF0	ADIF

Address: FFE3H After reset: 00H R/W

Symbol	7	6	5	4	<3>	<2>	1	<0>
IF1H	0	0	0	0	PIF9	PIF8	0	DMUIF

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Caution Be sure to clear bits 1 and 4 to 7 of IF1H to 0.

(2) Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt servicing.

MK0L, MK0H, MK1L, and MK1H are set by a 1-bit or 8-bit memory manipulation instruction. When MK0L and MK0H, and MK1L and MK1H are combined to form 16-bit registers MK0 and MK1, they are set by a 16-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 18-3. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H) (1/2)

<1> 78K0/KB2-A

Address: FFE4H After reset: FFH R/W

Symbol	<7>	<6>	<5>	4	3	<2>	<1>	<0>
MK0L	SREMK6	PMK5	PMK4	1	1	PMK1	PMK0	LVIMK

Address: FFE5H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0H	TMMK010	TMMK000	TMMK50	TMMKH0	TMMKH1	CSIMK10	STMK6	SRMK6

Address: FFE6H After reset: FFH R/W

Symbol	<7>	<6>	5	4	<3>	2	<1>	<0>
MK1L	PMK7	PMK6	1	1	TMMK51	1	IICAMK0	ADMK

Address: FFE7H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	<0>
MK1H	1	1	1	1	1	1	1	DMUMK

XXMKX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Caution Be sure to set bits 3 and 4 of MK0L, bits 2, 4, and 5 of MK1L, and bits 1 to 7 of MK1H to 1.

Figure 18-3. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H) (2/2)

<2> 78K0/KC2-A

Address: FFE4H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0L	SREMK6	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK

Address: FFE5H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0H	TMMK010	TMMK000	TMMK50	TMMKH0	TMMKH1	CSIMK10	STMK6	SRMK6

Address: FFE6H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1L	PMK7	PMK6	RTCMK	KRMK	TMMK51	RTCIMK	IICAMK0	ADMK

Address: FFE7H After reset: FFH R/W

Symbol	7	6	5	4	<3>	<2>	1	<0>
MK1H	1	1	1	1	PMK9	PMK8	1	DMUMK

XXMKX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Caution Be sure to set bits 1 and 4 to 7 of MK1H to 1.

(3) Priority specification flag registers (PR0L, PR0H, PR1L, PR1H)

The priority specification flag registers are used to set the corresponding maskable interrupt priority order.

PR0L, PR0H, PR1L, and PR1H are set by a 1-bit or 8-bit memory manipulation instruction. If PR0L and PR0H, and PR1L and PR1H are combined to form 16-bit registers PR0 and PR1, they are set by a 16-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 18-4. Format of Priority Specification Flag Registers (PR0L, PR0H, PR1L, PR1H) (1/2)

<1> 78K0/KB2-A

Address: FFE8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	4	3	<2>	<1>	<0>
PR0L	SREPR6	PPR5	PPR4	1	1	PPR1	PPR0	LVIPR

Address: FFE9H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR0H	TMPR010	TMPR000	TMPR50	TMPRH0	TMPRH1	CSIPR10	STPR6	SRPR6

Address: FFEAH After reset: FFH R/W

Symbol	<7>	<6>	5	4	<3>	2	<1>	<0>
PR1L	PPR7	PPR6	1	1	TMPR51	1	IICAPR0	ADPR

Address: FFEBH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	<0>
PR1H	1	1	1	1	1	1	1	IICPR0

XXPRX	Priority level selection
0	High priority level
1	Low priority level

Caution Be sure to set bits 3 and 4 of PR0L, bits 2, 4, and 5 of PR1L, and bits 1 to 7 of PR1H to 1.

Figure 18-4. Format of Priority Specification Flag Registers (PR0L, PR0H, PR1L, PR1H) (2/2)

<2> 78K0/KC2-A

Address: FFE8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR0L	SREPR6	PPR5	PPR4	PPR3	PPR2	PPR1	PPR0	LVIPR

Address: FFE9H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR0H	TMPR010	TMPR000	TMPR50	TMPRH0	TMPRH1	CSIPR10	STPR6	SRPR6

Address: FFEAH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR1L	PPR7	PPR6	RTCPR	KRPR	TMPR51	RTCIPR	IICAPR0	ADPR

Address: FFEBH After reset: FFH R/W

Symbol	7	6	5	4	<3>	<2>	1	<0>
PR1H	1	1	1	1	PPR9	PPR8	1	DMUPR

XXPRX	Priority level selection
0	High priority level
1	Low priority level

Caution Be sure to set bits 1 and 4 to 7 of PR1H to 1.

(4) External interrupt rising edge enable register (EGP0, EGP1), external interrupt falling edge enable register (EGN0, EGN1)

These registers specify the valid edge for INTPn.

EGP0, EGP1, EGN0, and EGN1 are set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Remark n = 0, 1, 4 to 7: 78K0/KB2-A
n = 0 to 9: 78K0/KC2-A

Figure 18-5. Format of External Interrupt Rising Edge Enable Register (EGP0, EGP1) and External Interrupt Falling Edge Enable Register (EGN0, EGN1)

<1> 78K0/KB2-A

Address: FF48H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP0	EGP7	EGP6	EGP5	EGP4	0	0	EGP1	EGP0

Address: FF49H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN0	EGN7	EGN6	EGN5	EGN4	0	0	EGN1	EGN0

<2> 78K0/KC2-A

Address: FF48H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP0	EGP7	EGP6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

Address: FF49H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN0	EGN7	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

Address: FF4AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP1	0	0	0	0	0	0	EGP9	EGP8

Address: FF4BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN1	0	0	0	0	0	0	EGN9	EGN8

EGPn	EGNn	INTPn pin valid edge selection
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Caution 78K0/KB2-A: Be sure to clear bits 2 and 3 of EGP0 and EGN0 to 0
78K0/KC2-A: Be sure to clear bits 2 to 7 of EGP1 and EGN1 to 0

Remark n = 0, 1, 4 to 7: 78K0/KB2-A
n = 0 to 9: 78K0/KC2-A

Table 18-3 shows the ports corresponding to EGPn and EGNn.

Table 18-3. Ports Corresponding to EGPn and EGNn

Detection Enable Register		Edge Detection Port	Interrupt Request Signal
EGP0	EGN0	P120	INTP0
EGP1	EGN1	P35	INTP1
EGP2	EGN2	P34Note	INTP2Note
EGP3	EGN3	P33Note	INTP3Note
EGP4	EGN4	P32	INTP4
EGP5	EGN5	P31	INTP5
EGP6	EGN6	P13	INTP6
EGP7	EGN7	P12	INTP7
EGP8	EGN8	P02Note	INTP8Note
EGP9	EGN9	P42Note	INTP9Note

Note 78K0/KC2-A only.

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, 1, 4 to 7: 78K0/KB2-A
n = 0 to 9: 78K0/KC2-A

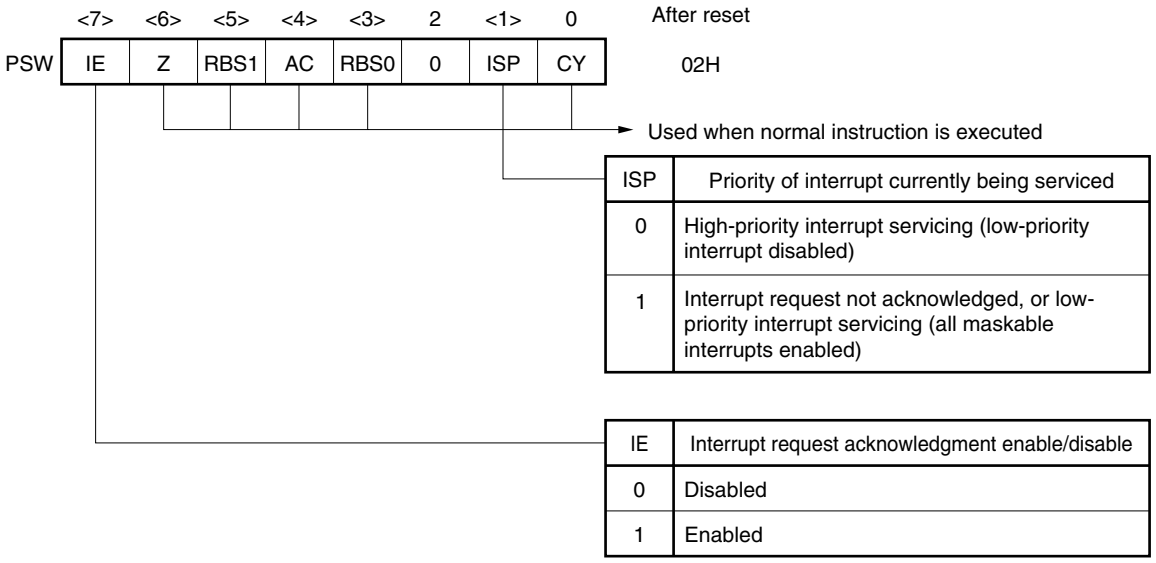
(5) Program status word (PSW)

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 ISP flag 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 ISP flag. 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 02H.

Figure 18-6. Format of Program Status Word



ISP	Priority of interrupt currently being serviced
0	High-priority interrupt servicing (low-priority interrupt disabled)
1	Interrupt request not acknowledged, or low-priority interrupt servicing (all maskable interrupts enabled)

IE	Interrupt request acknowledgment enable/disable
0	Disabled
1	Enabled

18.4 Interrupt Servicing Operations

18.4.1 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 (when the ISP flag is reset to 0).

The times from generation of a maskable interrupt request until vectored interrupt servicing is performed are listed in Table 18-4 below.

For the interrupt request acknowledgment timing, see **Figures 18-8** and **18-9**.

Table 18-4. Time from Generation of Maskable Interrupt Until Servicing

	Minimum Time	Maximum Time[Note]
When ××PR = 0	7 clocks	32 clocks
When ××PR = 1	8 clocks	33 clocks

Note If an interrupt request is generated just before a divide instruction, the wait time becomes longer.

Remark 1 clock: $1/f_{CPU}$ (f_{CPU}: 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.

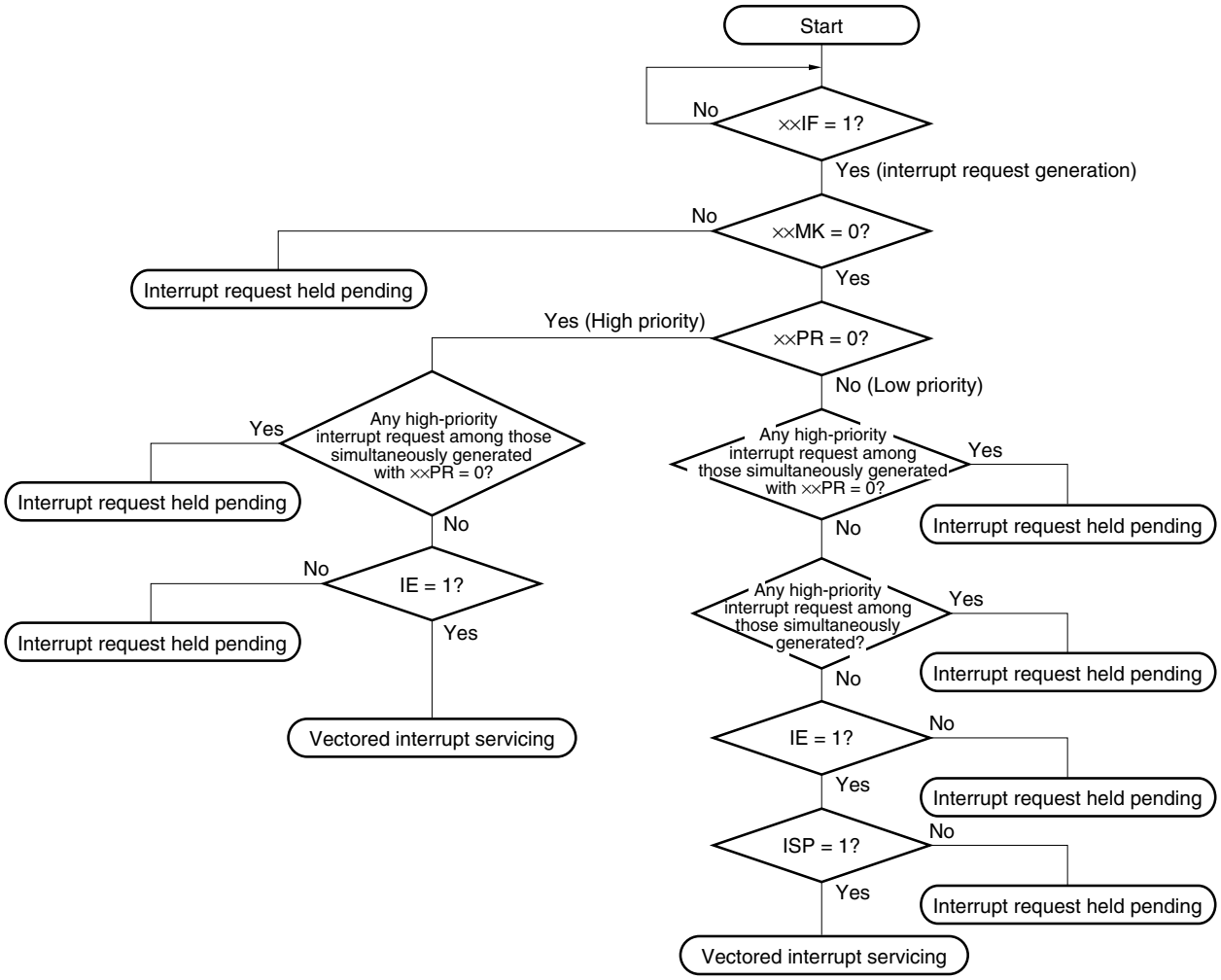
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 18-7 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 ISP flag. 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 18-7. Interrupt Request Acknowledgment Processing Algorithm



××IF: Interrupt request flag

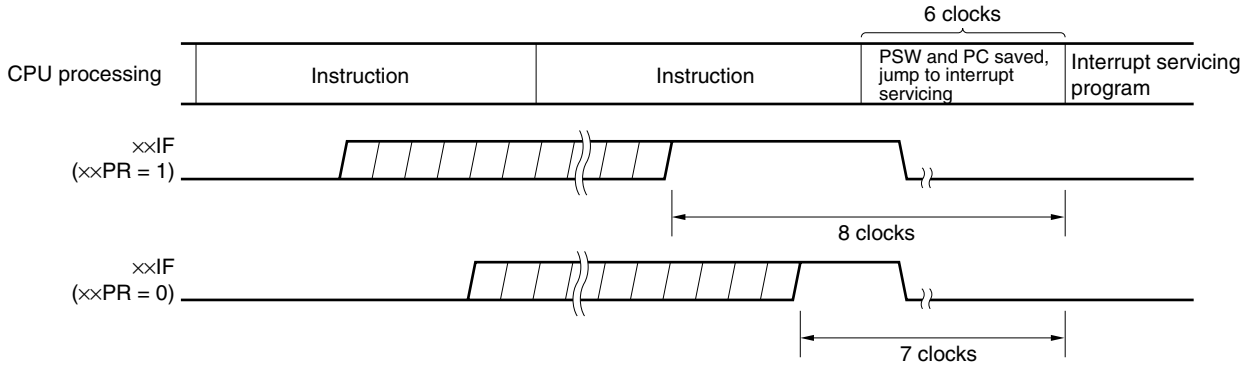
××MK: Interrupt mask flag

××PR: Priority specification flag

IE: Flag that controls acknowledgment of maskable interrupt request (1 = Enable, 0 = Disable)

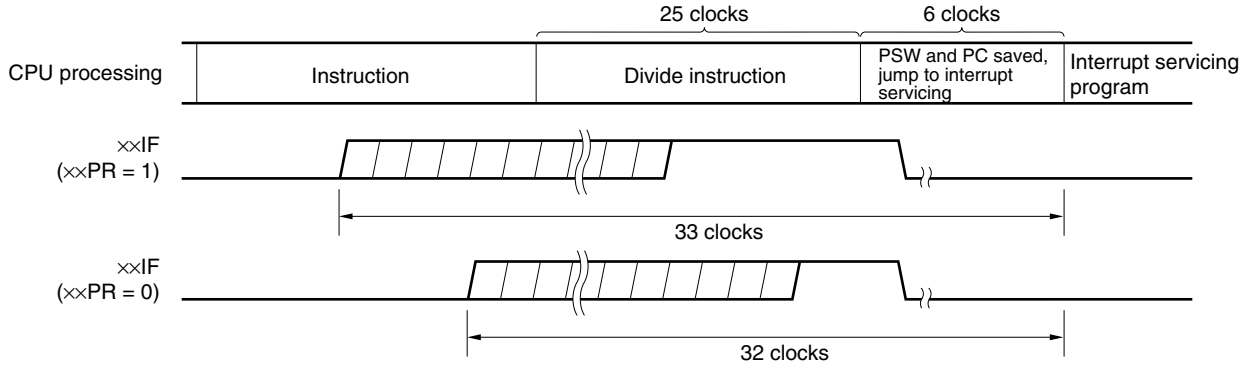
ISP: Flag that indicates the priority level of the interrupt currently being serviced (0 = high-priority interrupt servicing, 1 = No interrupt request acknowledged, or low-priority interrupt servicing)

Figure 18-8. Interrupt Request Acknowledgment Timing (Minimum Time)



Remark 1 clock: 1/f_{CPU} (f_{CPU}: CPU clock)

Figure 18-9. Interrupt Request Acknowledgment Timing (Maximum Time)



Remark 1 clock: 1/f_{CPU} (f_{CPU}: CPU clock)

18.4.2 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 (003EH, 003FH) 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.

18.4.3 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 18-5 shows relationship between interrupt requests enabled for multiple interrupt servicing and Figure 18-10 shows multiple interrupt servicing examples.

Table 18-5. Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing

Multiple Interrupt Request / Interrupt Being Serviced		Maskable Interrupt Request				Software Interrupt Request
		PR = 0		PR = 1		
		IE = 1	IE = 0	IE = 1	IE = 0	
Maskable interrupt	ISP = 0	○	×	×	×	○
	ISP = 1	○	×	○	×	○
Software interrupt		○	×	○	×	○

Remarks 1. ○: Multiple interrupt servicing enabled

2. ×: Multiple interrupt servicing disabled

3. ISP and IE are flags contained in the PSW.

ISP = 0: An interrupt with higher priority is being serviced.

ISP = 1: No interrupt request has been acknowledged, or an interrupt with a lower priority is being serviced.

IE = 0: Interrupt request acknowledgment is disabled.

IE = 1: Interrupt request acknowledgment is enabled.

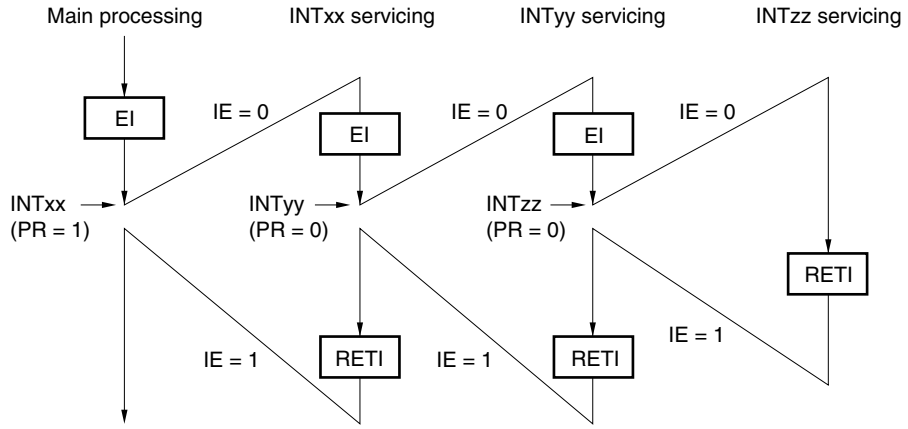
4. PR is a flag contained in PR0L, PR0H, PR1L, and PR1H.

PR = 0: Higher priority level

PR = 1: Lower priority level

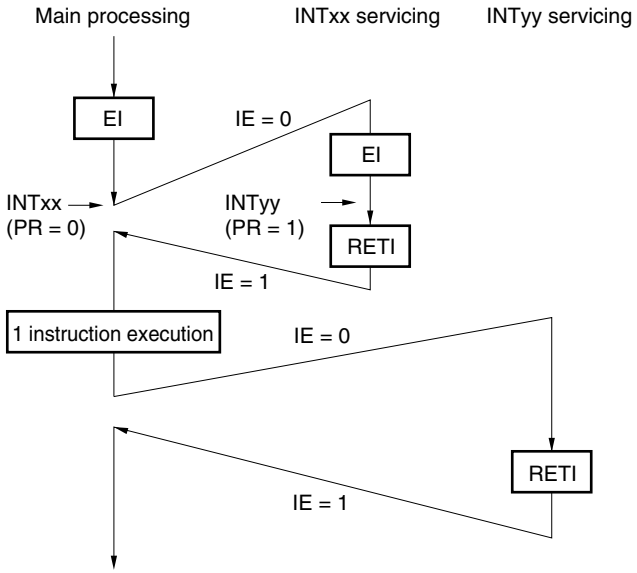
Figure 18-10. Examples of Multiple Interrupt Servicing (1/2)

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

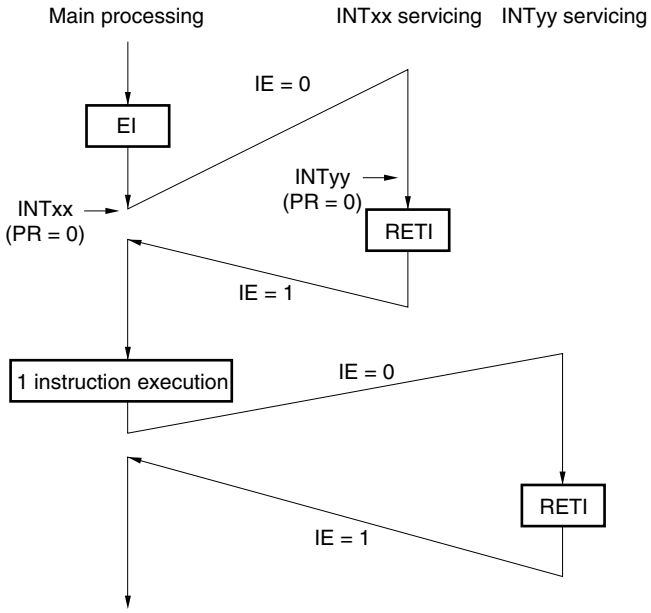


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 0: Higher priority level
PR = 1: Lower priority level
IE = 0: Interrupt request acknowledgment disabled

Figure 18-10. 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 = 0: Higher priority level
IE = 0: Interrupt request acknowledgment disabled

18.4.4 Interrupt request hold

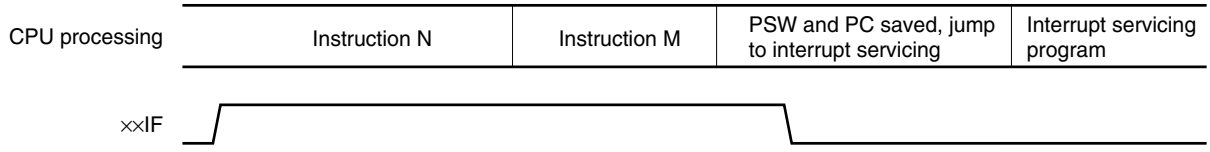
There are instructions where, even if an interrupt request is issued for them while another instruction is being executed, request acknowledgment is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- MOV PSW, #byte
- MOV A, PSW
- MOV PSW, A
- MOV1 PSW. bit, CY
- MOV1 CY, PSW. bit
- AND1 CY, PSW. bit
- OR1 CY, PSW. bit
- XOR1 CY, PSW. bit
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- PUSH PSW
- POP PSW
- BT PSW. bit, $addr16
- BF PSW. bit, $addr16
- BTCLR PSW. bit, $addr16
- EI
- DI
- Manipulation instructions for the IF0L, IF0H, IF1L, IF1H, MK0L, MK0H, MK1L, MK1H, PR0L, PR0H, PR1L, and PR1H 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 18-11 shows the timing at which interrupt requests are held pending.

Figure 18-11. 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 19 KEY INTERRUPT FUNCTION

Item	78K0/KB2-A	78K0/KC2-A
	30 pins	48 pins
Key interrupt	–	6 ch

19.1 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 (KRn).

Table 19-1. Assignment of Key Interrupt Detection Pins

Flag	Description
KRMn	Controls KRn signal in 1-bit units.

Remark n = 0, 5

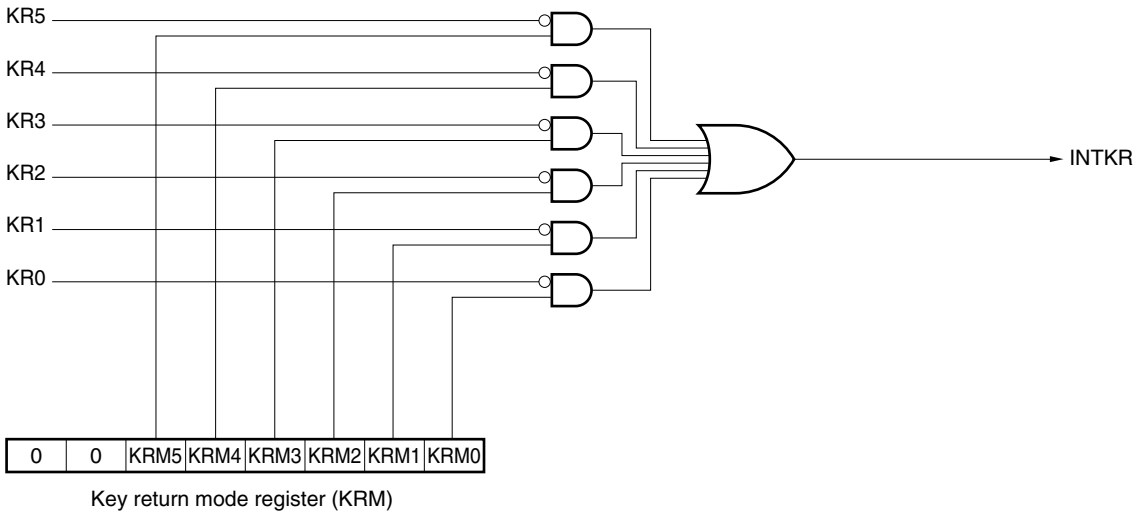
19.2 Configuration of Key Interrupt

The key interrupt includes the following hardware.

Table 19-2. Configuration of Key Interrupt

Item	Configuration
Control register	Key return mode register (KRM)

Figure 19-1. Block Diagram of Key Interrupt



Key return mode register (KRM)

19.3 Register Controlling Key Interrupt

(1) Key return mode register (KRM)

This register controls the KRMn bit using the KRn signal.

KRM is set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears KRM to 00H.

Figure 19-2. Format of Key Return Mode Register (KRM)

Address: FF6EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
KRM	0	0	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 KRMn bits used is set to 1, set bit n (PU7n) 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.

4. Be sure to set bits 6 and 7 of KRM to "0".

Remark n = 0, 5

CHAPTER 20 STANDBY FUNCTION

20.1 Standby Function and Configuration

20.1.1 Standby function

The standby function is mounted onto all 78K0/Kx2-A microcontroller products.

The standby function is designed to reduce the operating current of the system. 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, internal low-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.

Note The 78K0/KB2-A is not provided with a subsystem clock oscillator.

(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 subsystem clock oscillation cannot be stopped. 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. To reduce the operating current of the A/D converter block, execute the STOP instruction after performing the following processing:

- **Stop the A/D conversion operation: Clear bit 7 (ADCS) and bit 0 (ADCE) of the A/D converter mode register (ADM) to 0**
- **Stop the A/D converter voltage booster: Clear bit 1 (VRGV) of the analog reference voltage control register (ADVRC) to 0.**

20.1.2 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. When X1 clock oscillation starts with the internal high-speed oscillation clock or subsystem clock used as the CPU clock, the X1 clock oscillation stabilization time can be checked.

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, and WDT), the STOP instruction and MSTOP (bit 7 of MOC register) = 1 clear OSTC to 00H.

Figure 20-1. Format of Oscillation Stabilization Time Counter Status Register (OSTC)

Address: FFA3H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
OSTC	0	0	0	MOST11	MOST13	MOST14	MOST15	MOST16

MOST11	MOST13	MOST14	MOST15	MOST16	Oscillation stabilization time status		
						f_X = 10 MHz	f_X = 20 MHz
1	0	0	0	0	$2^{11}/f_X$ min.	204.8 μs min.	102.4 μs min.
1	1	0	0	0	$2^{13}/f_X$ min.	819.2 μs min.	409.6 μs min.
1	1	1	0	0	$2^{14}/f_X$ min.	1.64 ms min.	819.2 μs min.
1	1	1	1	0	$2^{15}/f_X$ min.	3.27 ms min.	1.64 ms min.
1	1	1	1	1	$2^{16}/f_X$ min.	6.55 ms min.	3.27 ms min.

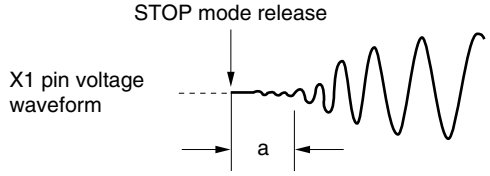
Cautions 1. After the above time has elapsed, the bits are set to 1 in order from MOST11 and remain 1.

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 OSTS to 05H.

Figure 20-2. Format of Oscillation Stabilization Time Select Register (OSTS)

Address: FFA4H After reset: 05H 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	1	$2^{11}/f_X$	204.8 μs	102.4 μs
0	1	0	$2^{13}/f_X$	819.2 μs	409.6 μs
0	1	1	$2^{14}/f_X$	1.64 ms	819.2 μs
1	0	0	$2^{15}/f_X$	3.27 ms	1.64 ms
1	0	1	$2^{16}/f_X$	6.55 ms	3.27 ms
Other than above			Setting prohibited		

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

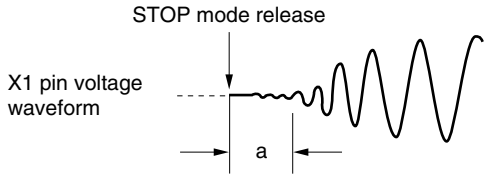
2. Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.

3. 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.

4. 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

20.2 Standby Function Operation

20.2.1 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[Note].

The operating statuses in the HALT mode are shown below.

Note The 78K0/KB2-A is not provided with a subsystem clock.

Table 20-1. 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_{RH})	When CPU Is Operating on X1 Clock (f_X)	When CPU Is Operating on External Main System Clock (f_{EXCLK})
System clock			Clock supply to the CPU is stopped		
	Main system clock	f_{RH}	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)	Status before HALT mode was set is retained
		f_{EXCLK}	Operates or stops by external clock input		Operation continues (cannot be stopped)
	Subsystem clock (f_{SUB})		Status before HALT mode was set is retained		
	Internal low-speed oscillation clock (f_{RL})		Status before HALT mode was set is retained		
CPU			Operation stopped		
Flash memory					
RAM			Status before HALT mode was set is retained		
Port (latch)					
16-bit timer/event counter 00			Operable		
8-bit timer/event counter		50			
		51			
8-bit timer		H0			
		H1			
Real-time counter (RTC)					
Watchdog timer			Operable. Clock supply to watchdog timer stops when “internal low-speed oscillator can be stopped by software” is set by option byte.		
Clock output			Operable		
A/D converter					
Operational amplifier					
Serial interface		UART6			
		CSI10			
		IICA			
Multiplier/divider					
Power-on-clear function					
Low-voltage detection function					
External interrupt					

Remarks 1. f_{RH}: Internal high-speed oscillation clock, f_X: X1 clock
f_{EXCLK}: External main system clock

2. The functions mounted depend on the product. See **1.5 Block Diagram** and **1.6 Outline of Functions**.

Table 20-1. Operating Statuses in HALT Mode (2/2)

HALT Mode Setting / Item			When HALT Instruction Is Executed While CPU Is Operating on Subsystem Clock
System clock			Clock supply to the CPU is stopped
	Main system clock	f_{RH}	Status before HALT mode was set is retained
		f_X	
		f_{EXCLK}	Operates or stops by external clock input
	Subsystem clock (f_{SUB})		Operation continues (cannot be stopped)
	Internal low-speed oscillation clock (f_{RL})		Status before HALT mode was set is retained
CPU			Operation stopped
Flash memory			
RAM			Status before HALT mode was set is retained
Port (latch)			
16-bit timer/event counter 00[Note]			Operable. However, operation disabled when peripheral hardware clock (f_{PRS}) is stopped.
8-bit timer/event counter		50[Note]	Operable
		51[Note]	
8-bit timer		H0	
		H1	
Real-time counter (RTC)			
Watchdog timer			Operable. Clock supply to watchdog timer stops when “internal low-speed oscillator can be stopped by software” is set by option byte.
Clock output			Operable
A/D converter			Not operable
Operational amplifier			Disables operation
Serial interface		UART6	Operable
		CSI10[Note]	
		IICA[Note]	
Multiplier/divider			
Power-on-clear function			
Low-voltage detection function			
External interrupt			

Note When the CPU is operating on the subsystem clock and the internal high-speed oscillation clock and high-speed system clock have been stopped, do not start operation of these functions on the external clock input from peripheral hardware pins.

