RENESAS

User's Manual

μPD780833Y Subseries

8-Bit Single-Chip Microcontrollers

μPD780833Y
μPD78F0833Y

Document No. U13892EJ2V0UM00 (2nd edition)
Date Published June 2001 N CP(K)

[MEMO]

NOTES FOR CMOS DEVICES

① **PRECAUTION AGAINST ESD FOR SEMICONDUCTORS**

Note:

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 once, when it has occurred. Environmental control must be adequate. When it is dry, humidifier should be used. It is recommended to avoid using insulators that easily build 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 bench and floor should be grounded. The operator should be grounded using wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with semiconductor devices on it.

② **HANDLING OF UNUSED INPUT PINS FOR CMOS**

Note:

No connection for CMOS device inputs can be cause of malfunction. If no connection is provided to the input pins, it is possible that an internal input level may be generated due to noise, etc., hence causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using a pull-up or pull-down circuitry. Each unused pin should be connected to V_{DD} or GND with a resistor, if it is considered to have a possibility of being an output pin. All handling related to the unused pins must be judged device by device and related specifications governing the devices.

③ **STATUS BEFORE INITIALIZATION OF MOS DEVICES**

Note:

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

The export of these products from Japan is regulated by the Japanese government. The export of some or all of these products may be prohibited without governmental license. To export or re-export some or all of these products from a country other than Japan may also be prohibited without a license from that country. Please call an NEC sales representative.

License not needed:	μPD78F0833YGC-8BT
The customer must judge the need for license:	μPD780833YGC-×××-8BT

The application circuits and their parameters are for reference only and are not intended for use in actual design-ins.

Regional Information

Some information contained in this document may vary from country to country. Before using any NEC product in your application, please contact the NEC office in your country to obtain a list of authorized representatives and distributors. They will verify:

- Device availability
- Ordering information
- Product release schedule
- Availability of related technical literature
- Development environment specifications (for example, specifications for third-party tools and components, host computers, power plugs, AC supply voltages, and so forth)
- Network requirements

In addition, trademarks, registered trademarks, export restrictions, and other legal issues may also vary from country to country.

NEC Electronics Inc. (U.S.)
Santa Clara, California
Tel: 408-588-6000
800-366-9782
Fax: 408-588-6130
800-729-9288

NEC Electronics (Germany) GmbH
Duesseldorf, Germany
Tel: 0211-65 03 02
Fax: 0211-65 03 490

NEC Electronics (UK) Ltd.
Milton Keynes, UK
Tel: 01908-691-133
Fax: 01908-670-290

NEC Electronics Italiana s.r.l.
Milano, Italy
Tel: 02-66 75 41
Fax: 02-66 75 42 99

NEC Electronics (Germany) GmbH
Benelux Office
Eindhoven, The Netherlands
Tel: 040-2445845
Fax: 040-2444580

NEC Electronics (France) S.A.
Velizy-Villacoublay, France
Tel: 01-3067-5800
Fax: 01-3067-5899

NEC Electronics (France) S.A.
Madrid Office
Madrid, Spain
Tel: 091-504-2787
Fax: 091-504-2860

NEC Electronics (Germany) GmbH
Scandinavia Office
Taeby, Sweden
Tel: 08-63 80 820
Fax: 08-63 80 388

NEC Electronics Hong Kong Ltd.
Hong Kong
Tel: 2886-9318
Fax: 2886-9022/9044

NEC Electronics Hong Kong Ltd.
Seoul Branch
Seoul, Korea
Tel: 02-528-0303
Fax: 02-528-4411

NEC Electronics Singapore Pte. Ltd.
Novena Square, Singapore
Tel: 253-8311
Fax: 250-3583

NEC Electronics Taiwan Ltd.
Taipei, Taiwan
Tel: 02-2719-2377
Fax: 02-2719-5951

NEC do Brasil S.A.
Electron Devices Division
Guarulhos-SP, Brasil
Tel: 11-6462-6810
Fax: 11-6462-6829

J01.2

Major Revisions in This Edition (1/2)

Page	Description
Throughout	Deletion of the following products. μPD780831Y, μPD780832Y Flashpro II
	CHAPTER 1 OUTLINE
p. 29	Modification of **1.5 78K/0 Series Lineup**
p. 33	Addition of timer overview.
	CHAPTER 2 PIN FUNCTION
p. 42	Modification of **Table 2-1 Types of Pin I/O Circuits and Recommended Connection of Unused Pins**
	CHAPTER 4 PORT FUNCTIONS
p. 86	Addition of **Caution 2** to **4.3 Registers Controlling Port Function**
p. 88	Modification of **Caution 2** in **(2) Pull-up resistor option registers (PU0, PU2 to PU7)**
	CHAPTER 5 CLOCK GENERATOR
p. 92	Addition of oscillation frequency in **5.1 Function of Clock Generator**
	CHAPTER 6 16-BIT TIMER/EVENT COUNTERS 00, 01
p. 102	Modification of **Caution 1** in **(2) 16-bit capture/compare registers 000, 001 (CR000, CR001)**
p. 103	Modification of **Caution** in **(3) 16-bit capture/compare registers 010, 011 (CR010, CR011**)
p. 117	Modification of **Figure 6-19 Timing of Pulse Width Measurement Operation by Free-Running Counter and One Capture Register (with Both Edges Specified)**
p. 119	Modification of **Figure 6-21 CR01n Capture Operation with Rising Edge Specified**
	Modification of **Figure 6-22 Timing of Pulse Width Measurement Operation by Free-Running Counter (with Both Edges Specified)**
p. 121	Modification of **Figure 6-24 Timing of Pulse Width Measurement Operation by Free-Running Counter and Two Capture Registers (with Rising Edge Specified)**
p. 122	Modification of **Figure 6-26 Timing of Pulse Width Measurement Operation by Means of Restart (with Rising Edge Specified)**
p. 126	Addition of **Note** to **6.5.6 One-shot pulse output operation**
p. 127	Modification of **Caution** in **Figure 6-32 Control Register Settings for One-Shot Pulse Output Operation Using Software Trigger**
p. 128	Addition of **Note** to **(2) One-shot pulse output by external trigger**
p. 129	Modification of **Caution** in **Figure 6-34 Control Register Settings for One-Shot Pulse Output Operation Using External Trigger**
	6.6 Notes on 16-Bit Timer/Event Counter
p. 132	Modification of **Figure 6-38 Capture Register Data Retention Timing**
p. 132	Addition of **(c) One-shot pulse output function**
p. 133	Modification of **Figure 6-39 Operation Timing of OVF0n Flag**
	CHAPTER 7 8-BIT TIMER/EVENT COUNTERS 50, 51, 52
p. 135	Addition of **7.1 Outline of 8-Bit Timer/Event Counters 50, 51, 52**
p. 138	Modification of **Caution** in **(2) 8-bit timer compare registers 50, 51, and 52 (CR50, CR51, and CR52)**
p. 140	Addition of **Caution 1** to **Figure 7-4 Format of 8-Bit Timer Mode Control Register 50 (TMC50)**
p. 141	Addition of **Caution** to **Figure 7-5 Format of 8-Bit Timer Mode Control Register 51 (TMC51)**
p. 142	Addition of **Caution 1** to **Figure 7-6 Format of 8-Bit Timer Mode Control Register 52 (TMC52)**
p. 151	Addition of **Caution** to **7.5.4 8-bit PWM output operation**
	CHAPTER 8 WATCH TIMER
p. 156	Addition of **8.1 Outline of Watch Timer**
p. 160	Addition of **Caution** to **Figure 8-3 Operation Timing of Watch Timer/Interval Timer**
	CHAPTER 9 WATCHDOG TIMER
p. 161	Addition of **9.1 Outline of Watchdog Timer**
p. 161	Modification of **Figure 9-1 Block Diagram of Watchdog Timer**
p. 164	Modification of **Caution** in **Figure 9-3 Format of Watchdog Timer Mode Register (WDTM)**
	CHAPTER 10 CLOCK OUTPUT CONTROLLER
p. 168	Addition of **10.1 Outline of Clock Output Controller**

Major Revisions in This Edition (2/2)

Page	Description
	CHAPTER 11 A/D CONVERTER
p. 183	Addition of **11.5 How to Read A/D Converter Characteristics Table**
pp. 187 to 189	Addition of items (11) to (14) to **11.6 Notes on A/D Converter**
p. 187	Addition of **Figure 11-14 Conversion Result Read Timing (When Conversion Result Is Undefined)**
p. 187	Addition of **Figure 11-15 Conversion Result Read Timing (When Conversion Result Is Normal)**
p. 188	Addition of **Figure 11-16 Example of Capacitor Connection to AV_{REF0} Pin**
p. 189	Addition of **Figure 11-17 Internal Equivalence Circuit of ANI00 to ANI70 Pins**
p. 189	Addition of **Table 11-3 Resistance and Capacitance Values for Equivalence Circuits (Reference Values)**
p. 189	Addition of **Figure 11-18 Example of Circuit When Signal Source Impedance Is High**
	CHAPTER 12 J1850 (Class 2)
p. 197	Addition of **Caution 3** and **4** to **Figure 12-4 Format of Class 2 Status Register 1 (C2ST1)**
p. 204	Addition of **Remark 2** to **Figure 12-8 Format of Class 2 Clock Selection Register (C2CLK)**
	12.3.6 Communication control based on CPU
pp. 228, 229	Addition of **Caution** and block diagram for **<2> When CRC is used** to **(2) Reception control (b) Example of reception control in block mode**
p. 230	Addition of **Caution** to **(3) Transmission control (a) Example of transmission control in normal mode**
pp. 231, 232	Modification of a block diagram in **(3) Transmission control (b) Example of transmission/reception control in block mode**
	CHAPTER 13 SERIAL INTERFACE UART0
p. 239	Addition of **Figure 13-2 Block Diagram of Baud Rate Generator**
p. 250	Addition of description on 5-bit counter to baud rate expression
p. 255	Modification of **(c) Transmission**
	CHAPTER 14 SERIAL INTERFACE SIO3
p. 259	Modification of **Figure 14-1 Block Diagram of Serial Interface SIO30**
	CHAPTER 15 SERIAL INTERFACE IIC0
p. 270	Modification of **Figure 15-3 Format of IIC Control Register 0 (IICC0)**
p. 274	Addition of **Note** to **Figure 15-4 Format of IIC Status Register 0 (IICS0)**
p. 277	Addition of **Remark 4** to **Figure 15-5 Format of IIC Transfer Clock Selection Register 0 (IICCL0)**
p. 287	Modification of timing diagram in **15.5.7 I^2C interrupt requests (INTIIC0)**
p. 311	Modification of **Figure 15-19 Master Operation Flow Chart**
p. 312	Modification of **Figure 15-20 Slave Operation Flow Chart**
	CHAPTER 16 INTERRUPT FUNCTIONS
p. 320	Addition of **Remark** to **16.2 Interrupt Sources and Configuration**
p. 321	Modification of **Table 16-1 Interrupt Source List**
p. 324	Change of **Note** in **Table 16-2 Flags Corresponding to Interrupt Request Sources**
p. 325	Addition of **Caution 2** to **Figure 16-2 Format of Interrupt Request Flag Register (IF0L, IF0H, IF1L, IF1H)**
	CHAPTER 18 RESET FUNCTION
p. 350	Modification of **Table 18-1 Hardware Statuses After Reset**
	CHAPTER 19 μPD78F0833Y
p. 352	Modification of **Table 19-1 Differences Between μPD78F0833Y and Mask ROM Versions**
p. 353	Modification of **Figure 19-1 Format of Memory Size Switching Register (IMS)**
p. 354	Modification of **Table 19-2 Communication Mode List**
	APPENDIX A DEVELOPMENT TOOLS
p. 390	Addition of SP78K0

The mark ★ shows major revised points.

INTRODUCTION

Target Readers

This manual is intended for users who wish to understand the functions of the μPD780833Y Subseries and to design and develop application systems and programs using these microcontrollers.
The target devices are the following μPD780833Y Subseries products.

μPD780833Y Subseries: μPD780833Y, 78F0833Y

Purpose

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

Organization

The μPD780833Y Subseries User's Manual is divided into two parts: this manual and the instruction manual (common to the 78K/0 Series).

μPD780833Y Subseries User's Manual (This manual)	78K/0 Series User's Manual Instructions
• Pin functions	• CPU functions
• Internal block functions	• Instruction set
• Interrupt functions	• Explanation of each instruction
• Other on-chip peripheral functions	

How To Read This Manual

It is assumed that the reader of this manual has general knowledge in the field of electrical engineering, logic circuits, and microcontrollers.

- To understand the functions in general:
  $\rightarrow$ Read this manual in the order of the contents.
- How to interpret the register format:
  $\rightarrow$ For a bit whose number is enclosed in a square, its bit name is defined as a reserved word in the RA78K0, and is defined in the header file named sfrbit.hin the CC78K0.
- When you know a register name and want to confirm its details:
  $\rightarrow$ Refer to **APPENDIX C.**

Conventions

Data significance:	Higher digits on the left and lower digits on the right	
Active low representation:	$\overline{\times\times\times}$ (overscore over pin or signal name)	
Note:	Footnote for item marked with **Note** in the text	
Caution:	Information requiring particular attention	
Remark:	Supplementary information	
Numerical representation:	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.
μPD780833Y Data Sheet	U15012E
μPD78F0833Y Data Sheet	U15013E
μPD780833Y Subseries User's Manual	This manual
78K/0 Series User's Manual Instructions	U12326E

- **Documents related to development tools (user's manuals)**

Document Name		Document No.
RA78K0 Assembler Package	Operation	U14445E
	Language	U14446E
	Structured Assembly Language	U11789E
CC78K0 C Compiler	Operation	U14297E
	Language	U14298E
CC78K0 C Compiler Application Note	Programming Know-How	U13034E
IE-78K0-NS		U13731E
IE-78001-R-A		To be prepared
IE-78K0-R-EX1		To be prepared
IE-780833-NS-EM4		To be prepared
SM78K0S, SM78K0 System Simulator Ver. 2.10 or Later Windows Based	Operation	U14611E
SM78K Series System Simulator Ver. 2.10 or Later	External Part User Open Interface Specifications	U15006E
ID78K0-NS Integrated Debugger Ver. 2.00 or Later Windows Based	Operation	U14379E
ID78K0 Integrated Debugger Windows Based	Reference	U11539E
	Guide	U11649E

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

- **Documents related to embedded software (user's manuals)**

Document Name		Document No.
78K/0 Series Real-Time OS	Fundamental	U11537E
	Installation	U11536E
78K/0 Series OS MX78K0	Fundamental	U12257E

- **Other related documents**

Document Name	Document No.
SEMICONDUCTOR SELECTION GUIDE - Products & Packages - (CD-ROM)	X13769E
Semiconductor Device Mounting Technology Manual	C10535E
Quality Guides on NEC Semiconductor Devices	C11531E
NEC Semiconductor Device Reliability and Quality Control	C10983E
Guide to Prevent Damage for Semiconductor Devices by Electrostatic Discharge (ESD)	C11892E

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

CONTENTS

CHAPTER 1 OUTLINE 26

1.1 Features 26
1.2 Applications 26
1.3 Ordering Information 26
1.4 Pin Configuration (Top View) 27
1.5 78K/0 Series Lineup 29
1.6 Block Diagram 31
1.7 Overview of Functions 32

CHAPTER 2 PIN FUNCTION 34

2.1 Pin Function List 34
2.2 Description of Pin Functions 37
2.2.1 P00 to P07 (Port 0) 37
2.2.2 P20 to P27 (Port 2) 37
2.2.3 P30 to P36 (Port 3) 38
2.2.4 P40 to P47 (Port 4) 38
2.2.5 P50 to P57 (Port 5) 38
2.2.6 P64 to P67 (Port 6) 38
2.2.7 P70 to P75 (Port 7) 39
2.2.8 P80 to P87 (Port 8) 39
2.2.9 P90 to P97 (Port 9) 40
2.2.10 C2RX and C2TX 40
2.2.11 AV_{REF0} and AV_{REF1} 40
2.2.12 AV_{DD0} 40
2.2.13 AVSS0 and AVSS1 40
2.2.14 $\overline{\text{RESET}}$ 40
2.2.15 X1 and X2 40
2.2.16 V_{DD0} and V_{DD1} 40
2.2.17 V_{SS0} and V_{SS1} 40
2.2.18 V_{PP} (μPD78F0833Y only) 40
2.2.19 IC (mask ROM version only) 41
2.3 Pin I/O Circuits and Recommended Connection of Unused Pins 42

CHAPTER 3 CPU ARCHITECTURE 46

3.1 Memory Space 46
3.1.1 Internal program memory space 48
3.1.2 Internal data memory space 50
3.1.3 Special Function Register (SFR) area 50
3.1.4 Data memory addressing 51
3.2 Processor Registers 53
3.2.1 Control registers 53
3.2.2 General-purpose registers 56
3.2.3 Special function registers (SFRs) 57

3.3 Instruction Address Addressing **61**
3.3.1 Relative addressing 61
3.3.2 Immediate addressing 62
3.3.3 Table indirect addressing 63
3.3.4 Register addressing 63
3.4 Operand Address Addressing **64**
3.4.1 Implied addressing 64
3.4.2 Register addressing 65
3.4.3 Direct addressing 66
3.4.4 Short direct addressing 67
3.4.5 Special function register (SFR) addressing 68
3.4.6 Register indirect addressing 69
3.4.7 Based addressing 70
3.4.8 Based indexed addressing 71
3.4.9 Stack addressing 71

CHAPTER 4 PORT FUNCTIONS **72**

4.1 Port Functions **72**
4.2 Port Configuration **75**
4.2.1 Port 0 75
4.2.2 Port 2 77
4.2.3 Port 3 78
4.2.4 Port 4 80
4.2.5 Port 5 81
4.2.6 Port 6 82
4.2.7 Port 7 83
4.2.8 Port 8 84
4.2.9 Port 9 85
4.3 Registers Controlling Port Function **86**
4.4 Port Function Operations **91**
4.4.1 Writing to I/O port 91
4.4.2 Reading from I/O port 91
4.4.3 Operations on I/O port 91

CHAPTER 5 CLOCK GENERATOR **92**

5.1 Function of Clock Generator **92**
5.2 Configuration of Clock Generator **92**
5.3 Register Controlling Clock Generator **93**
5.4 System Clock Oscillator **94**
5.4.1 System clock oscillator 94
5.4.2 Frequency divider 95
5.5 Operation of Clock Generator **96**
5.6 Changing CPU Clock Setting **96**
5.6.1 Time required for switchover between CPU clocks 96
5.6.2 CPU clock switching procedure 97

CHAPTER 6 16-BIT TIMER/EVENT COUNTERS 00, 01 98

6.1 Outline of 16-Bit Timer/Event Counters 00, 01 98
6.2 Function of 16-Bit Timer/Event Counters 00, 01 98
6.3 Configuration of 16-Bit Timer/Event Counters 00, 01 99
6.4 Registers Controlling 16-Bit Timer/Event Counter 00, 01 103
6.5 Operation of 16-Bit Timer/Event Counters 00, 01 113
6.5.1 Interval timer operation 113
6.5.2 PPG output operation 115
6.5.3 Pulse width measurement operation 116
6.5.4 External event counter operation 123
6.5.5 Square-wave output operation 124
6.5.6 One-shot pulse output operation 126
6.6 Notes on 16-Bit Timer/Event Counter 131

CHAPTER 7 8-BIT TIMER/EVENT COUNTERS 50, 51, 52 135

7.1 Outline of 8-Bit Timer/Event Counters 50, 51, 52 135
7.2 Function of 8-Bit Timer/Event Counters 50, 51, 52 135
7.3 Configuration of 8-Bit Timer/Event Counters 50, 51, 52 136
7.4 Registers Controlling 8-Bit Timer/Event Counters 50, 51, 52 139
7.5 Operation of 8-Bit Timer/Event Counters 50, 51, 52 146
7.5.1 Interval timer (8-bit) operation 146
7.5.2 External event counter operation 149
7.5.3 Square-wave output (8-bit resolution) operation 150
7.5.4 8-bit PWM output operation 151
7.5.5 Interval timer (16-bit) operation 154
7.6 Notes on 8-Bit Timer/Event Counters 155

CHAPTER 8 WATCH TIMER 156

8.1 Outline of Watch Timer 156
8.2 Function of Watch Timer 156
8.3 Configuration of Watch Timer 157
8.4 Registers Controlling Watch Timer 158
8.5 Operation of Watch Timer 159
8.5.1 Watch timer operation 159
8.5.2 Interval timer operation 159

CHAPTER 9 WATCHDOG TIMER 161

9.1 Outline of Watchdog Timer 161
9.2 Function of Watchdog Timer 161
9.3 Configuration of Watchdog Timer 163
9.4 Registers Controlling Watchdog Timer 163
9.5 Operation of Watchdog Timer 166
9.5.1 Watchdog timer operation 166
9.5.2 Interval timer operation 167

CHAPTER 10 CLOCK OUTPUT CONTROLLER 168

10.1 Outline of Clock Output Controller 168
10.2 Function of Clock Output Controller 168
10.3 Configuration of Clock Output Controller 169
10.4 Registers Controlling Clock Output Controller 169
10.5 Operation of Clock Output Controller 171