Remarks 1. f_{RH}: Internal high-speed oscillation clock, f_X: X1 clock
f_{EXCLK}: External main system clock

2. The functions mounted depend on the product. See **1.5 Block Diagram** and **1.6 Outline of Functions**.

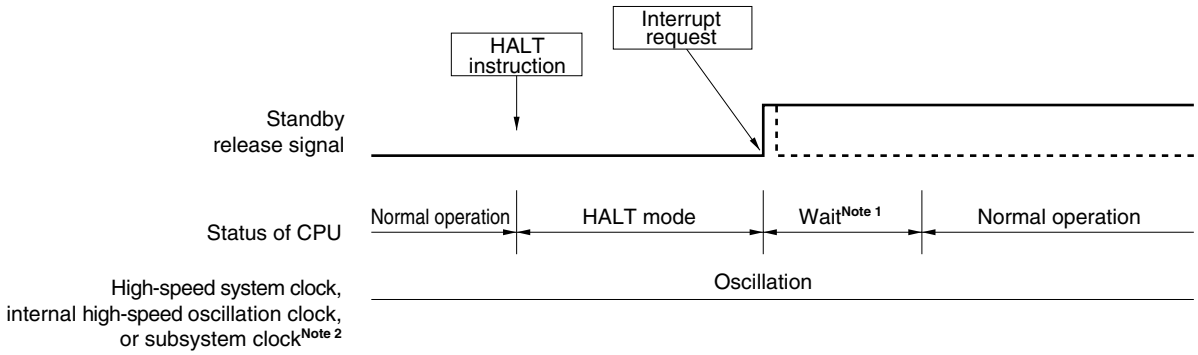
(2) HALT mode release

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 20-3. HALT Mode Release by Interrupt Request Generation



Notes 1. The wait time is as follows:

- When vectored interrupt servicing is carried out: 11 or 12 clocks
- When vectored interrupt servicing is not carried out: 4 or 5 clocks

2. The 78K0/KB2-A is not provided with a subsystem clock.

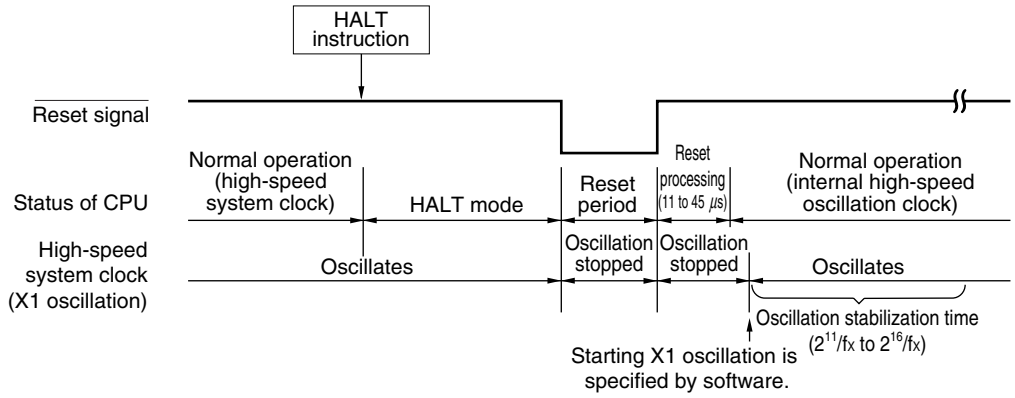
Remark The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

(b) Release by reset signal generation

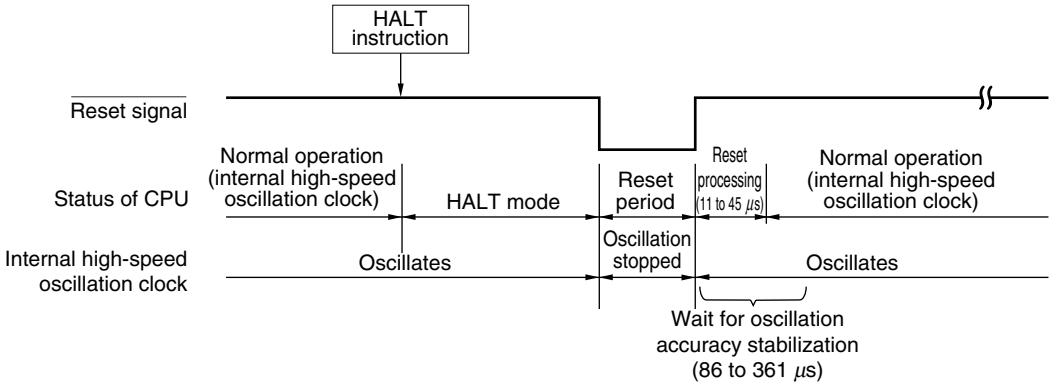
When the reset signal is generated, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 20-4. HALT Mode Release by Reset

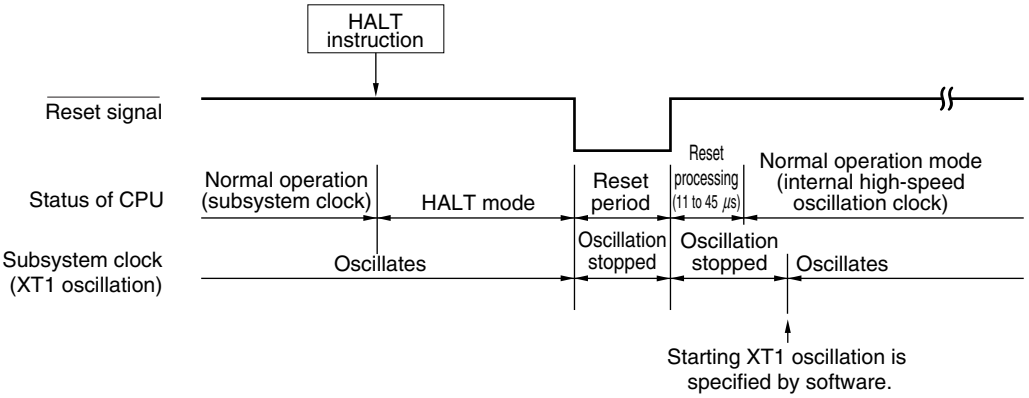
(1) When high-speed system clock is used as CPU clock



(2) When internal high-speed oscillation clock is used as CPU clock



(3) When subsystem clock is used as CPU clock[Note]



Note The 78K0/KB2 is not provided with a subsystem clock.

Remark fx: X1 clock oscillation frequency

Table 20-2. Operation in Response to Interrupt Request in HALT Mode

Release Source	MK××	PR××	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Next address instruction execution
	0	0	1	×	Interrupt servicing execution
	0	1	0	1	Next address instruction execution
	0	1	×	0	
	0	1	1	1	Interrupt servicing execution
	1	×	×	×	HALT mode held
Reset	–	–	×	×	Reset processing

×: don't care

20.2.2 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 20-3. 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_{RH})	When CPU Is Operating on X1 Clock (f_X)	When CPU Is Operating on External Main System Clock (f_{EXCLK})
System clock			Clock supply to the CPU is stopped		
	Main system clock	f_{RH}	Stopped		
		f_X			
		f_{EXCLK}	Input invalid		
	Subsystem clock (f_{SUB})		Status before STOP mode was set is retained		
	Internal low-speed oscillation clock (f_{RL})		Status before STOP mode was set is retained		
CPU			Operation stopped		
Flash memory					
RAM			Status before STOP mode was set is retained		
Port (latch)					
16-bit timer/event counter 00Note			Operation stopped		
8-bit timer/event counter		50Note	Operable only when TI50 is selected as the count clock		
		51Note	Operable only when TI51 is selected as the count clock		
8-bit timer		H0	Operable only when TM50 output is selected as the count clock during 8-bit timer/event counter 50 operation		
		H1	Operable only when f_{RL}, $f_{RL}/2^7$, $f_{RL}/2^9$ is selected as the count clock		
Real-time counter (RTC)			Operable only when subsystem clock is selected as the count clock		
Watchdog timer			Operable. Clock supply to watchdog timer stops when “internal low-speed oscillator can be stopped by software” is set by option byte.		
Clock output			Operable only when subsystem clock is selected as the count clock		
A/D converter			Operation stopped		
Operational amplifier			Operable		
Serial interface	UART6		Operable only when TM50 output is selected as the serial clock during 8-bit timer/event counter 50 operation		
	CSI10Note		Operable only when external clock is selected as the serial clock		
	IICANote		Wakeup by address match operable		
Multiplier/divider			Operation stopped		
Power-on-clear function			Operable		
Low-voltage detection function					
External interrupt					

Note Do not start operation of these functions on the external clock input from peripheral hardware pins in the stop mode.

Remarks 1. f_{RH}: Internal high-speed oscillation clock, f_X: X1 clock
f_{EXCLK}: External main system clock

2. The functions mounted depend on the product. See **1.5 Block Diagram** and **1.6 Outline of Functions**.

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. Even if "internal low-speed oscillator can be stopped by software" is selected by the option byte, the internal low-speed oscillation clock continues in the STOP mode in the status before the STOP mode is set. To stop the internal low-speed oscillator's oscillation in the STOP mode, stop it by software 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), switch the CPU clock to the internal high-speed oscillation clock before the execution of the STOP instruction using the following procedure.

<1> Set RSTOP to 0 (starting oscillation of the internal high-speed oscillator) → <2> Set MCM0 to 0 (switching the CPU from X1 oscillation to internal high-speed oscillation) → <3> Check that MCS is 0 (checking the CPU clock) → <4> Check that RSTS is 1 (checking internal high-speed oscillation operation) → <5> Execute the STOP instruction

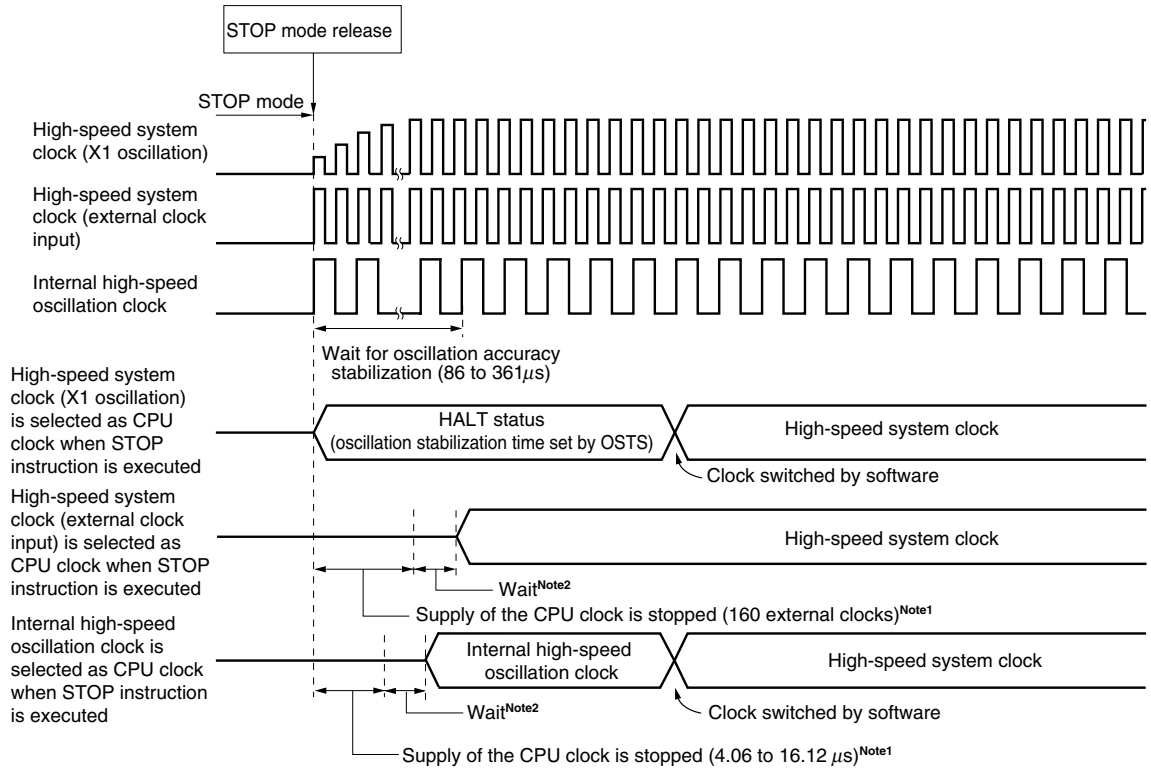
Before changing the CPU clock from the internal high-speed oscillation clock to the high-speed system clock (X1 oscillation) after the STOP mode is released, check the oscillation stabilization time with the oscillation stabilization time counter status register (OSTC).

4. If the STOP instruction is executed when AMPH = 1, supply of the CPU clock is stopped for 4.06 to 16.12 μs after the STOP mode is released when the internal high-speed oscillation clock is selected as the CPU clock, or for the duration of 160 external clocks when the high-speed system clock (external clock input) is selected as the CPU clock.

5. Execute the STOP instruction after having confirmed that the internal high-speed oscillator is operating stably (RSTS = 1).

(2) STOP mode release

Figure 20-5. Operation Timing When STOP Mode Is Released (When Unmasked Interrupt Request Is Generated)



Notes 1. When AMPH = 1

2. The wait time is as follows:

- When vectored interrupt servicing is carried out: 17 or 18 clocks
- When vectored interrupt servicing is not carried out: 11 or 12 clocks

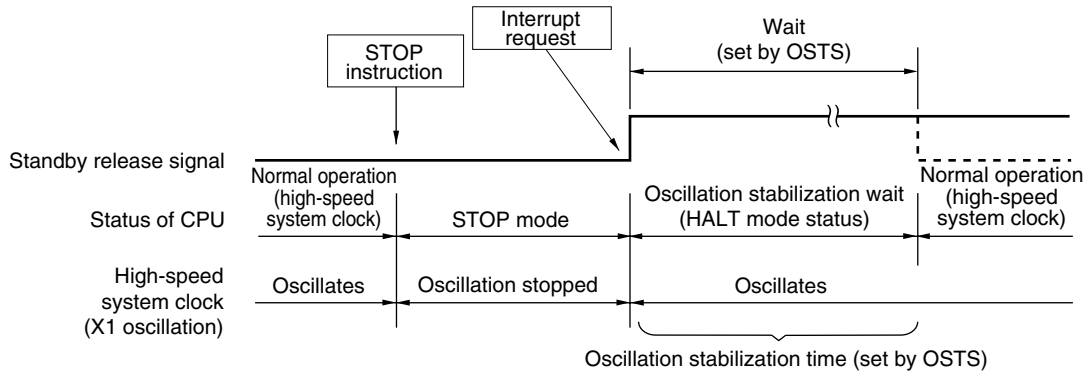
The STOP mode can be released by the following two sources.

(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.

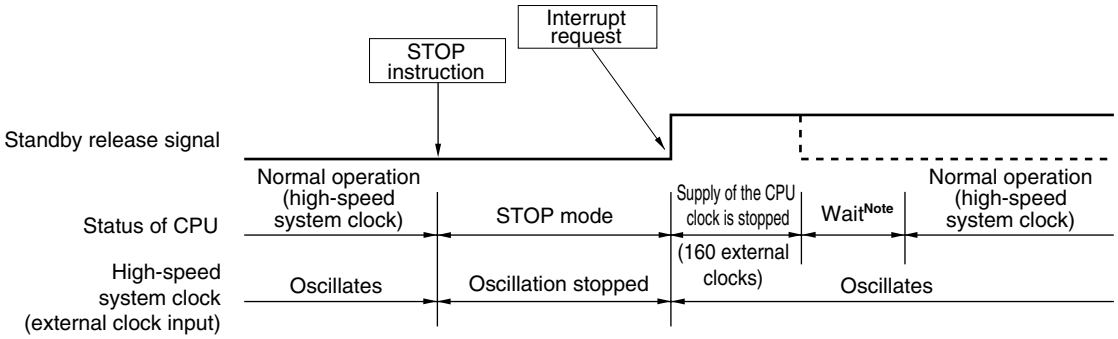
Figure 20-6. STOP Mode Release by Interrupt Request Generation (1/2)

(1) When high-speed system clock (X1 oscillation) is used as CPU clock

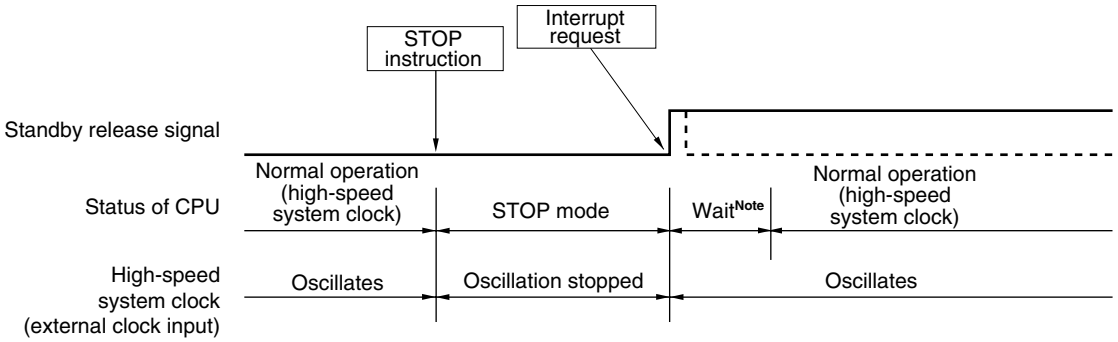


(2) When high-speed system clock (external clock input) is used as CPU clock

- When AMPH = 1



- When AMPH = 0



Note The wait time is as follows:

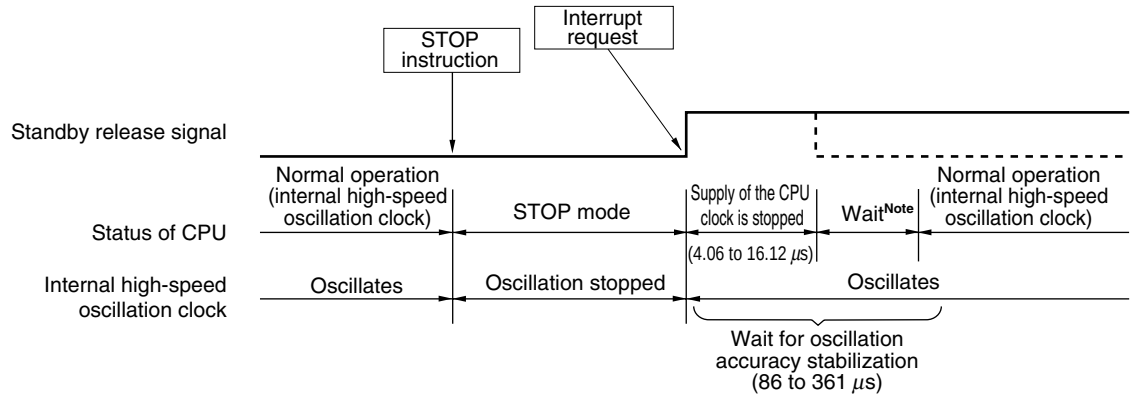
- When vectored interrupt servicing is carried out: 17 or 18 clocks
- When vectored interrupt servicing is not carried out: 11 or 12 clocks

Remark The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

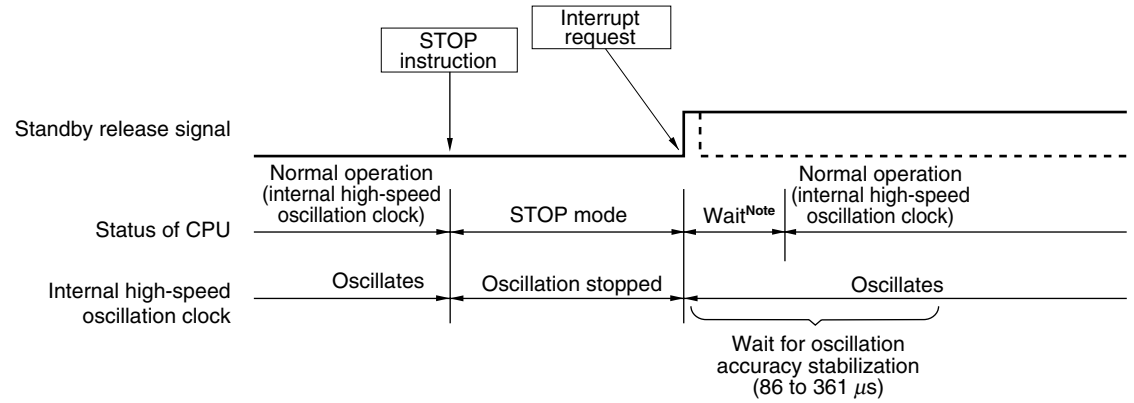
Figure 20-6. STOP Mode Release by Interrupt Request Generation (2/2)

(3) When internal high-speed oscillation clock is used as CPU clock

- When AMPH = 1



- When AMPH = 0



Note The wait time is as follows:

- When vectored interrupt servicing is carried out: 17 or 18 clocks
- When vectored interrupt servicing is not carried out: 11 or 12 clocks

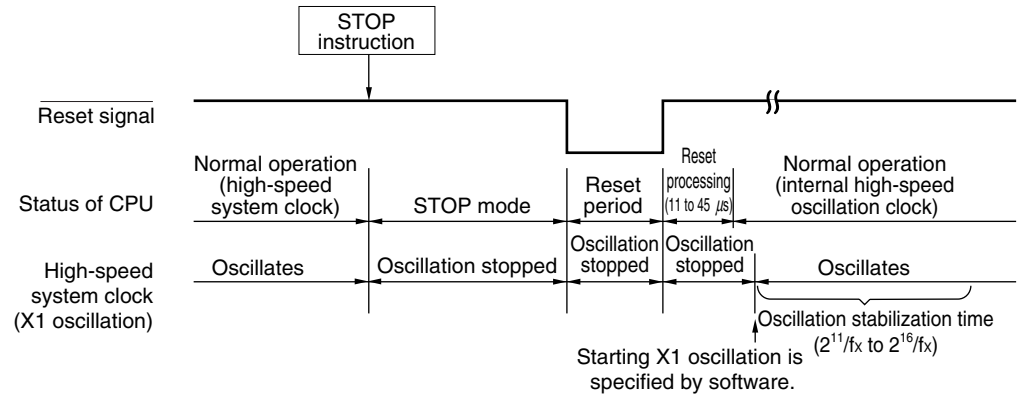
Remark The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

(b) Release by reset signal generation

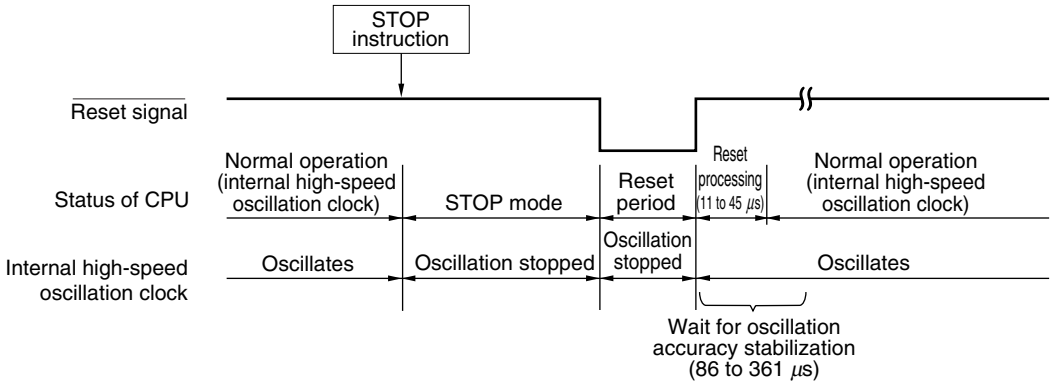
When the reset signal is generated, STOP mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 20-7. 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

Table 20-4. Operation in Response to Interrupt Request in STOP Mode

Release Source	MK××	PR××	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Next address instruction execution
	0	0	1	×	Interrupt servicing execution
	0	1	0	1	Next address instruction execution
	0	1	×	0	
	0	1	1	1	Interrupt servicing execution
	1	×	×	×	STOP mode held
Reset	–	–	×	×	Reset processing

×: don't care

CHAPTER 21 RESET FUNCTION

The reset function is mounted onto all 78K0/Kx2-A microcontroller products.

The following four operations are available to generate a reset signal.

(1) External reset input via $\overline{\text{RESET}}$ pin
(2) Internal reset by watchdog timer program loop detection
(3) Internal reset by comparison of supply voltage and detection voltage of power-on-clear (POC) circuit
(4) Internal reset by comparison of supply voltage and detection voltage of low-power-supply detector (LVI)

External and internal resets have no functional differences. In both cases, program execution starts at the address at 0000H and 0001H when the reset signal is generated.

A reset is applied when a low level is input to the $\overline{\text{RESET}}$ pin, the watchdog timer overflows, or by POC and LVI circuit voltage detection, and each item of hardware is set to the status shown in Tables 21-1 and 21-2. Each pin is high impedance during reset signal generation or during the oscillation stabilization time just after a reset release.

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 21-2** to **21-4**) 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 22 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 23 LOW-VOLTAGE DETECTOR**) after reset processing.

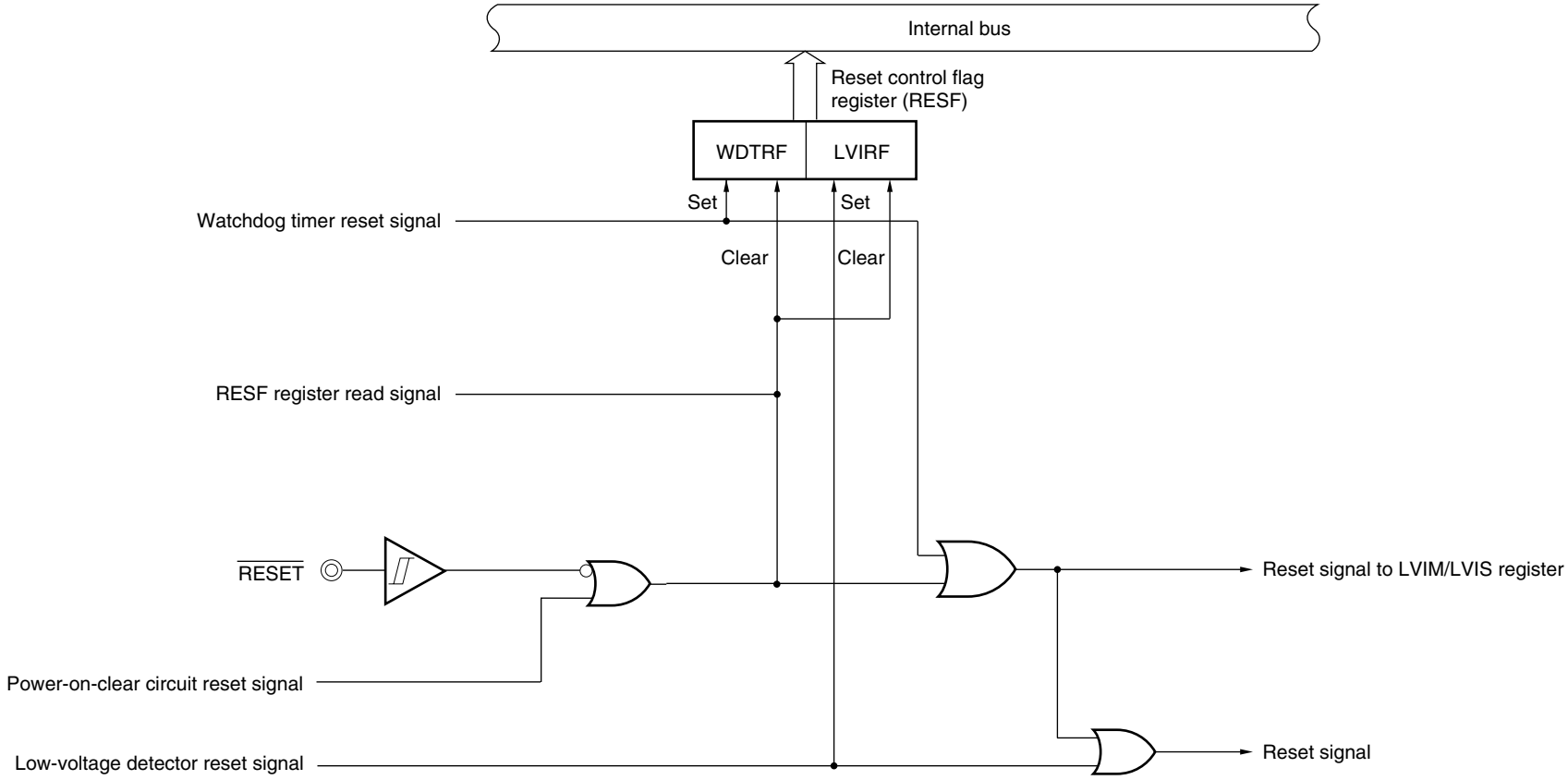
Cautions 1. For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin.

2. During reset signal generation, the X1 clock, XT1 clock[Note], internal high-speed oscillation clock, and internal low-speed oscillation clock stop oscillating. External main system clock input become invalid.

3. When the STOP mode is released by a reset, the STOP mode contents are held during reset input. However, the port pins become high-impedance.

Note The 78K0/KB2-A is not provided with XT1 clock.

Figure 21-1. 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 selection register

Figure 21-2. Timing of Reset by $\overline{\text{RESET}}$ Input

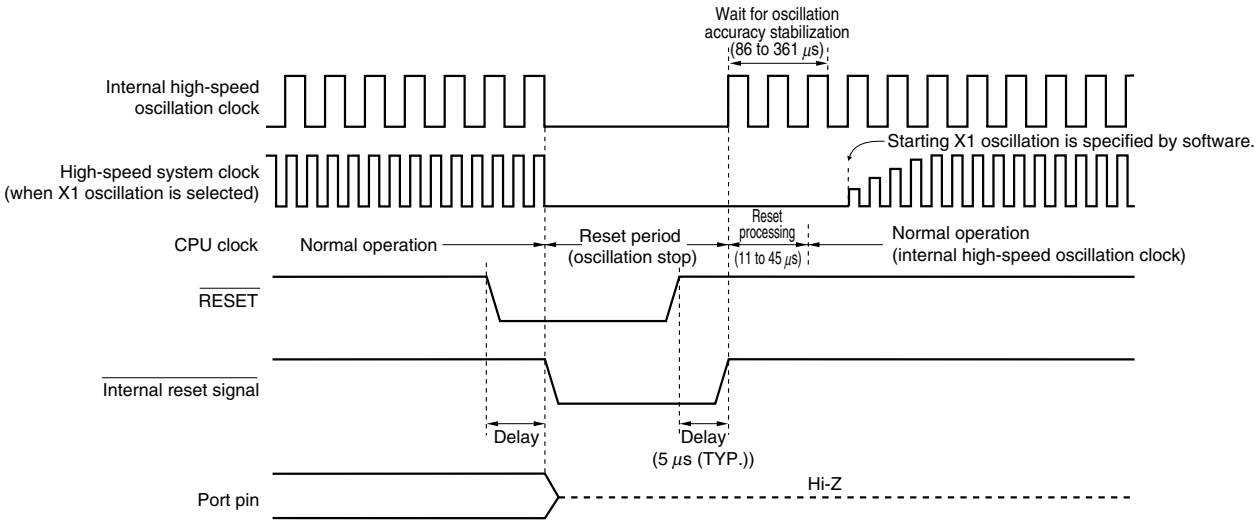
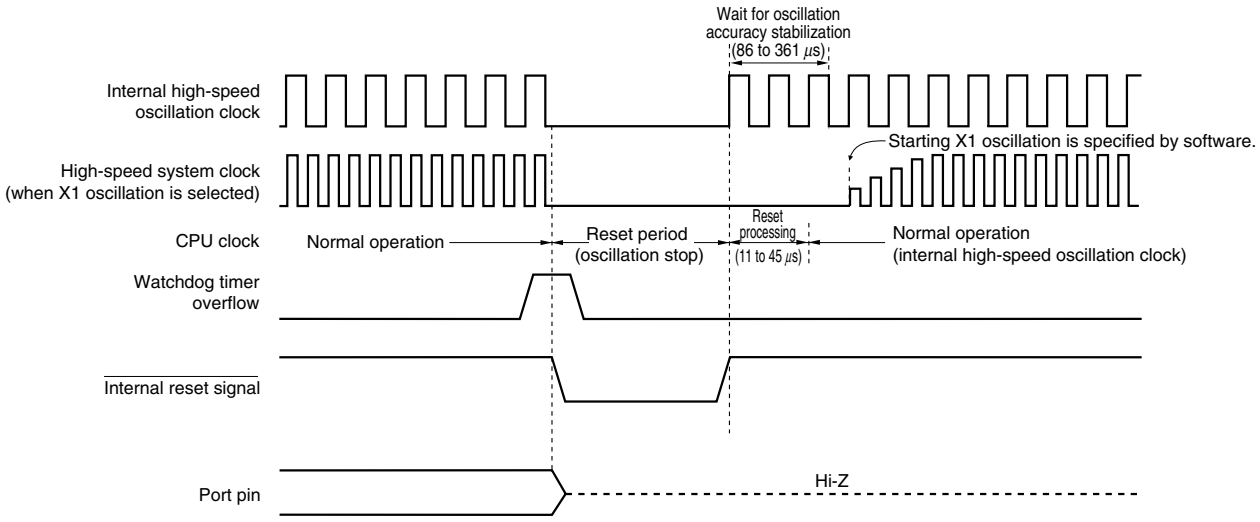
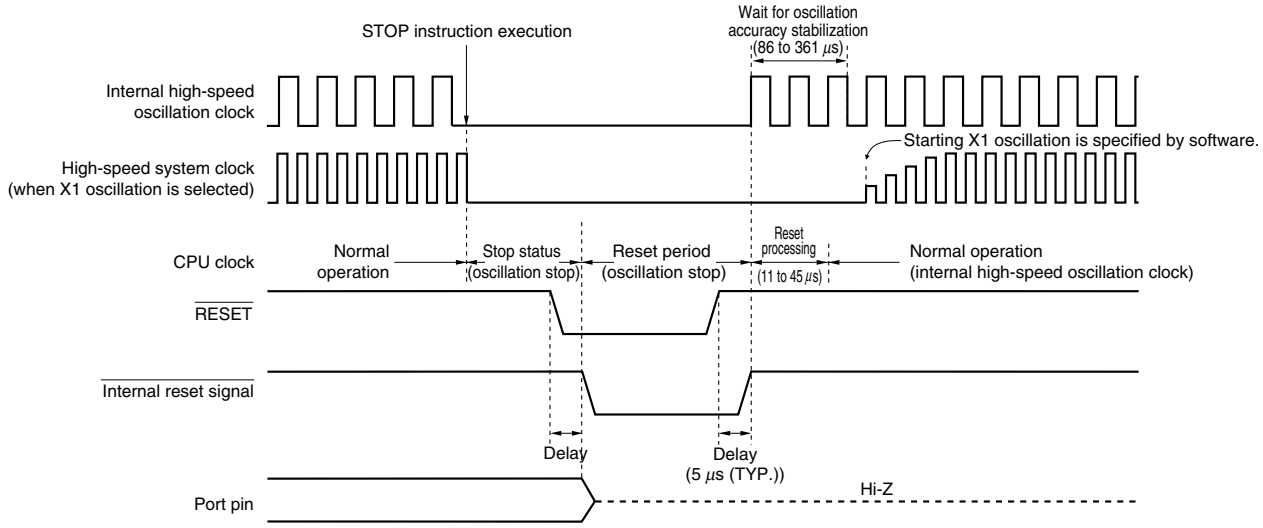


Figure 21-3. Timing of Reset Due to Watchdog Timer Overflow



Caution A watchdog timer internal reset resets the watchdog timer.

Figure 21-4. Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input



Remarks 1. For the reset timing of the power-on-clear circuit and low-voltage detector, see **CHAPTER 22 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 23 LOW-VOLTAGE DETECTOR**.

Table 21-1. Operation Statuses During Reset Period

Item		During Reset Period
System clock		Clock supply to the CPU is stopped.
Main system clock	f_{RH}	Operation stopped
	f_X	Operation stopped (pin is I/O port mode)
	f_{EXCLK}	Clock input invalid (pin is I/O port mode)
Subsystem clock (f_{SUB})		Operation stopped (pin is I/O port mode)
Internal low-speed oscillation clock (f_{RL})		Operation stopped
CPU		
Flash memory		
RAM		
Port (latch)		
16-bit timer/event counter 00		
8-bit timer/event counter	50	
	51	
8-bit timer	H0	
	H1	
Real-time counter (RTC)		
Watchdog timer		
Clock output		
A/D converter		
Serial interface	UART6	
	CSI10	
	IICA	
Multiplier/divider		
Power-on-clear function		Operable
Low-voltage detection function		Operation stopped
External interrupt		

Remarks 1. f_{RH}: Internal high-speed oscillation clock, f_X: X1 clock
f_{EXCLK}: External main system clock

2. The functions mounted depend on the product. See **1.5 Block Diagram** and **1.6 Outline of Functions**.

Table 21-2. Hardware Statuses After Reset Acknowledgment (1/4)

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)		02H
RAM	Data memory	Undefined[Note 2]
	General-purpose registers	Undefined[Note 2]
Port registers (P0 to P4, P6 to P8, P12) (output latches)		00H
Port mode registers (PM0 to PM4, PM6 to PM8, PM12)		FFH
Pull-up resistor option registers (PU0, PU1, PU3, PU4, PU7, PU12)		00H
Internal memory size switching register (IMS)		CFH[Note 3]

Notes 1. During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

2. When a reset is executed in the standby mode, the pre-reset status is held even after reset.

3. The initial values of the internal memory size switching register (IMS) after a reset release is constant (IMS = CFH) in all products of the 78K0/Kx2-A microcontrollers, regardless of the internal memory capacity. Therefore, set the value corresponding to each product as indicated in Table 3-1.

Remark The special function register (SFR) mounted depend on the product. See **3.2.3 Special function registers (SFRs)**.

Table 21-2. Hardware Statuses After Reset Acknowledgment (2/4)

Hardware		Status After Reset Acknowledgment[Note 1]
Clock operation mode select register (OSCCTL)		00H
Processor clock control register (PCC)		01H
Internal oscillation mode register (RCM)		80H
Main OSC control register (MOC)		80H
Main clock mode register (MCM)		00H
Oscillation stabilization time counter status register (OSTC)		00H
Oscillation stabilization time select register (OSTS)		05H
16-bit timer/event counter 00	Timer counter 00 (TM00)	0000H
	Capture/compare registers 000, 010 (CR000, CR010)	0000H
	Mode control register 00 (TMC00)	00H
	Prescaler mode register 00 (PRM00)	00H
	Capture/compare control register 00 (CRC00)	00H
	Timer output control register 00 (TOC00)	00H
8-bit timer/event counters 50, 51	Timer counters 50, 51 (TM50, TM51)	00H
	Compare registers 50, 51 (CR50, CR51)	00H
	Timer clock selection registers 50, 51 (TCL50, TCL51)	00H
	Mode control registers 50, 51 (TMC50, TMC51)	00H
8-bit timers H0, H1	Compare registers 00, 10, 01, 11 (CMP00, CMP10, CMP01, CMP11)	00H
	Mode registers (TMHMD0, TMHMD1)	00H
	Carrier control register 1 (TMCYC1)[Note 2]	00H
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

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. 8-bit timer H1 only.

Remark The special function register (SFR) mounted depend on the product. See **3.2.3 Special function registers (SFRs)**.

Table 21-2. Hardware Statuses After Reset Acknowledgment (3/4)

Hardware		Status After Reset Acknowledgment[Note 1]
Clock output	Clock output selection register (CKS)	00H
Watchdog timer	Enable register (WDTE)	1AH/9AH[Note 2]
A/D converter	12-bit A/D conversion result register (ADCR)	0000H
	8-bit A/D conversion result register (ADCRH)	00H
	A/D converter mode register (ADM)	00H
	A/D converter mode register 1(ADM1)	00H
	Analog reference voltage control register (ADVRC)	00H
	Analog input channel specification register (ADS)	00H
	A/D port configuration register (ADPC)	10H
Serial interface UART6	Receive buffer register 6 (RXB6)	FFH
	Transmit buffer register 6 (TXB6)	FFH
	Asynchronous serial interface operation mode register 6 (ASIM6)	01H
	Asynchronous serial interface reception error status register 6 (ASIS6)	00H
	Asynchronous serial interface transmission status register 6 (ASIF6)	00H
	Clock selection register 6 (CKSR6)	00H
	Baud rate generator control register 6 (BRGC6)	FFH
	Asynchronous serial interface control register 6 (ASICL6)	16H
	Input switch control register (ISC)	00H
Serial interface CSI10	Transmit buffer register 10 (SOTB10)	00H
	Serial I/O shift register 10 (SIO10)	00H
	Serial operation mode register 10 (CSIM10)	00H
	Serial clock selection register 10 (CSIC10)	00H
Serial interface IICA	Shift register (IICA)	00H
	Status register 0 (IICAS0)	00H
	Flag register 0 (IICAF0)	00H
	Control register 0 (IICACTL0)	00H
	Control register 1 (IICACTL1)	00H
	Low-level width setting register (IICWL)	FFH
	High-level width setting register (IICWH)	FFH
	Slave address register 0 (SVA0)	00H
Multiplier/divider	Remainder data register 0 (SDR0)	0000H
	Multiplication/division data register A0 (MDA0H, MDA0L)	0000H
	Multiplication/division data register B0 (MDB0)	0000H
	Multiplier/divider control register 0 (DMUC0)	00H
Key interrupt	Key return mode register (KRM)	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.

Remark The special function register (SFR) mounted depend on the product. See **3.2.3 Special function registers (SFRs)**.

Table 21-2. Hardware Statuses After Reset Acknowledgment (4/4)

Hardware		Status After Reset Acknowledgment[Note 1]
Reset function	Reset control flag register (RESF)	00H[Note 2]
Low-voltage detector	Low-voltage detection register (LVIM)	00H[Note 2]
	Low-voltage detection level selection register (LVIS)	00H[Note 2]
Interrupt	Request flag registers 0L, 0H, 1L, 1H (IF0L, IF0H, IF1L, IF1H)	00H
	Mask flag registers 0L, 0H, 1L, 1H (MK0L, MK0H, MK1L, MK1H)	FFH
	Priority specification flag registers 0L, 0H, 1L, 1H (PR0L, PR0H, PR1L, PR1H)	FFH
	External interrupt rising edge enable register (EGP0, EGP1)	00H
	External interrupt falling edge enable register (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 WDT	Reset by LVI
RESF	WDTRF flag	Cleared (0)	Cleared (0)	Set (1)	Held
	LVIRF flag			Held	Set (1)
LVIM		Cleared (00H)	Cleared (00H)	Cleared (00H)	Held
LVIS					

Remark The special function register (SFR) mounted depend on the product. See **3.2.3 Special function registers (SFRs)**.

21.1 Register for Confirming Reset Source

Many internal reset generation sources exist in the 78K0/Kx2-A microcontrollers. 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 21-5. Format of Reset Control Flag Register (RESF)

Address: FFACH After reset: 00H[Note] R

Symbol	7	6	5	4	3	2	1	0
RESF	0	0	0	WDTRF	0	0	0	LVIRF

WDTRF	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.

Note The value after reset varies depending on the reset source.

Caution Do not read data by a 1-bit memory manipulation instruction.

The status of RESF when a reset request is generated is shown in Table 21-3.

Table 21-3. RESF Status When Reset Request Is Generated

Flag \ Reset Source	$\overline{\text{RESET}}$ Input	Reset by POC	Reset by WDT	Reset by LVI
WDTRF	Cleared (0)	Cleared (0)	Set (1)	Held
LVIRF			Held	Set (1)

CHAPTER 22 POWER-ON-CLEAR CIRCUIT

22.1 Functions of Power-on-Clear Circuit

The power-on-clear circuit (POC) is mounted onto all 78K0/Kx2-A microcontroller products.
The power-on-clear circuit has the following functions.

- Generates internal reset signal at power on.
  In the 1.59 V POC mode (option byte: POCMODE = 0), the reset signal is released when the supply voltage (V_{DD}) exceeds 1.59 V ±0.15 V.
  In the 2.7 V/1.59 V POC mode (option byte: POCMODE = 1), the reset signal is released when the supply voltage (V_{DD}) exceeds 2.7 V ±0.2 V.

- Compares supply voltage (V_{DD}) and detection voltage (V_{POC} = 1.59 V ±0.15 V), generates internal reset signal when $V_{DD} < V_{POC}$.

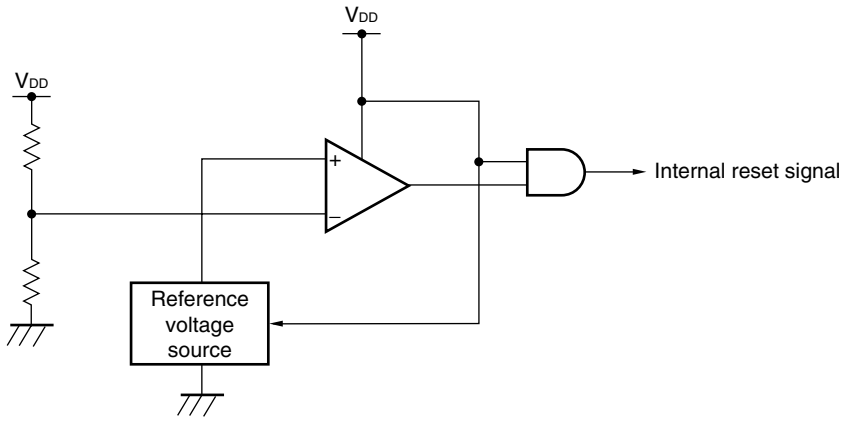
Caution If an internal reset signal is generated in the POC circuit, the reset control flag register (RESF) is cleared to 00H.

Remark 78K0/Kx2-A microcontrollers incorporate 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) or low-voltage-detector (LVI). 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 21 RESET FUNCTION**.

22.2 Configuration of Power-on-Clear Circuit

The block diagram of the power-on-clear circuit is shown in Figure 22-1.

Figure 22-1. Block Diagram of Power-on-Clear Circuit



22.3 Operation of Power-on-Clear Circuit

(1) In 1.59 V POC mode (option byte: POCMODE = 0)

- An internal reset signal is generated on power application. When the supply voltage (V_{DD}) exceeds the detection voltage (V_{POC} = 1.59 V ±0.15 V), the reset status is released.
- The supply voltage (V_{DD}) and detection voltage (V_{POC} = 1.59 V ±0.15 V) are compared. When $V_{DD} < V_{POC}$, the internal reset signal is generated. It is released when $V_{DD} \geq V_{POC}$.

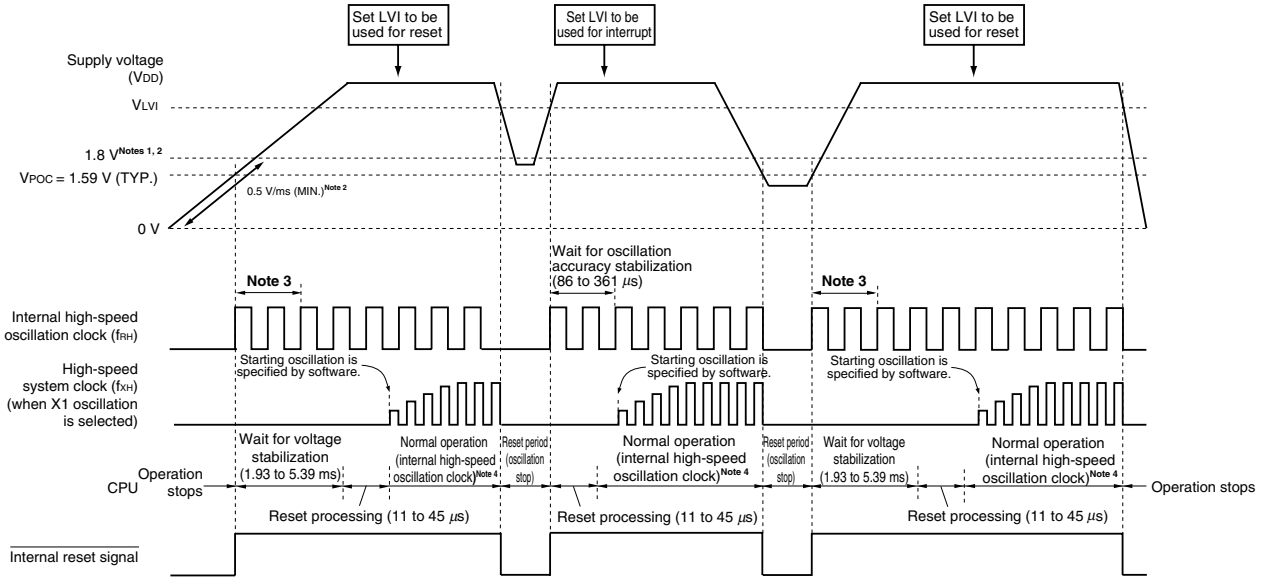
(2) In 2.7 V/1.59 V POC mode (option byte: POCMODE = 1)

- An internal reset signal is generated on power application. When the supply voltage (V_{DD}) exceeds the detection voltage (V_{DDPOC} = 2.7 V ±0.2 V), the reset status is released.
- The supply voltage (V_{DD}) and detection voltage (V_{POC} = 1.59 V ±0.15 V) are compared. When $V_{DD} < V_{POC}$, the internal reset signal is generated. It is released when $V_{DD} \geq V_{DDPOC}$.

The timing of generation of the internal reset signal by the power-on-clear circuit and low-voltage detector is shown below.

Figure 22-2. Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector (1/2)

(1) In 1.59 V POC mode (option byte: POCMODE = 0)



Notes 1. The guaranteed operation range is 1.8 V ≤ V_{DD} ≤ 5.5 V. To set the voltage range below the guaranteed operation range to the reset state when the supply voltage falls, use the reset function of the low-voltage detector, or input a low level to the $\overline{\text{RESET}}$ pin.

2. If the voltage rises to 1.8 V at a rate slower than 0.5 V/ms (MIN.) on power application, input a low level to the $\overline{\text{RESET}}$ pin after power application and before the voltage reaches 1.8 V, or set the 2.7 V/1.59 V POC mode by using an option byte (POCMODE = 1).

3. The oscillation accuracy stabilization time of the internal high-speed oscillation clock is included in the internal voltage stabilization time.

4. The CPU clock can be switched from the internal high-speed oscillation clock to the high-speed system clock or to the subsystem clock Note 5. To use the X1 clock, use the OSTC register to confirm the lapse of the oscillation stabilization time. To use the XT1 clock Note 5, use the timer function for confirmation of the lapse of the stabilization time.

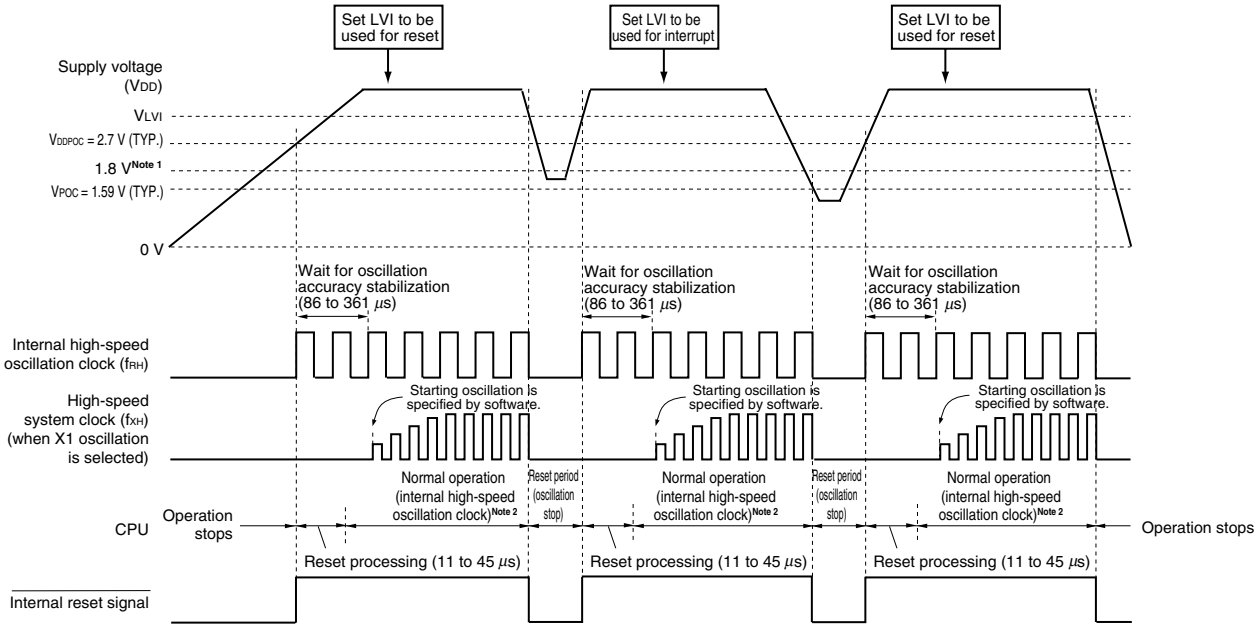
5. The 78K0/KB2-A is not provided with subsystem clock and XT1 clock.

Caution Set the low-voltage detector by software after the reset status is released (see CHAPTER 23 LOW-VOLTAGE DETECTOR).

Remark V_{LVI}: LVI detection voltage
V_{POC}: POC detection voltage

Figure 22-2. Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector (2/2)

(2) In 2.7 V/1.59 V POC mode (option byte: POCMODE = 1)



Notes 1. The guaranteed operation range is 1.8 V ≤ V_{DD} ≤ 5.5 V. To set the voltage range below the guaranteed operation range to the reset state when the supply voltage falls, use the reset function of the low-voltage detector, or input a low level to the $\overline{\text{RESET}}$ pin.

2. The CPU clock can be switched from the internal high-speed oscillation clock to the high-speed system clock or subsystem clock Note 3. To use the X1 clock, use the OSTC register to confirm the lapse of the oscillation stabilization time. To use the XT1 clock Note 3, use the timer function for confirmation of the lapse of the stabilization time.

3. The 78K0/KB2-A is not provided with subsystem clock and XT1 clock.

Cautions 1. Set the low-voltage detector by software after the reset status is released (see CHAPTER 23 LOW-VOLTAGE DETECTOR).

2. A voltage oscillation stabilization time of 1.93 to 5.39 ms is required after the supply voltage reaches 1.59 V (TYP.). If the supply voltage rises from 1.59 V (TYP.) to 2.7 V (TYP.) within 1.93 ms, the power supply oscillation stabilization time of 0 to 5.39 ms is automatically generated before reset processing.

Remark V_{LVI}: LVI detection voltage
V_{POC}: POC detection voltage

22.4 Cautions for Power-on-Clear Circuit

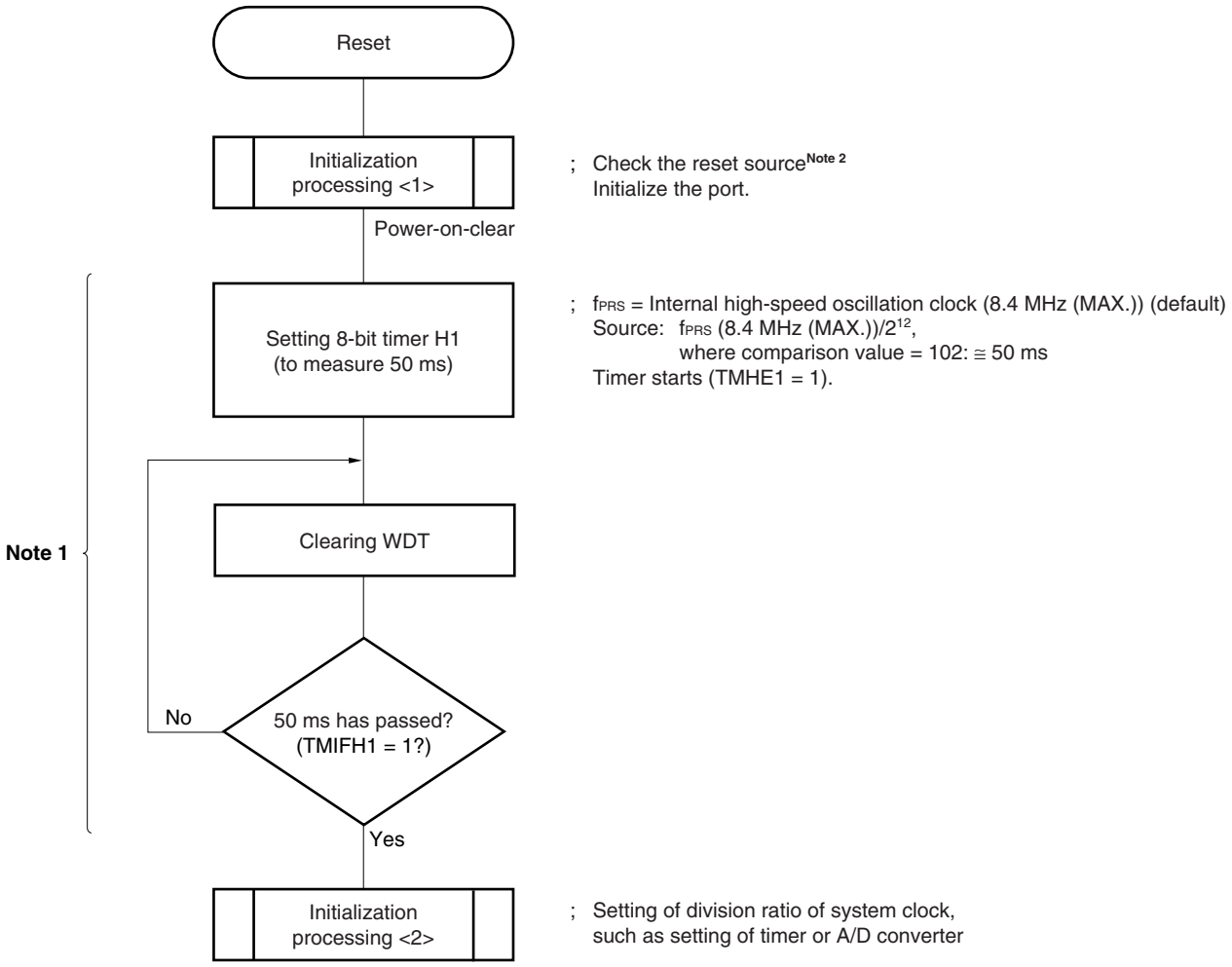
In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the POC detection voltage (V_{POC}), the system may be repeatedly reset and released from the reset status. In this case, the time from release of reset to the start of the operation of the microcontroller 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 22-3. Example of Software Processing After Reset Release (1/2)

• If supply voltage fluctuation is 50 ms or less in vicinity of POC detection voltage

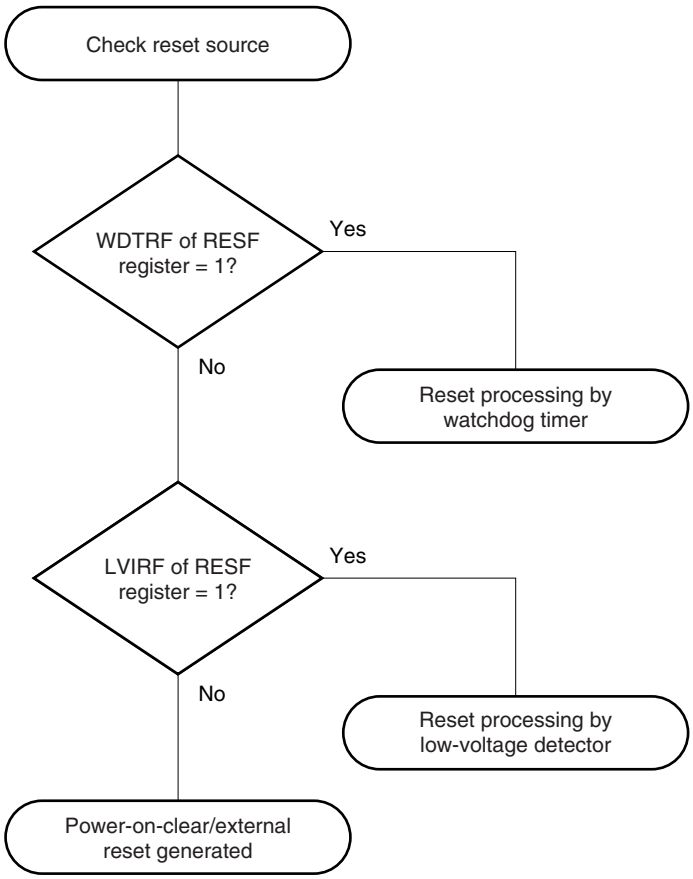


Notes 1. If reset is generated again during this period, initialization processing <2> is not started.

2. A flowchart is shown on the next page.

Figure 22-3. Example of Software Processing After Reset Release (2/2)

• Checking reset source



CHAPTER 23 LOW-VOLTAGE DETECTOR

23.1 Functions of Low-Voltage Detector

The low-voltage detector (LVI) is mounted onto all 78K0/Kx2-A microcontroller products.
The low-voltage detector 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 (TYP.): fixed), and generates an internal reset or internal interrupt signal.
- The supply voltage (V_{DD}) or input voltage from an external input pin (EXLVI) can be selected by software.
- Reset or interrupt function can be selected by software.
- Detection levels (16 levels) of supply voltage can be changed by software.
- Operable in STOP mode.

The reset and interrupt signals are generated as follows depending on selection by software.

Selection of Level Detection of Supply Voltage (V_{DD}) (LVISEL = 0)		Selection Level Detection of Input Voltage from External Input Pin (EXLVI) (LVISEL = 1)	
Selects reset (LVIMD = 1).	Selects interrupt (LVIMD = 0).	Selects reset (LVIMD = 1).	Selects interrupt (LVIMD = 0).
Generates an internal reset signal when $V_{DD} < V_{LVI}$ and releases the reset signal when $V_{DD} \geq V_{LVI}$.	Generates an internal interrupt signal when V_{DD} drops lower than V_{LVI} ($V_{DD} < V_{LVI}$) or when V_{DD} becomes V_{LVI} or higher ($V_{DD} \geq V_{LVI}$).	Generates an internal reset signal when EXLVI < V_{EXLVI} and releases the reset signal when EXLVI ≥ V_{EXLVI}.	Generates an internal interrupt signal when EXLVI drops lower than V_{EXLVI} (EXLVI < V_{EXLVI}) or when EXLVI becomes V_{EXLVI} or higher (EXLVI ≥ V_{EXLVI}).

Remark LVISEL: Bit 2 of low-voltage detection register (LVIM)
LVIMD: Bit 1 of LVIM

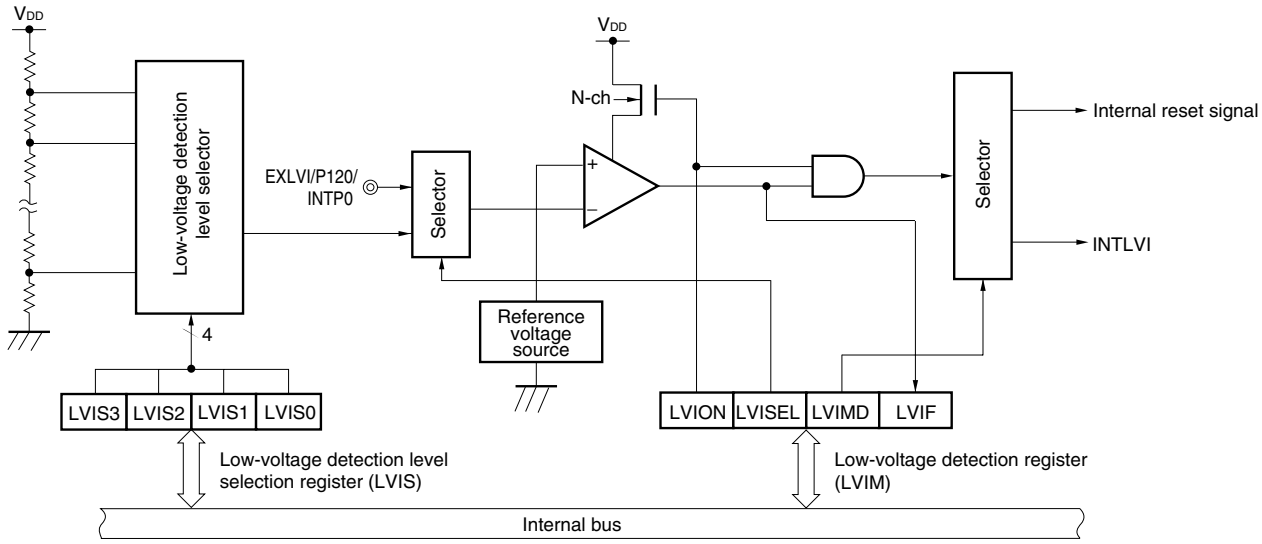
While the low-voltage detector is operating, whether the supply voltage or the input voltage from an external input pin is more than or less than the detection level can be checked by reading the low-voltage detection flag (LVIF: bit 0 of LVIM).

When the low-voltage detector is used to reset, bit 0 (LVIRF) of the reset control flag register (RESF) is set to 1 if reset occurs. For details of RESF, see **CHAPTER 21 RESET FUNCTION**.

23.2 Configuration of Low-Voltage Detector

The block diagram of the low-voltage detector is shown in Figure 23-1.

Figure 23-1. Block Diagram of Low-Voltage Detector



23.3 Registers Controlling Low-Voltage Detector

The low-voltage detector is controlled by the following registers.

- Low-voltage detection register (LVIM)
- Low-voltage detection level selection 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.

The generation of a reset signal other than an LVI reset clears this register to 00H.

Figure 23-2. Format of Low-Voltage Detection Register (LVIM)

Address: FFBEH 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 operation is disabled • LVISEL = 1: Input voltage from external input pin (EXLVI) ≥ detection voltage (V_{EXLVI}), or when 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. This bit is cleared to 00H upon a reset other than an LVI reset.

2. Bit 0 is read-only.

3. LVION, LVIMD, and LVISEL are cleared to 0 in the case of a reset other than an LVI reset. These are not cleared to 0 in the case of an LVI reset.

4. When LVION is set to 1, operation of the comparator in the LVI circuit is started. Use software to wait for an operation stabilization time (10 μs (MIN.)) from when LVION is set to 1 until operation is stabilized. After operation has stabilized, the external input of 200 μs (MIN.) (Minimum pulse width: 200 μs (MIN.)) is required from when a state below LVI detection voltage has been entered, until LVIF is set (1).

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 using LVI as an interrupt, if LVION is cleared (0) in a state below the LVI detection voltage, an INTLVI signal is generated and LVIIF becomes 1.

(2) Low-voltage detection level selection 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.

The generation of a reset signal other than an LVI reset clears this register to 00H.

Figure 23-3. Format of Low-Voltage Detection Level Selection Register (LVIS)

Address: FFBFH After reset: 00H[Note] 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} (4.24 V ±0.1 V)
0	0	0	1	V_{LVI1} (4.09 V ±0.1 V)
0	0	1	0	V_{LVI2} (3.93 V ±0.1 V)
0	0	1	1	V_{LVI3} (3.78 V ±0.1 V)
0	1	0	0	V_{LVI4} (3.62 V ±0.1 V)
0	1	0	1	V_{LVI5} (3.47 V ±0.1 V)
0	1	1	0	V_{LVI6} (3.32 V ±0.1 V)
0	1	1	1	V_{LVI7} (3.16 V ±0.1 V)
1	0	0	0	V_{LVI8} (3.01 V ±0.1 V)
1	0	0	1	V_{LVI9} (2.85 V ±0.1 V)
1	0	1	0	V_{LVI10} (2.70 V ±0.1 V)
1	0	1	1	V_{LVI11} (2.55 V ±0.1 V)
1	1	0	0	V_{LVI12} (2.39 V ±0.1 V)
1	1	0	1	V_{LVI13} (2.24 V ±0.1 V)
1	1	1	0	V_{LVI14} (2.08 V ±0.1 V)
1	1	1	1	V_{LVI15} (1.93 V ±0.1 V)

Note The value of LVIS is not reset but retained as is, upon a reset by LVI. It is cleared to 00H upon other resets.

Cautions 1. Be sure to clear bits 4 to 7 to "0".

2. Do not change the value of LVIS during LVI operation.

3. When an input voltage from the external input pin (EXLVI) is detected, the detection voltage (V_{EXLVI} = 1.21 V (TYP.)) 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 PM12 to FFH.

Figure 23-4. Format of Port Mode Register 12 (PM12)

Address: FF2CH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM12	1	1	1	PM124[Note]	PM123[Note]	PM122	PM121	PM120

PM12n	P12n pin I/O mode selection (n = 0 to 4)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Note For the 78K0/KB2-A, be sure to set bit 3 and 4 of PM12 to “1”.

23.4 Operation of Low-Voltage Detector

The low-voltage detector can be used in the following two modes.

(1) Used as reset (LVIMD = 1)

- If LVISEL = 0, compares the supply voltage (V_{DD}) and detection voltage (V_{LVI}), generates an internal reset signal when $V_{DD} < V_{LVI}$, and releases internal reset when $V_{DD} \geq V_{LVI}$.
- If LVISEL = 1, compares the input voltage from external input pin (EXLVI) and detection voltage (V_{EXLVI} = 1.21 V (TYP.)), generates an internal reset signal when EXLVI < V_{EXLVI}, and releases internal reset when EXLVI ≥ V_{EXLVI}.

(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 (TYP.)). When EXLVI drops lower than V_{EXLVI} (EXLVI < V_{EXLVI}) or when EXLVI becomes V_{EXLVI} or higher (EXLVI ≥ V_{EXLVI}), generates an interrupt signal (INTLVI).

While the low-voltage detector is operating, whether the supply voltage or the input voltage from an external input pin is more than or less than the detection level can be checked by reading the low-voltage detection flag (LVIF: bit 0 of LVIM).

Remark LVIMD: Bit 1 of low-voltage detection register (LVIM)
LVISEL: Bit 2 of LVIM

23.4.1 When used as reset

(1) When detecting level of supply voltage (V_{DD})

- 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 an operation stabilization time (10 μs (MIN.)).
  - <6> Wait until it is checked that (supply voltage (V_{DD}) ≥ detection voltage (V_{LVI})) by bit 0 (LVIF) of LVIM.
  - <7> Set bit 1 (LVIMD) of LVIM to 1 (generates reset when the level is detected).

  Figure 23-5 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

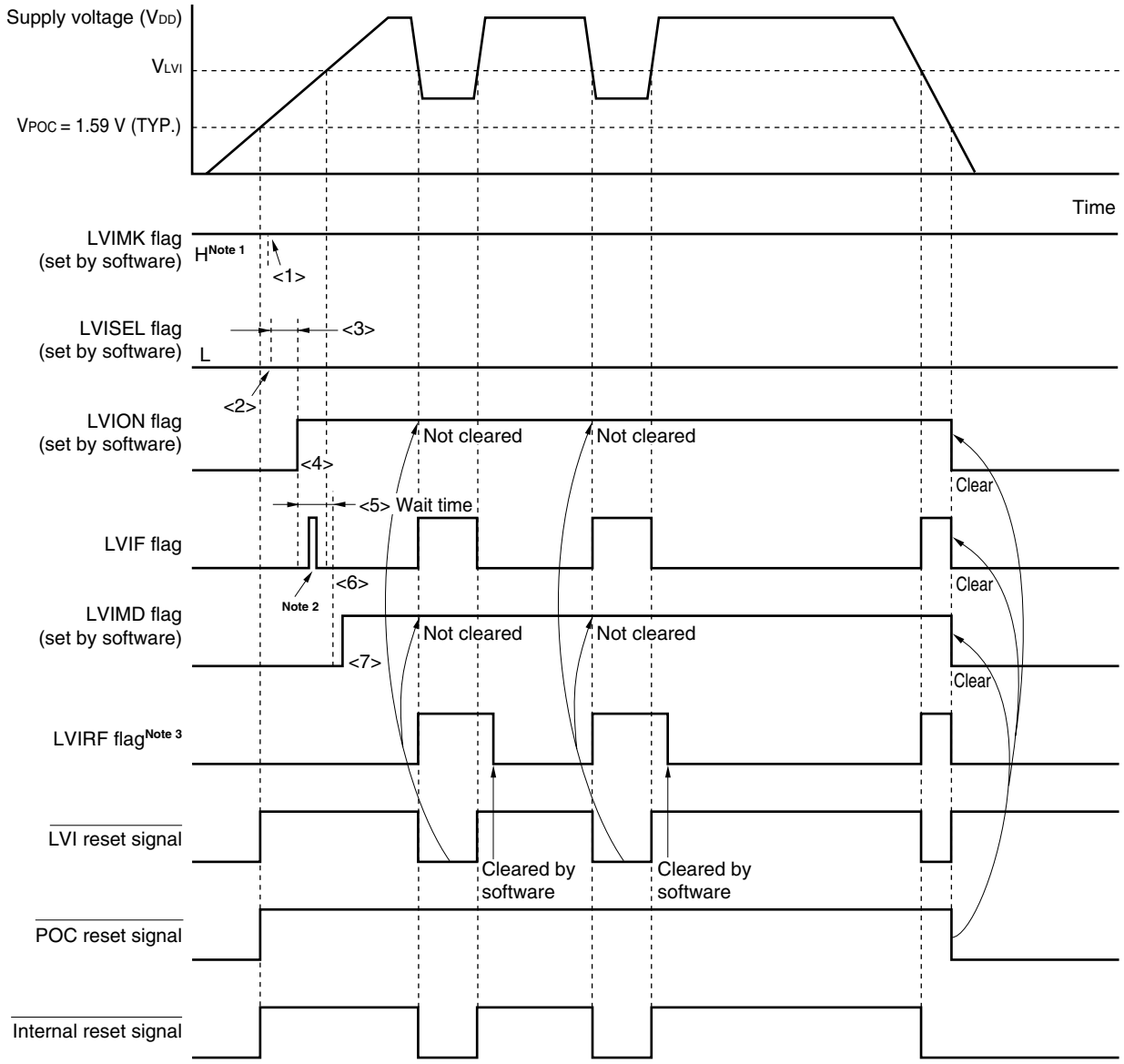
  Either of the following procedures must be executed.

  - 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 23-5. Timing of Low-Voltage Detector Internal Reset Signal Generation (Detects Level of Supply Voltage (V_{DD})) (1/2)

(1) In 1.59 V POC mode (option byte: POCMODE = 0)



Notes 1. The LVIMK flag is set to "1" by reset signal generation.