CHAPTER 11 A/D CONVERTER 172

11.1 Function of A/D Converter 173
11.2 Configuration of A/D Converter 175
11.3 Registers Controlling A/D Converter 177
11.4 Operation of A/D Converter 179
11.4.1 Basic operations of A/D converter 179
11.4.2 Input voltage and conversion results 181
11.4.3 A/D converter operation mode 182
11.5 How to Read A/D Converter Characteristics Table 183
11.6 Notes on A/D Converter 184

CHAPTER 12 J1850 (Class 2) 190

12.1 Protocol Overview 190
12.2 Class 2 Internal Register Configuration 191
12.3 Operation Timing 205
12.3.1 Reception 205
12.3.2 Transmission 210
12.3.3 Error detection 218
12.3.4 Interrupt operation 220
12.3.5 Propagation delay correction (for correcting the standard transmit time) 222
12.3.6 Communication control based on CPU 226
12.4 Standby Function 233
12.4.1 Sleep state 233
12.4.2 Wake-up state 235
12.4.3 Differences between the wake-up and sleep states 236
12.5 Class 2 Cautions 237

CHAPTER 13 SERIAL INTERFACE UART0 238

13.1 Function of Serial Interface 238
13.2 Configuration of Serial Interface 240
13.3 Registers Controlling Serial Interface 241
13.4 Operation of Serial Interface 245
13.4.1 Operation stop mode 245
13.4.2 Asynchronous serial interface (UART) mode 246

CHAPTER 14 SERIAL INTERFACE SIO3 258

14.1 Function of Serial Interface 259
14.2 Configuration of Serial Interface 260
14.3 Register Controlling Serial Interface 261
14.4 Operation of Serial Interface 262
14.4.1 Operation stop mode 262
14.4.2 3-wire serial I/O mode 263

CHAPTER 15 SERIAL INTERFACE IIC0 265

15.1 Function of Serial Interface 265
15.2 Configuration of Serial Interface 268
15.3 Registers Controlling Serial Interface 269
15.4 Function of I^{2}C Bus Mode 280
15.4.1 Pin configuration 280
15.5 I^{2}C Bus Definitions and Control Methods 281
15.5.1 Start conditions 281
15.5.2 Addresses 282
15.5.3 Transfer direction specification 282
15.5.4 Acknowledge ($\overline{\text{ACK}}$) signal 283
15.5.5 Stop condition 284
15.5.6 Wait signal ($\overline{\text{WAIT}}$) 285
15.5.7 I^{2}C interrupt requests (INTIIC0) 287
15.5.8 Interrupt request (INTIIC0) generation timing and wait control 304
15.5.9 Address match detection method 305
15.5.10 Error detection 305
15.5.11 Extension code 305
15.5.12 Arbitration 306
15.5.13 Wake-up function 307
15.5.14 Communication reservation 308
15.5.15 Other cautions 310
15.5.16 Communication operations 311
15.6 Timing Charts 313

CHAPTER 16 INTERRUPT FUNCTION 320

16.1 Interrupt Types 320
16.2 Interrupt Sources and Configuration 320
16.3 Interrupt Function Control Registers 324
16.4 Interrupt Servicing Operations 330
16.4.1 Non-maskable interrupt request acknowledgement operation 330
16.4.2 Maskable interrupt acknowledge operation 333
16.4.3 Software interrupt request acknowledgement operation 335
16.4.4 Multiple interrupt servicing 336
16.4.5 Interrupt request hold 339

CHAPTER 17 STANDBY FUNCTION **340**

17.1 Standby Function and Configuration **340**
17.1.1 Standby function 340
17.1.2 Standby function control register 341
17.2 Standby Function Operations **342**
17.2.1 HALT mode 342
17.2.2 STOP mode 345

CHAPTER 18 RESET FUNCTION **348**

18.1 Reset Function **348**

CHAPTER 19 μPD78F0833Y **352**

19.1 Memory Size Switching Register **353**
19.2 Internal Expansion RAM Size Switching Register **353**
19.3 Flash Memory Programming **354**
19.3.1 Selection of communication mode 354
19.3.2 Flash memory programming function 355
19.3.3 Flashpro III connection 355
19.4 Flash Memory Programming by Self-Write **356**
19.4.1 Flash memory configuration 356
19.4.2 Flash programming mode control register (FLPMC) 357
19.4.3 Self-write procedure 357
19.4.4 CPU resources 360
19.4.5 Entry ram area 360
19.4.6 Self-write subroutines 362
19.4.7 Self-write circuit configuration 372

CHAPTER 20 INSTRUCTION SET **373**

20.1 Conventions **374**
20.1.1 Operand identifiers and specification methods 374
20.1.2 Description of "operation" column 375
20.1.3 Description of "flag operation" column 375
20.2 Operation List **376**
20.3 Instructions Listed by Addressing Type **384**

APPENDIX A DEVELOPMENT TOOLS **387**

A.1 Language Processing Software **390**
A.2 Flash Memory Writing Tools **391**
A.3 Debugging Tools **392**
A.3.1 Hardware 392
A.3.2 Software 394
A.4 System Upgrade from Former In-Circuit Emulator for 78K/0 Series to IE-78001-R-A **395**

APPENDIX B EMBEDDED SOFTWARE 398

APPENDIX C REGISTER INDEX 400

C.1 Register Index (In Alphabetical Order with Respect to Register Names) 400
C.2 Register Index (In Alphabetical Order with Respect to Register Symbol) 403

★ APPENDIX D REVISION HISTORY 406

LIST OF FIGURES (1/6)

Figure No.	Title	Page
2-1	Pin I/O Circuits	44
3-1	Memory Map (μPD780833Y)	46
3-2	Memory Map (μPD78F0833Y)	47
3-3	Data Memory Addressing (μPD780833Y)	51
3-4	Data Memory Addressing (μPD78F0833Y)	52
3-5	Format of Program Counter	53
3-6	Format of Program Status Word	53
3-7	Stack Pointer Format	55
3-8	Data Saved to Stack Memory	55
3-9	Data Restored from Stack Memory	55
3-10	General-Purpose Register Configuration	56
4-1	Port Types	72
4-2	Block Diagram of P00 to P07	76
4-3	Block Diagram of P20 to P27	77
4-4	Block Diagram of P30 to P32 and P34 to P36	78
4-5	Block Diagram of P33	79
4-6	Block Diagram of P40 to P47	80
4-7	Block Diagram of Falling Edge Detector	80
4-8	Block Diagram of P50 to P57	81
4-9	Block Diagram of P64 to P67	82
4-10	Block Diagram of P70 to P75	83
4-11	Block Diagram of P80 to P87	84
4-12	Block Diagram of P90 to P97	85
4-13	Format of Port Mode Registers (PM0, PM2 to PM9)	87
4-14	Format of Pull-Up Resistor Option Registers (PU0, PU2 to PU7)	89
4-15	Format of Memory Expansion Mode Register (MEM)	90
5-1	Block Diagram of Clock Generator	92
5-2	Format of Processor Clock Control Register (PCC)	93
5-3	External Circuit of System Clock Oscillator	94
5-4	Examples of Incorrect Oscillator Connection	94
5-5	CPU Clock Switching	97
6-1	Block Diagram of 16-Bit Timer/Event Counter 00	99
6-2	Block Diagram of 16-Bit Timer/Event Counter 01	100
6-3	Format of 16-Bit Timer Mode Control Register 00 (TMC00)	104
6-4	Format of 16-Bit Timer Mode Control Register 01 (TMC01)	105
6-5	Format of Capture/Compare Control Register 00 (CRC00)	106
6-6	Format of Capture/Compare Control Register 01 (CRC01)	107
6-7	Format of 16-Bit Timer Output Control Register 00 (TOC00)	108
6-8	Format of 16-Bit Timer Output Control Register 01 (TOC01)	109
6-9	Format of Prescaler Mode Register 00 (PRM00)	110

LIST OF FIGURES (2/6)

Figure No.	Title	Page
6-10	Format of Prescaler Mode Register 01 (PRM01)	111
6-11	Format of Port Mode Register 3 (PM3)	112
6-12	Format of Port Mode Register 7 (PM7)	112
6-13	Control Register Settings for Interval Timer Operation	113
6-14	Configuration Diagram of Interval Timer	114
6-15	Timing of Interval Timer Operation	114
6-16	Control Register Settings for PPG Output Operation	115
6-17	Control Register Settings for Pulse Width Measurement with Free-Running Counter and One Capture Register	116
6-18	Configuration Diagram for Pulse Width Measurement by Free-Running Counter	117
6-19	Timing of Pulse Width Measurement Operation by Free-Running Counter and One Capture Register (with Both Edges Specified)	117
6-20	Control Register Settings for Measurement of Two Pulse Widths with Free-Running Counter	118
6-21	CR01n Capture Operation with Rising Edge Specified	119
6-22	Timing of Pulse Width Measurement Operation by Free-Running Counter (with Both Edges Specified)	119
6-23	Control Register Settings for Pulse Width Measurement with Free-Running Counter and Two Capture Registers	120
6-24	Timing of Pulse Width Measurement Operation by Free-Running Counter and Two Capture Registers (with Rising Edge Specified)	121
6-25	Control Register Settings for Pulse Width Measurement by Means of Restart	122
6-26	Timing of Pulse Width Measurement Operation by Means of Restart (with Rising Edge Specified)	122
6-27	Control Register Settings in External Event Counter Mode	123
6-28	Configuration Diagram of External Event Counter	124
6-29	External Event Counter Operation Timing (with Rising Edge Specified)	124
6-30	Control Register Settings in Square-Wave Output Mode	125
6-31	Square-Wave Output Operation Timing	125
6-32	Control Register Settings for One-Shot Pulse Output Operation Using Software Trigger	127
6-33	Timing of One-Shot Pulse Output Operation Using Software Trigger	128
6-34	Control Register Settings for One-Shot Pulse Output Operation Using External Trigger	129
6-35	Timing of One-Shot Pulse Output Operation Using External Trigger (Clear and Start by Valid Edge of TI00n, When Rising Edge Is Specified)	130
6-36	Start Timing for 16-Bit Timer Counter 0n	131
6-37	Timing After Change of Compare Register During Timer Count Operation	131
6-38	Capture Register Data Retention Timing	132
6-39	Operation Timing of OVF0n Flag	133
7-1	Block Diagram of 8-Bit Timer/Event Counter 50	136
7-2	Block Diagram of 8-Bit Timer/Event Counter 51	137
7-3	Block Diagram of 8-Bit Timer/Event Counter 52	137
7-4	Format of 8-Bit Timer Mode Control Register 50 (TMC50)	140
7-5	Format of 8-Bit Timer Mode Control Register 51 (TMC51)	141
7-6	Format of 8-Bit Timer Mode Control Register 52 (TMC52)	142

LIST OF FIGURES (3/6)

Figure No.	Title	Page
7-7	Format of Timer Clock Selection Register 50 (TCL50)	143
7-8	Format of Timer Clock Selection Register 51 (TLC51)	144
7-9	Format of Timer Clock Selection Register 52 (TCL52)	144
7-10	Format of Port Mode Register 2 (PM2)	145
7-11	Interval Timer Operation Timing	146
7-12	External Event Counter Operation Timing (with Rising Edge Specified)	149
7-13	Square-Wave Output Operation Timing	150
7-14	PWM Output Operation Timing	152
7-15	Operation Timing When CR5n Is Changed	153
7-16	16-Bit Resolution Cascade Mode	154
7-17	Start Timing of 8-Bit Timer/Counter	155
7-18	Timing After Changing Values of Compare Registers During Timer Count Operation	155
8-1	Block Diagram of Watch Timer	156
8-2	Format of Watch Timer Operation Mode Register 0 (WTNM0)	158
8-3	Operation Timing of Watch Timer/Interval Timer	160
9-1	Block Diagram of Watchdog Timer	161
9-2	Format of Watchdog Timer Clock Selection Register (WDCS)	163
9-3	Format of Watchdog Timer Mode Register (WDTM)	164
9-4	Format of Oscillation Stabilization Time Selection Register (OSTS)	165
10-1	Block Diagram of Clock Output Controller	168
10-2	Format of Clock Output Selection Register (CKS)	170
10-3	Format of Port Mode Register 7 (PM7)	170
10-4	Remote Control Output Application Example	171
11-1	Block Diagram of A/D Converter (AD00)	173
11-2	Block Diagram of A/D Converter (AD01)	174
11-3	Format of A/D Converter Mode Register 00 (ADM00)	177
11-4	Format of Analog Input Channel Specification Register 00 (ADS00)	178
11-5	Basic Operation of A/D Converter	180
11-6	Relationship Between Analog Input Voltage and A/D Conversion Result	181
11-7	A/D Conversion by Software Start	182
11-8	Overall Error	183
11-9	Quantization Error	183
11-10	Example of Reducing Current Consumption in Standby Mode	184
11-11	Analog Input Pin Connection	185
11-12	A/D Conversion End Interrupt Request Generation Timing	186
11-13	AV_{DD0} Pin Connection	186
11-14	Conversion Result Read Timing (When Conversion Result Is Undefined)	187
11-15	Conversion Result Read Timing (When Conversion Result Is Normal)	187
11-16	Example of Capacitor Connection to AV_{REF0} Pin	188

LIST OF FIGURES (4/6)

Figure No.	Title	Page
11-17	Internal Equivalence Circuit of ANI00 to ANI70 Pins	189
11-18	Example of Circuit When Signal Source Impedance Is High	189
12-1	Example of Message	190
12-2	Format of Class 2 Control Register 1 (C2CT1)	192
12-3	Format of Class 2 Control Register 2 (C2CT2)	195
12-4	Format of Class 2 Status Register 1 (C2ST1)	196
12-5	Format of Class 2 Status Register 2 (C2ST2)	198
12-6	Format of Class 2 Rise Time Propagation Delay Correction Register (C2PDR)	201
12-7	Format of Class 2 Fall Time Propagation Delay Correction Register (C2PDF)	202
12-8	Format of Class 2 Clock Selection Register (C2CLK)	203
12-9	Reception Status Transition Diagram (Normal Mode)	206
12-10	Reception Status Transition Diagram (×4 Mode)	207
12-11	Standard Reception Timing (Normal Mode)	208
12-12	Transmission	210
12-13	Transmission Status Transition Diagram (Transmission Status Flag)	211
12-14	Transmission Start Timing	212
12-15	Short Detection (at Transmission Only)	214
12-16	Collision 1 (If Clock Speed Varies for Same Symbol)	216
12-17	Collision 2 (Loss in Contention with Different Symbols)	217
12-18	Interrupt Flag Status Transition Diagram	221
12-19	Waveforms of C2TX and C2RX Pins (Normal Mode: PDR0 to PDR4 = 0 and PDF0 to PDF4 = 0)	222
12-20	Waveforms of C2TX and C2RX Pins (Normal Mode: PDR0 to PDR4 = 26 and PDF0 to PDF4 = 15)	223
12-21	Waveforms of C2TX and C2RX Pins (×4 Mode: PDRX0 to PDRX2 = 0 and PDFX0 to PDFX2 = 0)	224
12-22	Waveforms of C2TX and C2RX Pins (×4 Mode: PDRX0 to PDRX2 = 2 and PDFX0 to PDFX2 = 4)	225
13-1	Block Diagram of Serial Interface UART0	238
13-2	Block Diagram of Baud Rate Generator	239
13-3	Format of Asynchronous Serial Interface Mode Register 0 (ASIM0)	242
13-4	Format of Asynchronous Serial Interface Status Register 0 (ASIS0)	243
13-5	Format of Baud Rate Generator Control Register 0 (BRGC0)	244
13-6	Error Tolerance (When k = 0), Including Sampling Errors	251
13-7	Format of Transmit/Receive Data in Asynchronous Serial Interface	252
13-8	Timing of Asynchronous Serial Interface Transmit Completion Interrupt Request	255
13-9	Timing of Asynchronous Serial Interface Reception Completion Interrupt Request	256
13-10	Receive Error Timing	257
14-1	Block Diagram of Serial Interface SIO30	259
14-2	Format of Serial Operation Mode Register 30 (CSIM30)	261
14-3	Timing of 3-Wire Serial I/O Mode	264

LIST OF FIGURES (5/6)

Figure No.	Title	Page
15-1	Block Diagram of Serial Interface IIC0	266
15-2	Example of Serial Bus Configuration Using I^{2}C Bus	267
15-3	Format of IIC Control Register 0 (IICC0)	270
15-4	Format of IIC Status Register 0 (IICS0)	273
15-5	Format of IIC Transfer Clock Selection Register 0 (IICCL0)	276
15-6	Format of IIC Function Expansion Register 0 (IICX0)	278
15-7	Pin Configuration Diagram	280
15-8	Serial Data Transfer Timing of I^{2}C Bus	281
15-9	Start Conditions	281
15-10	Address	282
15-11	Transfer Direction Specification	282
15-12	$\overline{\text{ACK}}$ Signal	283
15-13	Stop Condition	284
15-14	Wait Signal	285
15-15	Arbitration Timing Example	306
15-16	Communication Reservation Timing	309
15-17	Timing for Acknowledging Communication Reservations	309
15-18	Communication Reservation Protocol	310
15-19	Master Operation Flow Chart	311
15-20	Slave Operation Flow Chart	312
15-21	Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave)	314
15-22	Example of Slave to Master Communication (When 9-Clock Wait Is Selected for Both Master and Slave)	317
16-1	Basic Configuration of Interrupt Function	322
16-2	Format of Interrupt Request Flag Register (IF0L, IF0H, IF1L, IF1H)	325
16-3	Format of Interrupt Mask Flag Register (MK0L, MK0H, MK1L, MK1H)	326
16-4	Format of Priority Specification Flag Register (PR0L, PR0H, PR1L, PR1H)	327
16-5	Format of External Interrupt Rising Edge Enable Register (EGP) and External Interrupt Falling Edge Enable Register (EGN)	328
16-6	Format of Program Status Word (PSW)	329
16-7	Flowchart of Non-Maskable Interrupt Request Generation Through to Acknowledgement	331
16-8	Non-Maskable Interrupt Request Acknowledgement Timing	331
16-9	Non-Maskable Interrupt Request Acknowledgement Operation	332
16-10	Interrupt Request Acknowledgement Processing Algorithm	334
16-11	Interrupt Request Acknowledgement Timing (Minimum Time)	335
16-12	Interrupt Request Acknowledgement Timing (Maximum Time)	335
16-13	Multiple Interrupt Examples	337
16-14	Interrupt Request Hold	339
17-1	Format of Oscillation Stabilization Time Selection Register (OSTS)	341
17-2	HALT Mode Release by Interrupt Request Generation	343
17-3	HALT Mode Release by $\overline{\text{RESET}}$ Input	344

LIST OF FIGURES (6/6)

Figure No.	Title	Page
17-4	STOP Mode Release by Interrupt Request Generation	346
17-5	STOP Mode Release by $\overline{\text{RESET}}$ Input	347
18-1	Block Diagram of Reset Function	348
18-2	Timing of Reset by $\overline{\text{RESET}}$ Input	349
18-3	Timing of Reset Due to Watchdog Timer Overflow	349
18-4	Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input	349
19-1	Format of Memory Size Switching Register (IMS)	353
19-2	Format of Internal Expansion RAM Size Switching Register (IXS)	353
19-3	Format of Communication Mode Selection	354
19-4	Connection of Flashpro III Using 3-Wire Serial I/O Mode	355
19-5	Flash Memory Configuration	356
19-6	Format of Flash Programming Mode Control Register (FLMPC)	357
19-7	Self-Write Flowchart	358
19-8	Self-Write Timing	359
19-9	Configuration of Self-Write Circuit	372
A-1	Development Tool Configuration	388
A-2	EV-9200GC-80 Drawing (for reference only)	396
A-3	EV-9200GC-80 Footprints (for reference only)	397

LIST OF TABLES (1/2)

Table No.	Title	Page
2-1	Types of Pin I/O Circuits and Recommended Connection of Unused Pins	42
3-1	Vector Table	49
3-2	Special Function Register List	58
4-1	Port Functions	73
4-2	Port Configuration	75
5-1	Configuration of Clock Generator	92
5-2	Relationship of CPU Clock and Minimum Instruction Execution Time	93
5-3	Maximum Time Required for CPU Clock Switchover	96
6-1	Configuration of 16-Bit Timer/Event Counters	99
6-2	TI00n Pin Valid Edge and Capture/Compare Register Capture Trigger	102
6-3	TI01n Pin Valid Edge and Capture/Compare Register Capture Trigger	102
7-1	Configuration of 8-Bit Timer/Event Counter	136
8-1	Interval Time of Interval Timer	157
8-2	Configuration of Watch Timer	157
8-3	Interval Time of Interval Timer	159
9-1	Watchdog Timer Program Loop Detection Time	162
9-2	Interval Time	162
9-3	Configuration of Watchdog Timer	163
9-4	Watchdog Timer Program Loop Detection Time	166
9-5	Interval Time of Interval Timer	167
10-1	Configuration of Clock Output Controller	169
11-1	AD00 and AD01 Name Differences	172
11-2	Configuration of A/D Converter	175
11-3	Resistance and Capacitance Values for Equivalence Circuits (Reference Values)	189
12-1	Examples of Symbols	190
12-2	Types of Errors and Flags	218
12-3	Waveforms Causing Symbol Errors	219
12-4	Interrupt Request Flag	220
12-5	Minimum Time for Flag Change	221
12-6	Differences Between Wake-Up and Sleep States	236
13-1	Configuration of Serial Interface UART0	240
13-2	Relationship Between 5-Bit Counter’s Source Clock and “n” Value	250
13-3	Relationship Between System Clock and Baud Rate	250

LIST OF TABLES (2/2)

Table No.	Title	Page
13-4	Causes of Receive Errors	257
14-1	SIO30 and SIO31 Name Differences	258
14-2	Configuration of Serial Interface SIO30	260
15-1	Configuration of Serial Interface IIC0	268
15-2	Selectable Transfer Clocks	278
15-3	INTIIC0 Timing and Wait Control	304
15-4	Extension Code Bit Definitions	305
15-5	Status During Arbitration and Interrupt Request Generation Timing	307
15-6	Wait Periods	308
16-1	Interrupt Source List	321
16-2	Flags Corresponding to Interrupt Request Sources	324
16-3	Time from Generation of Maskable Interrupt Until Servicing	333
16-4	Interrupt Request Enabled for Multiple Interrupt During Interrupt Servicing	336
17-1	HALT Mode Operating Statuses	342
17-2	Operation After HALT Mode Release	344
17-3	STOP Mode Operating Status	345
17-4	Operation After STOP Mode Release	347
18-1	Hardware Statuses After Reset	350
19-1	Differences Between μPD78F0833Y and Mask ROM Versions	352
19-2	Communication Mode List	354
19-3	Main Functions of Flash Memory Programming	355
19-4	Entry RAM Area	360
19-5	List of Self-Write Subroutines	362
20-1	Operand Identifiers and Specification Methods	374
A-1	Upgrade Method from Former In-Circuit Emulator for 78K/0 Series to IE-78001-R-A	395

CHAPTER 1 OUTLINE

1.1 Features

- On-chip J1850 (Class 2) bus controller
- Internal memory

★

Item / Part Number	Program Memory		Data Memory		
	Mask ROM	Flash Memory	High-Speed RAM	Expansion RAM	Buffer RAM (Class 2)
μPD780833Y	60 KB	–	1024 bytes	2048 bytes	32 bytes
μPD78F0833Y	–	60 KB			

- Minimum instruction execution time changeable from high speed (0.48 μs) to low speed (7.68 μs)
- Instruction set suited to system control
  - Bit manipulation possible in all address spaces
  - Multiply and divide instructions
- I/O ports: 65 (three N-ch open-drain ports, eight TTL input/CMOS output ports)
- 8-bit resolution A/D converter: 8 channels × 2
- Serial interface: 4 channels
  - 3-wire serial I/O mode: 2 channels
  - UART mode: 1 channel
  - I^{2}C bus mode: 1 channel
- Timer:

★

Item / Part Number	16-Bit Timer/ Event Counter	8-Bit Timer/ Event Counter	Watch Timer	Watchdog Timer
μPD780833Y	2 channels	3 channels	1 channel	1 channel
μPD78F0833Y				

- Vectored interrupts: 30
- Power supply voltage: V_{DD} = 4.5 to 5.5 V

1.2 Applications

Car audios, etc.

★ 1.3 Ordering Information

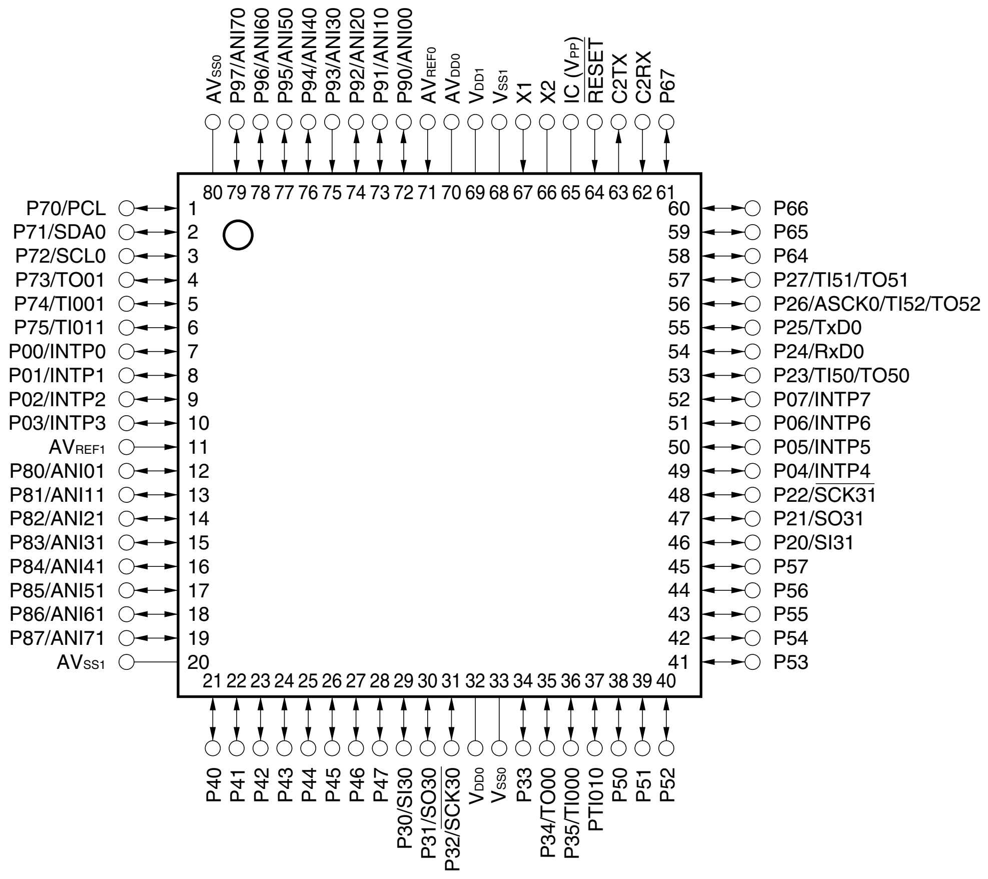
Part Number	Package	Internal ROM
μPD780833YGC-×××-8BT	80-pin plastic QFP (14 × 14)	Mask ROM
μPD780F0833YGC-8BT	80-pin plastic QFP (14 × 14)	Flash memory

Remark ××× indicates ROM code suffix.

1.4 Pin Configuration (Top View)

★ • **80-pin plastic QFP (14 × 14)**

μPD780833YGC-×××-8BT
μPD78F0833YGC-8BT



Cautions 1. Connect the IC (Internally Connected) pin directly to V_{SS0} or V_{SS1}.

2. Connect the AV_{DD0} pin to V_{DD0}.

3. Connect the AV_{SS0} and AV_{SS1} pins to V_{SS0}.

Remarks 1. When these microcontrollers are used in applications where the noise generated inside the microcontroller needs to be reduced, the implementation of noise reduction measures, such as supplying voltage to V_{DD0} and V_{DD1} independently and connecting V_{SS0} and V_{SS1} to different ground lines, is recommended.

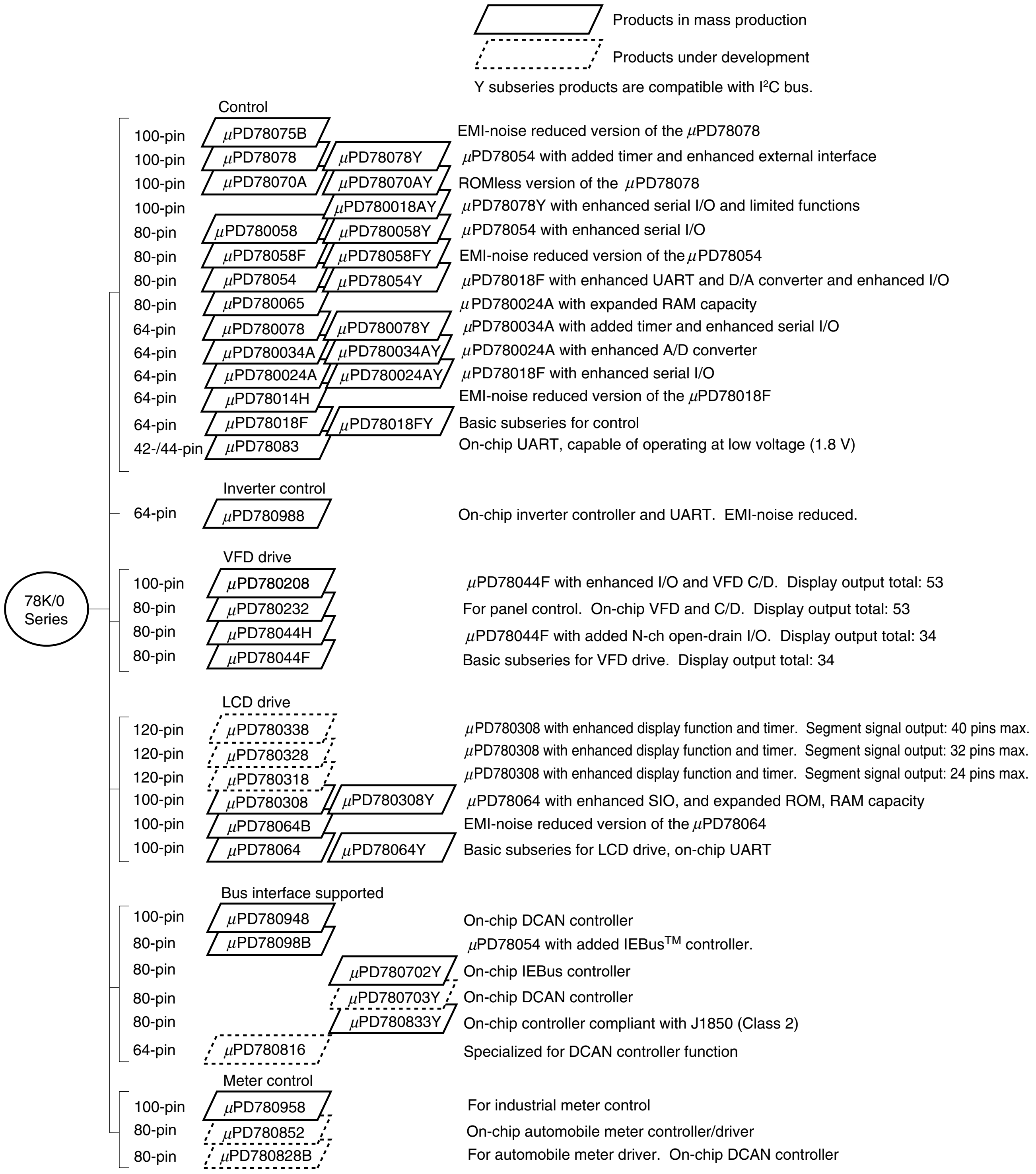
2. The pin connections in parentheses applies to the μPD78F0833Y only.

ANI00 to ANI70,	
ANI01 to ANI71:	Analog input
ASCK0:	Asynchronous serial clock
AV_{DD0}:	Analog power supply
AV_{REF0}, AV_{REF1}:	Analog reference voltage
AV_{SS0}, AV_{SS1}:	Analog ground
C2RX:	Class 2 receive data
C2TX:	Class 2 transmit data
IC:	Internally connected
INTP0 to INTP7:	External interrupt input
P00 to P07:	Port 0
P20 to P27:	Port 2
P30 to P36:	Port 3
P40 to P47:	Port 4
P50 to P57:	Port 5
P64 to P67:	Port 6
P70 to P75:	Port 7
P80 to P87:	Port 8
P90 to P97:	Port 9

PCL:	Programmable clock
RxD0:	Receive data
$\overline{\text{RESET}}$:	Reset
$\overline{\text{SCK30}}$, $\overline{\text{SCK31}}$:	Serial clock
SCL0:	Serial clock
SDA0:	Serial data
SI30, SI31:	Serial onput
SO30, SO31:	Serial output
TI000, TI010,	
TI001, TI011,	
TI50, TI51, TI52:	Timer input
TO00, TO01,	
TI50, TO51, TO52:	Timer output
TxD0:	Transmit data
X1, X2:	Crystal
V_{DD0}, V_{DD1}:	Power supply
V_{PP}:	Programming power supply
V_{SS0}, V_{SS1}:	Ground

★ 1.5 78K/0 Series Lineup

The products in the 78K/0 Series are listed below. The names enclosed in boxes are subseries names.



▱	Products in mass production		
⬚	Products under development		
	Y subseries products are compatible with I²C bus.		

78K/0 Series

	Control		
100-pin	μPD78075B		EMI-noise reduced version of the μPD78078
100-pin	μPD78078	μPD78078Y	μPD78054 with added timer and enhanced external interface
100-pin	μPD78070A	μPD78070AY	ROMless version of the μPD78078
100-pin		μPD780018AY	μPD78078Y with enhanced serial I/O and limited functions
80-pin	μPD780058	μPD780058Y	μPD78054 with enhanced serial I/O
80-pin	μPD78058F	μPD78058FY	EMI-noise reduced version of the μPD78054
80-pin	μPD78054	μPD78054Y	μPD78018F with enhanced UART and D/A converter and enhanced I/O
80-pin	μPD780065		μPD780024A with expanded RAM capacity
64-pin	μPD780078	μPD780078Y	μPD780034A with added timer and enhanced serial I/O
64-pin	μPD780034A	μPD780034AY	μPD780024A with enhanced A/D converter
64-pin	μPD780024A	μPD780024AY	μPD78018F with enhanced serial I/O
64-pin	μPD78014H		EMI-noise reduced version of the μPD78018F
64-pin	μPD78018F	μPD78018FY	Basic subseries for control
42-/44-pin	μPD78083		On-chip UART, capable of operating at low voltage (1.8 V)
	Inverter control		
64-pin	μPD780988		On-chip inverter controller and UART. EMI-noise reduced.
	VFD drive		
100-pin	μPD780208		μPD78044F with enhanced I/O and VFD C/D. Display output total: 53
80-pin	μPD780232		For panel control. On-chip VFD and C/D. Display output total: 53
80-pin	μPD78044H		μPD78044F with added N-ch open-drain I/O. Display output total: 34
80-pin	μPD78044F		Basic subseries for VFD drive. Display output total: 34
	LCD drive		
120-pin	μPD780338		μPD780308 with enhanced display function and timer. Segment signal output: 40 pins max.
120-pin	μPD780328		μPD780308 with enhanced display function and timer. Segment signal output: 32 pins max.
120-pin	μPD780318		μPD780308 with enhanced display function and timer. Segment signal output: 24 pins max.
100-pin	μPD780308	μPD780308Y	μPD78064 with enhanced SIO, and expanded ROM, RAM capacity
100-pin	μPD78064B		EMI-noise reduced version of the μPD78064
100-pin	μPD78064	μPD78064Y	Basic subseries for LCD drive, on-chip UART
	Bus interface supported		
100-pin	μPD780948		On-chip DCAN controller
80-pin	μPD78098B		μPD78054 with added IEBus™ controller.
80-pin		μPD780702Y	On-chip IEBus controller
80-pin		μPD780703Y	On-chip DCAN controller
80-pin		μPD780833Y	On-chip controller compliant with J1850 (Class 2)
64-pin	μPD780816		Specialized for DCAN controller function
	Meter control		
100-pin	μPD780958		For industrial meter control
80-pin	μPD780852		On-chip automobile meter controller/driver
80-pin	μPD780828B		For automobile meter driver. On-chip DCAN controller

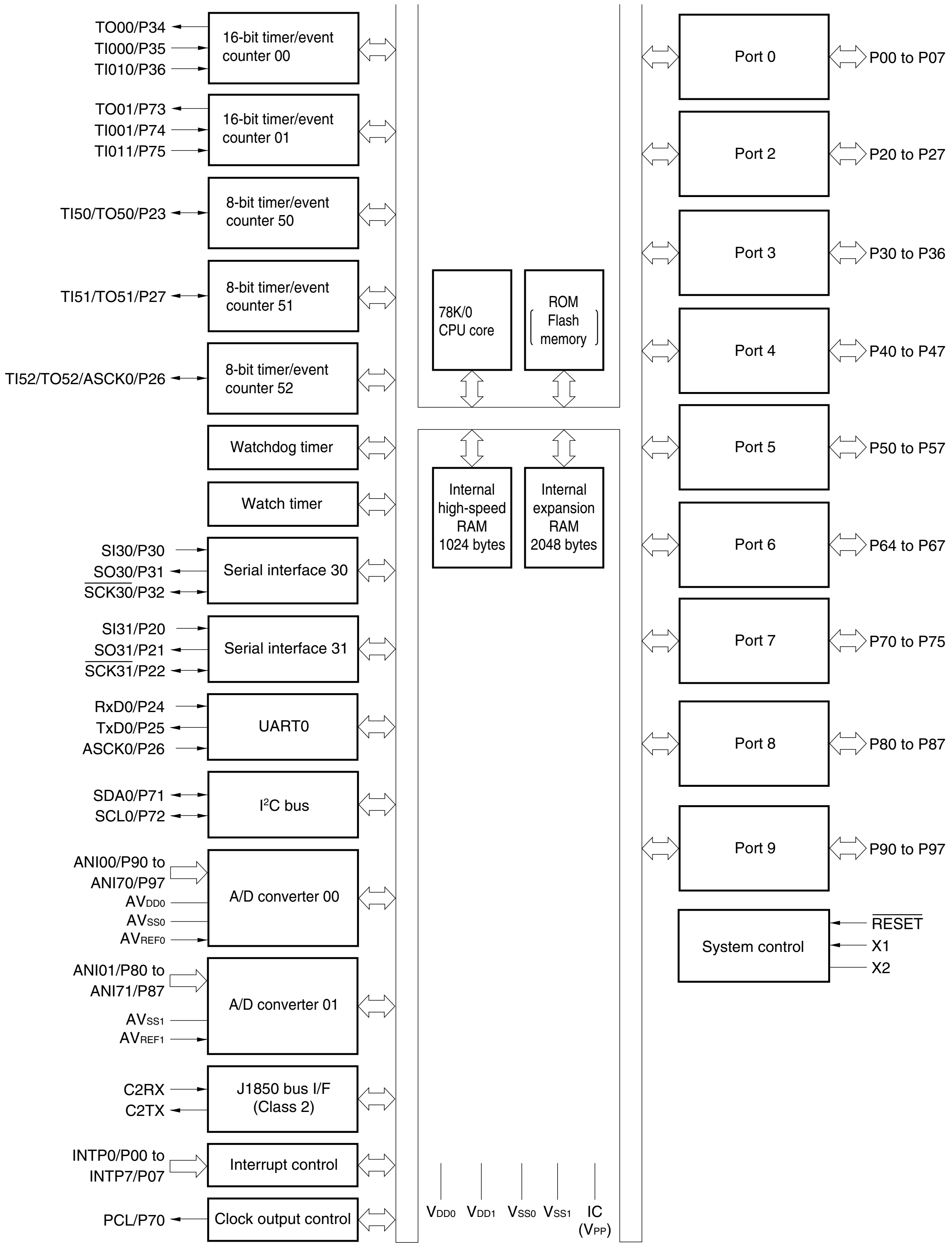
Remark VFD (Vacuum Fluorescent Display) is referred to as FIP™ (Fluorescent Indicator Panel) in some documents, but the functions of the two are the same.

- **Y Subseries**

Function / Subseries Name		ROM Capacity (Bytes)	Timer				8-Bit A/D	10-Bit A/D	8-Bit D/A	Serial Interface	I/O	V_{DD} MIN. Value	External Expansion
			8-Bit	16-Bit	Watch	WDT							
Control	μPD78078Y	48 K to 60 K	4 ch	1 ch	1 ch	1 ch	8 ch	–	2 ch	3 ch (UART: 1 ch, I^2C: 1 ch)	88	1.8 V	√
	μPD78070AY	–									61	2.7 V	
	μPD780018AY	48 K to 60 K							–	3 ch (I^2C: 1 ch)	88		
	μPD780058Y	24 K to 60 K	2 ch						2 ch	3 ch (time division UART: 1 ch, I^2C: 1 ch)	68	1.8 V	
	μPD78058FY	48 K to 60 K								3 ch (UART: 1 ch, I^2C: 1 ch)	69	2.7 V	
	μPD78054Y	16 K to 60 K										2.0 V	
	μPD780078Y	48 K to 60 K		2 ch			–	8 ch	–	4 ch (UART: 2 ch, I^2C: 1 ch)	52	1.8 V	
	μPD780034AY	8 K to 32 K		1 ch						3 ch (UART: 1 ch, I^2C: 1 ch)	51		
	μPD780024AY						8 ch	–					
	μPD78018FY	8 K to 60 K								2 ch (I^2C: 1 ch)	53		
LCD drive	μPD780308Y	48 K to 60 K	2 ch	1 ch	1 ch	1 ch	8 ch	–	–	3 ch (time division UART: 1 ch, I^2C: 1 ch)	57	2.0 V	–
	μPD78064Y	16 K to 32 K								2 ch (UART: 1 ch, I^2C: 1 ch)			
For bus interface	μPD780702Y	60 K	3 ch	2 ch	1 ch	1 ch	16 ch	–	–	4 ch (UART: 1 ch, I^2C: 1 ch)	67	3.5 V	–
	μPD780703Y												
	μPD780833Y										65	4.5 V	

Remark The functions of non Y subseries and Y subseries products are the same, except for the serial interface.

1.6 Block Diagram



★ **Remarks 1.** The internal ROM capacity varies depending on the version.