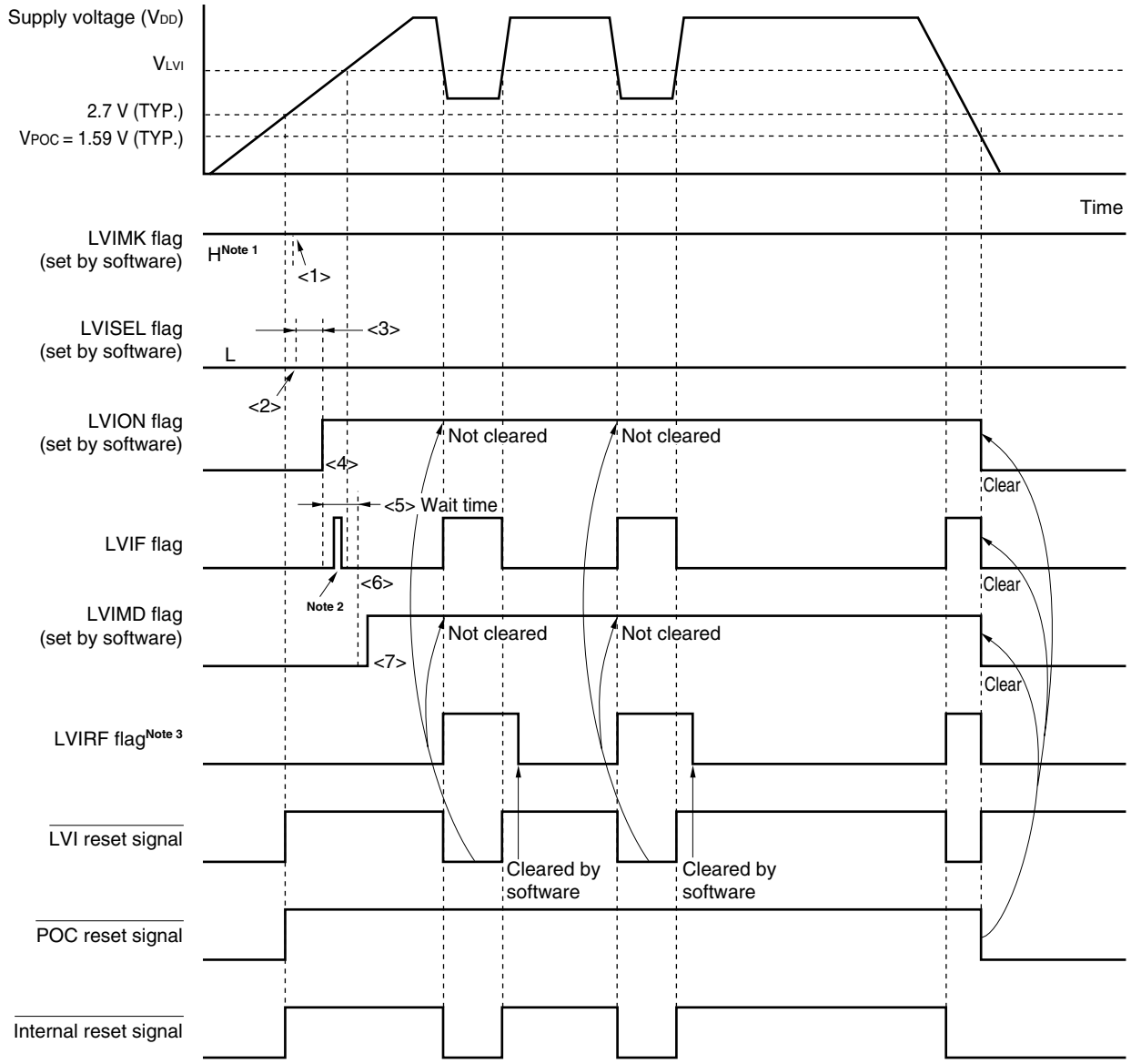
2. The LVIF flag may be set (1).

3. LVIRF is bit 0 of the reset control flag register (RESF). For details of RESF, see **CHAPTER 21 RESET FUNCTION**.

Remark <1> to <7> in Figure 23-5 above correspond to <1> to <7> in the description of "When starting operation" in **23.4.1 (1) When detecting level of supply voltage (V_{DD})**.

Figure 23-5. Timing of Low-Voltage Detector Internal Reset Signal Generation (Detects Level of Supply Voltage (V_{DD})) (2/2)

(2) In 2.7 V/1.59 V POC mode (option byte: POCMODE = 1)



Notes 1. The LVIMK flag is set to "1" by reset signal generation.

2. 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 21 RESET FUNCTION**.

Remark <1> to <7> in Figure 23-5 above correspond to <1> to <7> in the description of "When starting operation" in **23.4.1 (1) When detecting level of supply voltage (V_{DD})**.

(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 an operation stabilization time (10 μs (MIN.)).
  - <5> Wait until it is checked that (input voltage from external input pin (EXLVI) ≥ detection voltage (V_{EXLVI} = 1.21 V (TYP.))) by bit 0 (LVIF) of LVIM.
  - <6> Set bit 1 (LVIMD) of LVIM to 1 (generates reset signal when the level is detected).

  Figure 23-6 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.

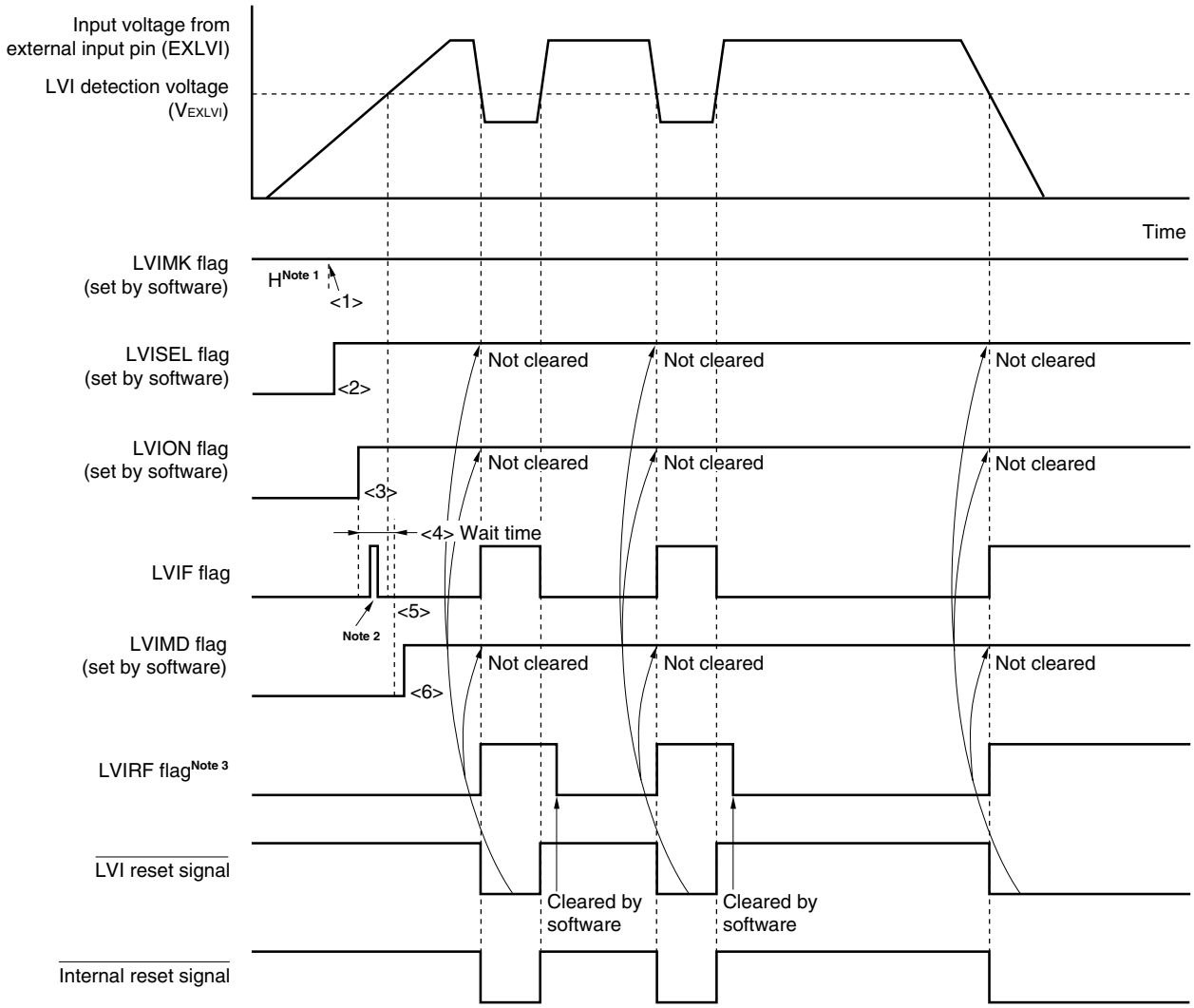
  Cautions 1. <1> must always be executed. When LVIMK = 0, an interrupt may occur immediately after the processing in <3>.
  2. If input voltage from external input pin (EXLVI) ≥ detection voltage (V_{EXLVI} = 1.21 V (TYP.)) when LVIMD is set to 1, an internal reset signal is not generated.
  3. Input 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 23-6. Timing of Low-Voltage Detector Internal Reset Signal Generation (Detects Level of Input Voltage from External Input Pin (EXLVI))



Notes 1. The LVIMK flag is set to “1” by reset signal generation.

2. 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 21 RESET FUNCTION**.

Remark <1> to <6> in Figure 23-6 above correspond to <1> to <6> in the description of **“**When starting operation” in **23.4.1 (2) When detecting level of input voltage from external input pin (EXLVI)**.

23.4.2 When used as interrupt

(1) When detecting level of supply voltage (V_{DD})

- 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> Clear bit 1 (LVIMD) of LVIM to 0 (generates interrupt signal when the level is detected) (default value).
  - <5> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).
  - <6> Use software to wait for an operation stabilization time (10 μs (MIN.)).
  - <7> 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.
  - <8> Clear the interrupt request flag of LVI (LVIIF) to 0.
  - <9> Release the interrupt mask flag of LVI (LVIMK).
  - <10> Execute the EI instruction (when vector interrupts are used).

  Figure 23-7 shows the timing of the interrupt signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <9> above.

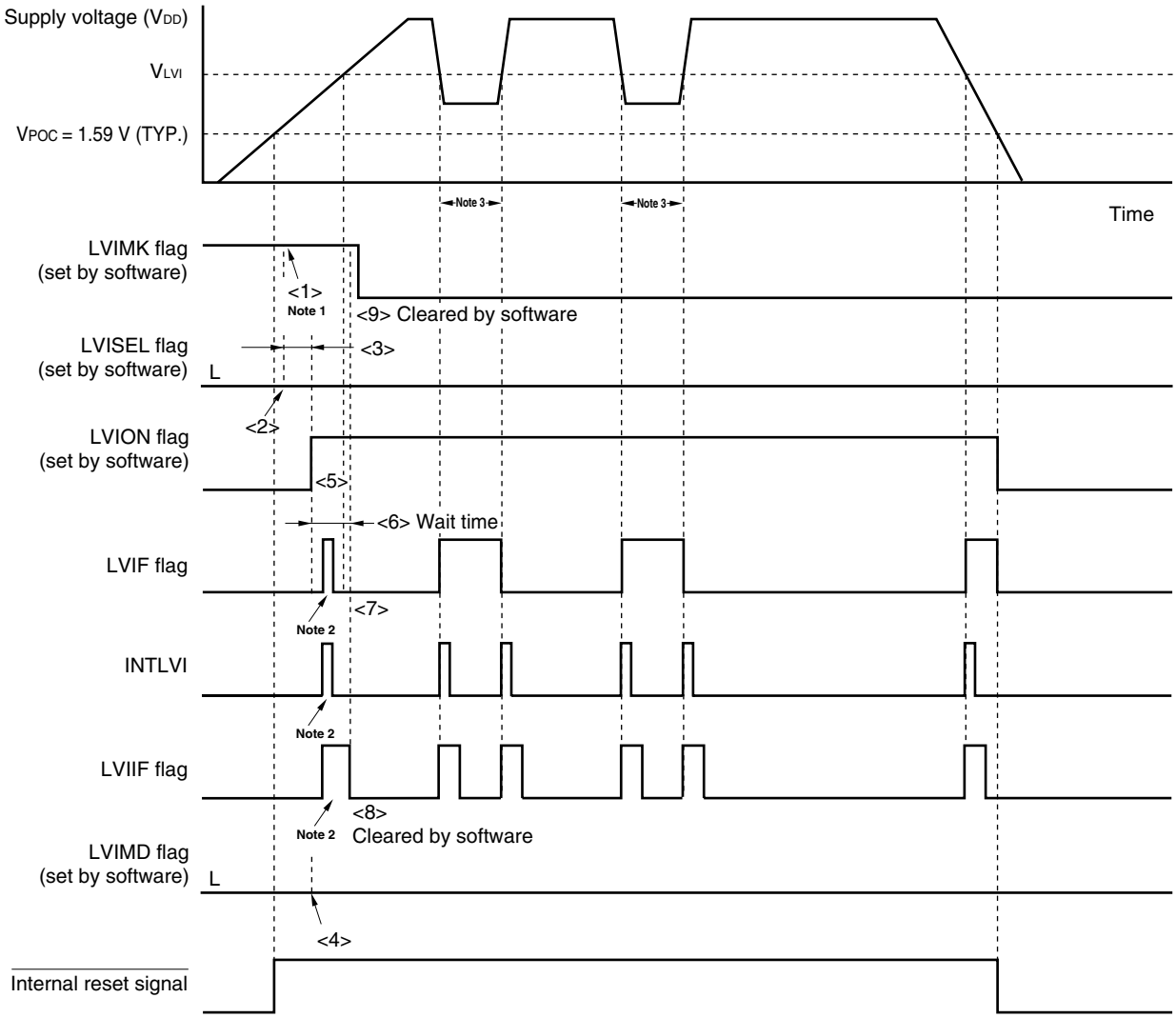
- When stopping operation
  Either of the following procedures must be executed.

  - When using 8-bit memory manipulation instruction:
    Write 00H to LVIM.

  - When using 1-bit memory manipulation instruction:
    Clear LVION to 0.

Figure 23-7. Timing of Low-Voltage Detector Interrupt Signal Generation (Detects Level of Supply Voltage (V_{DD})) (1/2)

(1) In 1.59 V POC mode (option byte: POCMODE = 0)



Notes 1. The LVIMK flag is set to "1" by reset signal generation.

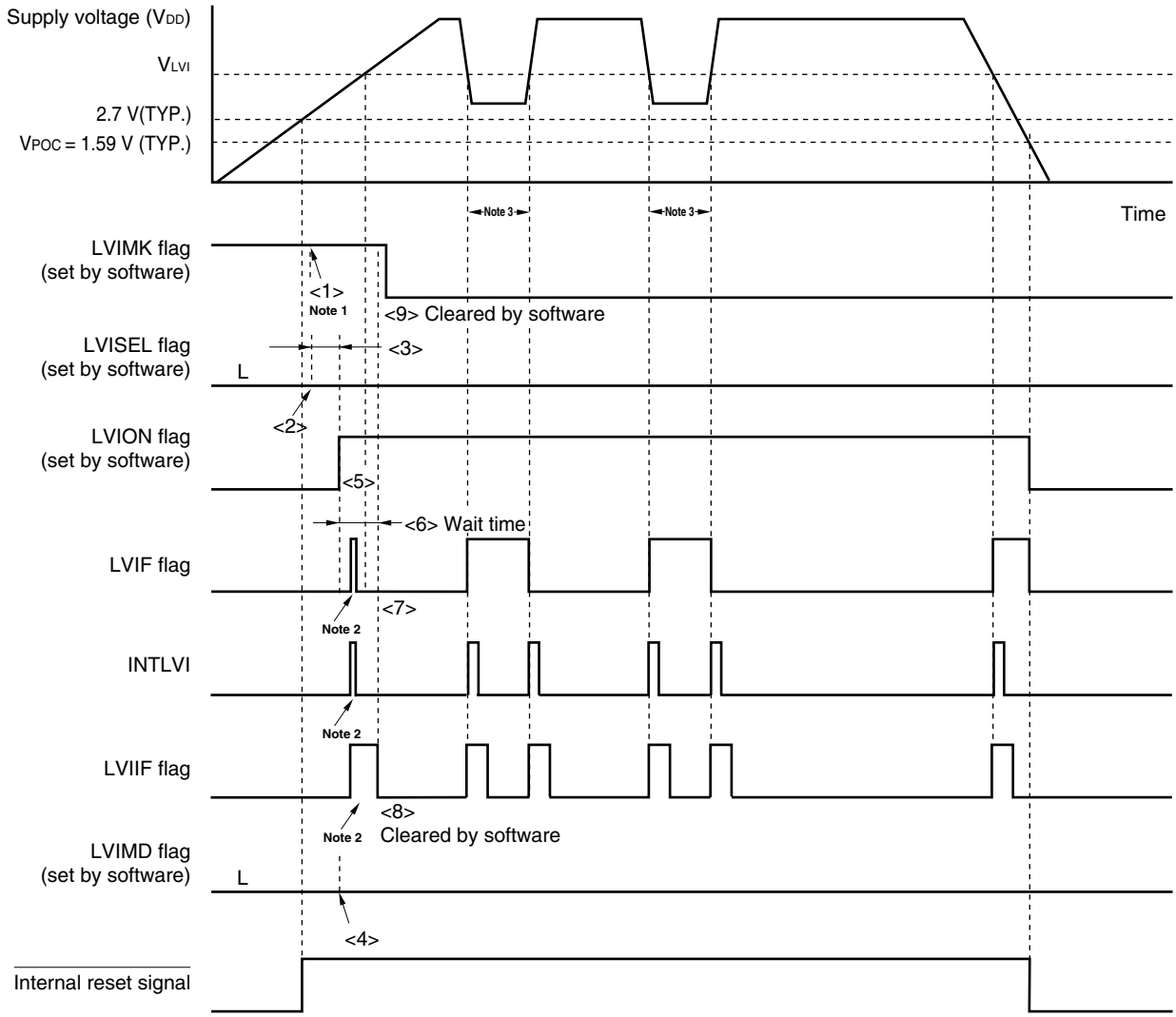
2. The interrupt request signal (INTLVI) is generated and the LVIF and LVIIF flags may be set (1).

3. If LVION is cleared (0) in a state below the LVI detection voltage, an INTLVI signal is generated and LVIIF becomes 1.

Remark <1> to <9> in Figure 23-7 above correspond to <1> to <9> in the description of "When starting operation" in **23.4.2 (1) When detecting level of supply voltage (V_{DD})**.

Figure 23-7. Timing of Low-Voltage Detector Interrupt Signal Generation (Detects Level of Supply Voltage (V_{DD})) (2/2)

(2) In 2.7 V/1.59 V POC mode (option byte: POCMODE = 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 LVION is cleared (0) in a state below the LVI detection voltage, an INTLVI signal is generated and LVIIF becomes 1.

Remark <1> to <9> in Figure 23-7 above correspond to <1> to <9> in the description of "When starting operation" in **23.4.2 (1) When detecting level of supply voltage (V_{DD})**.

(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> Clear bit 1 (LVIMD) of LVIM to 0 (generates interrupt signal when the level is detected) (default value).
  - <4> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).
  - <5> Use software to wait for an operation stabilization time (10 μs (MIN.)).
  - <6> Confirm that "input voltage from external input pin (EXLVI) $\geq$ detection voltage (V_{EXLVI} = 1.21 V (TYP.)" when detecting the falling edge of EXLVI, or "input voltage from external input pin (EXLVI) < detection voltage (V_{EXLVI} = 1.21 V (TYP.)" when detecting the rising edge of EXLVI, at bit 0 (LVIF) of LVIM.
  - <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 23-8 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.

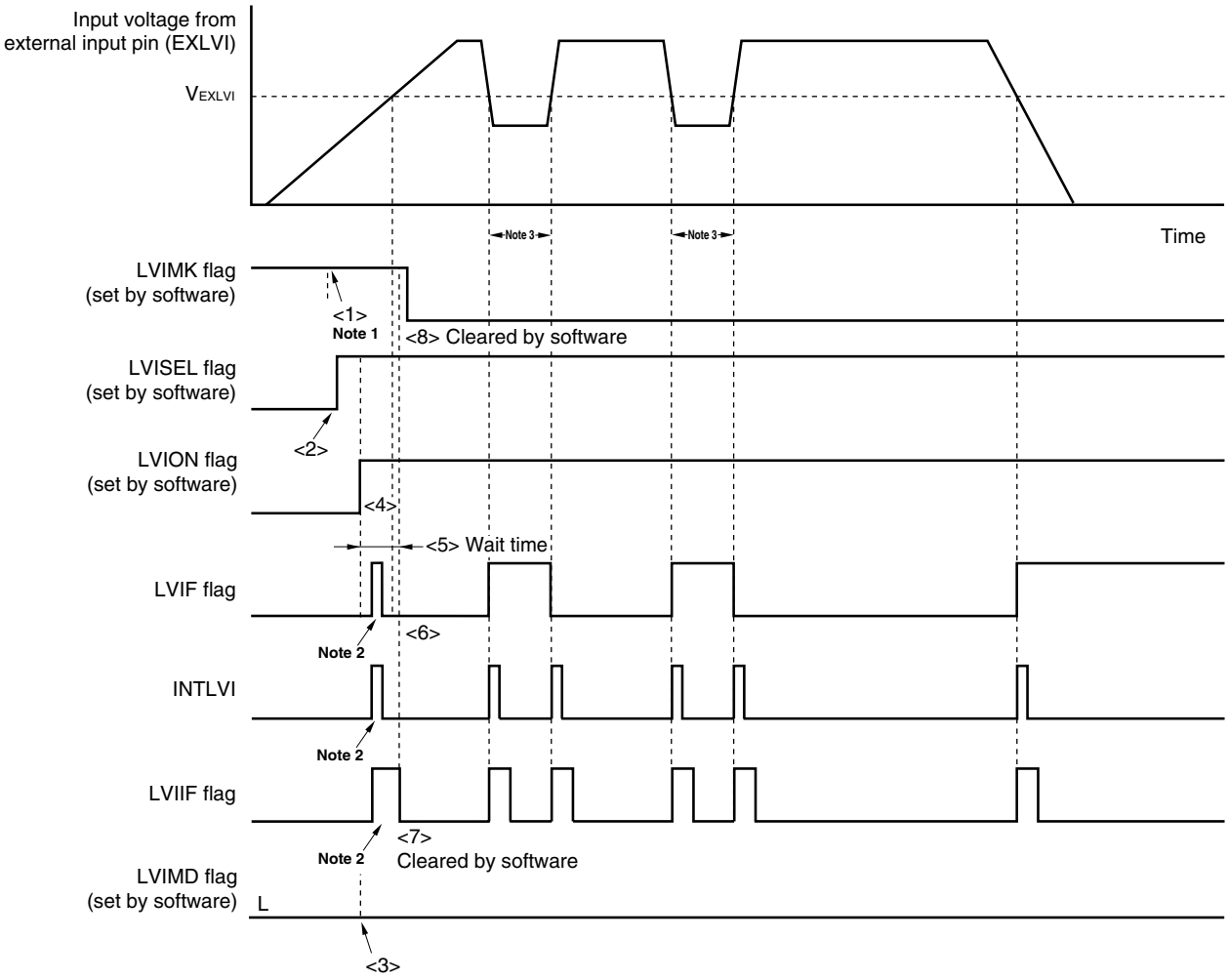
  Caution Input voltage from external input pin (EXLVI) must be EXLVI < V_{DD}.

- 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 23-8. Timing of Low-Voltage Detector Interrupt Signal Generation (Detects Level of Input Voltage from External Input Pin (EXLVI))



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 LVION is cleared (0) in a state below the LVI detection voltage, an INTLVI signal is generated and LVIIF becomes 1.

Remark <1> to <8> in Figure 23-8 above correspond to <1> to <8> in the description of "When starting operation" in **235.4.2 (2) When detecting level of input voltage from external input pin (EXLVI)**.

23.5 Cautions for Low-Voltage Detector

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.

(1) When used as reset

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 action (1) below.

(2) When used as interrupt

Interrupt requests may be frequently generated. Take (b) of action (2) below.

<Action>

(1) When used as reset

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 23-9**).

(2) When used as interrupt

(a) 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 0 (LVIIF) of interrupt request flag register 0L (IF0L) to 0.

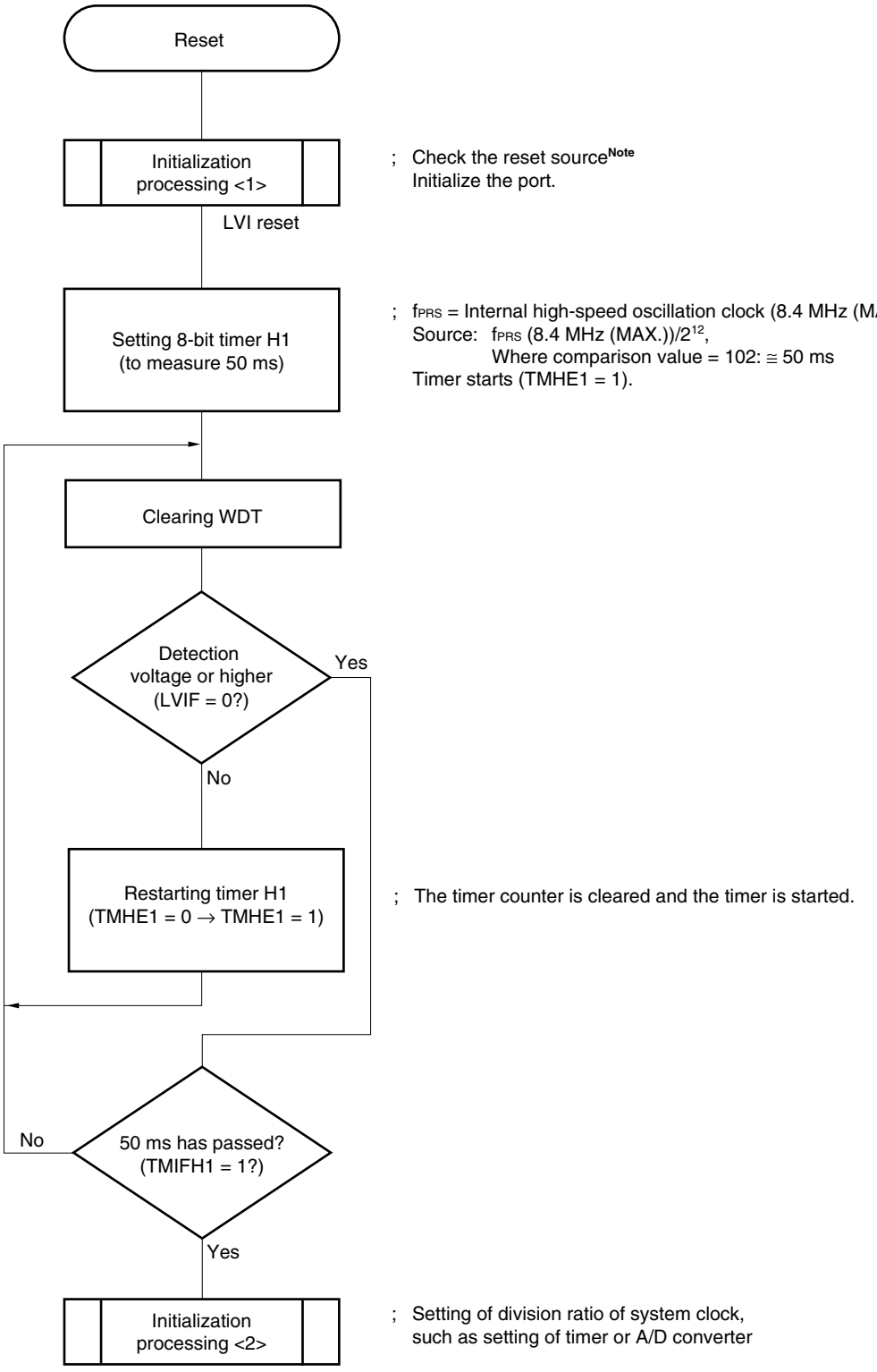
(b) In a system where the supply voltage fluctuation period is long in the vicinity of the LVI detection voltage, wait for the supply voltage fluctuation period, confirm that "supply voltage (V_{DD}) ≥ detection voltage (V_{LVI})" when detecting the falling edge of V_{DD}, or "supply voltage (V_{DD}) < detection voltage (V_{LVI})" when detecting the rising edge of V_{DD}, using the LVIF flag, and clear the LVIIF flag to 0.

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 23-9. Example of Software Processing After Reset Release (1/2)

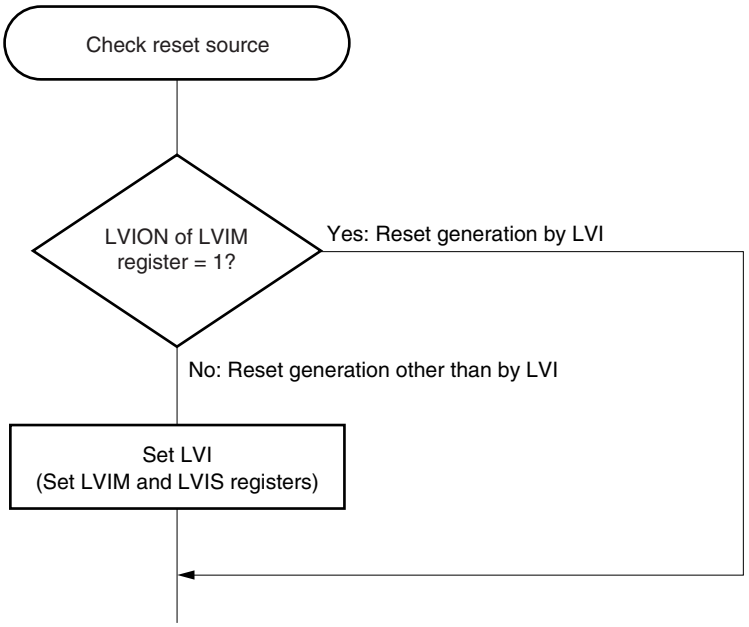
• If supply voltage fluctuation is 50 ms or less in vicinity of LVI detection voltage



Note A flowchart is shown on the next page.

Figure 23-9. Example of Software Processing After Reset Release (2/2)

• Checking reset source



CHAPTER 24 OPTION BYTE

24.1 Functions of Option Bytes

The flash memory at 0080H to 0084H of the 78K0/Kx2-A microcontrollers is an option byte area. When power is turned on or when the device is restarted from the reset status, the device automatically references the option bytes and sets specified functions. When using the product, be sure to set the following functions by using the option bytes.

When the boot swap operation is used during self-programming, 0080H to 0084H are switched to 1080H to 1084H. Therefore, set values that are the same as those of 0080H to 0084H to 1080H to 1084H in advance.

Caution Be sure to set 00H to 0082H and 0083H (0082H/1082H and 0083H/1083H when the boot swap function is used).

(1) 0080H/1080H

- ○ Internal low-speed oscillator operation
  - Can be stopped by software
  - Cannot be stopped
- ○ Watchdog timer overflow time setting
- ○ Watchdog timer counter operation
  - Enabled counter operation
  - Disabled counter operation
- ○ Watchdog timer window open period setting

Caution Set a value that is the same as that of 0080H to 1080H because 0080H and 1080H are switched during the boot swap operation.

(2) 0081H/1081H

- ○ Selecting POC mode
  - During 2.7 V/1.59 V POC mode operation (POCMODE = 1)
    The device is in the reset state upon power application and until the supply voltage reaches 2.7 V (TYP.). It is released from the reset state when the voltage exceeds 2.7 V (TYP.). After that, POC is not detected at 2.7 V but is detected at 1.59 V (TYP.).
    If the supply voltage rises to 1.8 V after power application at a rate slower than 0.5 V/ms (MIN.), use of the 2.7 V/1.59 V POC mode is recommended.
  - During 1.59 V POC mode operation (POCMODE = 0)
    The device is in the reset state upon power application and until the supply voltage reaches 1.59 V (TYP.). It is released from the reset state when the voltage exceeds 1.59 V (TYP.). After that, POC is detected at 1.59 V (TYP.), in the same manner as on power application.

Caution POCMODE can only be written by using a dedicated flash memory programmer. It cannot be set during self-programming or boot swap operation during self-programming. However, because the value of 1081H is copied to 0081H during the boot swap operation, it is recommended to set a value that is the same as that of 0081H to 1081H when the boot swap function is used.

(3) 0084H/1084H

○ On-chip debug operation control

- Disabling on-chip debug operation
- Enabling on-chip debug operation and erasing data of the flash memory in case authentication of the on-chip debug security ID fails
- Enabling on-chip debug operation and not erasing data of the flash memory even in case authentication of the on-chip debug security ID fails

Caution Set 00H to 1084H because 0084H and 1084H are switched during the boot operation.

24.2 Format of Option Byte

The format of the option byte is shown below.

Figure 24-1. Format of Option Byte (1/2)

Address: 0080H/1080H[Note]

7	6	5	4	3	2	1	0
0	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	LSROSC

WINDOW1	WINDOW0	Watchdog timer window open period
0	0	25%
0	1	50%
1	0	75%
1	1	100%

WDTON	Operation control of watchdog timer counter/illegal access detection
0	Counter operation disabled (counting stopped after reset), illegal access detection operation disabled
1	Counter operation enabled (counting started after reset), illegal access detection operation enabled

WDCS2	WDCS1	WDCS0	Watchdog timer overflow time
0	0	0	$2^{10}/f_{RL}$ (3.88 ms)
0	0	1	$2^{11}/f_{RL}$ (7.76 ms)
0	1	0	$2^{12}/f_{RL}$ (15.52 ms)
0	1	1	$2^{13}/f_{RL}$ (31.03 ms)
1	0	0	$2^{14}/f_{RL}$ (62.06 ms)
1	0	1	$2^{15}/f_{RL}$ (124.12 ms)
1	1	0	$2^{16}/f_{RL}$ (248.24 ms)
1	1	1	$2^{17}/f_{RL}$ (496.48 ms)

LSROSC	Internal low-speed oscillator operation
0	Can be stopped by software (stopped when 1 is written to bit 1 (LSRSTOP) of RCM register)
1	Cannot be stopped (not stopped even if 1 is written to LSRSTOP bit)

Note Also set a value that is the same as that of 0080H to 1080H because 0080H and 1080H are switched during the boot swap operation.

Cautions 1. The combination of WDCS2 = WDCS1 = WDCS0 = 0 and WINDOW1 = WINDOW0 = 0 is prohibited.

2. Setting WINDOW1 = WINDOW0 = 0 is prohibited when using the watchdog timer at 1.8 V ≤ V_{DD} < 2.7 V.

3. The watchdog timer continues its operation during self-programming and EEPROM emulation of the flash memory. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.

4. If LSROSC = 0 (oscillation can be stopped by software), the count clock is not supplied to the watchdog timer in the HALT and STOP modes, regardless of the setting of bit 0 (LSRSTOP) of the internal oscillation mode register (RCM).
When 8-bit timer H1 operates with the internal low-speed oscillation clock, the count clock is supplied to 8-bit timer H1 even in the HALT/STOP mode.

5. Be sure to clear bit 7 to 0.

Remarks 1. f_{RL}: Internal low-speed oscillation clock frequency

2. ( ): f_{RL} = 264 kHz (MAX.)

Figure 24-1. Format of Option Byte (2/2)

Address: 0081H/1081H[Notes 1, 2]

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

POCMODE	POC mode selection
0	1.59 V POC mode (default)
1	2.7 V/1.59 V POC mode

Notes 1. POCMODE can only be written by using a dedicated flash memory programmer. It cannot be set during self-programming or boot swap operation during self-programming. However, because the value of 1081H is copied to 0081H during the boot swap operation, it is recommended to set a value that is the same as that of 0081H to 1081H when the boot swap function is used.

2. To change the setting for the POC mode, set the value to 0081H again after batch erasure (chip erasure) of the flash memory. The setting cannot be changed after the memory of the specified block is erased.

Caution Be sure to clear bits 7 to 1 to "0".

Address: 0082H/1082H, 0083H/1083H[Note]

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

Note Be sure to set 00H to 0082H and 0083H, as these addresses are reserved areas. Also set 00H to 1082H and 1083H because 0082H and 0083H are switched with 1082H and 1083H when the boot swap operation is used.

Address: 0084H/1084H[Note]

7	6	5	4	3	2	1	0
0	0	0	0	0	0	OCDEN1	OCDEN0

OCDEN1	OCDEN0	On-chip debug operation control
0	0	Operation disabled
0	1	Setting prohibited
1	0	Operation enabled. Does not erase data of the flash memory in case authentication of the on-chip debug security ID fails.
1	1	Operation enabled. Erases data of the flash memory in case authentication of the on-chip debug security ID fails.

Note Also set 00H to 1084H because 0084H and 1084H are switched during the boot swap operation.

Remark For the on-chip debug security ID, see **CHAPTER 26 ON-CHIP DEBUG FUNCTION**.

Here is an example of description of the software for setting the option bytes.

```
OPT      CSEG  AT 0080H
OPTION:  DB    30H      ; Enables watchdog timer operation (illegal access detection operation),
                        ; Window open period of watchdog timer: 50%,
                        ; Overflow time of watchdog timer: 2^10/fRL,
                        ; Internal low-speed oscillator can be stopped by software.
         DB    00H      ; 1.59 V POC mode
         DB    00H      ; Reserved area
         DB    00H      ; Reserved area
         DB    00H      ; On-chip debug operation disabled
```

Remark Referencing of the option byte is performed during reset processing. For the reset processing timing, see **CHAPTER 21 RESET FUNCTION**.

CHAPTER 25 FLASH MEMORY

The 78K0/Kx2-A microcontrollers incorporates the flash memory to which a program can be written, erased, and overwritten while mounted on the board.

25.1 Internal Memory Size Switching Register

Select the internal memory capacity using the internal memory size switching register (IMS).

IMS is set by an 8-bit memory manipulation instruction.

Reset signal generation sets IMS to CFH.

Caution Be sure to set each product to the values shown in Table 25-1 after a reset release.

Figure 25-1. Format of Internal Memory Size Switching Register (IMS)

Address: FFF0H After reset: CFH R/W

Symbol	7	6	5	4	3	2	1	0
IMS	RAM2	RAM1	RAM0	0	ROM3	ROM2	ROM1	ROM0

RAM2	RAM1	RAM0	Internal high-speed RAM capacity selection
1	1	0	1024 bytes
Other than above			Setting prohibited

ROM3	ROM2	ROM1	ROM0	Internal ROM capacity selection
0	1	0	0	16 KB
1	0	0	0	32 KB
Other than above				Setting prohibited

Table 25-1. Internal Memory Size Switching Register Settings

Part number		IMS Setting
78K0/KB2-A	78K0/KC2-A	
μPD78F0590	μPD78F0592	C4H
μPD78F0591	μPD78F0593	C8H

25.2 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 78K0/Kx2-A microcontrollers have 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

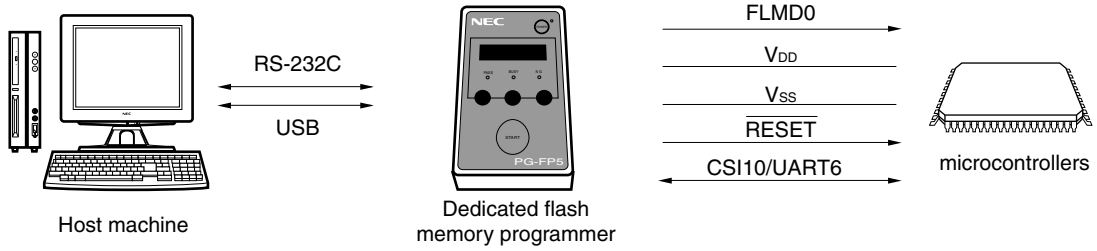
Data can be written to the flash memory with a dedicated program adapter (FA series) before the 78K0/Kx2-A microcontrollers are mounted on the target system.

Remark The FA series is a product of Naito Densei Machida Mfg. Co., Ltd.

25.3 Programming Environment

The environment required for writing a program to the flash memory of the 78K0/Kx2-A microcontrollers are illustrated below.

Figure 25-2. Environment for Writing Program to Flash Memory



A host machine that controls the dedicated flash memory programmer is necessary.

To interface between the dedicated flash memory programmer and the 78K0/Kx2-A microcontrollers, CSI10 or UART6 is used for manipulation such as writing and erasing. To write the flash memory off-board, a dedicated program adapter (FA series) is necessary.

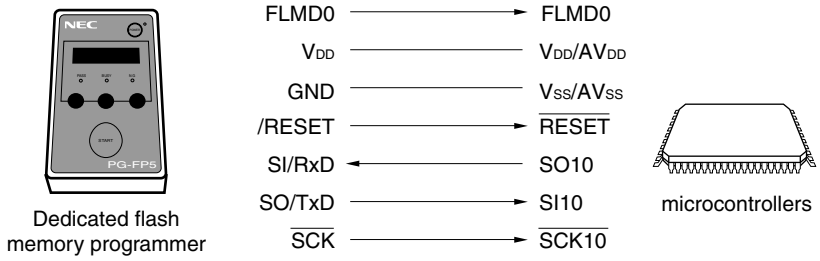
25.4 Communication Mode

Communication between the dedicated flash memory programmer and the 78K0/Kx2-A microcontrollers is established by serial communication via CSI10 or UART6 of the 78K0/Kx2-A microcontrollers.

(1) CSI10

Transfer rate: 2.4 kHz to 2.5 MHz

Figure 25-3. Communication with Dedicated Flash memory programmer (CSI10)

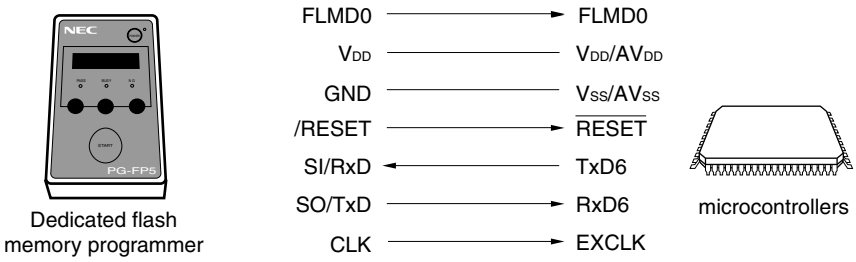


Caution: Only the P60/SCLA0/SCK10 and P61/SDAA0/SI10 pins (when used as the CSI10 pins SCK10 and SI10) can be used for communicating with the dedicated flash memory programmer. The P31/INTP5/OCD1A(/SCK10) and P32/INTP4/OCD1B(/SI10) pins cannot be used for communicating with the dedicated flash memory programmer.

(2) UART6

Transfer rate: 115200 bps

Figure 25-4. Communication with Dedicated Flash memory programmer (UART6)



The dedicated flash memory programmer generates the following signals for the 78K0/Kx2-A microcontrollers. For details, refer to the user's manual for the PG-FP5, FL-PR5, etc.

Table 25-2. Pin Connection

Dedicated Flash memory programmer			microcontrollers	Connection	
Signal Name	I/O	Pin Function	Pin Name	CSI10	UART6
FLMD0	Output	Mode signal	FLMD0	◎	◎
V_{DD}	I/O	V_{DD} voltage generation/power monitoring	V_{DD}, AV_{DD}	◎	◎
GND	–	Ground	V_{SS}, AV_{SS}	◎	◎
CLK	Output	Clock output to microcontrollers	EXCLK/X2/P122	×Note 1	○Note 2
/RESET	Output	Reset signal	$\overline{\text{RESET}}$	◎	◎
SI/RxD	Input	Receive signal	SO10/TxD6	◎	◎
SO/TxD	Output	Transmit signal	SI10/RxD6	◎	◎
SCK	Output	Transfer clock	$\overline{\text{SCK10}}$	◎	×

Notes 1. Only the internal high-speed oscillation clock (f_{RH}) can be used when CSI10 is used.

2. Only the X1 clock (f_X) or external main system clock (f_{EXCLK}) can be used when UART6 is used.

Remark ◎: Be sure to connect the pin.

○: The pin does not have to be connected if the signal is generated on the target board.

×: The pin does not have to be connected.

25.5 Connection of Pins on Board

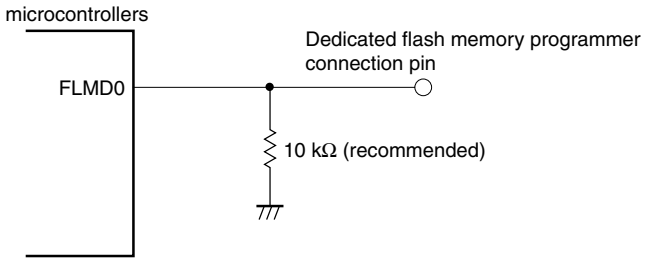
To write the flash memory on-board, connectors that connect the dedicated flash memory programmer must be provided on the target system. First provide a function that selects the normal operation mode or flash memory programming mode on the board.

When the flash memory programming mode is set, all the pins not used for programming the flash memory are in the same status as immediately after reset. Therefore, if the external device does not recognize the state immediately after reset, the pins must be handled as described below.

25.5.1 FLMD0 pin

In the normal operation mode, 0 V is input to the FLMD0 pin. In the flash memory programming mode, the V_{DD} write voltage is supplied to the FLMD0 pin. An FLMD0 pin connection example is shown below.

Figure 25-5. FLMD0 Pin Connection Example



25.5.2 Serial interface pins

The pins used by each serial interface are listed below.

Table 25-3. Pins Used by Each Serial Interface

Serial Interface	Pins Used
CSI10	SO10, SI10, $\overline{\text{SCK10}}$
UART6	TxD6, RxD6

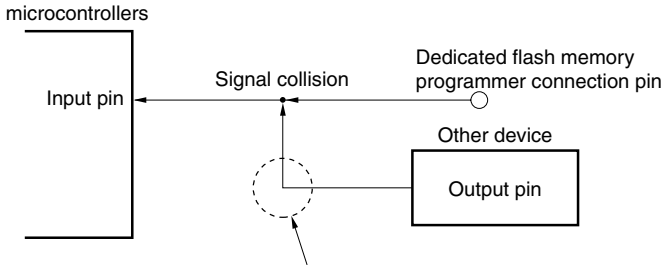
Caution: Only the P60/SCLA0/$\overline{\text{SCK10}}$ and P61/SDAA0/SI10 pins (when used as the CSI10 pins $\overline{\text{SCK10}}$ and SI10) can be used for communicating with the dedicated flash memory programmer. The P31/INTP5/OCD1A(/$\overline{\text{SCK10}}$) and P32/INTP4/OCD1B(/SI10) pins cannot be used for communicating with the dedicated flash memory programmer.

To connect the dedicated flash memory programmer to the pins of a serial interface that is connected to another device on the board, care must be exercised so that signals do not collide or that the other device does not malfunction.

(1) Signal collision

If the dedicated flash memory programmer (output) is connected to a pin (input) of a serial interface connected to another device (output), signal collision takes place. To avoid this collision, either isolate the connection with the other device, or make the other device go into an output high-impedance state.

Figure 25-6. Signal Collision (Input Pin of Serial Interface)

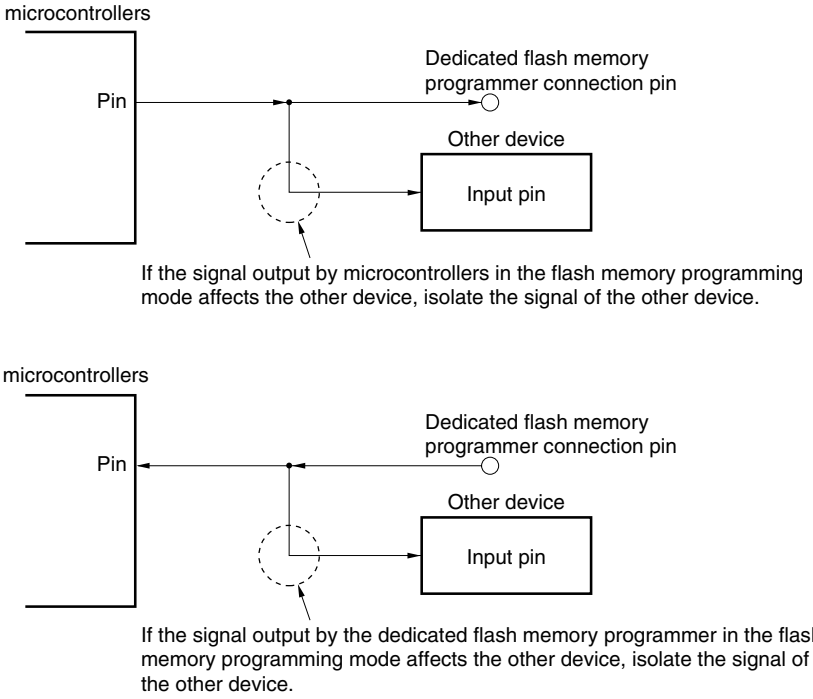


In the flash memory programming mode, the signal output by the device collides with the signal sent from the dedicated flash programmer. Therefore, isolate the signal of the other device.

(2) Malfunction of other device

If the dedicated flash memory programmer (output or input) is connected to a pin (input or output) of a serial interface connected to another device (input), a signal may be output to the other device, causing the device to malfunction. To avoid this malfunction, isolate the connection with the other device.

Figure 25-7. Malfunction of Other Device

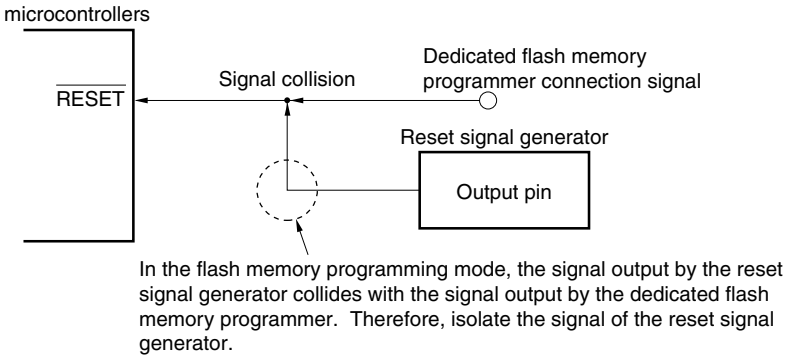


25.5.3 $\overline{\text{RESET}}$ pin

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, signal collision takes place. To prevent this collision, isolate the connection with the reset signal generator.

If the reset signal is input from the user system while the flash memory programming mode is set, the flash memory will not be correctly programmed. Do not input any signal other than the reset signal of the dedicated flash memory programmer.

Figure 25-8. Signal Collision ($\overline{\text{RESET}}$ Pin)



25.5.4 Port pins

When the flash memory programming mode is set, all the pins not used for flash memory programming enter the same status as that immediately after reset. If external devices connected to the ports do not recognize the port status immediately after reset, the port pin must be connected to V_{DD} or V_{SS} via a resistor.

25.5.5 REGC pin

Connect the REGC pin to GND via a capacitor (0.47 to 1 μF: recommended) in the same manner as during normal operation.

25.5.6 Other signal pins

Connect X1 and X2 in the same status as in the normal operation mode when using the on-board clock.

To input the operating clock from the dedicated flash memory programmer, however, connect CLK of the programmer to EXCLK/X2/P122.

Cautions 1. Only the internal high-speed oscillation clock (f_{RH}) can be used when CSI10 is used.

2. Only the X1 clock (f_X) or external main system clock (f_{EXCLK}) can be used when UART6 is used.

3. Connect P31/INTP5/OCD1A and P121/X1/OCD0A as follows when writing the flash memory with a flash memory programmer.

- **P31/INTP5/OCD1A: Connect to V_{SS} via a resistor.**
- **P121/X1/OCD0A: Open or connect to V_{SS} via a resistor.**

25.5.7 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, 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, even when using the on-board supply voltage.

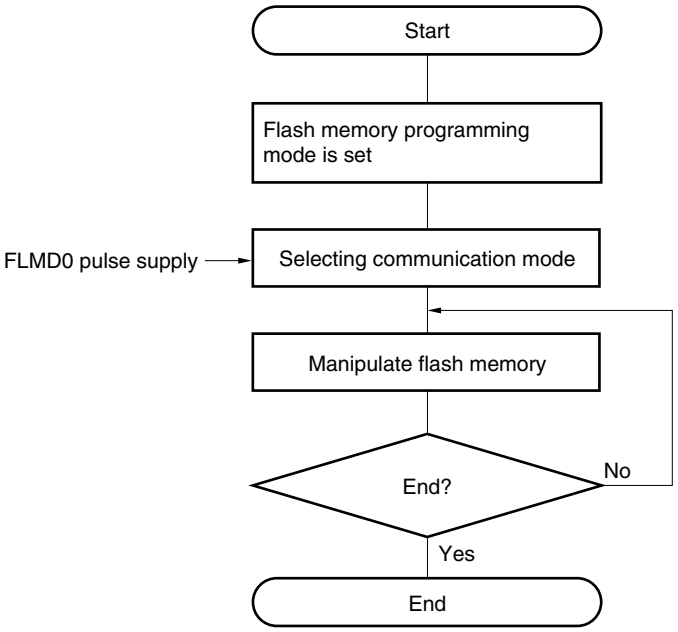
Supply the same other power supplies (AV_{DD}, AV_{REFM}, and AV_{SS}) as those in the normal operation mode.

25.6 Programming Method

25.6.1 Controlling flash memory

The following figure illustrates the procedure to manipulate the flash memory.

Figure 25-9. Flash Memory Manipulation Procedure



25.6.2 Flash memory programming mode

To rewrite the contents of the flash memory by using the dedicated flash memory programmer, set the 78K0/Kx2-A microcontrollers in the flash memory programming mode. To set the mode, set the FLMD0 pin to V_{DD} and clear the reset signal.

Change the mode by using a jumper when writing the flash memory on-board.

Figure 25-10. Flash Memory Programming Mode

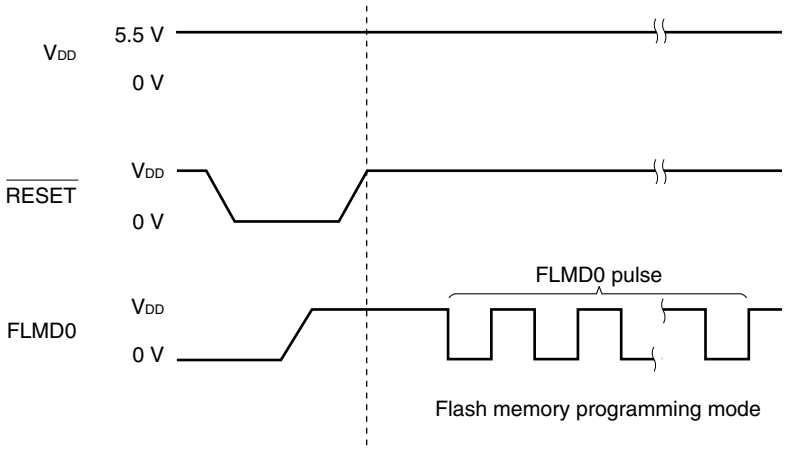


Table 25-4. Relationship Between FLMD0 Pin and Operation Mode After Reset Release

FLMD0	Operation Mode
0	Normal operation mode
V_{DD}	Flash memory programming mode

25.6.3 Selecting communication mode

In the 78K0/Kx2-A microcontrollers, a communication mode is selected by inputting pulses to the FLMD0 pin after the dedicated flash memory programming mode is entered. These FLMD0 pulses are generated by the flash memory programmer.

The following table shows the relationship between the number of pulses and communication modes.

Table 25-5. Communication Modes

Communication Mode	Standard Setting[Note 1]				Pins Used	Peripheral Clock	Number of FLMD0 Pulses
	Port	Speed	Frequency	Multiply Rate			
UART (UART6)	UART-Ext-Osc	115,200 bps[Note 3]	2 to 20 MHz[Note 2]	1.0	TxD6, RxD6	f_X	0
	UART-Ext-FP5CK					f_{EXCLK}	3
3-wire serial I/O (CSI10)	CSI-Internal-OSC	2.4 kHz to 2.5 MHz	–		$\overline{SO10}$, SI10, $\overline{SCK10}$	f_{RH}	8

Notes 1. Selection items for Standard settings on GUI of the flash memory programmer.

2. The possible setting range differs depending on the voltage. For details, refer to the chapter of electrical specifications.

3. 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.

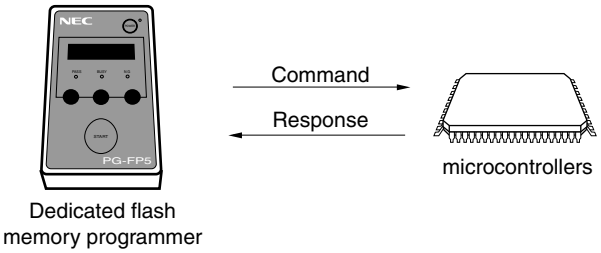
Caution When UART6 is selected, the receive clock is calculated based on the reset command sent from the dedicated flash memory programmer after the FLMD0 pulse has been received.

Remark f_X: X1 clock
f_{EXCLK}: External main system clock
f_{RH}: Internal high-speed oscillation clock

25.6.4 Communication commands

The 78K0/Kx2-A microcontrollers communicate with the dedicated flash memory programmer by using commands. The signals sent from the flash memory programmer to the 78K0/Kx2-A microcontrollers are called commands, and the signals sent from the 78K0/Kx2-A microcontrollers to the dedicated flash memory programmer are called response.

Figure 25-11. Communication Commands



The flash memory control commands of the 78K0/Kx2-A microcontrollers are listed in the table below. All these commands are issued from the programmer and the 78K0/Kx2-A microcontrollers perform processing corresponding to the respective commands.

Table 25-6. 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	Status	Gets the current operating status (status data).
	Silicon Signature	Gets information (such as the part number and flash memory configuration).
	Version Get	Gets version and 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.
	Oscillating Frequency Set	Specifies an oscillation frequency.

The 78K0/Kx2-A microcontrollers return a response for the command issued by the dedicated flash memory programmer. The response names sent from the 78K0/Kx2-A microcontrollers are listed below.

Table 25-7. Response Names

Response Name	Function
ACK	Acknowledges command/data.
NAK	Acknowledges illegal command/data.

25.7 Security Settings

The 78K0/Kx2-A microcontrollers support a security function that prohibits rewriting the user program written to the internal flash memory, so that the program cannot be changed by an unauthorized person.

The operations shown below can be performed using the Security Set command. The security setting is valid when the programming mode is set next.

- Disabling batch erase (chip erase)
  Execution of the block erase and batch erase (chip erase) commands for entire blocks in the flash memory is prohibited by this setting during on-board/off-board programming. Once execution of the batch erase (chip erase) command is prohibited, all of the prohibition settings (including prohibition of batch erase (chip erase)) can no longer be cancelled.

  Caution After the security setting for the batch erase is set, erasure cannot be performed for the device. In addition, even if a write command is executed, data different from that which has already been written to the flash memory cannot be written, because the erase command is disabled.

- Disabling block erase
  Execution of the block erase command for a specific block in the flash memory is prohibited during on-board/off-board programming. However, blocks can be erased by means of self programming.

- Disabling write
  Execution of the write and block erase commands for entire blocks in the flash memory is prohibited during on-board/off-board programming. However, blocks can be written by means of self programming.

- Disabling rewriting boot cluster 0
  Execution of the block erase command and write command on boot cluster 0 (0000H to 0FFFH) in the flash memory is prohibited by this setting. Execution of the batch erase (chip erase) command is also prohibited by this setting.

  Caution If a security setting that rewrites boot cluster 0 has been applied, the rewriting of boot cluster 0 and the batch erase (chip erase) will not be executed for the device.

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.

Prohibition of erasing blocks and writing is cleared by executing the batch erase (chip erase) command.

Table 25-8 shows the relationship between the erase and write commands when the 78K0/Kx2-A microcontroller security function is enabled.

Table 25-8. 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.

Table 25-9 shows how to perform security settings in each programming mode.

Table 25-9. 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 set 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		Cannot be disabled after set.

25.8 Flash Memory Programming by Self-Programming

The 78K0/Kx2-A microcontrollers support 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 a 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. To execute interrupt servicing, restore the normal operation mode after self-programming has been stopped, and execute the EI instruction. After the self-programming mode is later restored, self-programming can be resumed.

Remark For details of the self-programming function and the self-programming library, refer to **78K0 Microcontrollers Self Programming Library Type01 User's Manual (U18274E)**.

Cautions 1. The self-programming function cannot be used when the CPU operates with the subsystem clock.

2. Oscillation of the internal high-speed oscillator is started during self programming, regardless of the setting of the RSTOP flag (bit 0 of the internal oscillation mode register (RCM)). Oscillation of the internal high-speed oscillator cannot be stopped even if the STOP instruction is executed.

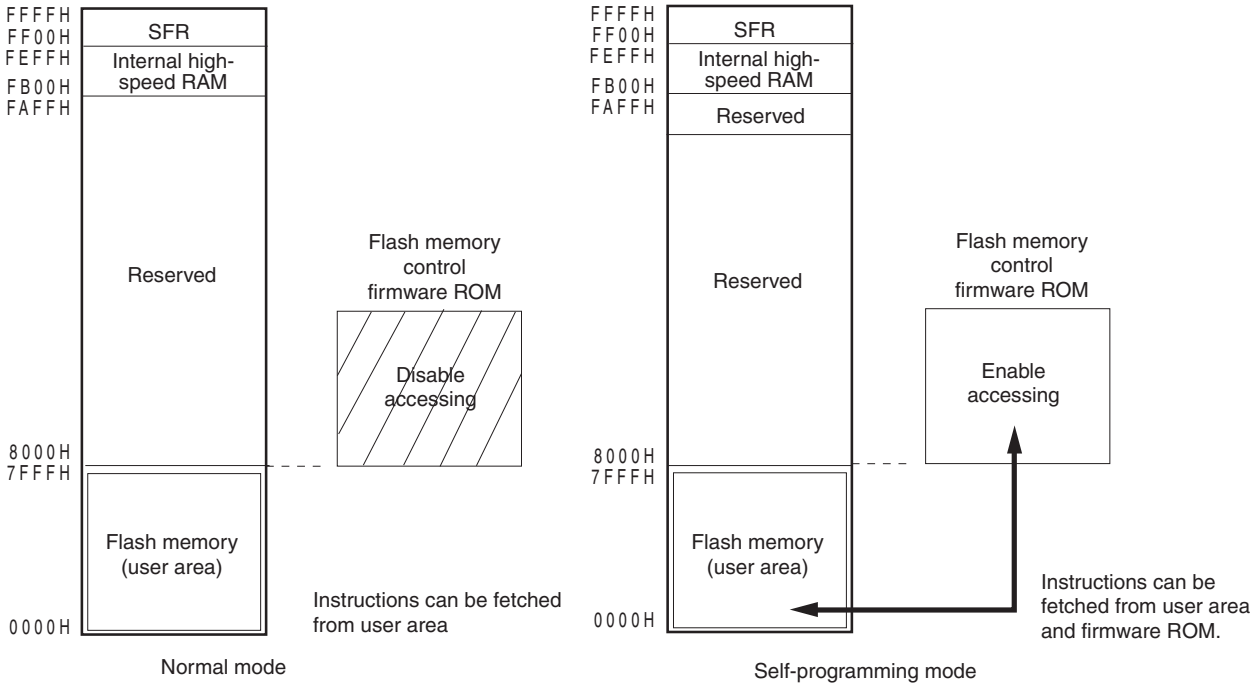
3. Input a high level to the FLMD0 pin during self-programming.

4. Be sure to execute the DI instruction before starting self-programming.
The self-programming function checks the interrupt request flags (IF0L, IF0H, IF1L, and IF1H). If an interrupt request is generated, self-programming is stopped.

5. Self-programming is also stopped by an interrupt request that is not masked even in the DI status. To prevent this, mask the interrupt by using the interrupt mask flag registers (MK0L, MK0H, MK1L, and MK1H).

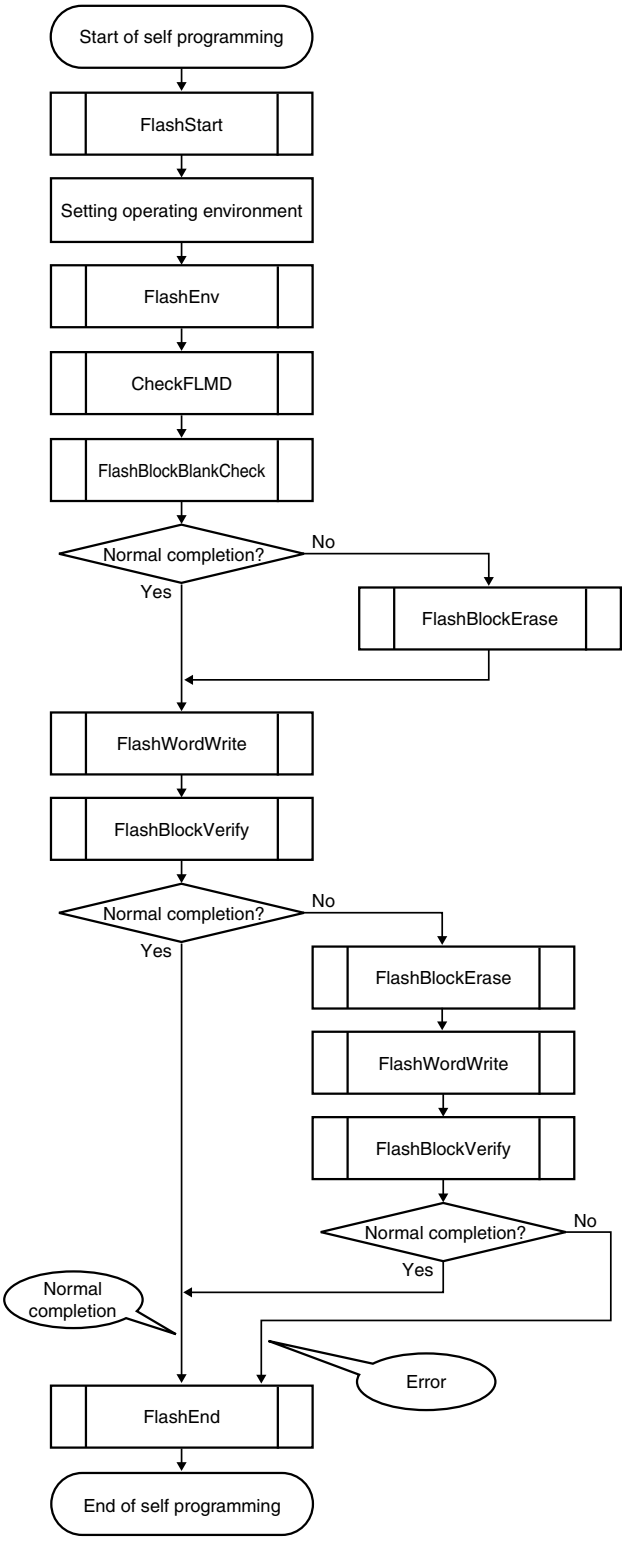
6. Allocate the entry program for self-programming in the area of 0000H to 7FFFH.

Figure 25-12. Operation Mode and Memory Map for Self-Programming (μPD78F0593)



The following figure illustrates a flow of rewriting the flash memory by using a self-programming library.

Figure 25-13. Flow of Self Programming (Rewriting Flash Memory)



Remark For details of the self-programming library, refer to **78K0 Microcontrollers Self Programming Library Type01 User's Manual (U18274E)**.

The following table shows the processing time and interrupt response time for the self-programming library.

Table 25-10. Processing Time for Self Programming Library (Reference Values) (1/3)

(1) When internal high-speed oscillation clock is used and entry RAM is located outside short direct addressing range

Library Name		Processing Time (μs)			
		Normal Model of C Compiler		Static Model of C Compiler/Assembler	
		Min.	Max.	Min.	Max.
Self programming start library		4.0	4.5	4.0	4.5
Initialize library		1105.9	1106.6	1105.9	1106.6
Mode check library		905.7	906.1	904.9	905.3
Block blank check library		12776.1	12778.3	12770.9	12772.6
Block erase library		26050.4	349971.3	26045.3	349965.6
Word write library		1180.1 + 203 × w	1184.3 + 2241 × w	1172.9 + 203 × w	1176.3 + 2241 × w
Block verify library		25337.9	25340.2	25332.8	25334.5
Self programming end library		4.0	4.5	4.0	4.5
Get information library	Option value: 03H	1072.9	1075.2	1067.5	1069.1
	Option value: 04H	1060.2	1062.6	1054.8	1056.6
	Option value: 05H	1023.8	1028.2	1018.3	1022.1
Set information library		70265.9	759995.0	70264.9	759994.0
EEPROM write library		1316.8 + 347 × w	1320.9 + 2385 × w	1309.0 + 347 × w	1312.4 + 2385 × w

(2) When internal high-speed oscillation clock is used and entry RAM is located in short direct addressing range

Library Name		Processing Time (μs)			
		Normal Model of C Compiler		Static Model of C Compiler/Assembler	
		Min.	Max.	Min.	Max.
Self programming start library		4.0	4.5	4.0	4.5
Initialize library		449.5	450.2	449.5	450.2
Mode check library		249.3	249.7	248.6	248.9
Block blank check library		12119.7	12121.9	12114.6	12116.3
Block erase library		25344.7	349266.4	25339.6	349260.8
Word write library		445.8 + 203 × w	449.9 + 2241 × w	438.5 + 203 × w	441.9 + 2241 × w
Block verify library		24682.7	24684.9	24677.6	24679.3
Self programming end library		4.0	4.5	4.0	4.5
Get information library	Option value: 03H	417.6	419.8	412.1	413.8
	Option value: 04H	405.0	407.4	399.5	401.3
	Option value: 05H	367.4	371.8	361.9	365.8
Set information library		69569.3	759297.3	69568.3	759296.2
EEPROM write library		795.1 + 347 × w	799.3 + 2385 × w	787.4 + 347 × w	790.8 + 2385 × w

Remarks 1. The above processing times are those when a write start address structure is located in the internal high-speed RAM and during stabilized operation of the internal high-speed oscillator (RSTS = 1).

2. RSTS: Bit 7 of the internal oscillation mode register (RCM)

3. w: Number of words in write data (1 word = 4 bytes)

Table 25-10. Processing Time for Self Programming Library (Reference Values) (2/3)

(3) When high-speed system clock (X1 oscillation or external clock input) is used and entry RAM is located outside short direct addressing range

Library Name		Processing Time (μs)			
		Normal Model of C Compiler		Static Model of C Compiler/Assembler	
		Min.	Max.	Min.	Max.
Self programming start library		34/f_{CPU}			
Initialize library		55/f_{CPU} + 594			
Mode check library		36/f_{CPU} + 495		30/f_{CPU} + 495	
Block blank check library		179/f_{CPU} + 6429		136/f_{CPU} + 6429	
Block erase library		179/f_{CPU} + 19713	179/f_{CPU} + 268079	136/f_{CPU} + 19713	136/f_{CPU} + 268079
Word write library		333/f_{CPU} + 647 + 136 × w	333/f_{CPU} + 647 + 1647 × w	272/f_{CPU} + 647 + 136 × w	272/f_{CPU} + 647 + 1647 × w
Block verify library		179/f_{CPU} + 13284		136/f_{CPU} + 13284	
Self programming end library		34/f_{CPU}			
Get information library	Option value: 03H	180/f_{CPU} + 581		134/f_{CPU} + 581	
	Option value: 04H	190/f_{CPU} + 574		144/f_{CPU} + 574	
	Option value: 05H	350/f_{CPU} + 535		304/f_{CPU} + 535	
Set information library		80/f_{CPU} + 43181	80/f_{CPU} + 572934	72/f_{CPU} + 43181	72/f_{CPU} + 572934
EEPROM write library		333/f_{CPU} + 729 + 209 × w	333/f_{CPU} + 729 + 1722 × w	268/f_{CPU} + 729 + 209 × w	268/f_{CPU} + 729 + 1722 × w

Remarks 1. The above processing times are those when a write start address structure is located in the internal high-speed RAM and during stabilized operation of the internal high-speed oscillator (RSTS = 1).