2. The item in parentheses applies to the μPD78F0833Y only.

1.7 Overview of Functions

★

Item \ Part Number		μPD780833Y	μPD78F0833Y
Internal memory	ROM	60 KB (Mask ROM)	60 KB (Flash memory)
	High-speed RAM	1024 bytes	
	Expansion RAM	2048 bytes	
Memory space		64 KB	
General-purpose registers		8 bits × 32 registers (8 bits × 8 registers × 4 banks)	
Minimum instruction execution time		On-chip minimum instruction execution time variation function 0.48 μs/0.96 μs/1.92 μs/3.84 μs/7.68 μs (4.19 MHz operation)	
Instruction set		• 16-bit operation • Multiply/divide (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulation (set, reset, test, and Boolean operation) • BCD adjust, etc.	
I/O ports		Total: 65 • CMOS I/O: 54 • TTL input/CMOS output: 8 • N-ch open-drain I/O: 3	
A/D converter		• 8-bit resolution × 8 channels × 2	
Serial interface		• 3-wire serial I/O mode: 2 channels • UART mode: 1 channel • I^{2}C bus mode: 1 channel	
Timers		• 16-bit timer/event counter: 2 channels • 8-bit timer/event counter: 3 channels • Watch timer: 1 channel • Watchdog timer: 1 channel	
Timer outputs		5 (8-bit PWM outputs: 3)	
Clock output		32.8 kHz, 65.5 kHz, 130.9 kHz, 261.9 kHz, 523.6 kHz, 1.05 MHz, 2.10 MHz, 4.19 MHz (4.19 MHz with system clock)	
Bus controller		J1850 (Class 2) bus interface	
Vectored interrupts	Maskable	Internal: 19, external: 9	
	Non-maskable	Internal: 1	
	Software	1	
Power supply voltage		V_{DD} = 4.5 to 5.5 V	
Operating ambient temperature		T_A = –40 to +85°C	
Package		80-pin plastic QFP (14 × 14)	

★ An overview of the timer/event counters is shown below. (For details, refer to **CHAPTER 6 16-BIT TIMER/EVENT COUNTERS 00**, **01**, **CHAPTER 7 8-BIT TIMER/EVENT COUNTERS 50**, **51**, **52**, **CHAPTER 8 WATCH TIMER**, and **CHAPTER 9 WATCHDOG TIMER**.)

		16-Bit Timer/Event Counters 00, 01	8-Bit Timer/Event Counters 50, 51, 52	Watch Timer	Watchdog Timer
Operation Mode	Interval timer	2 channels	3 channels	1 channel**Note 1**	1 channel**Note 2**
	External event counter	√	√	–	–
Function	Timer output	√	√	–	–
	PPG output	√	–	–	–
	PWM output	–	√	–	–
	Pulse width measurement	√	–	–	–
	Square-wave output	√	√	–	–
	One-shot pulse output	√	–	–	–
	Interrupt request	√	√	√	√

Notes 1. The watch timer can perform both watch timer and interval timer functions at the same time.

2. The watchdog timer can perform either watchdog timer or interval timer.

CHAPTER 2 PIN FUNCTION

2.1 Pin Function List

(1) Port pins (1/2)

Pin Name	I/O	Function		After Reset	Alternate Function
P00 to P07	I/O	Port 0 8-bit I/O port Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.		Input	INTP0 to INTP7
P20	I/O	Port 2 8-bit I/O port Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.		Input	SI31
P21					SO31
P22					$\overline{\text{SCK31}}$
P23					TI50/TO50
P24					RxD0
P25					TxD0
P26					ASCK0/TI52/TO52
P27					TI51/TO5
P30	I/O	Port 3 7-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	SI30
P31					SO30
P32					$\overline{\text{SCK30}}$
P33			N-ch open-drain I/O port LEDs can be driven directly.		—
P34			Use of an on-chip pull-up resistor can be specified by a software setting.		TO00
P35					TI000
P36					TI010
P40 to P47	I/O	Port 4 8-bit I/O port Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting. The interrupt request flag (KRIF) is set to 1 by falling edge detection.		Input	—
P50 to P57	I/O	Port 5 8-bit I/O port TTL level input/CMOS output 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	—
P64 to P67	I/O	Port 6 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	—

(1) Port pins (2/2)

Pin Name	I/O	Function		After Reset	Alternate Function
P70	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	PCL
P71			Use of N-ch open-drain I/O port		SDA0
P72					SCL0
P73			Use of an on-chip pull-up resistor can specified by a software setting.		TO01
P74					TI001
P75					TI011
P80 to P87	I/O	Port 8 8-bit I/O port Input/output can be specified in 1-bit units.		Input	ANI01 to ANI71
P90 to P97	I/O	Port 9 8-bit I/O port Input/output can be specified in 1-bit units.		Input	ANI00 to ANI70

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

Pin Name	I/O	Function	After Reset	Alternate Function
INTP0 to INTP7	Input	External interrupt request input for which the valid edge can be specified (rising edge, falling edge, or both rising and falling edges)	Input	P00 to P07
SI30		Serial interface SIO30 serial data input	Input	P30
SO30	Output	Serial interface SIO30 serial data output	Input	P31
SI31	Input	Serial interface SIO31 serial data input	Input	P20
SO31	Output	Serial interface SIO31 serial data output	Input	P21
SDA0	I/O	Serial interface IIC0 serial data input/output	Input	P71
$\overline{SCK30}$		Serial interface SIO30 serial clock input/output	Input	P32
$\overline{SCK31}$		Serial interface SIO31 serial clock input/output	Input	P22
SCL0		Serial interface IIC0 serial clock input/output	Input	P72
RxD0	Input	Asynchronous serial interface serial data input	Input	P24
TxD0	Output	Asynchronous serial interface serial data output	Input	P25
ASCK0	Input	Asynchronous serial interface serial clock input	Input	P26/TI52/TO52
TI000		External count clock input to 16-bit timer/event counter 00 Capture trigger input to 16-bit timer/event counter 00 capture register (CR000 and CR010)	Input	P35
TI010		Capture trigger input to 16-bit timer/event counter 00 capture register (CR000)		P36
TI001		External count clock input to 16-bit timer/event counter 01 Capture trigger input to 16-bit timer/event counter 01 capture register (CR001 and CR011)		P74
TI011		Capture trigger input to 16-bit timer/event counter 01 capture register (CR001)		P75
TI50		External count clock input to 8-bit timer counter/event counter 50		P23/TO50
TI51		External count clock input to 8-bit timer counter/event counter 51		P27/TO51

(2) Non-port pins (2/2)

Pin Name	I/O	Function	After Reset	Alternate Function
TI52	Input	External count clock input to 8-bit timer counter/event counter 52	Input	P26/ASCK0/TO52
TO00	Output	16-bit timer/event counter 00 output		P34
TO01		16-bit timer/event counter 01 output		P73
TO50		8-bit timer/event counter 50 output		P23/TI50
TO51		8-bit timer/event counter 51 output		P27/T151
TO52		8-bit timer/event counter 52 output		P26/ASCK0/TI52
TO00		16-bit timer/event counter output		P34
TO01		16-bit timer/event counter output		P73
PCL		Clock output		P70
ANI00 to ANI70	Input	A/D converter (AD00) analog input		P90 to P97
ANI01 to ANI71		A/D converter (AD01) analog input		P80 to P87
AV_{REF0}		A/D converter (AD00) reference voltage input	—	—
AV_{REF1}		A/D converter (AD01) analog power supply and reference voltage input		—
AV_{DD0}	—	A/D converter (AD00) analog power supply		—
AV_{SS0}		A/D converter (AD00) ground potential. Set the same potential as that of V_{SS0} or V_{SS1}.		—
AV_{SS1}		A/D converter (AD01) ground potential. Set the same potential as that of V_{SS0} or V_{SS1}.		—
C2RX	Input	Class 2 data input		—
C2TX	Output	Class 2 data output		—
$\overline{\text{RESET}}$	Input	System reset input		—
X1		Connection of crystal resonator for oscillation		—
X2	—			—
V_{DD0}		Positive power supply		—
V_{SS0}		Ground potential		—
V_{DD1}		Positive power supply other than port		—
V_{SS1}		Ground potential other than port		—
IC		Internally connected. (Mask ROM version only)		—
V_{PP}		High-voltage application for program write/verify (μPD78F0833Y only)		—

2.2 Description of Pin Functions

2.2.1 P00 to P07 (Port 0)

These pins constitute an 8-bit I/O port. In addition to I/O port pins, they also function as external interrupt inputs. The following operation modes can be specified in 1-bit units.

(1) Port mode

These pins function as an 8-bit I/O port. Input or output can be specified in 1-bit units by means of port mode register 0 (PM0). An on-chip pull-up resistor can be connected by setting pull-up resistor option register 0 (PU0).

(2) Control mode

These pins function as external interrupt request inputs.
INTP0 to INTP7 are external interrupt request input pins for which the valid edges can be selected (rising edge, falling edge, or both rising and falling edges).

2.2.2 P20 to P27 (Port 2)

These pins constitute an 8-bit I/O port. In addition to I/O port pins, they also function as serial interface data I/O, clock I/O, and timer I/O.
The following operation modes can be specified in 1-bit units.

(1) Port mode

These pins function as an 8-bit I/O port. Input or output can be specified in 1-bit units by means of port mode register 2 (PM2). An on-chip pull-up resistor can be connected by setting pull-up resistor option register 2 (PU2).

(2) Control mode

These pins function as serial interface data I/O, clock I/O, and timer I/O.

(a) SI31, SO31

These are the I/O pin for the serial data of the serial interface.

(b) $\overline{\text{SCK31}}$

This is the I/O pin for the serial clock of the serial interface.

(c) RxD0, TxD0

These are the I/O pin for the serial data of the asynchronous serial interface.

(d) ASCK0

This is the I/O pin for the serial clock of the asynchronous serial interface.

(e) TI50, TI51, TI52

These are external count clock input pins for the 8-bit timer/event counter (TM50, TM51, TM52).

(f) TO50, T51, TO52

These are 8-bit timer output pins.

2.2.3 P30 to P36 (Port 3)

These pins constitute a 7-bit I/O port. In addition to I/O port pins, they also function as serial interface data I/O and clock I/O.

LEDs can be directly driven.

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

(1) Port mode

These pins function as a 7-bit I/O port. Input or output can be specified in 1-bit units by means of port mode register 3 (PM3). P33 is N-ch open drain. An on-chip pull-up resistor can be connected by setting pull-up resistor option register 3 (PU3).

(2) Control mode

These pins function as serial interface data I/O, clock I/O, and timer I/O.

(a) SI30, SO30

These are the I/O pin for the serial data of the serial interface.

(b) $\overline{\text{SCK30}}$

This is the I/O pin for the serial clock of the serial interface.

(c) TI000

This is the external count clock input pin to 16-bit timer/event counter 00 (TM00) and capture trigger signal input pin to the TM00 capture register (CR000 and CR010).

(d) TI010

This is the capture trigger signal input pin to the capture register (CR000) of 16-bit timer/event counter 00 (TM00).

(e) TO00

This is the 16-bit timer/event counter 00 (TM00) timer output pin.

2.2.4 P40 to P47 (Port 4)

These pins constitute an 8-bit I/O port. Input or output can be specified in 1-bit units by means of port mode register 4 (PM4). On-chip pull-up resistors can be used by setting pull-up resistor option register 4 (PU4).

2.2.5 P50 to P57 (Port 5)

These pins constitute an 8-bit I/O port. TTL level input. Input or output can be specified in 1-bit units by means of port mode register 5 (PM5). An on-chip pull-up resistor can be connected by setting pull-up resistor option register 5 (PU5).

2.2.6 P64 to P67 (Port 6)

These pins constitute a 4-bit I/O port. Input or output can be specified in 1-bit units by means of port mode register 6 (PM6). An on-chip pull-up resistor can be connected by setting pull-up resistor option register 6 (PU6).

2.2.7 P70 to P75 (Port 7)

These pins constitute a 6-bit I/O port. In addition to I/O port pins, they also function as serial interface data I/O, clock input/output, and timer input/output.

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

(1) Port mode

These pins function as a 6-bit I/O port. Input or output can be specified in 1-bit units by means of port mode register 7 (PM7). P71 to P72 are N-ch open drain I/O port pins. An on-chip pull-up resistor can be connected to P70, P73 to P75 by setting pull-up resistor option register 7 (PU7). P74 and P75 are also function as the16-bit timer/event counter capture trigger signal input pins with a valid edge input.

(2) Control mode

These pins function as serial interface data I/O, clock I/O and timer I/O.

(a) SDA0

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

(b) SCL0

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

(c) TI001

This is the external count clock input pin to 16-bit timer/event counter 01 (TM01) and capture trigger signal input pin to the capture register (CR001 and CR011) of the 16-bit timer/event counter.

(d) TI011

This is the capture trigger signal input pin to the capture register (CR001) of 16-bit timer/event counter (TM01).

(e) TO01

This is the 16-bit timer/event counter 01 (TM01) timer output pin.

(f) PCL

This is the clock output pin.

2.2.8 P80 to P87 (Port 8)

There pins constitute an 8-bit I/O port. In addition to I/O port pins, they also function as A/D converter analog inputs. The following operation modes can be specified in 1-bit units.

(1) Port mode

These pins function as an 8-bit I/O port. Input or output can be specified in 1-bit units by means of port mode register 8 (PM8).

(2) Control mode

These pins function as A/D converter analog input pins (ANI01 to ANI71).

2.2.9 P90 to P97 (Port 9)

These pins constitute an 8-bit I/O port. In addition to I/O port pins, they also function as A/D converter analog inputs. The following operation modes can be specified in 1-bit units.

(1) Port mode

These pins function as an 8-bit I/O port. Input or output can be specified in 1-bit units by means of port mode register 9 (PM9).

(2) Control mode

These pins function as A/D converter analog input pins (ANI00 to ANI70).

2.2.10 C2RX and C2TX

This is the J1850 (Class 2) data I/O pin.

2.2.11 AV_{REF0} and AV_{REF1}

AV_{REF0} is an A/D converter (AD00) reference voltage input pin.

AV_{REF1} is an A/D converter (AD01) reference voltage input pin. This pin also functions as an analog power supply pin.

When no A/D converter is used, connect these pins to V_{DD0}.

2.2.12 AV_{DD0}

This is an analog power supply pin of the A/D converter (AD00). Always use the same voltage as that of the V_{DD0} pin even when AD00 is not used.

2.2.13 AV_{SS0} and AV_{SS1}

AV_{SS0} is a ground voltage pin the of the A/D converter (AD00).

AV_{SS1} is a ground voltage pin of the A/D converter (AD01). Always use the same voltage as that of the V_{SS0} pin even when no A/D converter is used.

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

This is a low-level active system reset input pin.

2.2.15 X1 and X2

These are crystal resonator connection pins for main system clock oscillation. When the clock is being supplied externally, input the clock signal to X1 and its inverted signal to X2.

2.2.16 V_{DD0} and V_{DD1}

V_{DD0} is a positive power supply pin for ports.

V_{DD1} is a positive power supply pin for other than port.

2.2.17 V_{SS0} and V_{SS1}

V_{SS0} is a ground potential pin for ports.

V_{SS1} is a ground potential pin for other than port.

2.2.18 V_{PP} (μPD78F0833Y only)

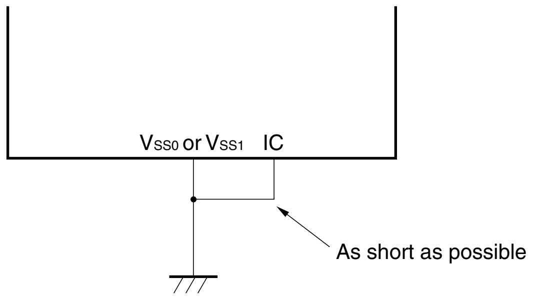
This is a high-voltage application pin for flash memory programming mode setting and program write/verify. Connect directly to V_{SS0} or V_{SS1} in the normal operation mode.

2.2.19 IC (mask ROM version only)

The IC (Internally Connected) pin is provided to set the test mode to check the μPD780833Y Subseries at shipment. Connect it directly to V_{SS0} or V_{SS1} using the shortest possible wire in the normal operation mode.

When a voltage difference is produced between the IC pin and V_{SS0} pin or V_{SS1} pin, because the wiring between those two pins is too long or an external noise is input to the IC pin, the user's program may not operate normally.

- **Connect IC pin to V_{SS0} pin or V_{SS1} pin directly.**



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

The I/O circuit type of each pin and recommended connection of unused pins are shown in Table 2-1.
For the I/O circuit configuration of each type, refer to Figure 2-1.

★ **Table 2-1. Types of Pin I/O Circuits and Recommended Connection of Unused Pins (1/2)**

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00/INTP0 to P07/INTP7	8-C	I/O	Input: Independently connect to V_{SS0} via a resistor. Output: Leave open.
P20/SI31			Input: Independently connect to V_{DD0} or V_{SS0} via a resistor. Output: Leave open.
P21/SO31	5-H		
P22/$\overline{SCK31}$	8-C		
P23/TI50/TO50			
P24/RxD0			
P25/TxD0	5-H		
P26/ASCK0/TI52/TO52	8-C		
P27/TI51/TO51			
P30/SI30			
P31/SO30	5-H		
P32/SCK30	8-C		
P33	13-P		Input: Independently connect to V_{DD0} via a resistor. Output: Leave open.
P34/TO00	5-H		Input: Independently connect to V_{DD0} or V_{SS0} via a resistor. Output: Leave open.
P35/TI000	8-C		
P36/TI010			
P40 to P47	5-H		Input: Independently connect to V_{DD0} via a resistor. Output: Leave open.
P50 to P57	5-T		Input: Independently connect to V_{DD0} or V_{SS0} via a resistor. Output: Leave open.
P64 to P67	5-H		
P70/PCL			
P71/SDA0	13-R		Input: Independently connect to V_{DD0} via a resistor. Output: Leave open.
P72/SCL0			
P73/TO01	5-H		Input: Independently connect to V_{DD0} or V_{SS0} via a resistor. Output: Leave open.
P74/TI001	8-C		
P75/TI011			
P80/ANI01 to P87/ANI71	11-E		
P90/ANI00 to P97/ANI70			

Table 2-1. Types of Pin I/O Circuits and Recommended Connection of Unused Pins (2/2)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
C2RX	2	Input	Connect to V_{SS0} via a resistor.
C2TX	3-B	Output	Leave open.
$\overline{\text{RESET}}$	2	Input	—
AV_{DD0}	—	—	Connect to V_{DD0}.
AV_{REF0}			
AV_{REF1}			
AV_{SS0}			Connect to V_{SS0}.
AV_{SS1}			
IC (μPD780833Y only)			Connect directly to V_{DD0} or V_{SS0}.
V_{PP} (μPD78F0833Y only)			

Figure 2-1. Pin I/O Circuits (1/2)

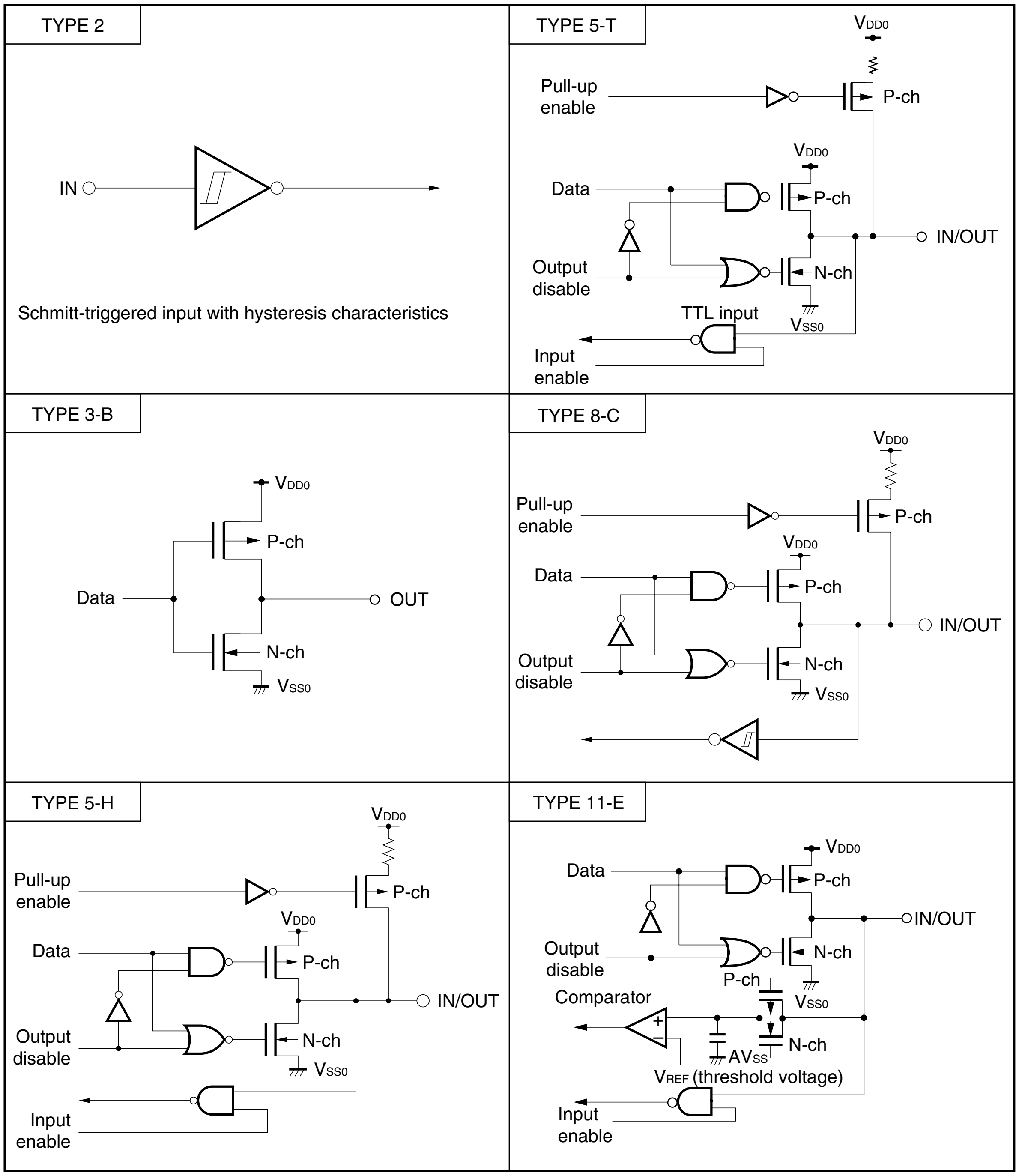
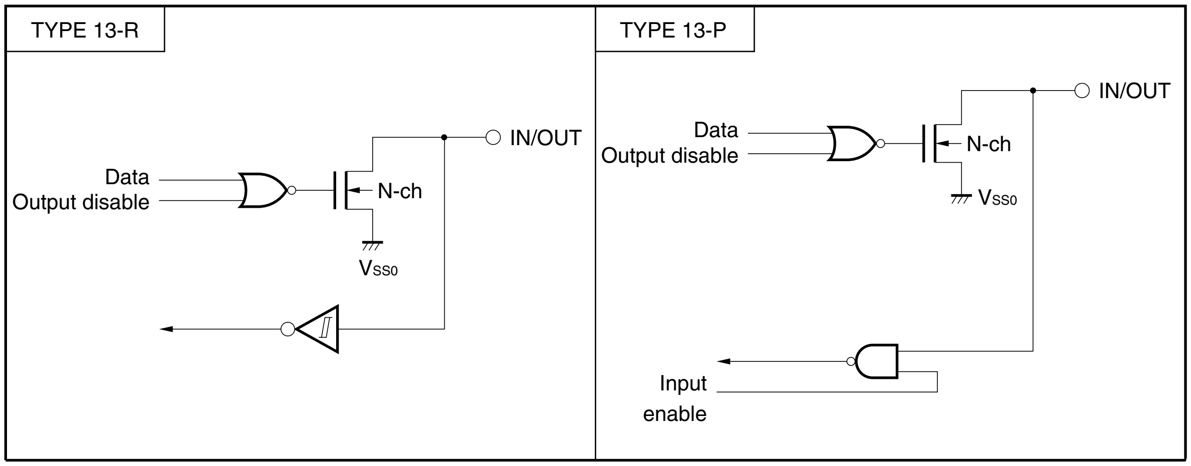


Figure 2-1. Pin I/O Circuits (2/2)



CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

Products in the μPD780833Y Subseries can each access 64 KB of memory space.
Figures 3-1 and 3-2 show memory maps.

★ **Caution The initial value of the internal expansion RAM size switching register (IXS) is set to 0CH. Be sure to set this value to 08H before use.**

Figure 3-1. Memory Map (μPD780833Y)

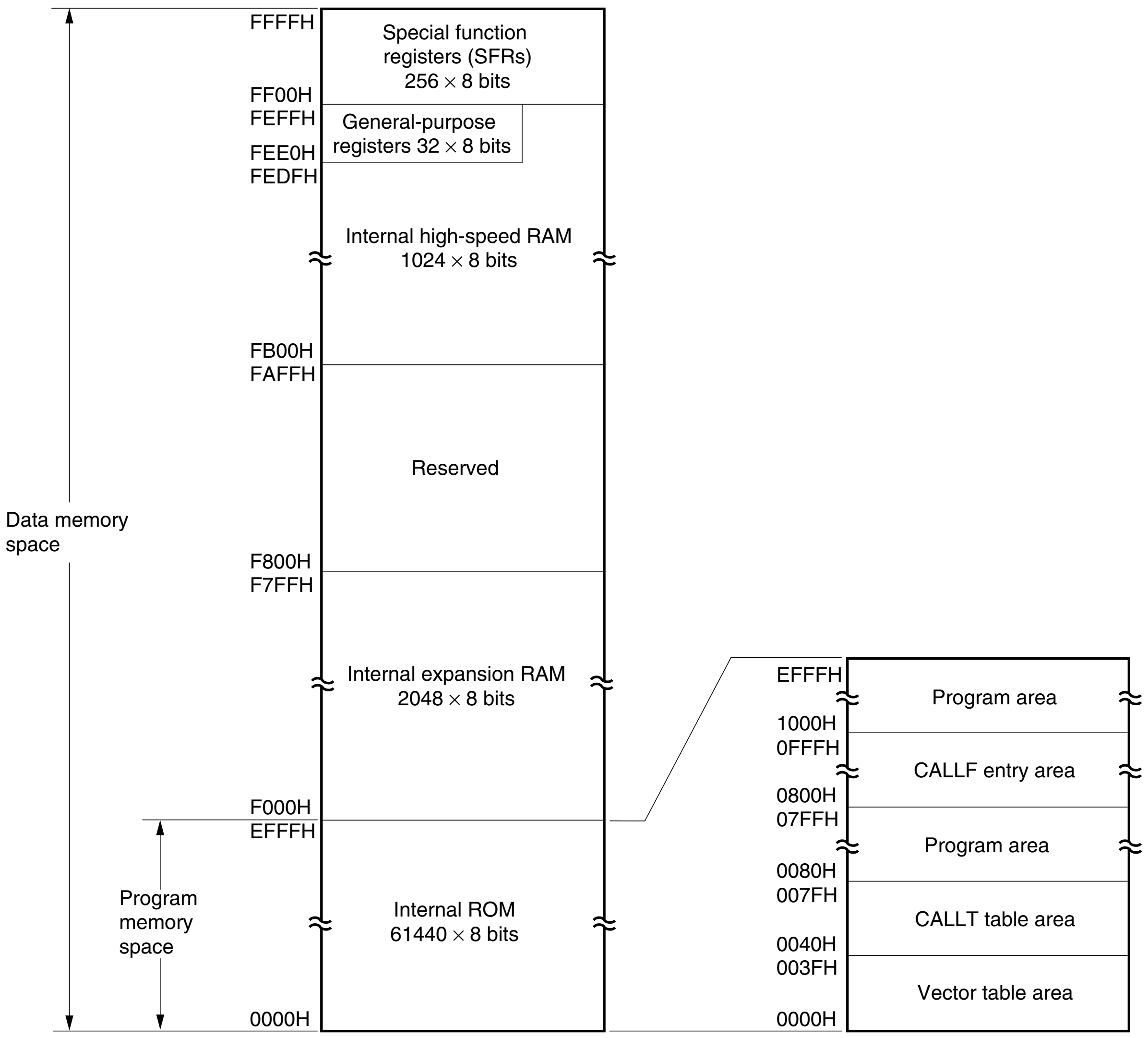
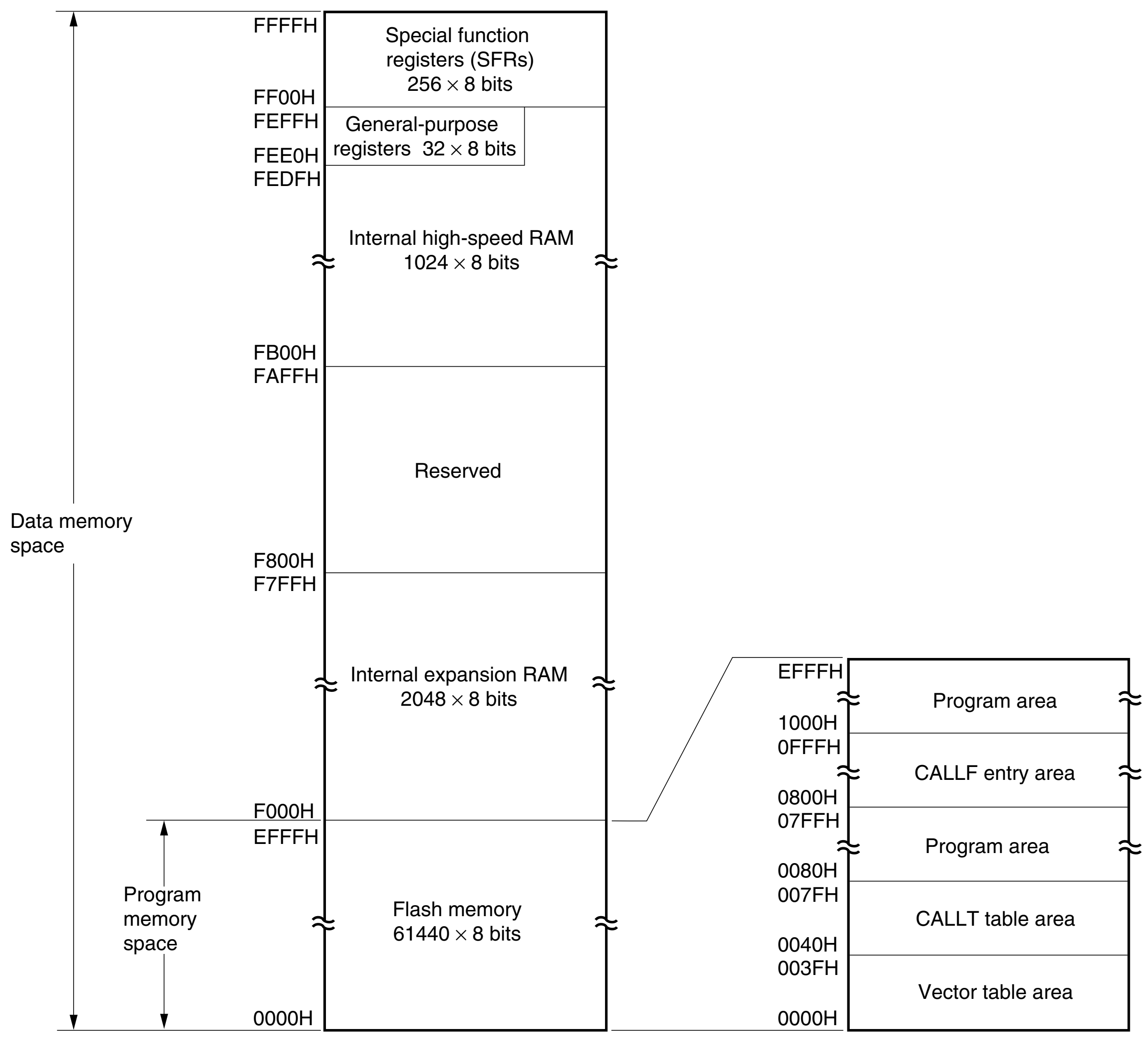


Figure 3-2. Memory Map (μPD78F0833Y)



3.1.1 Internal program memory space

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

The μPD780833Y Subseries products incorporate an on-chip ROM (or flash memory).

Product	Structure	Capacity
μPD780833Y	Mask ROM	61440 × 8 bits (0000H-EFFFH)
μPD78F0833Y	Flash memory	

The internal program memory space is divided into the following three 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 $\overline{\text{RESET}}$ input or generation of each interrupt request are stored in the vector table area. In a 16-bit address, the lower 8 bits are stored at even addresses and the higher 8 bits are stored at odd addresses.

Table 3-1. Vector Table

Vector Table Address	Interrupt Source
0000H	$\overline{\text{RESET}}$ input
0004H	INTWDT
0006H	INTP0
0008H	INTP1
000AH	INTP2
000CH	INTP3
000EH	INTP4
0010H	INTP5
0012H	INTP6
0014H	INTP7
0016H	INTSER0
0018H	INTSR0
001AH	INTST0
001CH	INTCSI30
001EH	INTCSI31
0020H	INTIIC0
0022H	INTC2
0024H	INTWINI0
0026H	INTTM000
0028H	INTTM010
002AH	INTTM001
002CH	INTTM011
002EH	INTAD00
0030H	INTAD01
0034H	INTWTN0
0036H	INTKR
0038H	INTTM50
003AH	INTTM51
003CH	INTTM52
003EH	BRK

(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) CALLF instruction entry area

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

3.1.2 Internal data memory space

In the μPD780833Y Subseries, the following on-chip RAM is included.

(1) Internal high-speed RAM

This RAM is configured as 1024 × 8 bits at addresses FB00H to FEFFH, from which the 32-byte area of FEE0H to FEFFH is allocated as 4 general-purpose register banks with eight 1-bit registers as 1 bank.

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

(2) Internal expansion RAM

This RAM includes 2048 bytes at addresses F000H to F7FFH, and is allocated as internal expansion RAM.

3.1.3 Special Function Register (SFR) area

On-chip peripheral hardware special function registers (SFRs) are allocated to the area of FF00H to FFFFH. (Refer to **Table 3-2 Special Function Register List** in **3.2.3 Special function registers (SFRs)**.)

Caution Do not access addresses to which an SFR is 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.

The address of the instruction to be executed next is addressed by the program counter (PC) (for details, see **3.3 Instruction Address Addressing**).

Several addressing modes are provided for addressing the memory relevant to the execution of instructions for the μPD780833Y Subseries, 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. Data memory addressing is illustrated in Figures 3-3 and 3-4. For details of each addressing mode, see **3.4 Operand Address Addressing**.

Figure 3-3. Data Memory Addressing (μPD780833Y)

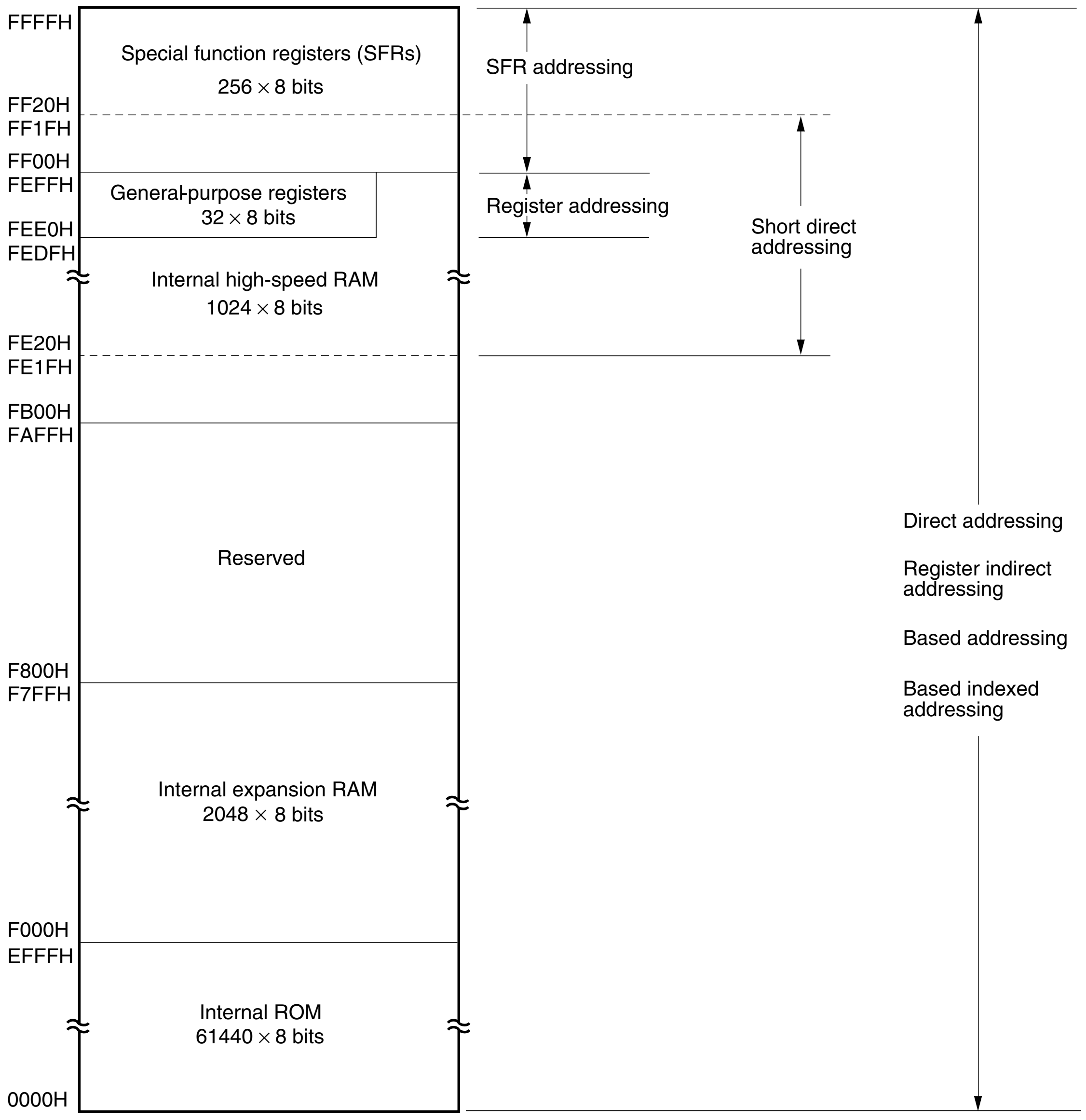
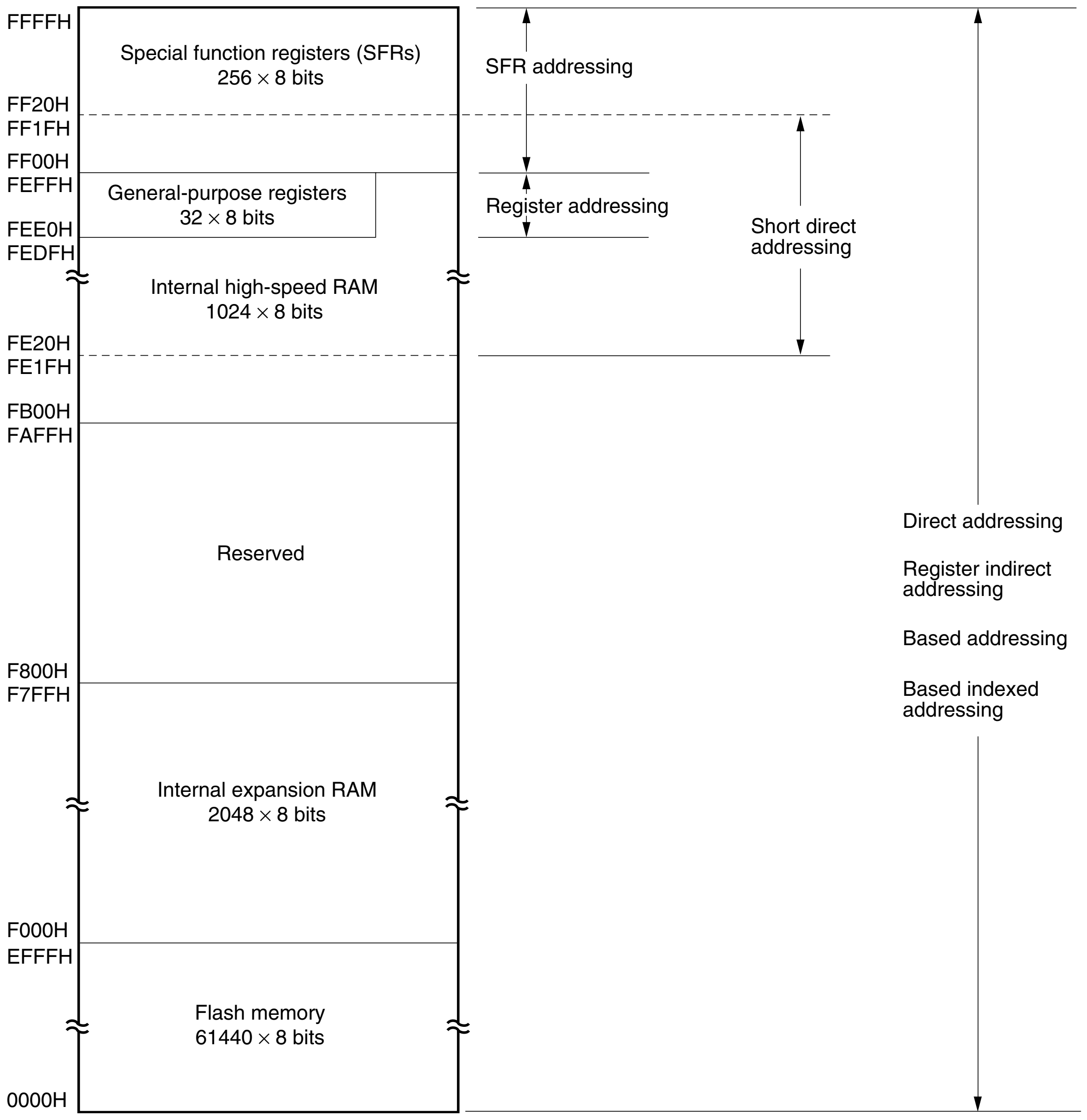

FFFFH
Special function registers (SFRs) 256 × 8 bits
SFR addressing
FF20H
FF1FH
FF00H
FEFFH
General-purpose registers 32 × 8 bits
Register addressing
Short direct addressing
FEE0H
FEDFH
Internal high-speed RAM 1024 × 8 bits
FE20H
FE1FH
FB00H
FAFFH
Reserved
Direct addressing
Register indirect addressing
Based addressing
Based indexed addressing
F800H
F7FFH
Internal expansion RAM 2048 × 8 bits
F000H
EFFFH
Internal ROM 61440 × 8 bits
0000H

Figure 3-4. Data Memory Addressing (μPD78F0833Y)



3.2 Processor Registers

The μPD780833Y Subseries products incorporate the following processor registers.