2. RSTS: Bit 7 of the internal oscillation mode register (RCM)

3. f_{CPU}: CPU operation clock frequency

4. w: Number of words in write data (1 word = 4 bytes)

Table 25-10. Processing Time for Self Programming Library (Reference Values) (3/3)

(4) When high-speed system clock (X1 oscillation or external clock input) is used and entry RAM is located in short direct addressing range

Library Name		Processing Time (μs)			
		Normal Model of C Compiler		Static Model of C Compiler/Assembler	
		Min.	Max.	Min.	Max.
Self programming start library		$34/f_{CPU}$			
Initialize library		$55/f_{CPU} + 272$			
Mode check library		$36/f_{CPU} + 173$		$30/f_{CPU} + 173$	
Block blank check library		$179/f_{CPU} + 6108$		$136/f_{CPU} + 6108$	
Block erase library		$179/f_{CPU} + 19371$	$179/f_{CPU} + 267738$	$136/f_{CPU} + 19371$	$136/f_{CPU} + 267738$
Word write library		$333/f_{CPU} + 247 + 136 \times w$	$333/f_{CPU} + 247 + 1647 \times w$	$272/f_{CPU} + 247 + 136 \times w$	$272/f_{CPU} + 247 + 1647 \times w$
Block verify library		$179/f_{CPU}+12964$		$136/f_{CPU}+12964$	
Self programming end library		$34/f_{CPU}$			
Get information library	Option value: 03H	$180/f_{CPU} + 261$		$134/f_{CPU} + 261$	
	Option value: 04H	$190/f_{CPU} + 254$		$144/f_{CPU} + 254$	
	Option value: 05H	$350/f_{CPU} + 213$		$304/f_{CPU} + 213$	
Set information library		$80/f_{CPU} + 42839$	$80/f_{CPU} + 572592$	$72/f_{CPU} + 42839$	$72/f_{CPU} + 572592$
EEPROM write library		$333/f_{CPU} + 516 + 209 \times w$	$333/f_{CPU} + 516 + 1722 \times w$	$268/f_{CPU} + 516 + 209 \times w$	$268/f_{CPU} + 516 + 1722 \times w$

Remarks 1. The above processing times are those when a write start address structure is located in the internal high-speed RAM and during stabilized operation of the internal high-speed oscillator (RSTS = 1).

2. RSTS: Bit 7 of the internal oscillation mode register (RCM)

3. f_{CPU}: CPU operation clock frequency

4. w: Number of words in write data (1 word = 4 bytes)

Table 25-11. Interrupt Response Time for Self Programming Library (Reference Values) (1/2)

(1) When internal high-speed oscillation clock is used

Library Name	Interrupt Response Time (μs (Max.))			
	Normal Model of C Compiler		Static Model of C Compiler/Assembler	
	Entry RAM location is outside short direct addressing range	Entry RAM location is in short direct addressing range	Entry RAM location is outside short direct addressing range	Entry RAM location is in short direct addressing range
Block blank check library	1100.9	431.9	1095.3	426.3
Block erase library	1452.9	783.9	1447.3	778.3
Word write library	1247.2	579.2	1239.2	571.2
Block verify library	1125.9	455.9	1120.3	450.3
Set information library	906.9	312.0	905.8	311.0
EEPROM write library	1215.2	547.2	1213.9	545.9

Remarks 1. The above interrupt response times are those during stabilized operation of the internal high-speed oscillator (RSTS = 1).

2. RSTS: Bit 7 of the internal oscillation mode register (RCM)

(2) When high-speed system clock is used (normal model of C compiler)

Library Name	Interrupt Response Time (μs (Max.))			
	RSTOP = 0, RSTS = 1		RSTOP = 1	
	Entry RAM location is outside short direct addressing range	Entry RAM location is in short direct addressing range	Entry RAM location is outside short direct addressing range	Entry RAM location is in short direct addressing range
Block blank check library	$179/f_{CPU} + 567$	$179/f_{CPU} + 246$	$179/f_{CPU} + 1708$	$179/f_{CPU} + 569$
Block erase library	$179/f_{CPU} + 780$	$179/f_{CPU} + 459$	$179/f_{CPU} + 1921$	$179/f_{CPU} + 782$
Word write library	$333/f_{CPU} + 763$	$333/f_{CPU} + 443$	$333/f_{CPU} + 1871$	$333/f_{CPU} + 767$
Block verify library	$179/f_{CPU} + 580$	$179/f_{CPU} + 259$	$179/f_{CPU} + 1721$	$179/f_{CPU} + 582$
Set information library	$80/f_{CPU} + 456$	$80/f_{CPU} + 200$	$80/f_{CPU} + 1598$	$80/f_{CPU} + 459$
EEPROM write library[Note]	$29/f_{CPU} + 767$ $333/f_{CPU} + 696$	$29/f_{CPU} + 447$ $333/f_{CPU} + 376$	$29/f_{CPU} + 767$ $333/f_{CPU} + 1838$	$29/f_{CPU} + 447$ $333/f_{CPU} + 700$

Note The longer value of the EEPROM write library interrupt response time becomes the Max. value, depending on the value of f_{CPU}.

Remarks 1. f_{CPU}: CPU operation clock frequency

2. RSTOP: Bit 0 of the internal oscillation mode register (RCM)

3. RSTS: Bit 7 of the internal oscillation mode register (RCM)

Table 25-11. Interrupt Response Time for Self Programming Library (Reference Values) (2/2)

(3) When high-speed system clock is used (static model of C compiler/assembler)

Library Name	Interrupt Response Time (μs (Max.))			
	RSTOP = 0, RSTS = 1		RSTOP = 1	
	Entry RAM location is outside short direct addressing range	Entry RAM location is in short direct addressing range	Entry RAM location is outside short direct addressing range	Entry RAM location is in short direct addressing range
Block blank check library	136/f_{CPU} + 567	136/f_{CPU} + 246	136/f_{CPU} + 1708	136/f_{CPU} + 569
Block erase library	136/f_{CPU} + 780	136/f_{CPU} + 459	136/f_{CPU} + 1921	136/f_{CPU} + 782
Word write library	272/f_{CPU} + 763	272/f_{CPU} + 443	272/f_{CPU} + 1871	272/f_{CPU} + 767
Block verify library	136/f_{CPU} + 580	136/f_{CPU} + 259	136/f_{CPU} + 1721	136/f_{CPU} + 582
Set information library	72/f_{CPU} + 456	72/f_{CPU} + 200	72/f_{CPU} + 1598	72/f_{CPU} + 459
EEPROM write library[Note]	19/f_{CPU} + 767 268/f_{CPU} + 696	19/f_{CPU} + 447 268/f_{CPU} + 376	19/f_{CPU} + 767 268/f_{CPU} + 1838	19/f_{CPU} + 447 268/f_{CPU} + 700

Note The longer value of the EEPROM write library interrupt response time becomes the Max. value, depending on the value of f_{CPU}.

Remarks 1. f_{CPU}: CPU operation clock frequency

2. RSTOP: Bit 0 of the internal oscillation mode register (RCM)

3. RSTS: Bit 7 of the internal oscillation mode register (RCM)

25.8.1 Boot swap function

If rewriting the boot area has failed during self-programming due to a power failure or some other cause, the data in the boot area may be lost and the program may not be restarted by resetting.

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 78K0/Kx2-A microcontrollers, 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.

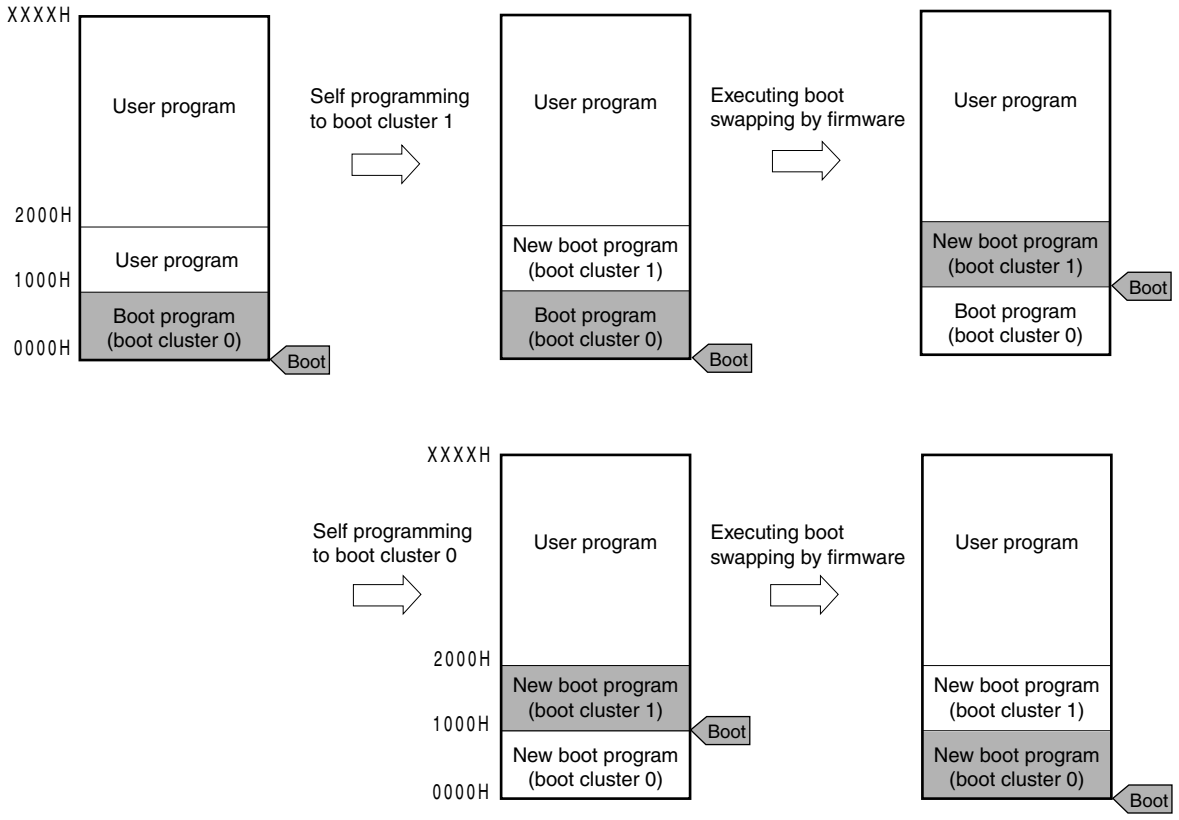
If the program has been correctly written to boot cluster 0, restore the original boot area by using the set information function of the firmware of the 78K0/Kx2-A microcontrollers.

Note A boot cluster is a 4 KB area and boot clusters 0 and 1 are swapped by the boot swap function.

Boot cluster 0 (0000H to 0FFFH): Original boot program area
Boot cluster 1 (1000H to 1FFFH): Area subject to boot swap function

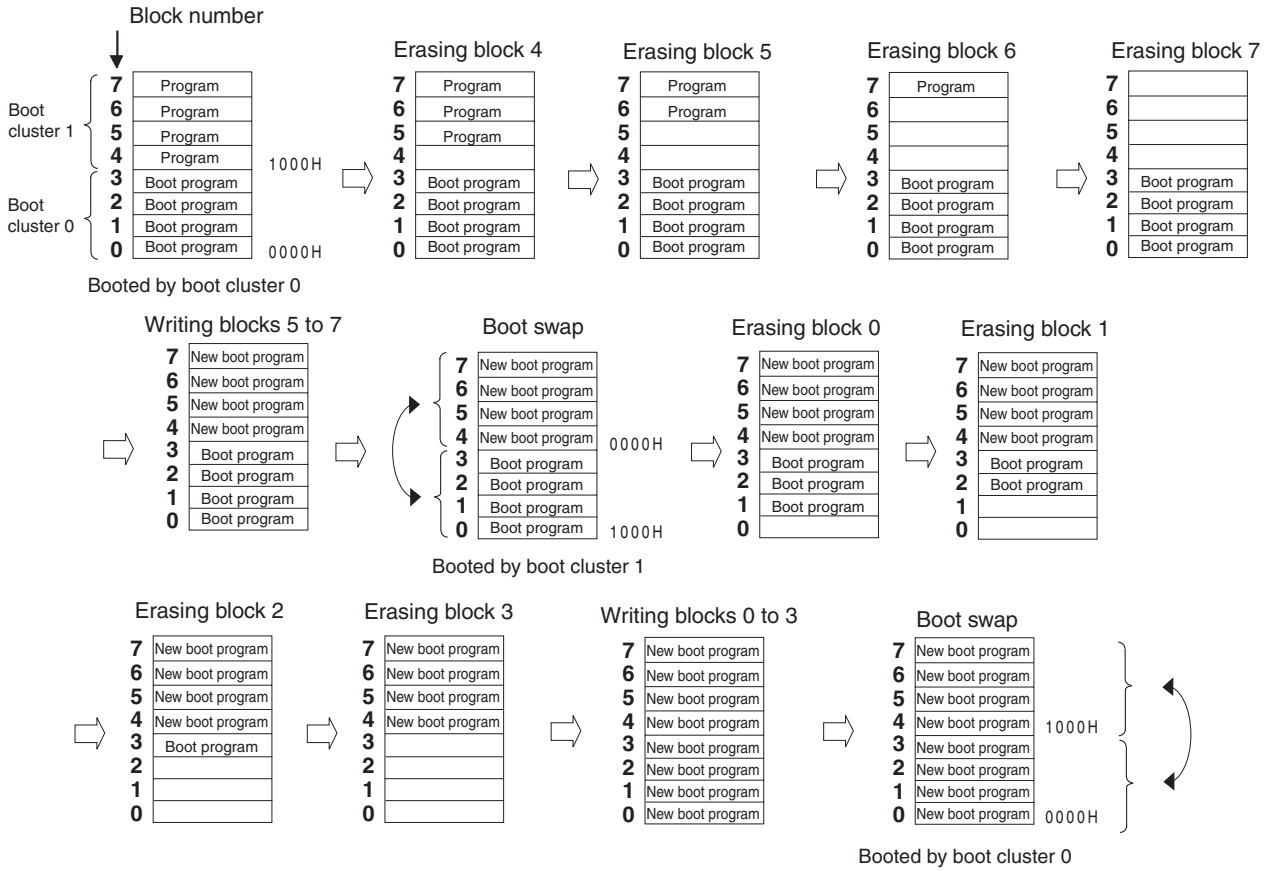
Caution When executing boot swapping, do not use the E.P.V command with the dedicated flash memory programmer.

Figure 25-14. Boot Swap Function



Remark Boot cluster 1 becomes 0000H to 0FFFH when a reset is generated after the boot flag has been set.

Figure 25-15. Example of Executing Boot Swapping

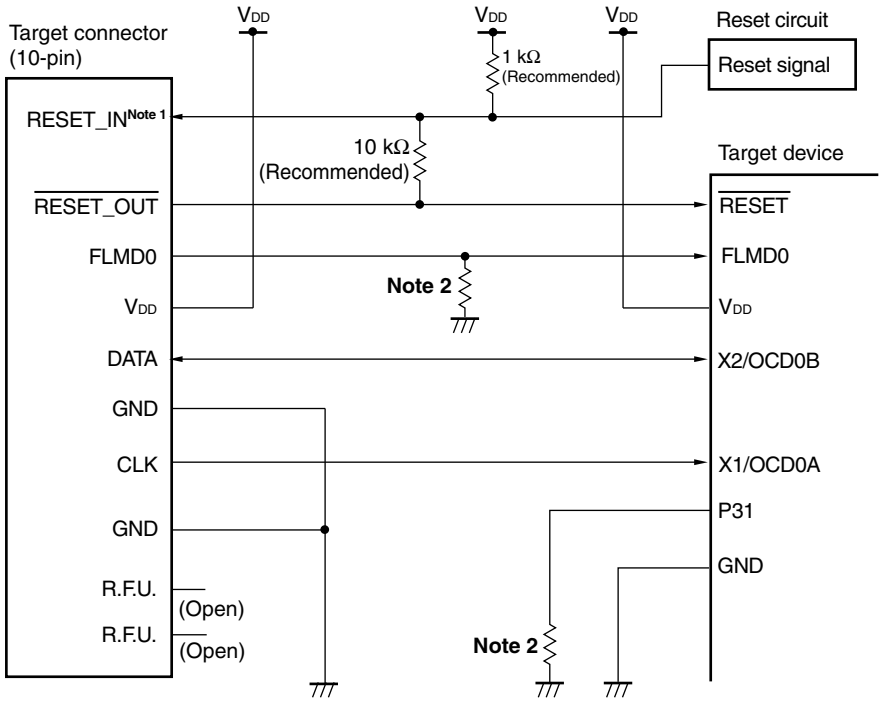


26.1 Connecting QB-MINI2 to 78K0/Kx2-A microcontrollers

The 78K0/Kx2-A microcontrollers use the V_{DD}, FLMD0, $\overline{RESET}$, OCD0A/X1 (or OCD1A/P31), OCD0B/X2 (or OCD1B/P32), and V_{SS} pins to communicate with the host machine via an on-chip debug emulator (QB-MINI2). Whether OCD0A/X1 and OCD1A/P31, or OCD0B/X2 and OCD1B/P32 are used can be selected.

Caution The 78K0/Kx2-A microcontrollers have an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. NEC Electronics is not liable for problems occurring when the on-chip debug function is used.

Figure 26-1. Connection Example of QB-MINI2 and 78K0/Kx2-A microcontrollers (When OCD0A/X1 and OCD0B/X2 Are Used)



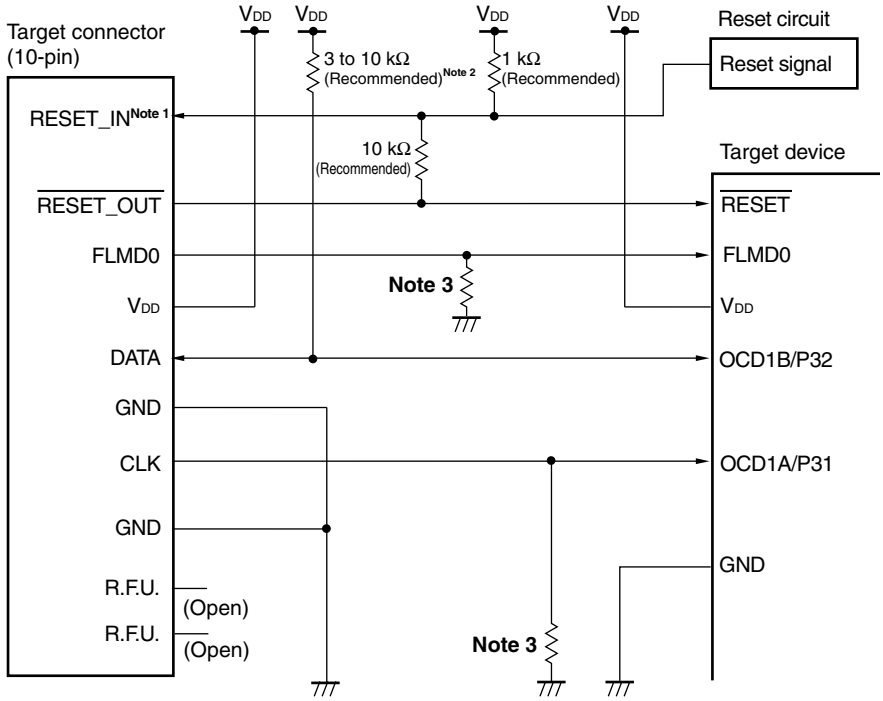
Notes 1. This connection is designed assuming that the reset signal is output from the N-ch open-drain buffer (output resistance: 100 Ω or less). For details, refer to **QB-MINI2 User's Manual (U18371E)**.

2. Make pull-down resistor 470 Ω or more (10 kΩ: recommended).

Cautions 1. Input the clock from the OCD0A/X1 pin during on-chip debugging.

2. Control the OCD0A/X1 and OCD0B/X2 pins by externally pulling down the OCD1A/P31 pin or by using an external circuit using the P130 pin (that outputs a low level when the device is reset).

Figure 26-2. Connection Example of QB-MINI2 and 78K0/Kx2-A microcontrollers (When OCD1A/P31 and OCD1B/P32 Are Used)



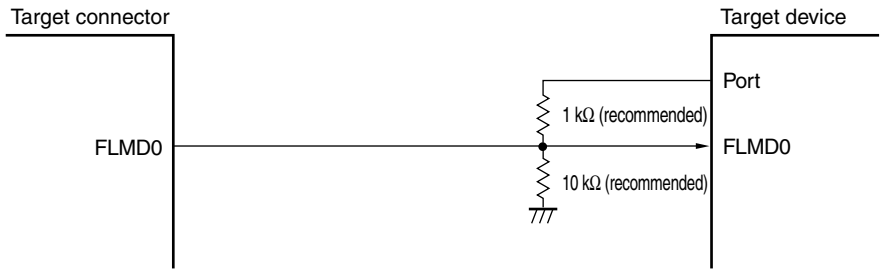
Notes 1. This connection is designed assuming that the reset signal is output from the N-ch open-drain buffer (output resistance: 100 Ω or less). For details, refer to **QB-MINI2 User's Manual (U18371E)**.

2. This is the processing of the pin when OCD1B/P32 is set as the input port (to prevent the pin from being left opened when not connected to QB-MINI2).

3. Make pull-down resistor 470 Ω or more (10 kΩ: recommended).

Connect the FLMD0 pin as follows when performing self programming by means of on-chip debugging.

Figure 26-3. Connection of FLMD0 Pin for Self Programming by Means of On-Chip Debugging



Caution When using the port that controls the FLMD0 pin, make sure that it satisfies the values of the high-level output current and FLMD0 supply voltage (minimum value: 0.8VDD) stated in CHAPTER 28 ELECTRICAL SPECIFICATIONS.

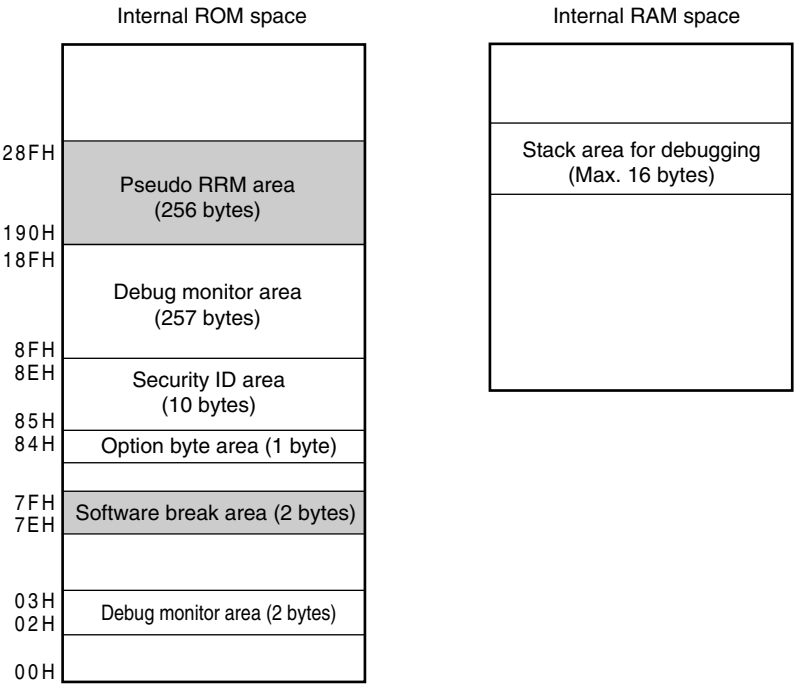
26.2 Reserved Area Used by QB-MINI2

QB-MINI2 uses the reserved areas shown in Figure 26-4 below to implement communication with the 78K0/Kx2-A microcontrollers, or each debug function. The shaded reserved areas are used for the respective debug functions to be used, and the other areas are always used for debugging. These reserved areas can be secured by using user programs and compiler options.

When using a boot swap operation during self programming, set the same value to boot cluster 1 beforehand.

For details on reserved area, refer to **QB-MINI2 User's Manual (U18371E)**.

Figure 26-4. Reserved Area Used by QB-MINI2



Remark Shaded reserved areas: Area used for the respective debug functions to be used
Other reserved areas: Areas always used for debugging

CHAPTER 27 INSTRUCTION SET

This chapter lists each instruction set of the 78K0/Kx2-A microcontrollers in table form. For details of each operation and operation code, refer to the separate document **78K/0 Series Instructions User's Manual (U12326E)**.

27.1 Conventions Used in Operation List

27.1.1 Operand identifiers and specification methods

Operands are written in the "Operand" column of each instruction in accordance with the specification method of the instruction operand identifier (refer to the assembler specifications for details). When there are two or more methods, select one of them. Uppercase letters and the symbols #, !, $ and [ ] are keywords and must be written as they are. Each symbol has the following meaning.

- #: Immediate data specification
- !: Absolute address specification
- $: Relative address specification
- [ ]: Indirect address specification

In the case of immediate data, describe an appropriate numeric value or a label. When using a label, be sure to write the #, !, $, and [ ] 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 specification.

Table 27-1. Operand Identifiers and Specification Methods

Identifier	Specification 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[Note]
sfrp	Special function register symbol (16-bit manipulatable register even addresses only)[Note]
saddr	FE20H to FF1FH Immediate data or labels
saddrp	FE20H to FF1FH Immediate data or labels (even address only)
addr16	0000H to FFFFH Immediate data or labels (Only even addresses for 16-bit data transfer instructions)
addr11	0800H to 0FFFH Immediate data or labels
addr5	0040H to 007FH Immediate data or labels (even address 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 Addresses from FFD0H to FFDFH cannot be accessed with these operands.

Remark For special function register symbols, see **Table 3-6 Special Function Register List**.

27.1.2 Description of operation column

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
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: Higher 8 bits and lower 8 bits of 16-bit register
$\wedge$: Logical product (AND)
$\vee$: Logical sum (OR)
$\veebar$: Exclusive logical sum (exclusive OR)
$\overline{\ \ }$: Inverted data
addr16: 16-bit immediate data or label
jdisp8: Signed 8-bit data (displacement value)

27.1.3 Description of flag operation column

(Blank): Not affected
0: Cleared to 0
1: Set to 1
×: Set/cleared according to the result
R: Previously saved value is restored

27.2 Operation List

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit data transfer	**MOV**	r, #byte	2	4	–	r ← byte	
		saddr, #byte	3	6	7	(saddr) ← byte	
		sfr, #byte	3	–	7	sfr ← byte	
		A, r Note 3	1	2	–	A ← r	
		r, A Note 3	1	2	–	r ← A	
		A, saddr	2	4	5	A ← (saddr)	
		saddr, A	2	4	5	(saddr) ← A	
		A, sfr	2	–	5	A ← sfr	
		sfr, A	2	–	5	sfr ← A	
		A, !addr16	3	8	9	A ← (addr16)	
		!addr16, A	3	8	9	(addr16) ← A	
		PSW, #byte	3	–	7	PSW ← byte	× × ×
		A, PSW	2	–	5	A ← PSW	
		PSW, A	2	–	5	PSW ← A	× × ×
		A, [DE]	1	4	5	A ← (DE)	
		[DE], A	1	4	5	(DE) ← A	
		A, [HL]	1	4	5	A ← (HL)	
		[HL], A	1	4	5	(HL) ← A	
		A, [HL + byte]	2	8	9	A ← (HL + byte)	
		[HL + byte], A	2	8	9	(HL + byte) ← A	
		A, [HL + B]	1	6	7	A ← (HL + B)	
		[HL + B], A	1	6	7	(HL + B) ← A	
		A, [HL + C]	1	6	7	A ← (HL + C)	
		[HL + C], A	1	6	7	(HL + C) ← A	
	XCH	A, r Note 3	1	2	–	A ↔ r	
		A, saddr	2	4	6	A ↔ (saddr)	
		A, sfr	2	–	6	A ↔ (sfr)	
		A, !addr16	3	8	10	A ↔ (addr16)	
		A, [DE]	1	4	6	A ↔ (DE)	
		A, [HL]	1	4	6	A ↔ (HL)	
		A, [HL + byte]	2	8	10	A ↔ (HL + byte)	
		A, [HL + B]	2	8	10	A ↔ (HL + B)	
		A, [HL + C]	2	8	10	A ↔ (HL + C)	

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM 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 processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
16-bit data transfer	**MOVW**	rp, #word	3	6	–	rp ← word	
		saddrp, #word	4	8	10	(saddrp) ← word	
		sfrp, #word	4	–	10	sfrp ← word	
		AX, saddrp	2	6	8	AX ← (saddrp)	
		saddrp, AX	2	6	8	(saddrp) ← AX	
		AX, sfrp	2	–	8	AX ← sfrp	
		sfrp, AX	2	–	8	sfrp ← AX	
		AX, rp Note 3	1	4	–	AX ← rp	
		rp, AX Note 3	1	4	–	rp ← AX	
		AX, !addr16	3	10	12	AX ← (addr16)	
		!addr16, AX	3	10	12	(addr16) ← AX	
	XCHW	AX, rp Note 3	1	4	–	AX ↔ rp	
8-bit operation	**ADD**	A, #byte	2	4	–	A, CY ← A + byte	× × ×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) + byte	× × ×
		A, r Note 4	2	4	–	A, CY ← A + r	× × ×
		r, A	2	4	–	r, CY ← r + A	× × ×
		A, saddr	2	4	5	A, CY ← A + (saddr)	× × ×
		A, !addr16	3	8	9	A, CY ← A + (addr16)	× × ×
		A, [HL]	1	4	5	A, CY ← A + (HL)	× × ×
		A, [HL + byte]	2	8	9	A, CY ← A + (HL + byte)	× × ×
		A, [HL + B]	2	8	9	A, CY ← A + (HL + B)	× × ×
		A, [HL + C]	2	8	9	A, CY ← A + (HL + C)	× × ×
	ADDC	A, #byte	2	4	–	A, CY ← A + byte + CY	× × ×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) + byte + CY	× × ×
		A, r Note 4	2	4	–	A, CY ← A + r + CY	× × ×
		r, A	2	4	–	r, CY ← r + A + CY	× × ×
		A, saddr	2	4	5	A, CY ← A + (saddr) + CY	× × ×
		A, !addr16	3	8	9	A, CY ← A + (addr16) + CY	× × ×
		A, [HL]	1	4	5	A, CY ← A + (HL) + CY	× × ×
		A, [HL + byte]	2	8	9	A, CY ← A + (HL + byte) + CY	× × ×
		A, [HL + B]	2	8	9	A, CY ← A + (HL + B) + CY	× × ×
		A, [HL + C]	2	8	9	A, CY ← A + (HL + C) + CY	× × ×

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Only when rp = BC, DE or HL

4. Except “r = A”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	**SUB**	A, #byte	2	4	–	A, CY ← A – byte	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) – byte	×	×	×
		A, r Note 3	2	4	–	A, CY ← A – r	×	×	×
		r, A	2	4	–	r, CY ← r – A	×	×	×
		A, saddr	2	4	5	A, CY ← A – (saddr)	×	×	×
		A, !addr16	3	8	9	A, CY ← A – (addr16)	×	×	×
		A, [HL]	1	4	5	A, CY ← A – (HL)	×	×	×
		A, [HL + byte]	2	8	9	A, CY ← A – (HL + byte)	×	×	×
		A, [HL + B]	2	8	9	A, CY ← A – (HL + B)	×	×	×
		A, [HL + C]	2	8	9	A, CY ← A – (HL + C)	×	×	×
	SUBC	A, #byte	2	4	–	A, CY ← A – byte – CY	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) – byte – CY	×	×	×
		A, r Note 3	2	4	–	A, CY ← A – r – CY	×	×	×
		r, A	2	4	–	r, CY ← r – A – CY	×	×	×
		A, saddr	2	4	5	A, CY ← A – (saddr) – CY	×	×	×
		A, !addr16	3	8	9	A, CY ← A – (addr16) – CY	×	×	×
		A, [HL]	1	4	5	A, CY ← A – (HL) – CY	×	×	×
		A, [HL + byte]	2	8	9	A, CY ← A – (HL + byte) – CY	×	×	×
		A, [HL + B]	2	8	9	A, CY ← A – (HL + B) – CY	×	×	×
		A, [HL + C]	2	8	9	A, CY ← A – (HL + C) – CY	×	×	×
	AND	A, #byte	2	4	–	A ← A ∧ byte	×		
		saddr, #byte	3	6	8	(saddr) ← (saddr) ∧ byte	×		
		A, r Note 3	2	4	–	A ← A ∧ r	×		
		r, A	2	4	–	r ← r ∧ A	×		
		A, saddr	2	4	5	A ← A ∧ (saddr)	×		
		A, !addr16	3	8	9	A ← A ∧ (addr16)	×		
		A, [HL]	1	4	5	A ← A ∧ (HL)	×		
		A, [HL + byte]	2	8	9	A ← A ∧ (HL + byte)	×		
		A, [HL + B]	2	8	9	A ← A ∧ (HL + B)	×		
		A, [HL + C]	2	8	9	A ← A ∧ (HL + C)	×		

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM 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 processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit operation	**OR**	A, #byte	2	4	–	A ← A ∨ byte	×
		saddr, #byte	3	6	8	(saddr) ← (saddr) ∨ byte	×
		A, r Note 3	2	4	–	A ← A ∨ r	×
		r, A	2	4	–	r ← r ∨ A	×
		A, saddr	2	4	5	A ← A ∨ (saddr)	×
		A, !addr16	3	8	9	A ← A ∨ (addr16)	×
		A, [HL]	1	4	5	A ← A ∨ (HL)	×
		A, [HL + byte]	2	8	9	A ← A ∨ (HL + byte)	×
		A, [HL + B]	2	8	9	A ← A ∨ (HL + B)	×
		A, [HL + C]	2	8	9	A ← A ∨ (HL + C)	×
	XOR	A, #byte	2	4	–	A ← A ⊻ byte	×
		saddr, #byte	3	6	8	(saddr) ← (saddr) ⊻ byte	×
		A, r Note 3	2	4	–	A ← A ⊻ r	×
		r, A	2	4	–	r ← r ⊻ A	×
		A, saddr	2	4	5	A ← A ⊻ (saddr)	×
		A, !addr16	3	8	9	A ← A ⊻ (addr16)	×
		A, [HL]	1	4	5	A ← A ⊻ (HL)	×
		A, [HL + byte]	2	8	9	A ← A ⊻ (HL + byte)	×
		A, [HL + B]	2	8	9	A ← A ⊻ (HL + B)	×
		A, [HL + C]	2	8	9	A ← A ⊻ (HL + C)	×
	CMP	A, #byte	2	4	–	A – byte	× × ×
		saddr, #byte	3	6	8	(saddr) – byte	× × ×
		A, r Note 3	2	4	–	A – r	× × ×
		r, A	2	4	–	r – A	× × ×
		A, saddr	2	4	5	A – (saddr)	× × ×
		A, !addr16	3	8	9	A – (addr16)	× × ×
		A, [HL]	1	4	5	A – (HL)	× × ×
		A, [HL + byte]	2	8	9	A – (HL + byte)	× × ×
		A, [HL + B]	2	8	9	A – (HL + B)	× × ×
		A, [HL + C]	2	8	9	A – (HL + C)	× × ×