3.2.1 Control registers

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

(1) Program counter (PC)

The program counter is a 16-bit register which holds the address information of the next program to be executed. In normal operation, the 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.

$\overline{\text{RESET}}$ input sets the program counter to the reset vector table values at addresses 0000H and 0001H.

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 to be set/reset by instruction execution. Program status word contents are automatically stacked upon interrupt request generation or PUSH PSW instruction execution and are automatically reset upon execution of the RETB, RETI and POP PSW instructions.

$\overline{\text{RESET}}$ input sets the 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 is set to the disable interrupt (DI) state, and only non-maskable interrupt requests become acknowledgeable. Other interrupt requests are all disabled.

When 1, the IE is set to the enable interrupt (EI) state and interrupt request acknowledge enable is controlled by the in-service priority flag (ISP), interrupt mask flags for various interrupt sources, and a priority specification flag.

The IE 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 selection flags (RBS0 and RBS1)

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

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

(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, acknowledgement of low-level vectored interrupt requests specified by a priority specification flag register (PR0L, PR0H, PR1L, PR1H) (refer to **16.3 (3) Priority specification flag register (PR0L, PR0H, PR1L, PR1H)**) is disabled. When it is 1, all interrupts are acknowledgeable. 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 manipulation 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 (FB00H to FEFFH) can be set as the stack area.

Figure 3-7. Stack Pointer Format

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

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

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

Caution Since RESET input makes SP contents undefined, be sure to initialize the SP before instruction execution.

Figure 3-8. Data Saved to Stack Memory

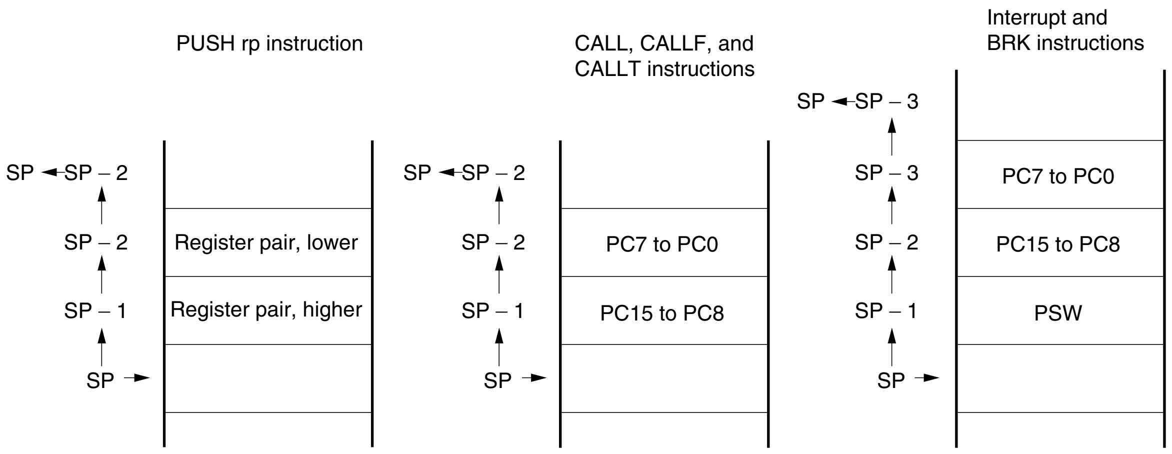
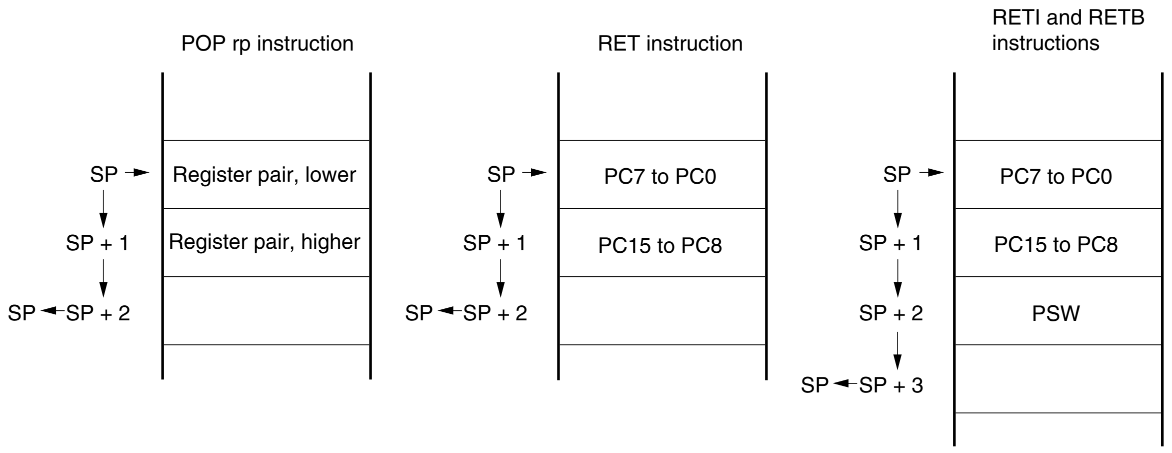


Figure 3-9. Data Restored from Stack Memory



3.2.2 General-purpose registers

The general-purpose registers are mapped at particular addresses (FEE0H to FEFFH) of the data memory. They consist of 4 banks, each bank consisting of eight 8-bit registers (X, A, C, B, E, D, L, and H).

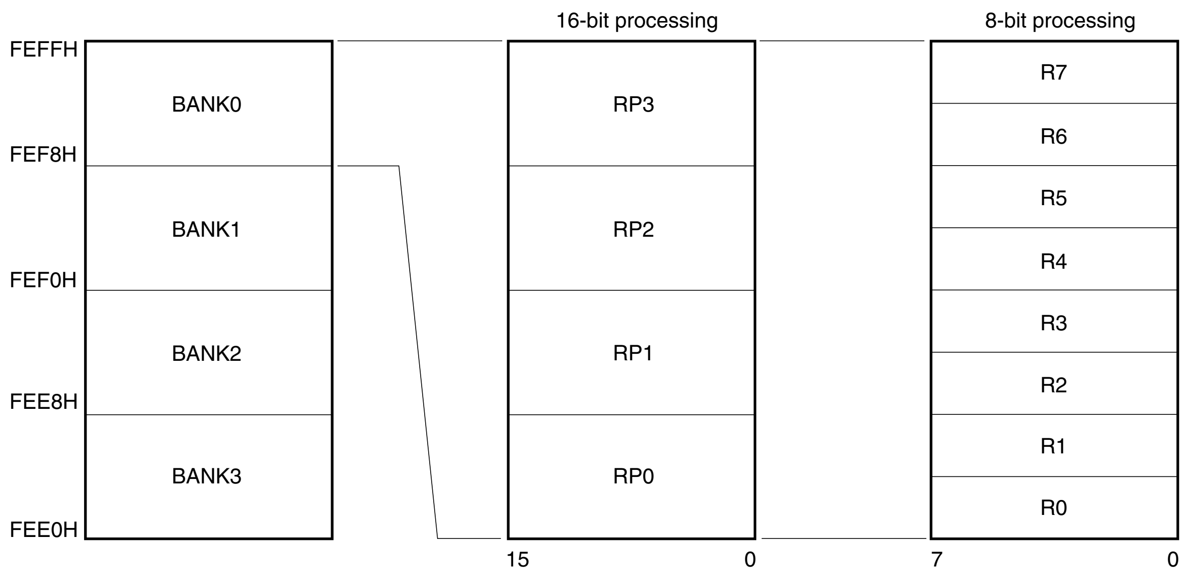
In addition to use as an 8-bit register, two 8-bit registers can also be used in pairs as a 16-bit register (AX, BC, DE, and HL).

They 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).

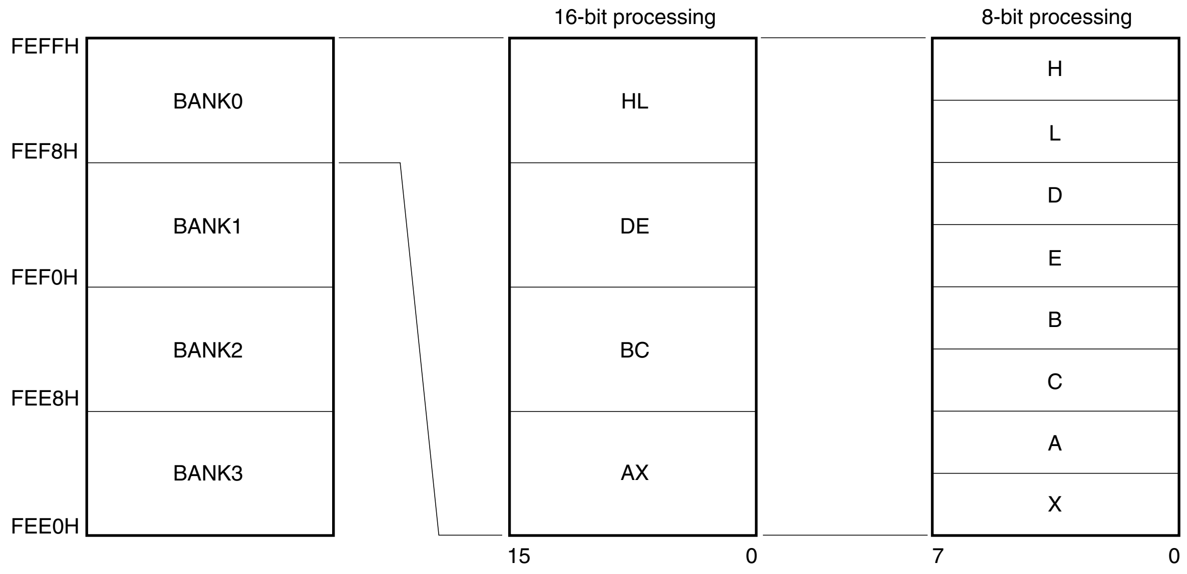
Register banks to be used for instruction execution are set by the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interrupts for each bank.

Figure 3-10. General-Purpose Register Configuration

(a) Absolute name



(b) Function name



3.2.3 Special function registers (SFRs)

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

The special function registers 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 unit, 1, 8, or 16 bits, depends on the special function register type.

Each manipulatable bit unit can be specified as follows.

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

Table 3-2 gives a list of 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 via the header file “sfrbit.h” in the CC78K0. When using the RA78K0, ID78K0-NS, ID78K0, or SM78K0, 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 enabled
  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 $\overline{\text{RESET}}$ input.

Table 3-2. Special Function Register List (1/3)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset
				1 Bit	8 Bits	16 Bits	
FF00H	Port 0	P0	R/W	√	√	—	00H
FF02H	Port 2	P2		√	√	—	
FF03H	Port 3	P3		√	√	—	
FF04H	Port 4	P4		√	√	—	Undefined
FF05H	Port 5	P5		√	√	—	
FF06H	Port 6	P6		√	√	—	
FF07H	Port 7	P7		√	√	—	00H
FF08H	Port 8	P8		√	√	—	
FF09H	Port 9	P9		√	√	—	
FF0AH	16-bit timer capture/compare register 000	CR000		—	—	√	0000H
FF0BH							
FF0CH	16-bit timer capture/compare register 010	CR010		—	—	√	
FF0DH							
FF0EH	16-bit timer counter 00	TM00	R	—	—	√	
FF0FH							
FF10H	16-bit timer capture/compare register 001	CR001	R/W	—	—	√	
FF11H							
FF12H	16-bit timer capture/compare register 011	CR011		—	—	√	
FF13H							
FF14H	16-bit timer counter 01	TM01	R	—	—	√	
FF15H							
FF17H	A/D conversion result register 00	ADCR00		—	√	—	Undefined
FF18H	Transmit shift register 0	TXS0	W	—	√	—	FFH
	Receive buffer register 0	RXB0	R	—	√	—	
FF1AH	Serial shift register 30	SIO30	R/W	—	√	—	00H
FF1BH	Serial shift register 31	SIO31		—	√	—	
FF1FH	IIC shift register 0	IIC0		—	√	—	
FF20H	Port mode register 0	PM0		√	√	—	FFH
FF22H	Port mode register 2	PM2		√	√	—	
FF23H	Port mode register 3	PM3		√	√	—	
FF24H	Port mode register 4	PM4		√	√	—	
FF25H	Port mode register 5	PM5		√	√	—	
FF26H	Port mode register 6	PM6		√	√	—	
FF27H	Port mode register 7	PM7		√	√	—	
FF28H	Port mode register 8	PM8		√	√	—	
FF29H	Port mode register 9	PM9		√	√	—	
FF30H	Pull-up resistor option register 0	PU0		√	√	—	00H
FF32H	Pull-up resistor option register 2	PU2		√	√	—	
FF33H	Pull-up resistor option register 3	PU3		√	√	—	
FF34H	Pull-up resistor option register 4	PU4		√	√	—	
FF35H	Pull-up resistor option register 5	PU5		√	√	—	

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset
					1 Bit	8 Bits	16 Bits	
FF36H	Pull-up resistor option register 6	PU6			√	√	—	00H
FF37H	Pull-up resistor option register 7	PU7			√	√	—	
FF40H	Clock output selection register	CKS			√	√	—	
FF41H	Watch timer mode control register 0	WTNM0			√	√	—	
FF42H	Watchdog timer clock selection register	WDCS			—	√	—	
FF47H	Memory expansion mode register	MEM			—	√	—	
FF48H	External interrupt rising edge enable register	EGP			√	√	—	
FF49H	External interrupt falling edge enable register	EGN			√	√	—	
FF60H	16-bit timer mode control register 00	TMC00			√	√	—	
FF61H	Prescaler mode register 00	PRM00			—	√	—	
FF62H	Capture/compare control register 00	CRC00			√	√	—	
FF63H	16-bit timer output control register 00	TOC00			√	√	—	
FF68H	16-bit timer mode control register 01	TMC01			√	√	—	
FF69H	Prescaler mode register 01	PRM01			—	√	—	
FF6AH	Capture/compare control register 01	CRC01			√	√	—	
FF6BH	16-bit timer output control register 01	TOC01			√	√	—	
FF70H	8-bit timer compare register 50	CR5	CR50		—	√	√	
FF71H	8-bit timer compare register 51		CR51		—	√		
FF72H	8-bit timer compare register 52	CR52			—	√	—	
FF74H	8-bit timer/counter 50	TM5	TM50	R	—	√	√	
FF75H	8-bit timer/counter 51		TM51		—	√		
FF76H	8-bit timer/counter 52	TM52			—	√	—	
FF78H	Timer clock selection register 50	TCL50		R/W	—	√	—	
FF79H	8-bit timer mode control register 50	TMC50			√	√	—	04H
FF7AH	Timer clock selection register 51	TCL51			—	√	—	00H
FF7BH	8-bit timer code control register 51	TMC51			√	√	—	04H
FF7CH	Timer clock selection register 52	TCL52			—	√	—	00H
FF7DH	8-bit timer mode control register 52	TMC52			√	√	—	04H
FF80H	A/D converter mode register 00	ADM00			√	√	—	00H
FF81H	Analog input channel specification register 00	ADS00			—	√	—	
FF88H	A/D converter mode register 01	ADM01			√	√	—	
FF89H	Analog input channel specification register 01	ADS01			—	√	—	
FF8BH	A/D conversion result register 01	ADCR01		R	—	√	—	Undefined
FFA0H	Asynchronous serial interface mode register 0	ASIM0		R/W	√	√	—	00H
FFA1H	Asynchronous serial interface status register 0	ASIS0		R	√	√	—	
FFA2H	Baud rate generator control register 0	BRGC0		R/W	—	√	—	
FFA7H	IIC function expansion register 0	IICX0			√	√	—	
FFA8H	IIC control register 0	IICC0			√	√	—	
FFA9H	IIC status register 0	IICS0		R	√	√	—	

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset
					1 Bit	8 Bits	16 Bits	
FFAAH	IIC transfer clock selection register 0	IICCL0		R/W	√	√	—	00H
FFABH	Slave address register 0	SVA0			—	√	—	
FFB0H	Serial operation mode register 30	CSIM30			√	√	—	
FFB8H	Serial operation mode register 31	CSIM31			√	√	—	
FFC0H	Class 2 control register 1	C2CT1		R/W	√	√	—	00H
FFC1H	Class 2 control register 2	C2CT2			√	√	—	
FFC2H	Class 2 status register 1	C2ST1		R	√	√	—	C0H
FFC3H	Class 2 status register 2	C2ST2			√	√	—	00H
FFC4H	Class 2 transmit FIFO register	C2TXFIFO		W	—	√	—	
	Class 2 receive FIFO register	C2RXFIFO		R	—	√	—	
FFC5H	Class 2 rise time propagation delay correction register	C2PDR		R/W	—	√	—	
FFC6H	Class 2 fall time propagation delay correction register	C2PDF			—	√	—	
FFC7H	Class 2 clock selection register	C2CLK			√	√	—	
FFCDH	Flash programming mode control register[Note 1]	FLPMC			√	√	—	08H[Note 2]
FFD0H to FFDFH	External access area[Note 3]				√	√	—	Undefined
FFE0H	Interrupt request flag register 0L	IF0	IF0L		√	√	√	00H
FFE1H	Interrupt request flag register 0H		IF0H		√	√		
FFE2H	Interrupt request flag register 1L	IF1	IF1L		√	√	√	
FFE3H	Interrupt request flag register 1H		IF1H		√	√		DFH[Note 4]
FFE4H	Interrupt mask flag register 0L	MK0	MK0L		√	√	√	FFH
FFE5H	Interrupt mask flag register 0H		MK0H		√	√		
FFE6H	Interrupt mask flag register 1L	MK1	MK1L		√	√	√	
FFE7H	Internal mask flag register 1H		MK1H		√	√		
FFE8H	Priority level specification flag register 0L	PR0	PR0L		√	√	√	
FFE9H	Priority level specification flag register 0H		PR0H		√	√		
FFEAH	Priority level specification flag register 1L	PR1	PR1L		√	√	√	
FFEBH	Priority level specification flag register 1H		PR1H		√	√		
FFF0H	Memory size switching register	IMS			—	√	—	CFH
FFF4H	Internal expansion RAM size switching register	IXS			—	√	—	0CH[Note 5]
FFF9H	Watchdog timer mode register	WDTM			√	√	—	00H
FFFAH	Oscillation stabilization time selection register	OSTS			—	√	—	04H
FFFBH	Processor clock control register	PCC			√	√	—	

Notes 1. Only the μPD78F0833Y incorporates this register.

2. This value changes depending on the status of the V_{PP} pin. It is set to 08H when high voltage is not applied to the V_{PP} pin and set to 0CH when high voltage is applied to the V_{PP} pin.

3. The external access area cannot be accessed by SFR addressing. Access it using the direct addressing method.

4. Bit 6 is fixed to 1 and bit 5 is fixed to 0 when reading.

5. The initial value is 0CH, but it must be set to 08H before use.

3.3 Instruction Address Addressing

An instruction address is determined by the program counter (PC) contents 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 the PC and branched by the following addressing. (For details of instructions, refer to the **78K/0 User's Manual Instruction (U12326E)**).

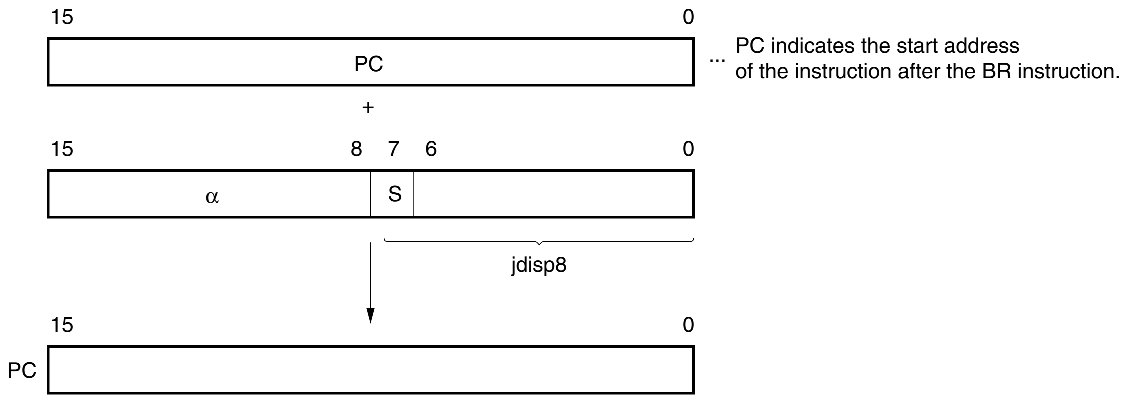
3.3.1 Relative addressing

[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]


15
0
PC
... PC indicates the start address of the instruction after the BR instruction.
+
15
8
7
6
0
α
S
jdisp8
15
0
PC

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

3.3.2 Immediate addressing

[Function]

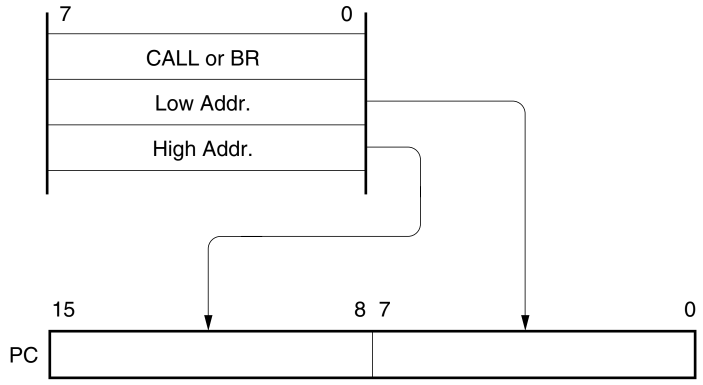
Immediate data in the instruction word is transferred to the program counter (PC) and branched.

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

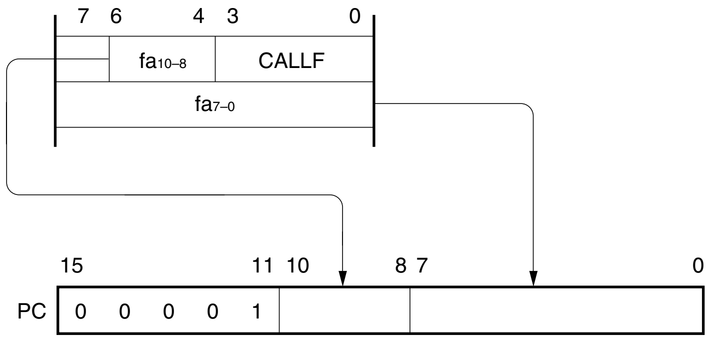
The CALL !addr16 and BR !addr16 instructions can be branched to the entire memory space. The CALLF !addr11 instruction is branched to the 0800H to 0FFFH area.

[Illustration]

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



In the case of the CALLF !addr11 instruction



3.3.3 Table indirect addressing

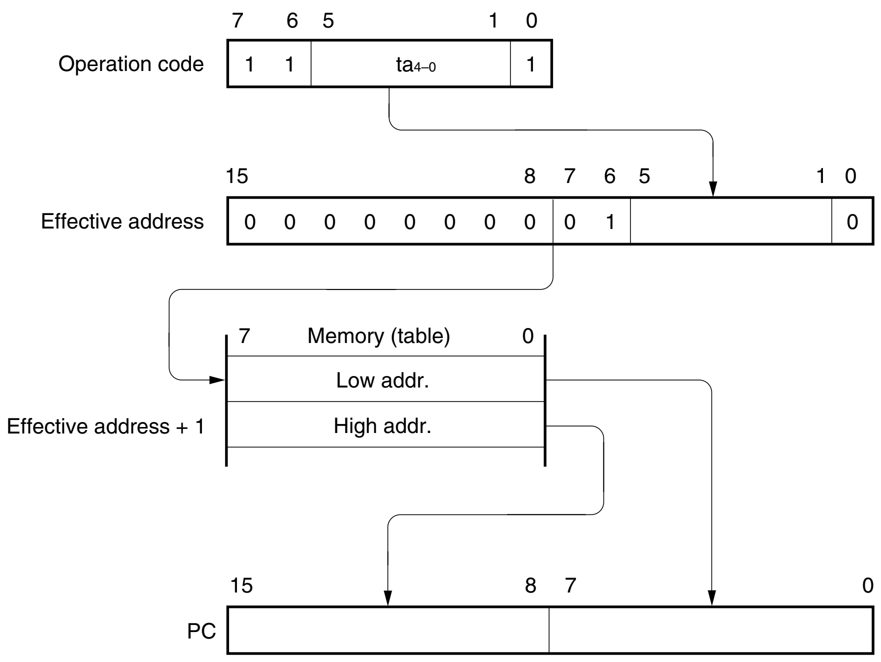
[Function]

The 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 stored in the memory table from 40H to 7FH, and allows branching to the entire memory space.

[Illustration]



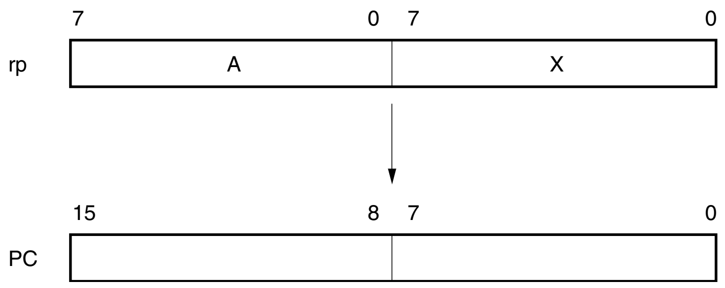
3.3.4 Register addressing

[Function]

The register pair (AX) contents to be specified by 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 (addressing) are available to specify the register and memory to undergo manipulation during instruction execution.

3.4.1 Implied addressing

[Function]

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

Of the μPD780833Y Subseries 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 employed using 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 register A and register X 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 specification code (Rn and RPn) of an instruction word in the registered bank specified by the register bank selection flag (RBS0 to RBS1).

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 by 3 bits in the operation code.

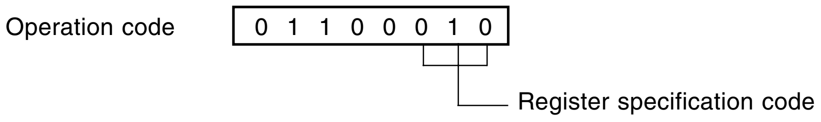
[Operand format]

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

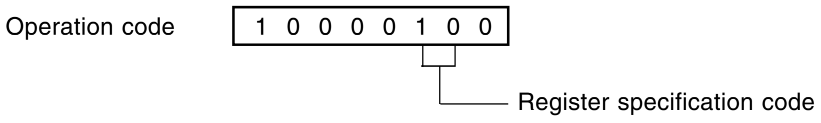
'r' and 'rp' can be described with absolute names (R0 to R7 and RP0 to RP3) as well as function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL).

[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 indicated by the immediate data in an instruction word is directly addressed.

[Operand format]

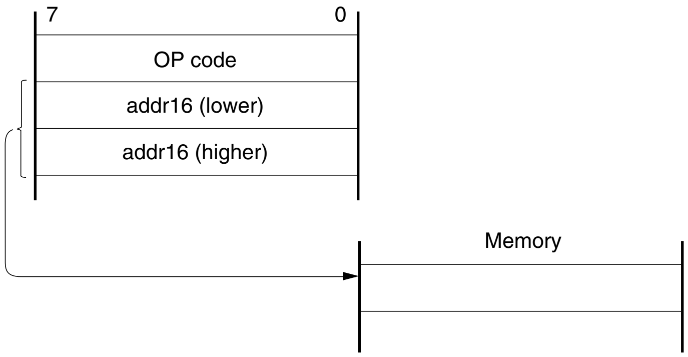
Identifier	Description
addr16	Label or 16-bit immediate data

[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 RAM and special function registers (SFRs) are mapped at FE20H to FEFFH and FF00H to FF1FH, respectively.

The SFR area (FF00H to FF1FH) to which short direct addressing is applied is part of the entire SFR area. Ports which are frequently accessed in a program and compare registers and capture registers of timer/event counters are mapped in this area and these SFRs can 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. Refer to the **[Illustration]** on the next page.

[Operand format]

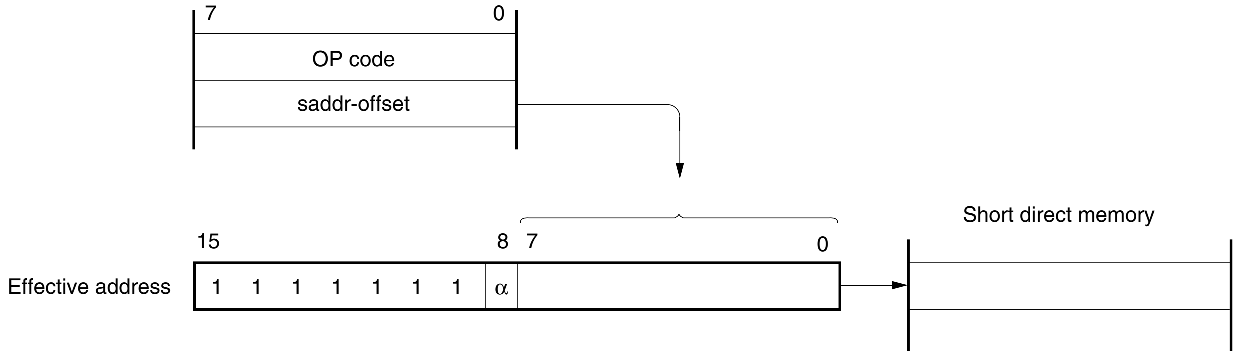
Identifier	Description
saddr	Label of FE20H to FF1FH immediate data
saddrp	Label of FE20H to FF1FH immediate data (even address only)

[Description example]

MOV 0FE30H, #50H; when setting saddr to FE30H and immediate data to 50H

Operation code	0 0 0 1 0 0 0 1	OP code
	0 0 1 1 0 0 0 0	30H (saddr-offset)
	0 1 0 1 0 0 0 0	50H (immediate data)

[Illustration]



When 8-bit immediate data is 20H to FFH, α = 0

When 8-bit immediate data is 00H to 1FH, α = 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, SFRs mapped at FF00H to FF1FH can be accessed with short direct addressing.

[Operand format]

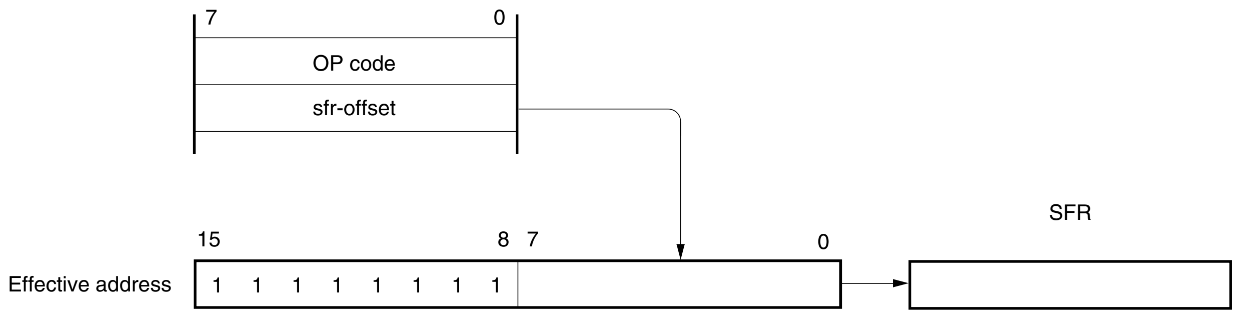
Identifier	Description
sfr	Special function register name
sfrp	16-bit manipulatable special function register name (even address only)

[Description example]

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

Operation code	1 1 1 1 0 1 1 0	OP code
	0 0 1 0 0 0 0 0	20H (sfr-offset)

[Illustration]



3.4.6 Register indirect addressing

[Function]

The register pair contents specified by a register pair specification code in an instruction word of a register bank specified by a register bank selection flag (RBS0 and RBS1) serve as an operand address for addressing the memory to be manipulated. This addressing can be carried out for all the memory spaces.

[Operand format]

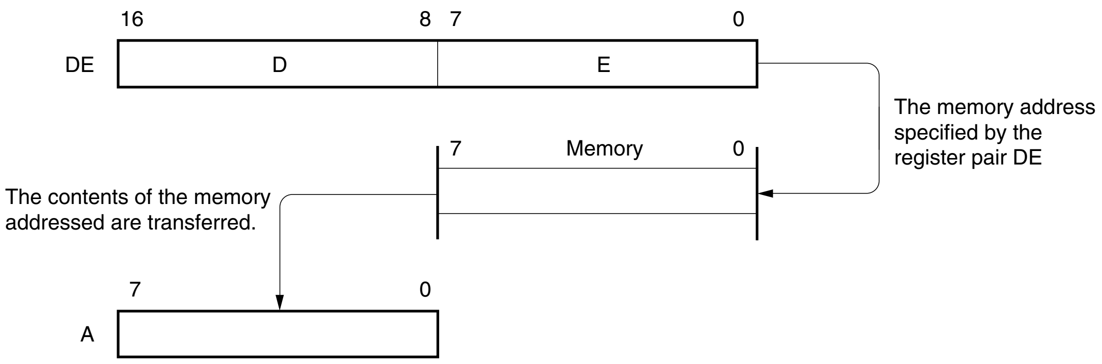
Identifier	Description
—	[DE], [HL]

[Description example]

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

Operation code $\boxed{1\ 0\ 0\ 0\ 0\ 1\ 0\ 1}$

[Illustration]


16
8
7
0
DE
D
E
The memory address specified by the register pair DE
7
Memory
0
The contents of the memory addressed are transferred.
7
0
A

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 an instruction word of the register bank specified by the register bank selection 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 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

3.4.8 Based indexed addressing

[Function]

The B or C register contents specified in an instruction are added to the contents of the base register, that is, the HL register pair in an instruction word of the register bank specified by the register bank selection 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 the memory spaces.

[Operand format]

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

[Description example]

In the case of MOV A, [HL + B]

Operation code $\boxed{1\ 0\ 1\ 0\ 1\ 0\ 1\ 1}$

3.4.9 Stack addressing

[Function]

The stack area is indirectly addressed by 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/restored upon generation of an interrupt request.
Stack addressing addresses the internal high-speed RAM area only.

[Description example]

In the case of PUSH DE

Operation code $\boxed{1\ 0\ 1\ 1\ 0\ 1\ 0\ 1}$

CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

The μPD780833Y Subseries products incorporate 65 I/O ports. Figure 4-1 shows the port configuration. Every port is capable of 1-bit and 8-bit manipulations and can carry out a wide variety of control operations. In addition to port functions, the ports can also serve as on-chip hardware I/O pins.

Figure 4-1. Port Types

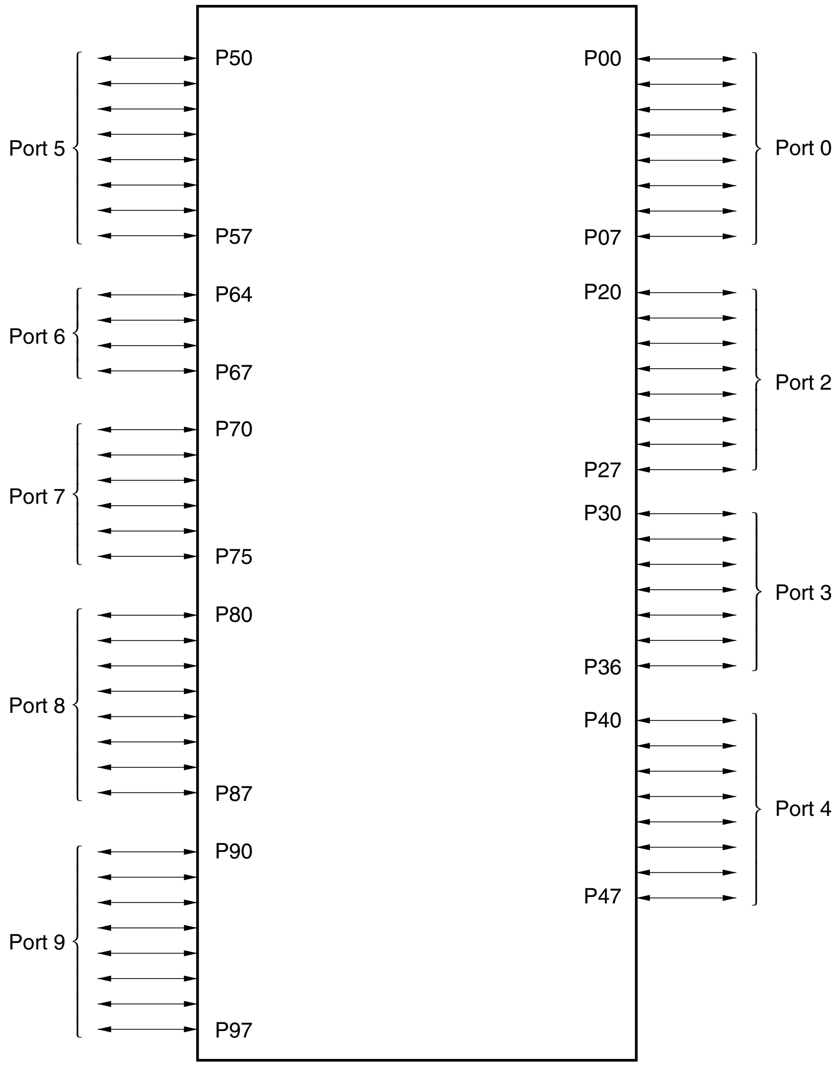


Table 4-1. Port Functions (1/2)

Pin Name	Function		Alternate Function
P00 to P07	Port 0 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.		INTP0 to INTP7
P20	Port 2 8-bit I/O port Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.		SI31
P21			SO31
P22			$\overline{\text{SCK31}}$
P23			TI50/TO50
P24			RxD0
P25			TxD0
P26			ASCK0/TI52/TO52
P27			TI51/TO51
P30	Port 3 7-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.	SI30
P31			SO30
P32			$\overline{\text{SCK30}}$
P33		N-ch open-drain I/O port. LEDs can be driven directly.	—
P34		Use of an on-chip pull-up resistor can be specified by a software setting.	TO00
P35			TI000
P36			TI010
P40 to P47	Port 4 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting. Interrupt request flag (KRIF) is set to 1 by falling edge detection.		—
P50 to P57	Port 5 8-bit I/O port. TTL level input/CMOS output. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.		—
P64 to P67	Port 6 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.		—

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

Pin Name	Function		Alternate Function
P70	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.	PCL
P71			
P72		N-ch open-drain I/O port.	SDA0
P73			SCL0
P74		Use of an on-chip pull-up resistor can be specified by a software setting.	TO01
P75			TI011
P80 to P87	Port 8 8-bit I/O port. Input/output can be specified in 1-bit units.		ANI01 to ANI71
P90 to P97	Port 9 8-bit I/O port. Input/output can be specified in 1-bit units.		ANI00 to ANI70

4.2 Port Configuration

The ports include the following hardware.

Table 4-2. Port Configuration

Item	Configuration
Control registers	Port mode register (PMm: m = 0, 2 to 9) Pull-up resistor option register (PUm,: m = 0, 2 to 7)
Ports	Total: 65 ports (I/O)
Pull-up resistors	Total: 46

4.2.1 Port 0

Port 0 is an 8-bit I/O port with an output latch. Input/output can be specified in 1-bit units using port mode register 0 (PM0). When port 0 is used as an input port, on-chip pull-up resistors can be connected in 6-bit units by setting pull-up resistor option register 0 (PU0).

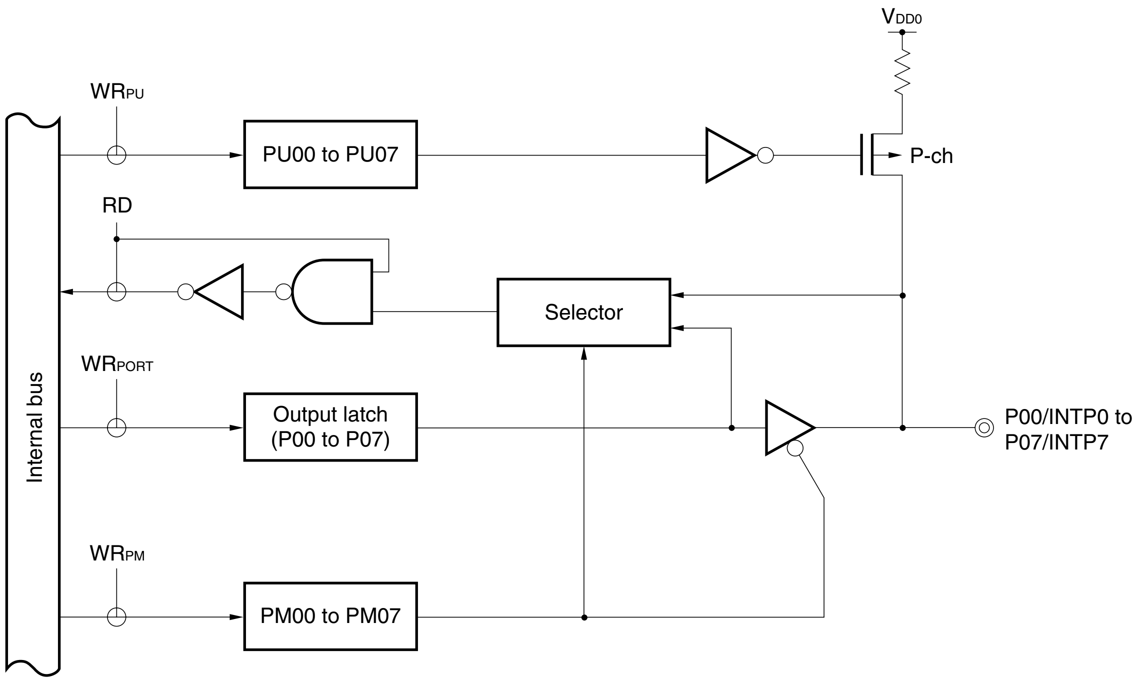
Port 0 can also be used for external interrupt request input.