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM 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 processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit operation	**ADDW**	AX, #word	3	6	–	AX, CY ← AX + word	×	×	×
	SUBW	AX, #word	3	6	–	AX, CY ← AX – word	×	×	×
	CMPW	AX, #word	3	6	–	AX – word	×	×	×
Multiply/ divide	**MULU**	X	2	16	–	AX ← A × X			
	DIVUW	C	2	25	–	AX (Quotient), C (Remainder) ← AX ÷ C			
Increment/ decrement	**INC**	r	1	2	–	r ← r + 1	×	×	
		saddr	2	4	6	(saddr) ← (saddr) + 1	×	×	
	DEC	r	1	2	–	r ← r – 1	×	×	
		saddr	2	4	6	(saddr) ← (saddr) – 1	×	×	
	INCW	rp	1	4	–	rp ← rp + 1			
	DECW	rp	1	4	–	rp ← rp – 1			
Rotate	**ROR**	A, 1	1	2	–	(CY, A_7 ← A_0, A_{m-1} ← A_m) × 1 time			×
	ROL	A, 1	1	2	–	(CY, A_0 ← A_7, A_{m+1} ← A_m) × 1 time			×
	RORC	A, 1	1	2	–	(CY ← A_0, A_7 ← CY, A_{m-1} ← A_m) × 1 time			×
	ROLC	A, 1	1	2	–	(CY ← A_7, A_0 ← CY, A_{m+1} ← A_m) × 1 time			×
	ROR4	[HL]	2	10	12	A_{3-0} ← $(HL)_{3-0}$, $(HL)_{7-4}$ ← A_{3-0}, $(HL)_{3-0}$ ← $(HL)_{7-4}$			
	ROL4	[HL]	2	10	12	A_{3-0} ← $(HL)_{7-4}$, $(HL)_{3-0}$ ← A_{3-0}, $(HL)_{7-4}$ ← $(HL)_{3-0}$			
BCD adjustment	**ADJBA**		2	4	–	Decimal Adjust Accumulator after Addition	×	×	×
	ADJBS		2	4	–	Decimal Adjust Accumulator after Subtract	×	×	×
Bit manipulate	**MOV1**	CY, saddr.bit	3	6	7	CY ← (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← sfr.bit			×
		CY, A.bit	2	4	–	CY ← A.bit			×
		CY, PSW.bit	3	–	7	CY ← PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← (HL).bit			×
		saddr.bit, CY	3	6	8	(saddr.bit) ← CY			
		sfr.bit, CY	3	–	8	sfr.bit ← CY			
		A.bit, CY	2	4	–	A.bit ← CY			
		PSW.bit, CY	3	–	8	PSW.bit ← CY	×	×	
		[HL].bit, CY	2	6	8	(HL).bit ← CY			

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
Bit manipulate	**AND1**	CY, saddr.bit	3	6	7	CY ← CY ∧ (saddr.bit)	×
		CY, sfr.bit	3	–	7	CY ← CY ∧ sfr.bit	×
		CY, A.bit	2	4	–	CY ← CY ∧ A.bit	×
		CY, PSW.bit	3	–	7	CY ← CY ∧ PSW.bit	×
		CY, [HL].bit	2	6	7	CY ← CY ∧ (HL).bit	×
	OR1	CY, saddr.bit	3	6	7	CY ← CY ∨ (saddr.bit)	×
		CY, sfr.bit	3	–	7	CY ← CY ∨ sfr.bit	×
		CY, A.bit	2	4	–	CY ← CY ∨ A.bit	×
		CY, PSW.bit	3	–	7	CY ← CY ∨ PSW.bit	×
		CY, [HL].bit	2	6	7	CY ← CY ∨ (HL).bit	×
	XOR1	CY, saddr.bit	3	6	7	CY ← CY ⊻ (saddr.bit)	×
		CY, sfr.bit	3	–	7	CY ← CY ⊻ sfr.bit	×
		CY, A.bit	2	4	–	CY ← CY ⊻ A.bit	×
		CY, PSW. bit	3	–	7	CY ← CY ⊻ PSW.bit	×
		CY, [HL].bit	2	6	7	CY ← CY ⊻ (HL).bit	×
	SET1	saddr.bit	2	4	6	(saddr.bit) ← 1	
		sfr.bit	3	–	8	sfr.bit ← 1	
		A.bit	2	4	–	A.bit ← 1	
		PSW.bit	2	–	6	PSW.bit ← 1	× × ×
		[HL].bit	2	6	8	(HL).bit ← 1	
	CLR1	saddr.bit	2	4	6	(saddr.bit) ← 0	
		sfr.bit	3	–	8	sfr.bit ← 0	
		A.bit	2	4	–	A.bit ← 0	
		PSW.bit	2	–	6	PSW.bit ← 0	× × ×
		[HL].bit	2	6	8	(HL).bit ← 0	
	SET1	CY	1	2	–	CY ← 1	1
	CLR1	CY	1	2	–	CY ← 0	0
	NOT1	CY	1	2	–	CY ← $\overline{CY}$	×

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
Call/return	**CALL**	!addr16	3	7	–	(SP – 1) ← (PC + 3)H, (SP – 2) ← (PC + 3)L, PC ← addr16, SP ← SP – 2	
	CALLF	!addr11	2	5	–	(SP – 1) ← (PC + 2)H, (SP – 2) ← (PC + 2)L, PC15 – 11 ← 00001, PC10 – 0 ← addr11, SP ← SP – 2	
	CALLT	[addr5]	1	6	–	(SP – 1) ← (PC + 1)H, (SP – 2) ← (PC + 1)L, PCH ← (addr5 + 1), PCL ← (addr5), SP ← SP – 2	
	BRK		1	6	–	(SP – 1) ← PSW, (SP – 2) ← (PC + 1)H, (SP – 3) ← (PC + 1)L, PCH ← (003FH), PCL ← (003EH), SP ← SP – 3, IE ← 0	
	RET		1	6	–	PCH ← (SP + 1), PCL ← (SP), SP ← SP + 2	
	RETI		1	6	–	PCH ← (SP + 1), PCL ← (SP), PSW ← (SP + 2), SP ← SP + 3	R R R
	RETB		1	6	–	PCH ← (SP + 1), PCL ← (SP), PSW ← (SP + 2), SP ← SP + 3	R R R
Stack manipulate	**PUSH**	PSW	1	2	–	(SP – 1) ← PSW, SP ← SP – 1	
		rp	1	4	–	(SP – 1) ← rpH, (SP – 2) ← rpL, SP ← SP – 2	
	POP	PSW	1	2	–	PSW ← (SP), SP ← SP + 1	R R R
		rp	1	4	–	rpH ← (SP + 1), rpL ← (SP), SP ← SP + 2	
	MOVW	SP, #word	4	–	10	SP ← word	
		SP, AX	2	–	8	SP ← AX	
		AX, SP	2	–	8	AX ← SP	
Unconditional branch	**BR**	!addr16	3	6	–	PC ← addr16	
		$addr16	2	6	–	PC ← PC + 2 + jdisp8	
		AX	2	8	–	PCH ← A, PCL ← X	
Conditional branch	**BC**	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if CY = 1	
	BNC	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if CY = 0	
	BZ	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if Z = 1	
	BNZ	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if Z = 0	

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
Conditional branch	**BT**	saddr.bit, $addr16	3	8	9	PC ← PC + 3 + jdisp8 if (saddr.bit) = 1	
		sfr.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if sfr.bit = 1	
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 1	
		PSW.bit, $addr16	3	–	9	PC ← PC + 3 + jdisp8 if PSW.bit = 1	
		[HL].bit, $addr16	3	10	11	PC ← PC + 3 + jdisp8 if (HL).bit = 1	
	BF	saddr.bit, $addr16	4	10	11	PC ← PC + 4 + jdisp8 if (saddr.bit) = 0	
		sfr.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if sfr.bit = 0	
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 0	
		PSW.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if PSW. bit = 0	
		[HL].bit, $addr16	3	10	11	PC ← PC + 3 + jdisp8 if (HL).bit = 0	
	BTCLR	saddr.bit, $addr16	4	10	12	PC ← PC + 4 + jdisp8 if (saddr.bit) = 1 then reset (saddr.bit)	
		sfr.bit, $addr16	4	–	12	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit	
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit	
		PSW.bit, $addr16	4	–	12	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	× × ×
		[HL].bit, $addr16	3	10	12	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit	
	DBNZ	B, $addr16	2	6	–	B ← B – 1, then PC ← PC + 2 + jdisp8 if B ≠ 0	
		C, $addr16	2	6	–	C ← C –1, then PC ← PC + 2 + jdisp8 if C ≠ 0	
		saddr, $addr16	3	8	10	(saddr) ← (saddr) – 1, then PC ← PC + 3 + jdisp8 if (saddr) ≠ 0	
CPU control	**SEL**	RBn	2	4	–	RBS1, 0 ← n	
	NOP		1	2	–	No Operation	
	EI		2	–	6	IE ← 1 (Enable Interrupt)	
	DI		2	–	6	IE ← 0 (Disable Interrupt)	
	HALT		2	6	–	Set HALT Mode	
	STOP		2	6	–	Set STOP Mode	

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

27.3 Instructions Listed by Addressing Type

(1) 8-bit instructions

MOV, XCH, ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP, MULU, DIVUW, INC, DEC, ROR, ROL, RORC, ROLC, ROR4, ROL4, PUSH, POP, DBNZ

First Operand \ Second Operand	#byte	A	r[Note]	sfr	saddr	!addr16	PSW	[DE]	[HL]	[HL + byte] [HL + B] [HL + C]	$addr16	1	None
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP		ROR ROL RORC ROLC	
r	MOV	MOV ADD ADDC SUB SUBC AND OR XOR CMP											INC DEC
B, C											DBNZ		
sfr	MOV	MOV											
saddr	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV									DBNZ		INC DEC
!addr16		MOV											
PSW	MOV	MOV											PUSH POP
[DE]		MOV											
[HL]		MOV											ROR4 ROL4
[HL + byte] [HL + B] [HL + C]		MOV											
X													MULU
C													DIVUW

Note Except “r = A”

(2) 16-bit instructions

MOVW, XCHW, ADDW, SUBW, CMPW, PUSH, POP, INCW, DECW

First Operand \ Second Operand	#word	AX	rp[Note]	sfrp	saddrp	!addr16	SP	None
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	MOVW	MOVW	
rp	MOVW	MOVW[Note]						INCW DECW PUSH POP
sfrp	MOVW	MOVW						
saddrp	MOVW	MOVW						
!addr16		MOVW						
SP	MOVW	MOVW						

Note Only when rp = BC, DE, HL

(3) Bit manipulation instructions

MOV1, AND1, OR1, XOR1, SET1, CLR1, NOT1, BT, BF, BTCLR

First Operand \ Second Operand	A.bit	sfr.bit	saddr.bit	PSW.bit	[HL].bit	CY	$addr16	None
A.bit						MOV1	BT BF BTCLR	SET1 CLR1
sfr.bit						MOV1	BT BF BTCLR	SET1 CLR1
saddr.bit						MOV1	BT BF BTCLR	SET1 CLR1
PSW.bit						MOV1	BT BF BTCLR	SET1 CLR1
[HL].bit						MOV1	BT BF BTCLR	SET1 CLR1
CY	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1			SET1 CLR1 NOT1

(4) Call instructions/branch instructions

CALL, CALLF, CALLT, BR, BC, BNC, BZ, BNZ, BT, BF, BTCLR, DBNZ

First Operand \ Second Operand	AX	!addr16	!addr11	[addr5]	$addr16
Basic instruction	BR	CALL BR	CALLF	CALLT	BR BC BNC BZ BNZ
Compound instruction					BT BF BTCLR DBNZ

(5) Other instructions

ADJBA, ADJBS, BRK, RET, RETI, RETB, SEL, NOP, EI, DI, HALT, STOP

CHAPTER 28 ELECTRICAL SPECIFICATIONS

Cautions 1. The 78K0/Kx2-A microcontrollers have an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. NEC Electronics is not liable for problems occurring when the on-chip debug function is used.

2. The pins mounted depend on the product as follows.

(1) Port functions

Port	78K0/KB2-A	78K0/KC2-A
	30 Pins	48 Pins
Port 0	–	P00 to P02
Port 1	P10 to P13	
Port 2	P20 to P25	P20 to P27
Port 3	P31, P32, P35	P31 to P35
Port 4	–	P40 to P42
Port 6	P60, P61	
Port 7	–	P70 to P75
Port 8	P80 to P83	
Port 12	P120 to P122	P120 to P124

(The remaining table is on the next page.)

(2) Non-port functions

Port		78K0/KB2-A	78K0/KC2-A
		30 Pins	48 Pins
Power supply, ground		V_{DD}, V_{SS}, AV_{REF}, AV_{DD} , AV_{SS}	V_{DD}, V_{SS}, AV_{REFP}, AV_{REFM}, AV_{DD}, AV_{SS}
Regulator		REGC	
Reset		$\overline{\text{RESET}}$	
Clock oscillation		X1, X2, EXCLK	X1, X2, XT1, XT2, EXCLK
Writing to flash memory		FLMD0	
Interrupt		INTP0, INTP1, INTP4 to INTP7	INTP0 to INTP9
Key interrupt		–	KR0 to KR5
Timer	TM00	TI000, TI010, TO00	
	TM50	TI50, TO50	
	TM51	TI51, TO51	
	TMH0	TOH0	
	TMH1	TOH1	
Serial	UART6	RxD6, TxD6	
	IICA	SCLA0, SDAA0	
	CSI10	$\overline{\text{SCK10}}$, SI10, SO10	$\overline{\text{SCK10}}$, SI10, SO10, $\overline{\text{SSI10}}$
A/D converter		ANI0 to ANI5, ANI8 to ANI11	ANI0 to ANI6, ANI8 to ANI11, ANI15
Clock output		–	PCL
Real-time counter output		–	RTC1HZ, RTCCL, RTCDIV
Low-voltage detector (LVI)		EXLVI	
On-chip debug function		OCD0A, OCD1A, OCD0B, OCD1B	

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

Absolute Maximum Ratings (T_A = 25°C) (1/2)

Parameter	Symbol	Conditions	Ratings	Unit
Supply voltage	V_{DD}		−0.5 to +6.5	V
	V_{SS}		−0.5 to +0.3	V
	AV_{DD}		−0.5 to V_{DD} +0.3[Note]	V
	AV_{SS}		−0.5 to +0.3	V
REGC pin input voltage	V_{IREGC}		–0.5 to +3.6 and –0.5 to V_{DD}	V
Input voltage	V_{I1}	P00 to P02, P10 to P13, P31 to P35, P40 to P42, P70 to P75, P120 to P124, X1, X2, XT1, XT2, $\overline{\text{RESET}}$, FLMD0	−0.3 to V_{DD} +0.3[Note]	V
	V_{I2}	P60, P61 (N-ch open drain)	−0.3 to +6.5[Note]	V
	V_{I3}	P20 to P27, P80 to P83	−0.3 to AV_{DD} +0.3 and −0.3 to V_{DD} +0.3[Note]	V
Output voltage	V_{O1}	Other than P20 to P27, P80 to P83	−0.3 to V_{DD} +0.3[Note]	V
	V_{O2}	P20 to P27, P80 to P83	−0.3 to AV_{DD} +0.3[Note]	V
Analog input voltage	V_{AN1}	ANI0 to ANI6, ANI8 to ANI11, ANI15, AMP0+, AMP1+, AMP2+, AMP0−, AMP1−, AMP2−	−0.3 to AV_{DD} +0.3[Note] and −0.3 to V_{DD} +0.3[Note]	V
Analog output voltage	V_{ANO1}	AMP0OUT, AMP1OUT, AMP2OUT	−0.3 to AV_{DD} +0.3	V
Anarog reference voltage input	AV_{REFP}		−0.3 to AV_{DD} +0.3[Note]	V
	AV_{REFM}		−0.3 to AV_{DD} +0.3[Note] and $AV_{REFM} \leq AV_{REFP}$[Note]	V

Note Must be 6.5 V or lower.

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

Absolute Maximum Ratings (T_A = 25°C) (2/2)

Parameter	Symbol	Conditions		Ratings	Unit
Output current, high	I_{OH}	Per pin	P00 to P02, P10 to P13, P31 to P35, P40 to P42, P70 to P75, P120	−10	mA
		Total of all pins −80 mA	P00 to P02, P40 to P42, P120	−25	mA
			P10 to P13, P31 to P35, P70 to P75	−55	mA
		Per pin	P20 to P27, P80 to P83	−0.5	mA
		Total of all pins		−2	mA
		Per pin	P121 to P124	−1	mA
		Total of all pins		−4	mA
		Per pin	AMP0OUT, AMP1OUT, AMP2OUT	−1	mA
		Total of all pins		−3	mA
Output current, low	I_{OL}	Per pin	P00 to P02, P10 to P13, P31 to P35, P40 to P42, P60, P61, P70 to P75, P120	30	mA
		Total of all pins 200 mA	P00 to P02, P40 to P42, P120	60	mA
			P10 to P13, P31 to P35, P60, P61, P70 to P75	140	mA
		Per pin	P20 to P27, P80 to P83	1	mA
		Total of all pins		5	mA
		Per pin	P121 to P124	4	mA
		Total of all pins		10	mA
		Per pin	AMP0OUT, AMP1OUT, AMP2OUT	1	mA
		Total of all pins		3	mA
Operating ambient temperature	T_A			−40 to +85	°C
Storage temperature	T_{stg}			−65 to +150	°C

Cautions 1. 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.

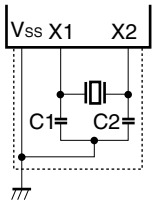
2. The value of the current that can be run per pin must satisfy the value of the current per pin and the total value of the currents of all pins.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

X1 Oscillator Characteristics

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic resonator, Crystal resonator		X1 clock oscillation frequency (f_X)[Note 1]	2.7 V ≤ V_{DD} ≤ 5.5 V	1.0[Note 2]		20.0	MHz
			1.8 V ≤ V_{DD} < 2.7 V	1.0		5.0	MHz

Notes 1. Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

2. It is 2.0 MHz (MIN.) when programming on the board via UART6.

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.

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

Internal Oscillator Characteristics

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

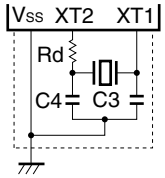
Resonator	Parameter	Conditions		MIN.	TYP.	MAX.	Unit
8 MHz internal oscillator	Internal high-speed oscillation clock frequency (f_{RH})[Note]	RSTS = 1	2.7 V ≤ V_{DD} ≤ 5.5 V	7.6	8.0	8.4	MHz
			1.8 V ≤ V_{DD} < 2.7 V	7.6	8.0	10.4	MHz
		RSTS = 0		2.48	5.6	9.86	MHz
240 kHz internal oscillator	Internal low-speed oscillation clock frequency (f_{RL})	2.7 V ≤ V_{DD} ≤ 5.5 V		216	240	264	kHz
		1.8 V ≤ V_{DD} < 2.7 V		192	240	264	kHz

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

Remark RSTS: Bit 7 of the internal oscillation mode register (RCM)

XT1 Oscillator Characteristics[Note 1]

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Crystal resonator		XT1 clock oscillation frequency (f_{XT})[Note 2]		32	32.768	35	kHz

Notes 1. The 78K0/KB2-A is not provided with the XT1 oscillator.

2. 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 figure 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.

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

DC Characteristics (1/5)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, AV_{DD} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high[Note 1]	I_{OH1}	Per pin for P00 to P02, P10 to P13, P31 to P35, P40 to P42, P70 to P75, P120	4.0 V ≤ V_{DD} ≤ 5.5 V			–3.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			–2.5	mA
			1.8 V ≤ V_{DD} < 2.7 V			–1.0	mA
		Total of P00 to P02, P40 to P42, P120[Note 3]	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 P10 to P13, P31 to P35, P70 to P75[Note 3]	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 the pins above[Note 3]	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
	I_{OH2}	Per pin for P20 to P27, P80 to P83	AV_{DD} = V_{DD}			–0.1	mA
		Per pin for P121 to P124				–0.1	mA
Output current, low[Note 2]	I_{OL1}	Per pin for P00 to P02, P10 to P13, P31 to P35, P40 to P42, P70 to P75, P120	4.0 V ≤ V_{DD} ≤ 5.5 V			8.5	mA
			2.7 V ≤ V_{DD} < 4.0 V			5.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			2.0	mA
		Per pin for P60, P61	4.0 V ≤ V_{DD} ≤ 5.5 V			15.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			5.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			2.0	mA
		Total of P00 to P02, P40 to P42, P120[Note 3]	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 P10 to P13, P31 to P35, P60, P61, P70 to P75[Note 3]	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 the pins above[Note 3]	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
	I_{OL2}	Per pin for P20 to P27, P80 to P83	AV_{DD} = V_{DD}			0.4	mA
		Per pin for P121 to P124				0.4	mA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from V_{DD} to an output pin.

2. Value of current at which the device operation is guaranteed even if the current flows from an output pin to GND.

3. Specification under conditions where the duty factor is 70% (time for which current is output is 0.7 × t and time for which current is not output is 0.3 × t, where t is a specific time). The total output current of the pins at a duty factor of other than 70% can be calculated by the following expression.

- Where the duty factor of I_{OH} is n%: Total output current of pins = (I_{OH} × 0.7)/(n × 0.01)

<Example> Where the duty factor is 50%, I_{OH} = –20.0 mA

Total output current of pins = (–20.0 × 0.7)/(50 × 0.01) = –28.0 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

DC Characteristics (2/5)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, AV_{DD} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P40, P41, P121 to P124, EXCLK		0.7V_{DD}		V_{DD}	V
	V_{IH2}	P00 to P02, P10 to P13, P31 to P35, P42, P70 to P75, P120, $\overline{\text{RESET}}$		0.8V_{DD}		V_{DD}	V
	V_{IH3}	P20 to P27, P80 to P83	AV_{DD} = V_{DD}	0.7AV_{DD}		AV_{DD}	V
	V_{IH4}	P60, P61		0.7V_{DD}		6.0	V
Input voltage, low	V_{IL1}	P40, P41, P60, P61, P121 to P124, EXCLK		0		0.3V_{DD}	V
	V_{IL2}	P00 to P02, P10 to P13, P31 to P35, P42, P70 to P75, P120, $\overline{\text{RESET}}$		0		0.2V_{DD}	V
	V_{IL3}	P20 to P27, P80 to P83	AV_{DD} = V_{DD}	0		0.3AV_{DD}	V
Output voltage, high	V_{OH1}	P00 to P02, P10 to P13, P31 to P35, P40 to P42, P70 to P75, P120	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = –3.0 mA	V_{DD} – 0.7			V
			2.7 V ≤ V_{DD} < 4.0 V, I_{OH1} = –2.5 mA	V_{DD} – 0.5			V
			1.8 V ≤ V_{DD} < 2.7 V, I_{OH1} = –1.0 mA	V_{DD} – 0.5			V
	V_{OH2}	P20 to P27, P80 to P83	AV_{DD} = V_{DD}, I_{OH2} = –100 μA	AV_{DD} – 0.5			V
		P121 to P124	I_{OH2} = –100 μA	V_{DD} – 0.5			V

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

DC Characteristics (3/5)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, AV_{DD} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output voltage, low	V_{OL1}	P00 to P02, P10 to P13, P31 to P35, P40 to P42, P70 to P75, P120	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 8.5 mA			0.7	V
			2.7 V ≤ V_{DD} < 4.0 V, I_{OL1} = 5.0 mA			0.7	V
			1.8 V ≤ V_{DD} < 2.7 V, I_{OL1} = 2.0 mA			0.5	V
			1.8 V ≤ V_{DD} < 2.7 V, I_{OL1} = 0.5 mA			0.4	V
	V_{OL2}	P20 to P27, P80 to P83	AV_{DD} = V_{DD}, I_{OL2} = 0.4 mA			0.4	V
		P121 to P124	I_{OL2} = 0.4 mA			0.4	V
	V_{OL3}	P60, P61	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} < 4.0 V, I_{OL1} = 5.0 mA			0.6	V
			2.7 V ≤ V_{DD} < 4.0 V, I_{OL1} = 3.0 mA			0.4	V
			1.8 V ≤ V_{DD} < 2.7 V, I_{OL1} = 2.0 mA			0.4	V
Input leakage current, high	I_{LIH1}	P00 to P02, P10 to P13, P31 to P35, P40 to P42, P60, P61, P70 to P75, P120, FLMD0, $\overline{\text{RESET}}$	V_I = V_{DD}			1	μA
	I_{LIH2}	P20 to P27, P80 to P83	V_I = AV_{DD}, AV_{DD} = V_{DD}			1	μA
	I_{LIH3}	P121 to 124 (X1, X2, XT1, XT2)	V_I = V_{DD} I/O port mode			1	μA
			OSC mode			20	μA
Input leakage current, low	I_{LIL1}	P00 to P02, P10 to P13, P31 to P35, P40 to P42, P60, P61, P70 to P75, P120, FLMD0, $\overline{\text{RESET}}$	V_I = V_{SS}			–1	μA
	I_{LIL2}	P20 to P27, P80 to P83	V_I = V_{SS}, AV_{DD} = V_{DD}			–1	μA
	I_{LIL3}	P121 to 124 (X1, X2, XT1, XT2)	V_I = V_{SS} I/O port mode			–1	μA
			OSC mode			–20	μA
Pull-up resistor	R_U	V_I = V_{SS}		10	20	100	kΩ
FLMD0 supply voltage	V_{IL}	In normal operation mode		0		0.2V_{DD}	V
	V_{IH}	In self-programming mode		0.8V_{DD}		V_{DD}	V

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

DC Characteristics (4/5)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, AV_{DD} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current[Note 1]	I_{DD1}	Operating mode	f_{XH} = 20 MHz, V_{DD} = 5.0 V [Note 2]	Square wave input		3.2	5.5	mA
				Resonator connection		4.5	6.9	mA
			f_{XH} = 20 MHz, V_{DD} = 3.0 V [Note 2]	Square wave input		3.2	5.5	mA
				Resonator connection		4.2	6.6	mA
			f_{XH} = 10 MHz, V_{DD} = 5.0 V [Notes 2, 3]	Square wave input		1.6	2.8	mA
				Resonator connection		2.3	3.9	mA
			f_{XH} = 10 MHz, V_{DD} = 3.0 V [Notes 2, 3]	Square wave input		1.5	2.7	mA
				Resonator connection		2.2	3.2	mA
			f_{XH} = 5 MHz, V_{DD} = 3.0 V [Notes 2, 3]	Square wave input		0.9	1.6	mA
				Resonator connection		1.3	2.0	mA
			f_{XH} = 5 MHz, V_{DD} = 2.0 V [Notes 2, 3]	Square wave input		0.7	1.4	mA
				Resonator connection		1.0	1.6	mA
			f_{RH} = 8 MHz, V_{DD} = 5.0 V [Note 4]			1.4	2.5	mA
			f_{SUB} = 32.768 kHz, V_{DD} = 5.0 V [Note 5]	Resonator connection		11	30	μA
			f_{SUB} = 32.768 kHz, V_{DD} = 3.0 V [Note 5]	Resonator connection		6	28	μA
	I_{DD2}	HALT mode	f_{XH} = 20 MHz, V_{DD} = 5.0 V [Note 2]	Square wave input		0.8	2.6	mA
				Resonator connection		2.0	4.4	mA
			f_{XH} = 20 MHz, V_{DD} = 3.0 V [Note 2]	Square wave input		0.8	2.6	mA
				Resonator connection		1.7	4.1	mA
			f_{XH} = 10 MHz, V_{DD} = 5.0 V [Notes 2, 3]	Square wave input		0.4	1.3	mA
				Resonator connection		1.0	2.4	mA
			f_{XH} = 10 MHz, V_{DD} = 3.0 V [Notes 2, 3]	Square wave input		0.4	1.3	mA
				Resonator connection		0.7	2.1	mA
			f_{XH} = 5 MHz, V_{DD} = 3.0 V [Notes 2, 3]	Square wave input		0.2	0.65	mA
				Resonator connection		0.5	1.1	mA
			f_{RH} = 8 MHz, V_{DD} = 5.0 V [Note 4]			0.4	1.2	mA
			f_{SUB} = 32.768 kHz, V_{DD} = 5.0 V [Note 5]	Resonator connection		9	25	μA
			f_{SUB} = 32.768 kHz, V_{DD} = 3.0 V [Note 5]	Resonator connection		4	24	μA
	I_{DD3}[Note 6]	STOP mode				1	20	μA
			T_A = –40 to +70 °C			1	10	μA

Remarks 1. f_{XH}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

2. f_{RH}: Internal high-speed oscillation clock frequency

3. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

(Notes on next page)

Notes 1. Total current flowing into the internal power supply (V_{DD}), including the peripheral operation current and the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. However, the current flowing into the pull-up resistors and the output current of the port are not included.

2. Not including the operating current of the 8 MHz internal oscillator, 240 kHz internal oscillator, and XT1 oscillator, and the current flowing into the A/D converter, operational amplifier, watchdog timer, and LVI circuit.

3. When AMPH (bit 0 of clock operation mode select register (OSCCTL)) = 0.

4. Not including the operating current of the X1 oscillator, XT1 oscillator, and 240 kHz internal oscillator, and the current flowing into the A/D converter, operational amplifier, watchdog timer, and LVI circuit.

5. Not including the operating current of the X1 oscillator, 8 MHz internal oscillator, and 240 kHz internal oscillator, and the current flowing into the A/D converter, operational amplifier, watchdog timer, and LVI circuit.

6. Not including the operating current of the 240 kHz internal oscillator and XT1 oscillation, and the current flowing into the A/D converter, operational amplifier, watchdog timer, and LVI circuit.

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

DC Characteristics (5/5)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, AV_{DD} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Watchdog timer operating current	I_{WDT} Note 1	During 240 kHz internal low-speed oscillation clock operation				5	10	μA
LVI operating current	I_{LVI} Note 2					9	18	μA
A/D converter operating current	I_{ADC} Note 3	During conversion at maximum speed	Normal mode 1	AV_{DD} = 5.0 V		1.7	3.4	mA
				AV_{DD} = 3.0 V		0.7	1.4	mA
			Normal mode 2	AV_{DD} = 2.3 V		0.5	1.2	mA
			Low-voltage mode	AV_{DD} = 1.8 V		0.3	0.8	mA
Operational amplifier operating current	I_{AMP} Note 4		Mode 1	AV_{DD} = 3.0 V		25	37.5	μA
			Mode 2			50	75	μA
			Mode 3			153	230	μA
RTC operating current	I_{RTC} Note 5	f_{SUB} = 32.768 kHz		V_{DD} = 3.0 V		0.2	1.0	μA
				V_{DD} = 2.0 V		0.2	1.0	μA

Notes 1. Current flowing only to the watchdog timer (including the operating current of the 240 kHz internal oscillator). The current value of the 78K0/Kx2-A microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{WDT} when the watchdog timer operates.

2. Current flowing only to the LVI circuit. The current value of the 78K0/Kx2-A microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{LVI} when the LVI circuit operates.

3. Current flowing only to the A/D converter (AV_{DD}). The current value of the 78K0/Kx2-A microcontrollers 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.

4. Current flowing only to the operational amplifier (AV_{DD}). The current value of the 78K0/Kx2-A microcontrollers is the sum of I_{DD1}, I_{DD2}, or I_{DD3} and I_{AMP} when the operational amplifier operates in an operation mode, HALT mode, or STOP mode.

5. Current flowing only to the real-time counter (excluding the operating current of the XT1 oscillator). The current value of the 78K0/Kx2-A microcontrollers 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.

Remarks 1. f_{XH}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

2. f_{RH}: Internal high-speed oscillation clock frequency

3. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

AC Characteristics

(1) Basic operation

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, AV_{DD} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{XP}) operation	2.7 V ≤ V_{DD} ≤ 5.5 V	0.1		32	μs
			1.8 V ≤ V_{DD} < 2.7 V	0.4 Note 1		32	μs
		Subsystem clock (f_{SUB}) operation Note 2		114	122	125	μs
Peripheral hardware clock frequency	f_{PRS}	f_{PRS} = f_{XH} (XSEL = 1)	4.0 V ≤ V_{DD} ≤ 5.5 V			20	MHz
			2.7 V ≤ V_{DD} < 4.0 V			20 Note 3	MHz
			1.8 V ≤ V_{DD} < 2.7 V			5	MHz
		f_{PRS} = f_{RH} (XSEL = 0)	2.7 V ≤ V_{DD} ≤ 5.5 V	7.6		8.4	MHz
			1.8 V ≤ V_{DD} < 2.7 V Note 4	7.6		10.4	MHz
External main system clock frequency	f_{EXCLK}		2.7 V ≤ V_{DD} ≤ 5.5 V	1.0 Note 5		20.0	MHz
			1.8 V ≤ V_{DD} < 2.7 V	1.0		5.0	MHz
External main system clock input high-level width, low-level width	t_{EXCLKH}, t_{EXCLKL}		4.0 V ≤ V_{DD} ≤ 5.5 V	24			ns
			2.7 V ≤ V_{DD} < 4.0 V	24			ns
			1.8 V ≤ V_{DD} < 2.7 V	96			ns
TI000, TI010, input high-level width, low-level width	t_{TIH0}, t_{TIL0}	4.0 V ≤ V_{DD} ≤ 5.5 V		$2/f_{sam}$ + 0.1 Note 6			μs
		2.7 V ≤ V_{DD} < 4.0 V		$2/f_{sam}$ + 0.2 Note 6			μs
		1.8 V ≤ V_{DD} < 2.7 V		$2/f_{sam}$ + 0.5 Note 6			μs
TI50, TI51 input frequency	f_{TI5}	4.0 V ≤ V_{DD} ≤ 5.5 V				10	MHz
		2.7 V ≤ V_{DD} < 4.0 V				10	MHz
		1.8 V ≤ V_{DD} < 2.7 V				5	MHz
TI50, TI51 input high-level width, low-level width	t_{TIH5}, t_{TIL5}	4.0 V ≤ V_{DD} ≤ 5.5 V		50			ns
		2.7 V ≤ V_{DD} < 4.0 V		50			ns
		1.8 V ≤ V_{DD} < 2.7 V		100			ns
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

Notes 1. 0.38 μs when operating with the 8 MHz internal oscillator.

2. The 78K0/KB2-A is not provided with a subsystem clock.

3. Characteristics of the main system clock frequency. Set the division clock to be set by a peripheral function to $f_{XH}/2$ (10 MHz) or less. The multiplier/divider, however, can operate on f_{XH} (20 MHz).

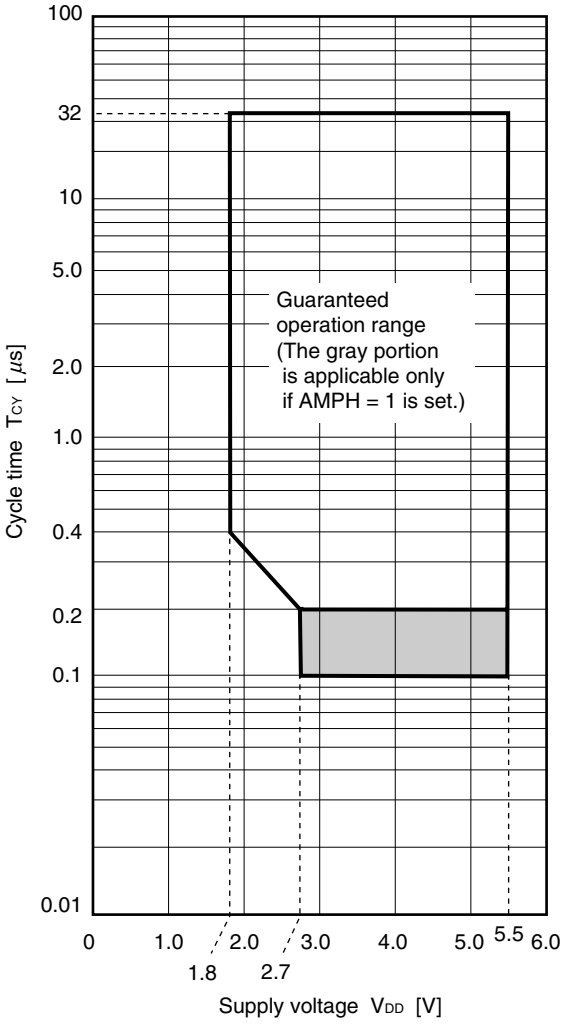
4. Characteristics of the main system clock frequency. Set the division clock to be set by a peripheral function to $f_{RH}/2$ or less.

5. 2.0 MHz (MIN.) when using UART6 during on-board programming.

6. Selection of f_{sam} = f_{PRS}, $f_{PRS}/4$, $f_{PRS}/256$ is possible using bits 0 and 1 (PRM000, PRM001) of prescaler mode registers 00 (PRM00). Note that when selecting the TI000 or TI001 valid edge as the count clock, f_{sam} = f_{PRS}.

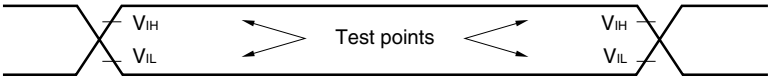
Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

T_{CY} vs. V_{DD} (Main System Clock Operation)

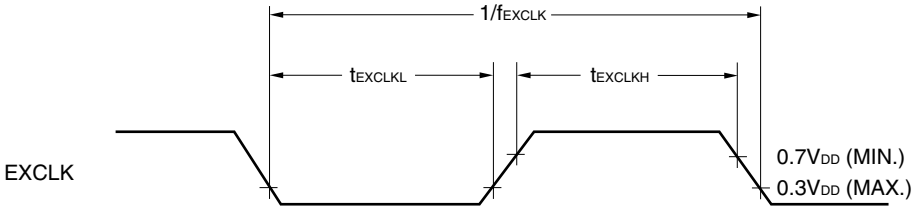


Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

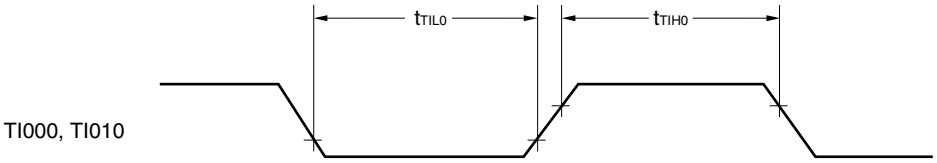
AC Timing Test Points

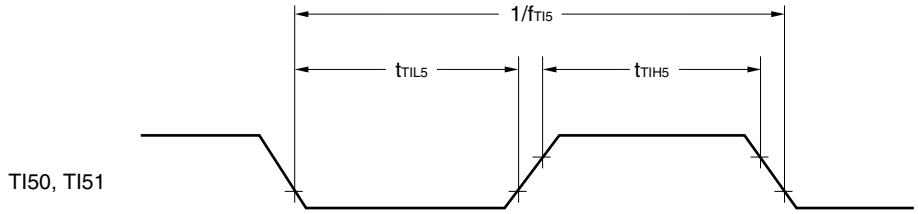


External Main System Clock Timing

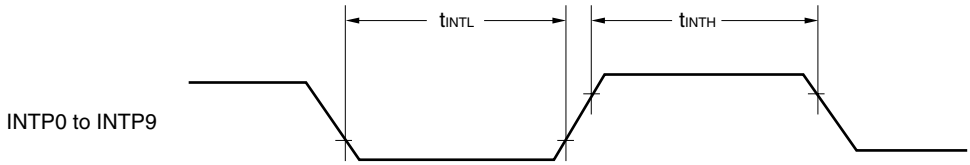


TI Timing



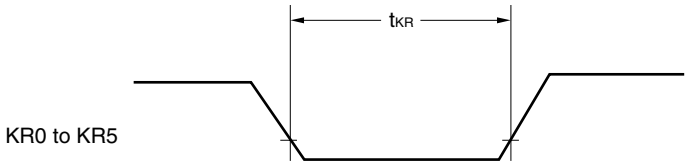


Interrupt Request Input Timing

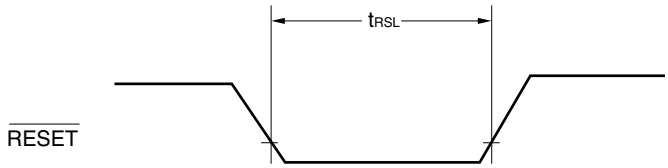


Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

Key Interrupt Input Timing



$\overline{\text{RESET}}$ Input Timing



Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

(2) Serial interface

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

(a) UART6 (dedicated baud rate generator output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					625	kbps

(b) IICA

Parameter	Symbol	Conditions	Standard Mode		Fast Mode		Unit
			MIN.	MAX.	MIN.	MAX.	
SCLA0 clock frequency	f_{SCL}	Fast mode: f_{PRS} ≥ 3.5 MHz, Normal mode: f_{PRS} ≥ 1 MHz	0	100	0	400	kHz
Setup time of start condition and stop condition	$t_{SU: STA}$		4.7	–	0.6	–	μs
Hold time[Note 1]	$t_{HD: STA}$		4.0	–	0.6	–	μs
Hold time when SCLA0 = "L"	t_{LOW}		4.7	–	1.3	–	μs
Hold time when SCLA0 = "H"	t_{HIGH}		4.0	–	0.6	–	μs
Data setup time (reception)	$t_{SU: DAT}$		250	–	100	–	ns
Data hold time (transmission)[Notes 2, 3]	$t_{HD: DAT}$		0	3.45	0	0.9	μs
Setup time of stop condition	$t_{SU: STO}$		4.0	–	0.6	–	μs
Bus free time between stop condition and start condition	t_{BUF}		4.7	–	1.3	–	μs
Rise time of SDAA0 and SCLA0 signals	t_R			1000	2.0 + 0.1C_b	300	ns
Fall time of SDAA0 and SCLA0 signals	t_F			300	2.0 + 0.1 C_b	300	ns
Total load capacitance value of each communication line (SCLA0, SDAA0)	C_b			400		400	pF

Notes 1. The first clock pulse is generated after this period when the start condition or 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. The data hold time differs depending on the setting of the IICA low-level width setting register (IICWL).

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

(c) CSI10 (master mode, $\overline{SCK10}$... internal clock output)[Note 1]

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{SCK10}$ cycle time	t_{KCY1}	4.0 V ≤ V_{DD} ≤ 5.5 V	160			ns
		2.7 V ≤ V_{DD} < 4.0 V	250			ns
		1.8 V ≤ V_{DD} < 2.7 V	500			ns
$\overline{SCK10}$ high-/low-level width	t_{KH1}, t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V	$t_{KCY1}/2 - 15$[Note 2]			ns
		2.7 V ≤ V_{DD} < 4.0 V	$t_{KCY1}/2 - 25$[Note 2]			ns
		1.8 V ≤ V_{DD} < 2.7 V	$t_{KCY1}/2 - 50$[Note 2]			ns
SI10 setup time (to $\overline{SCK10}$↑)	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V	55			ns
		2.7 V ≤ V_{DD} < 4.0 V	80			ns
		1.8 V ≤ V_{DD} < 2.7 V	170			ns
SI10 hold time (from $\overline{SCK10}$↑)	t_{KSI1}		30			ns
Delay time from $\overline{SCK10}$↓ to SO10 output	t_{KSO1}	C = 50 pF[Note 3]			40	ns

Notes 1. The master mode can be used only when bit 2 (ISC2) of the input switch control register (ISC) is set to 1.

2. This value is when high-speed system clock (f_{XH}) is used.

3. C is the load capacitance of the $\overline{SCK10}$ and SO10 output lines.

(d) CSI10 (slave mode, $\overline{SCK10}$... external clock input)

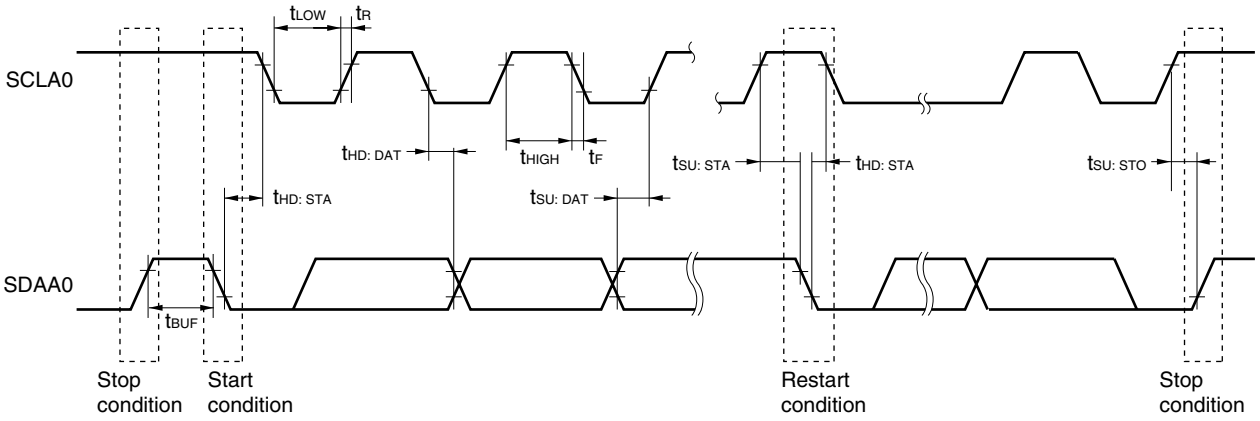
Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{SCK10}$ cycle time	t_{KCY2}			400			ns
$\overline{SCK10}$ high-/low-level width	t_{KH2}, t_{KL2}			$t_{KCY2}/2$			ns
SI10 setup time (to $\overline{SCK10}$↑)	t_{SIK2}			80			ns
SI10 hold time (from $\overline{SCK10}$↑)	t_{KSI2}			50			ns
Delay time from $\overline{SCK10}$↓ to SO10 output	t_{KSO2}	C = 50 pF[Note]	4.0 V ≤ V_{DD} ≤ 5.5 V			120	ns
			2.7 V ≤ V_{DD} < 4.0 V			120	ns
			1.8 V ≤ V_{DD} < 2.7 V			165	ns

Note C is the load capacitance of the SO10 output line.

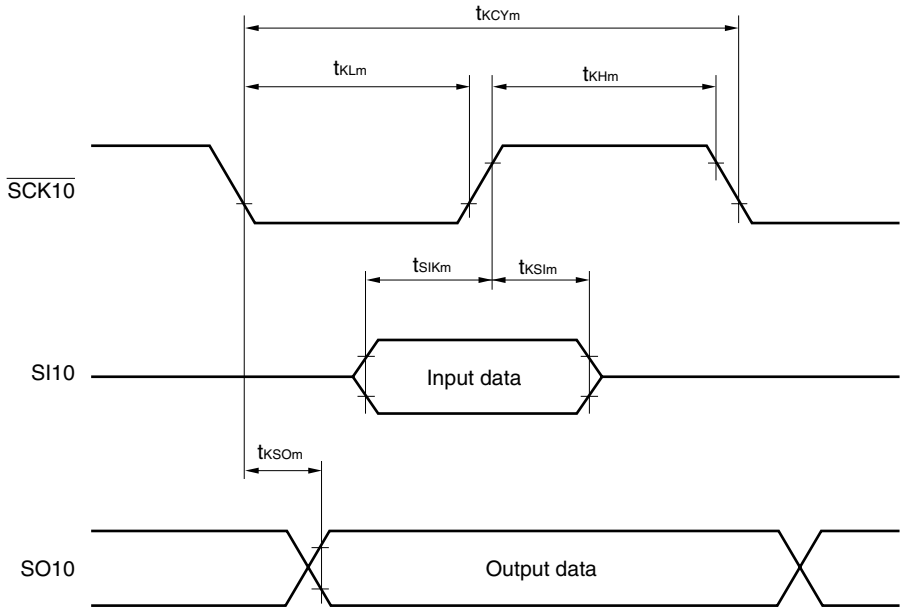
Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

Serial Transfer Timing

IICA:



CSI10:



Remark m = 1, 2

Analog Characteristics

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

(1) A/D Converter (1/2)

(a) T_A = –40 to +85°C, 2.3 V ≤ AV_{REFP} ≤ AV_{DD}, 2.3 V ≤ AV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = AV_{REFM} = 0 V

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	R_{ES}		12	12	12	bit
Overall error[Note 1]	A_{INL}			±2.0	±8.0	LSB
Conversion time	t_{CONV}	Normal mode 1, 2[Note 2]	5			μs
		Low-voltage mode[Note 2]	6.25			μs
Zero-scale error[Note 1]	E_{ZS}			±2.0	±6.0	LSB
Full-scale error[Note 1]	E_{FS}			±2.0	±6.0	LSB
Integral non-linearity error[Note 1]	I_{LE}				±4.0	LSB
Differential non-linearity error[Note 1]	D_{LE}			±1.0	±2.0	LSB
Analog input voltage	V_{AIN}	In 78K0/KB2-A	0		AV_{REF}	V
		When using AV_{REFM} in 78K0/KC2-A	AV_{REFM}		AV_{REFP}	V
Reference supply current	I_{REF}			46	220	μA

(b) T_A = –40 to +85°C, 1.8 V ≤ AV_{REFP} ≤ AV_{DD}, 1.8 V ≤ AV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = AV_{REFM} = 0 V

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	R_{ES}		12	12	12	bit
Overall error[Note 1]	A_{INL}	2.3 V ≤ AV_{REFP} ≤ 5.5 V		±2.0	±12	LSB
		1.8 V ≤ AV_{REFP} < 2.3 V		±3.0	±12	LSB
Conversion time	t_{CONV}	Normal mode 1, 2[Note 2]	5			μs
		Low-voltage mode[Note 2]	21			μs
Zero-scale error[Note 1]	E_{ZS}			±2.0	±10.0	LSB
Full-scale error[Note 1]	E_{FS}			±2.0	±10.0	LSB
Integral non-linearity error[Note 1]	I_{LE}				±8.0	LSB
Differential non-linearity error[Note 1]	D_{LE}				±3.0	LSB
Analog input voltage	V_{AIN}	In 78K0/KB2-A	0		AV_{REF}	V
		When using AV_{REFM} in 78K0/KC2-A	AV_{REFM}		AV_{REFP}	V
Reference supply current	I_{REF}			46	220	μA

Notes 1. Excludes quantization error (±1/2 LSB).

2. The voltage range that can be used on each mode is as follows.

Normal mode 1 : 2.7 V ≤ AV_{DD} ≤ 5.5 V

Normal mode 2 : 2.3 V ≤ AV_{DD} ≤ 5.5 V

Low-voltage mode : 1.8 V ≤ AV_{DD} ≤ 5.5 V

When using the A/D converter in Normal mode 2 or low-voltage mode, be sure to enable the A/D converter voltage booster (by setting VRGV to 1).

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

(1) A/D Converter (2/2)

(c) T_A = 0 to +50°C, 1.8 V ≤ AV_{REFP} ≤ AV_{DD}, 2.3 V ≤ AV_{DD} ≤ V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = AV_{REFM} = 0 V

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	R_{ES}		12	12	12	bit
Overall error[Note 1]	A_{INL}	2.3 V ≤ AV_{REFP} ≤ 3.6 V		±2.0	±8.0	LSB
		1.8 V ≤ AV_{REFP} < 2.3 V		±3.0	±8.0	LSB
Conversion time	t_{CONV}	Normal mode 1, 2[Note 2]	5			μs
		Low-voltage mode[Note 2]	6.25			μs
Zero-scale error[Note 1]	E_{ZS}			±2.0	±6.0	LSB
Full-scale error[Note 1]	E_{FS}			±2.0	±6.0	LSB
Integral non-linearity error[Note 1]	I_{LE}				±4.0	LSB
Differential non-linearity error[Note 1]	D_{LE}			±1.0	±2.0	LSB
Analog input voltage	V_{AIN}	In 78K0/KB2-A	0		AV_{REF}	V
		When using AV_{REFM} in 78K0/KC2-A	AV_{REFM}		AV_{REFP}	V
Reference supply current	I_{REF}			46	200	μA

Notes 1. Excludes quantization error (±1/2 LSB).

2. The voltage range that can be used on each mode is as follows.

Normal mode 1 : 2.7 V ≤ AV_{DD} ≤ 5.5 V

Normal mode 2 : 2.3 V ≤ AV_{DD} ≤ 5.5 V

Low-voltage mode : 1.8 V ≤ AV_{DD} ≤ 5.5 V

When using the A/D converter in Normal mode 2 or low-voltage mode, be sure to enable the A/D converter voltage booster (by setting VRGV to 1).

(2) Operational amplifier

(T_A = –40 to +85°C, 2.0 V ≤ AV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Common-mode input voltage	V_{IAMP}	AV_{DD} = 3.0 V		0		AV_{DD0} – 0.6	V
Input offset voltage	V_{IOAMP}					±10	mV
Maximum output voltage (high)	V_{OHAMP}	AV_{DD} = 3.0 V/2.3 V, I_{SOURCE} = – 500 μA		AV_{DD0} – 0.2			V
Maximum output voltage (low)	V_{OLAMP}	AV_{DD} = 3.0 V/2.3 V, I_{SOURCE} = 500 μA				0.1	V
Open-loop gain					100		dB
Slew rate	SR_{AMP}	AV_{DD} = 3.0 V	Mode 1		0.2		V/μs
			Mode 2		0.4		V/μs
			Mode 3		1.4		V/μs
Turn-on time	t_{ONAMP}	AV_{DD} = 3.0 V	Mode 1			60	μs
			Mode 2			30	μs
			Mode 3			10	μs

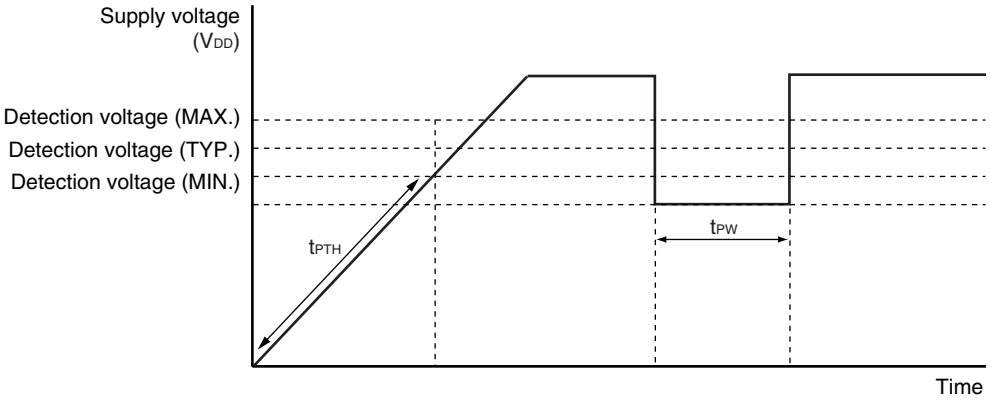
Note Time required until a state is entered where the DC and AC specifications of the operational amplifier 0 are satisfied after the operational amplifier 0 operation has been enabled (OPAMP0E = 1).

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

1.59 V POC Circuit Characteristics (T_A = –40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POC}		1.44	1.59	1.74	V
Power supply voltage rise inclination	t_{PTH}	V_{DD}: 0 V → change inclination of V_{POC}	0.5			V/ms
Minimum pulse width	t_{PW}		200			μs

1.59 V POC Circuit Timing

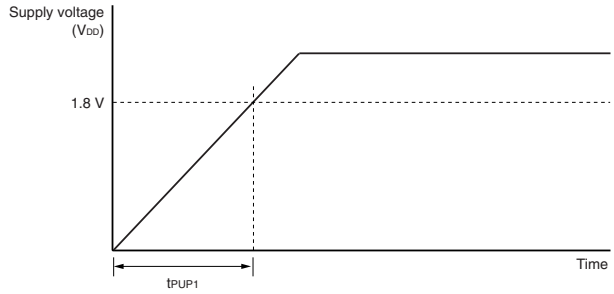


Supply Voltage Rise Time (T_A = –40 to +85°C, V_{SS} = 0 V)

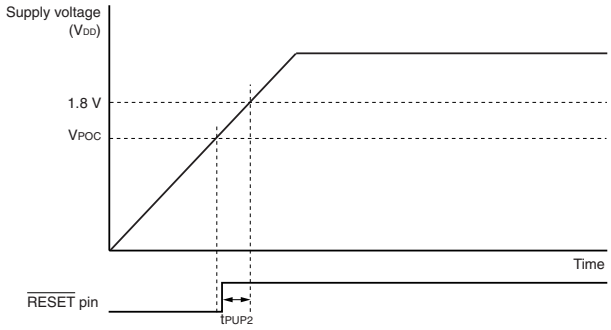
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Maximum time to rise to 1.8 V (V_{DD} (MIN.)) (V_{DD}: 0 V → 1.8 V)	t_{PUP1}	POCMODE (option byte) = 0, when $\overline{\text{RESET}}$ input is not used			3.6	ms
Maximum time to rise to 1.8 V (V_{DD} (MIN.)) (releasing $\overline{\text{RESET}}$ input → V_{DD}: 1.8 V)	t_{PUP2}	POCMODE (option byte) = 0, when $\overline{\text{RESET}}$ input is used			1.9	ms

Supply Voltage Rise Time Timing

- When $\overline{\text{RESET}}$ pin input is not used



- When $\overline{\text{RESET}}$ pin input is used



Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

2.7 V POC Circuit Characteristics (T_A = –40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage on application of supply voltage	V_{DDPOC}	POCMODE (option bye) = 1	2.50	2.70	2.90	V

Remark The operations of the POC circuit are as described below, depending on the POCMODE (option byte) setting.

Option Byte Setting	POC Mode	Operation
POCMODE = 0	1.59 V mode operation	A reset state is retained until V_{POC} = 1.59 V (TYP.) is reached after the power is turned on, and the reset is released when V_{POC} is exceeded. After that, POC detection is performed at V_{POC}, similarly as when the power was turned on. The power supply voltage must be raised at a time of t_{PUP1} or t_{PUP2} when POCMODE is 0.
POCMODE = 1	2.7 V/1.59 V mode operation	A reset state is retained until V_{DDPOC} = 2.7 V (TYP.) is reached after the power is turned on, and the reset is released when V_{DDPOC} is exceeded. After that, POC detection is performed at V_{POC} = 1.59 V (TYP.) and not at V_{DDPOC}. The use of the 2.7 V/1.59 V POC mode is recommended when the rise of the voltage, after the power is turned on and until the voltage reaches 1.8 V, is more relaxed than t_{PTH}.

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

LVI Circuit Characteristics (T_A = –40 to +85°C, $V_{POC} \le V_{DD} \le 5.5$ V, $AV_{REFP} \le AV_{DD} \le V_{DD}$, V_{SS} = 0 V)

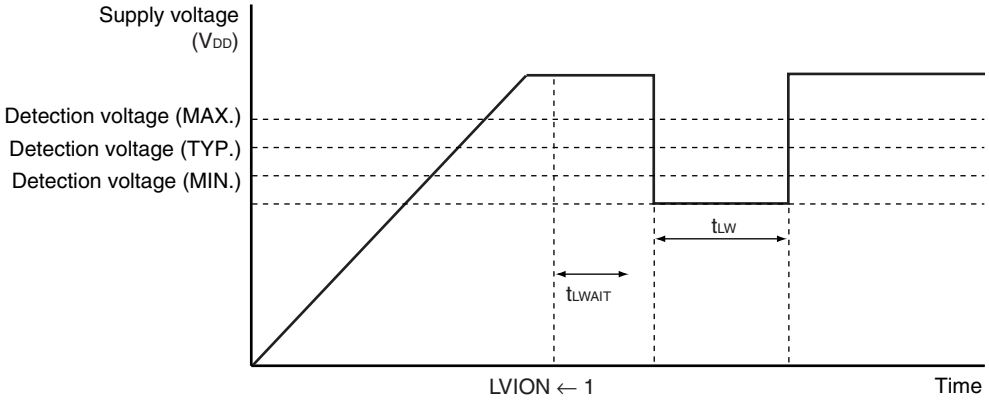
Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Supply voltage level	V_{LVI0}		4.14	4.24	4.34	V
		V_{LVI1}		3.99	4.09	4.19	V
		V_{LVI2}		3.83	3.93	4.03	V
		V_{LVI3}		3.68	3.78	3.88	V
		V_{LVI4}		3.52	3.62	3.72	V
		V_{LVI5}		3.37	3.47	3.57	V
		V_{LVI6}		3.22	3.32	3.42	V
		V_{LVI7}		3.06	3.16	3.26	V
		V_{LVI8}		2.91	3.01	3.11	V
		V_{LVI9}		2.75	2.85	2.95	V
		V_{LVI10}		2.60	2.70	2.80	V
		V_{LVI11}		2.45	2.55	2.65	V
		V_{LVI12}		2.29	2.39	2.49	V
		V_{LVI13}		2.14	2.24	2.34	V
		V_{LVI14}		1.98	2.08	2.18	V
		V_{LVI15}		1.83	1.93	2.03	V
	External input pin[Note 1]	EXLVI	EXLVI < V_{DD}, 1.8 V ≤ V_{DD} ≤ 5.5 V	1.11	1.21	1.31	V
Minimum pulse width		t_{LW}		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. Program the software to make the system wait for this time or longer.

Remark $V_{LVI(n-1)} > V_{LVIn}$: n = 1 to 15

LVI Circuit Timing

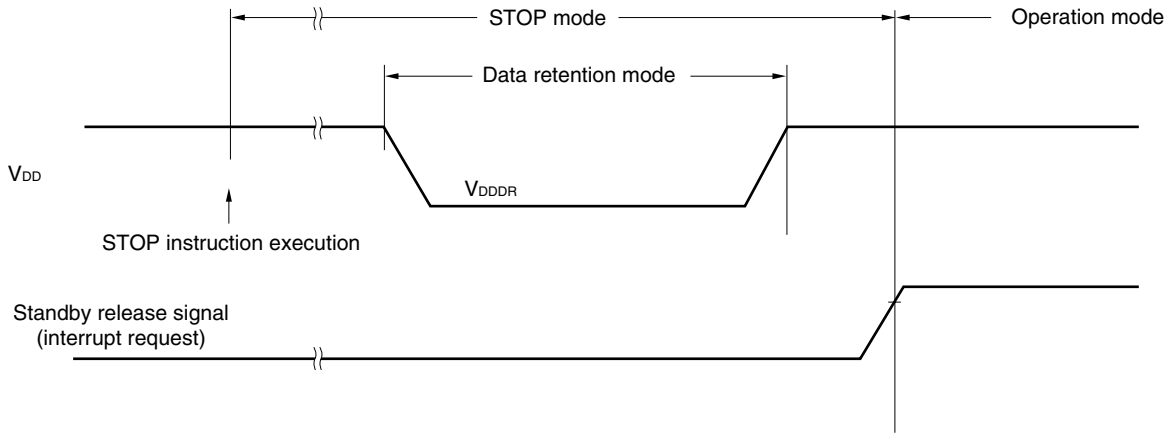


Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

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}		1.44[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.



Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

Flash Memory Programming Characteristics

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, AV_{REFP} ≤ AV_{DD} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

• Basic characteristics

Parameter		Symbol	Conditions			MIN.	TYP.	MAX.	Unit
V_{DD} supply current		I_{DD}	f_{XP} = 10 MHz (TYP.), 20 MHz (MAX.)				4.5	11.0	mA
Erase time Notes 1, 2	All block	T_{eraca}					20	200	ms
	Block unit	T_{erasa}					20	200	ms
Write time (in 8-bit units) Note 1		T_{wrwa}					10	100	μs
Number of rewrites per chip		C_{erwr}	1 erase + 1 write after erase = 1 rewrite Note 3	• When a flash memory programmer is used, and the libraries Note 4 provided by NEC Electronics are used • For program update	Retention: 15 years	1000			Times
				• When the EEPROM emulation libraries Note 5 provided by NEC Electronics are used • The rewritable ROM size: 4 KB • For data update	Retention: 5 years	10000			Times
				Conditions other than the above Note 6	Retention: 10 years	100			Times

Notes 1. Characteristic of the flash memory.

2. The prewrite time before erasure and the erase verify time (writeback time) are not included.

3. When a product is first written after shipment, “erase → write” and “write only” are both taken as one rewrite.

4. The sample library specified by the **78K0/Kx2 Flash Memory Self Programming User’s Manual** (Document No.: **U17516E**) is excluded.

5. The sample program specified by the **78K0/Kx2 EEPROM Emulation Application Note** (Document No.: **U17517E**) is excluded.

6. These include when the sample library specified by the **78K0/Kx2 Flash Memory Self Programming User’s Manual** (Document No.: **U17516E**) and the sample program specified by **the 78K0/Kx2 EEPROM Emulation Application Note** (Document No.: **U17517E**) are used.

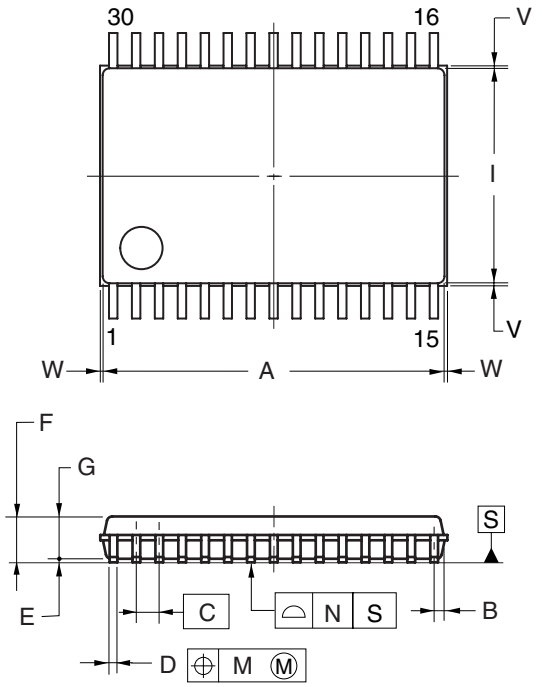
Remarks 1. f_{XP}: Main system clock oscillation frequency

2. For serial write operation characteristics, refer to **78K0/Kx2 Flash Memory Programming (Programmer) Application Note (**Document No.: **U17739E)**.

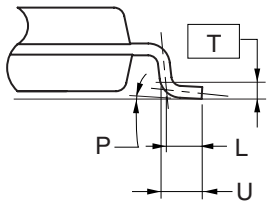
29.1 78K0/KB2-A

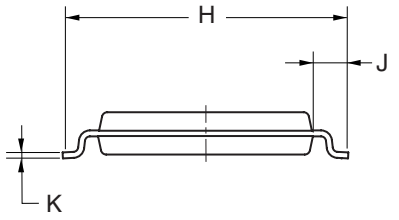
• μPD78F0590MC-CAB-AX, 78F0591MC-CAB-AX

30-PIN PLASTIC SSOP (7.62mm (300))



detail of lead end





(UNIT:mm)

ITEM	DIMENSIONS
A	9.70±0.10
B	0.30
C	0.65 (T.P.)
D	$0.22^{+0.10}_{-0.05}$
E	0.10±0.05
F	1.30±0.10
G	1.20
H	8.10±0.20
I	6.10±0.10
J	1.00±0.20
K	$0.15^{+0.05}_{-0.01}$
L	0.50
M	0.13
N	0.10
P	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
T	0.25(T.P.)
U	0.60±0.15
V	0.25 MAX.
W	0.15 MAX.

P30MC-65-CAB

NOTE

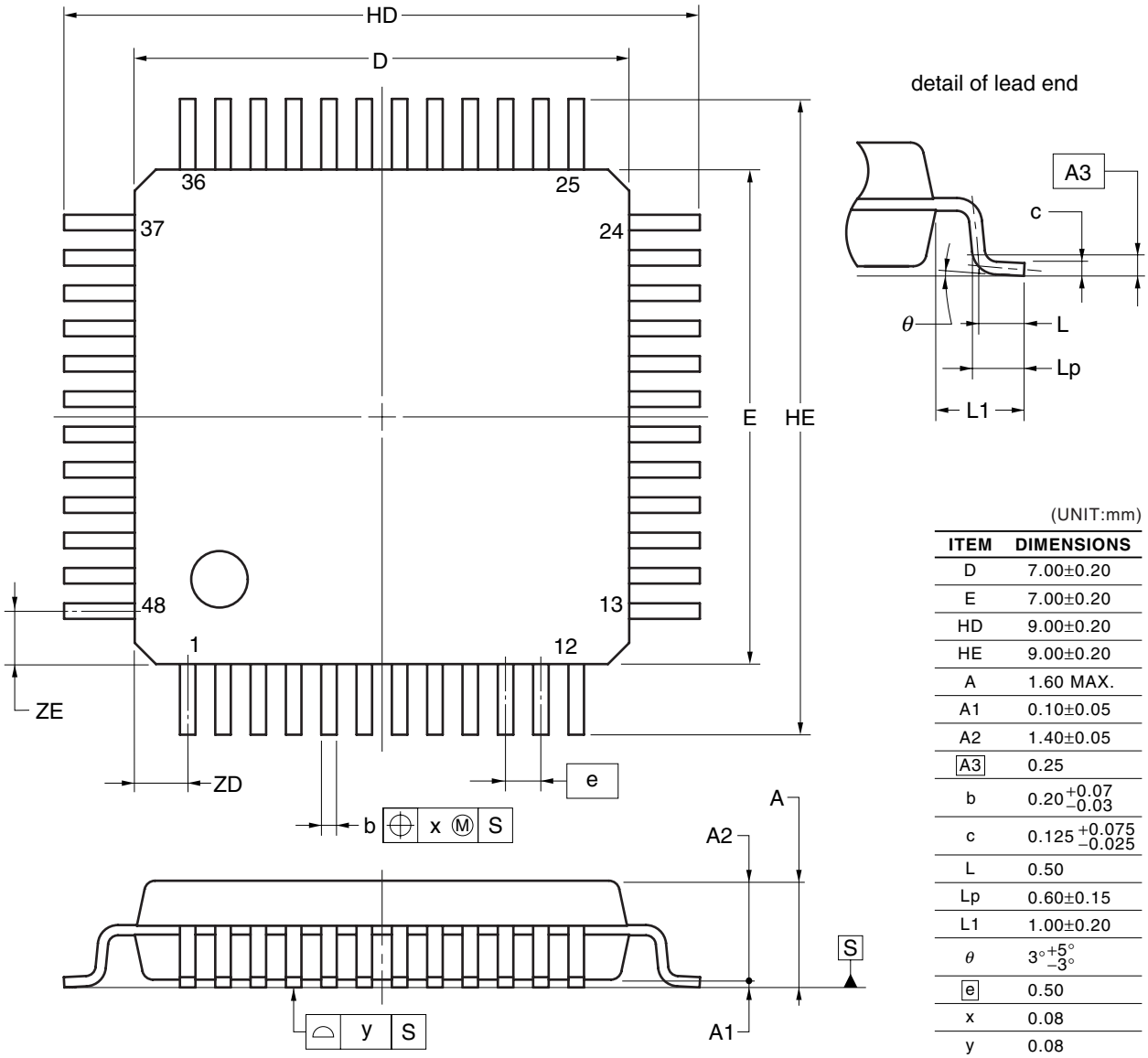
Each lead centerline is located within 0.13 mm of its true position (T.P.) at maximum material condition.

© NEC Electronics Corporation 2005

29.2 78K0/KC2-A

• μPD78F0592GA-GAM-AX, 78F0593GA-GAM-AX

48-PIN PLASTIC LQFP (FINE PITCH) (7x7)



(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	1.60 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	0.75
ZE	0.75

P48GA-50-GAM

NOTE
Each lead centerline is located within 0.08 mm of its true position at maximum material condition.

© NEC Electronics Corporation 2005

CHAPTER 30 CAUTIONS FOR WAIT

30.1 Cautions for Wait

This product has two internal system buses.

One is a CPU bus and the other is a peripheral bus that interfaces with the low-speed peripheral hardware.

Because the clock of the CPU bus and the clock of the peripheral bus are asynchronous, unexpected illegal data may be passed if an access to the CPU conflicts with an access to the peripheral hardware.

When accessing the peripheral hardware that may cause a conflict, therefore, the CPU repeatedly executes processing, until the correct data is passed.

As a result, the CPU does not start the next instruction processing but waits. If this happens, the number of execution clocks of an instruction increases by the number of wait clocks (for the number of wait clocks, see **Table 30-1**). This must be noted when real-time processing is performed.

30.2 Peripheral Hardware That Generates Wait

Table 30-1 lists the registers that issue a wait request when accessed by the CPU, and the number of CPU wait clocks.

Table 30-1. Registers That Generate Wait and Number of CPU Wait Clocks

Peripheral Hardware	Register	Access	Number of Wait Clocks
Serial interface UART6	ASIS6	Read	1 clock (fixed)
Serial interface IICA	IICAS0	Read	1 clock (fixed)

Caution When the peripheral hardware clock (f_{PRS}) is stopped, do not access the registers listed above using an access method in which a wait request is issued.

Remark The clock is the CPU clock (f_{CPU}).

Published by: NEC Electronics Corporation (http://www.necel.com/)
Contact: http://www.necel.com/support/