$\overline{\text{RESET}}$ input sets port 0 to input mode.

Figure 4-2 shows a block diagram of port 0.

Caution Because port 0 can also be used for external interrupt request input, when the port function output mode is specified and the output level is changed, the interrupt request flag is set. Thus, when the output mode is used, set the interrupt mask flag to 1.

Figure 4-2. Block Diagram of P00 to P07



PU: Pull-up resistor option register
PM: Port mode register
RD: Port 0 read signal
WR: Port 0 write signal

4.2.2 Port 2

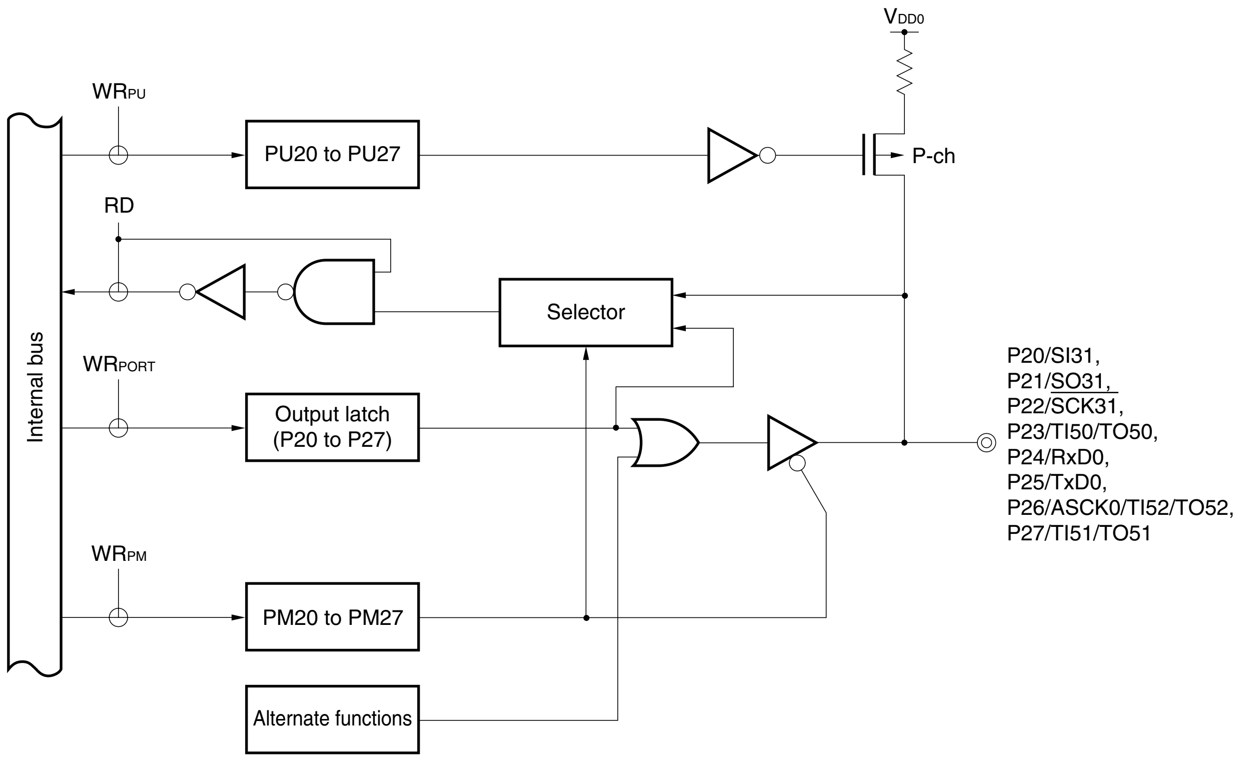
Port 2 is an 8-bit I/O port with an output latch. Input/output can be specified in 1-bit units using port mode register 2 (PM2). On-chip pull-up resistors can be connected to port 2 in 1-bit units by setting pull-up resistor option register 2 (PU2).

Port 2 can also be used for serial interface data I/O, clock I/O, and timer I/O.

$\overline{\text{RESET}}$ input sets port 2 to input mode.

Figure 4-3 shows a block diagram of port 2.

Figure 4-3. Block Diagram of P20 to P27



VDD0
WRPU
PU20 to PU27
P-ch
RD
Selector
Internal bus
WRPORT
Output latch (P20 to P27)
P20/SI31, P21/SO31, P22/SCK31, P23/TI50/TO50, P24/RxD0, P25/TxD0, P26/ASCK0/TI52/TO52, P27/TI51/TO51
WRPM
PM20 to PM27
Alternate functions

PU: Pull-up resistor option register
PM: Port mode register
RD: Port 2 read signal
WR: Port 2 write signal

4.2.3 Port 3

Port 3 is a 7-bit I/O port with an output latch. Input/output can be specified in 1-bit units using port mode register 3 (PM3).

On-chip pull-up resistors can be connected to port 3 in 1-bit units by setting pull-up resistor option register 3 (PU3).

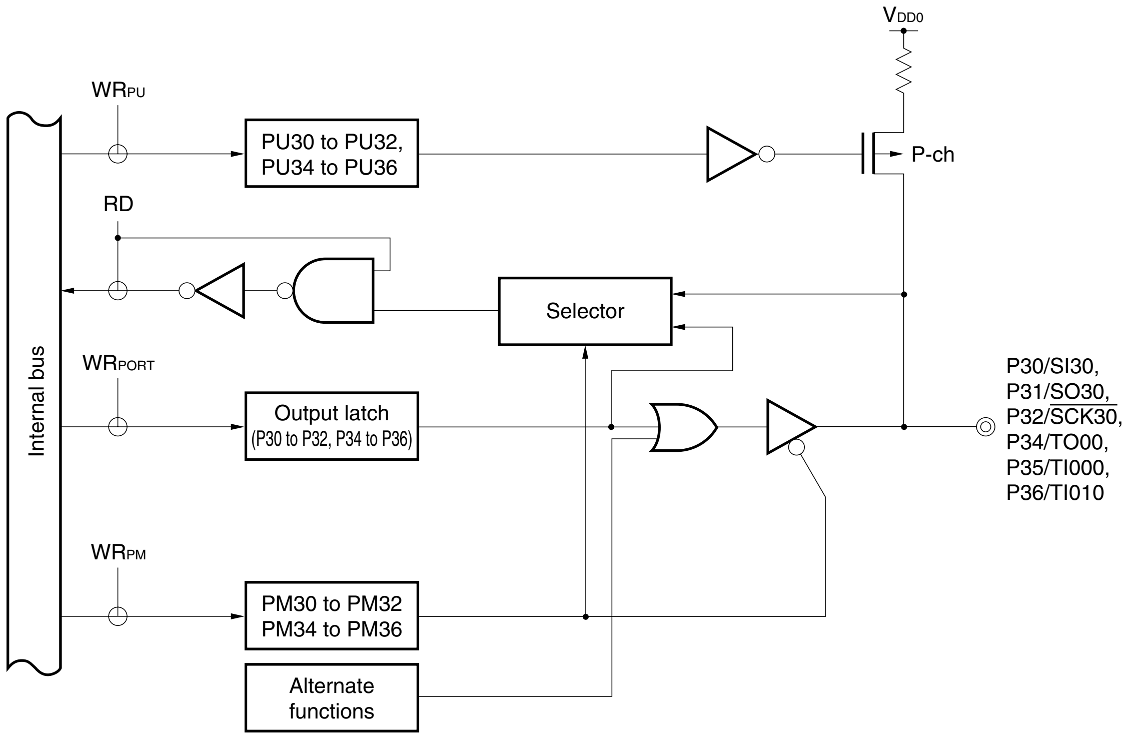
Pin P33 can drive LEDs directly.

Port 3 can also be used for serial interface data I/O, clock I/O and timer I/O.

$\overline{\text{RESET}}$ input sets port 3 to input mode.

Figures 4-4 and 4-5 show a block diagram of port 3.

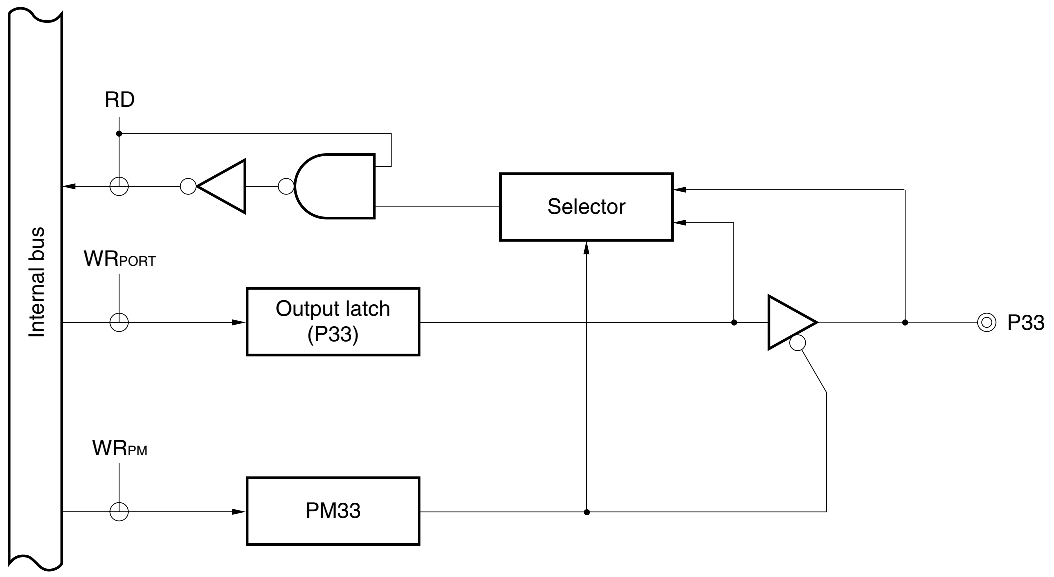
Figure 4-4. Block Diagram of P30 to P32 and P34 to P36



VDD0
WRPU
PU30 to PU32, PU34 to PU36
P-ch
RD
Selector
Internal bus
WRPORT
Output latch (P30 to P32, P34 to P36)
P30/SI30, P31/SO30, P32/SCK30, P34/TO00, P35/TI000, P36/TI010
WRPM
PM30 to PM32 PM34 to PM36
Alternate functions

PU: Pull-up resistor option register
PM: Port mode register
RD: Port 3 read signal
WR: Port 3 write signal

Figure 4-5. Block Diagram of P33



PM: Port mode register
RD: Port 3 read signal
WR: Port 3 write signal

4.2.4 Port 4

Port 4 is an 8-bit I/O port with an output latch. Input/output can be specified in 1-bit units using port mode register 4 (PM4). On-chip pull-up resistors can be connected to port 4 in 1-bit units by setting pull-up resistor option register 4 (PU4).

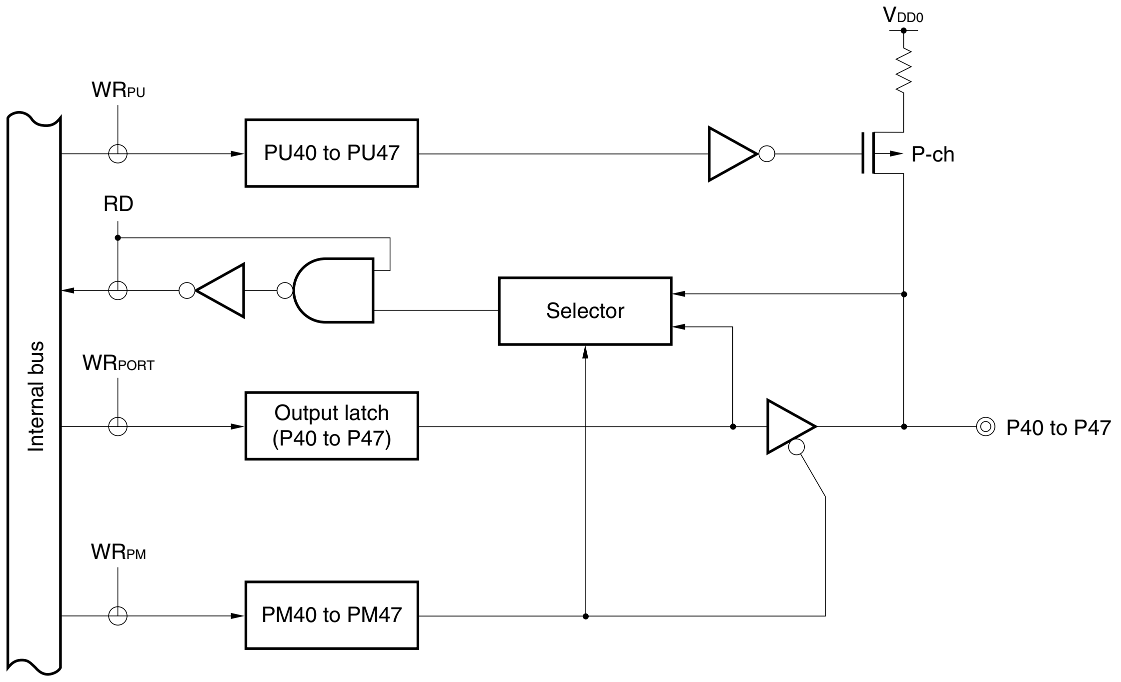
The interrupt request flag (KRIF) can be set to 1 by falling edge detection.

Falling edge detection is controlled by the memory expansion mode register (MEM).

$\overline{\text{RESET}}$ input sets port 4 to input mode.

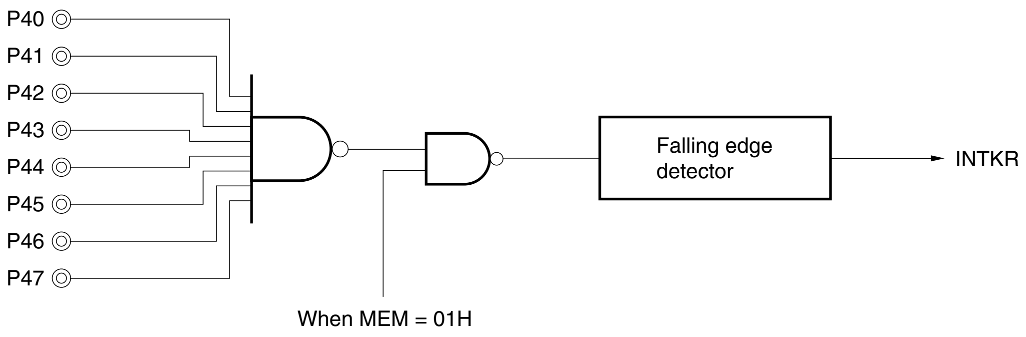
Figures 4-6 and 4-7 show a block diagram of port 4 and block diagram of the falling edge detector, respectively.

Figure 4-6. Block Diagram of P40 to P47



PU:	Pull-up resistor option register
PM:	Port mode register
RD:	Port 4 read signal
WR	Port 4 write signal

Figure 4-7. Block Diagram of Falling Edge Detector



4.2.5 Port 5

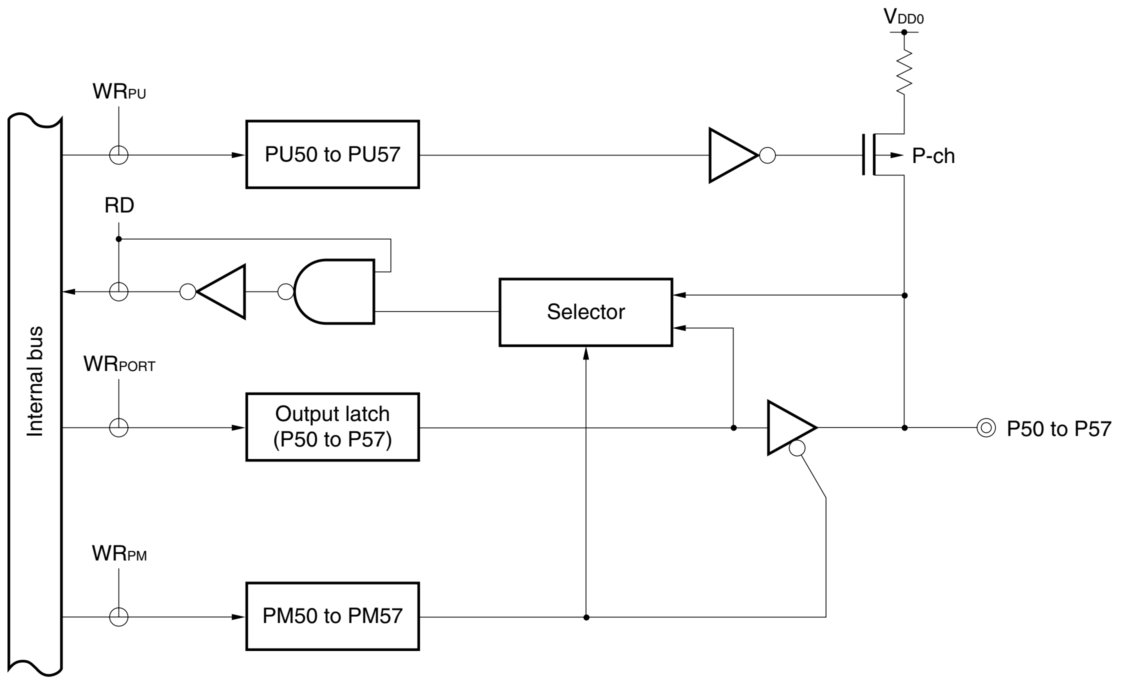
Port 5 is an 8-bit I/O port with an output latch. Input/output can be specified in 1-bit units using port mode register 5 (PM5). On-chip pull-up resistors can be connected to port 5 in 1-bit units by setting pull-up resistor option register 5 (PU5).

Port 5 is a TTL level input.

$\overline{\text{RESET}}$ input sets port 5 to input mode.

Figure 4-8 shows a block diagram of port 5.

Figure 4-8. Block Diagram of P50 to P57



PU: Pull-up resistor option register
PM: Port mode register
RD: Port 5 read signal
WR: Port 5 write signal

4.2.6 Port 6

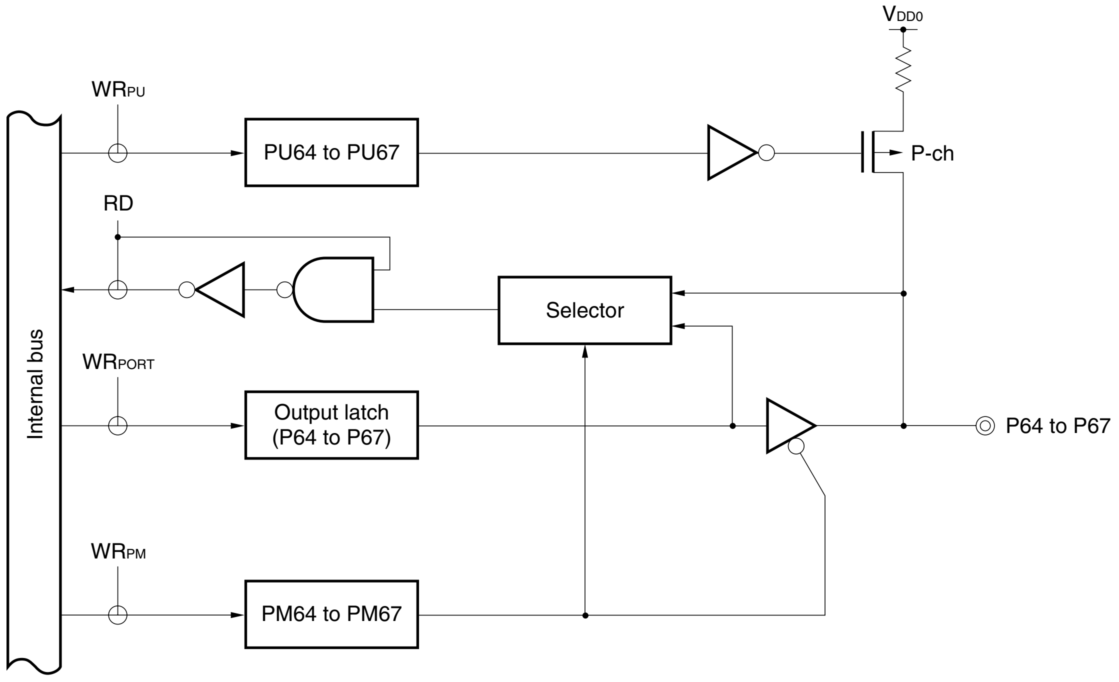
Port 6 is a 4-bit I/O port with an output latch. Input/output can be specified in 1-bit units using port mode register 6 (PM6).

On-chip pull-up resistors can be connected to port 6 in 1-bit units by setting pull-up resistor option register 6 (PU6).

$\overline{\text{RESET}}$ input sets port 6 to input mode.

Figure 4-9 shows a block diagram of port 6.

Figure 4-9. Block Diagram of P64 to P67



PU: Pull-up resistor option register
PM: Port mode register
RD: Port 6 read signal
WR: Port 6 write signal

4.2.7 Port 7

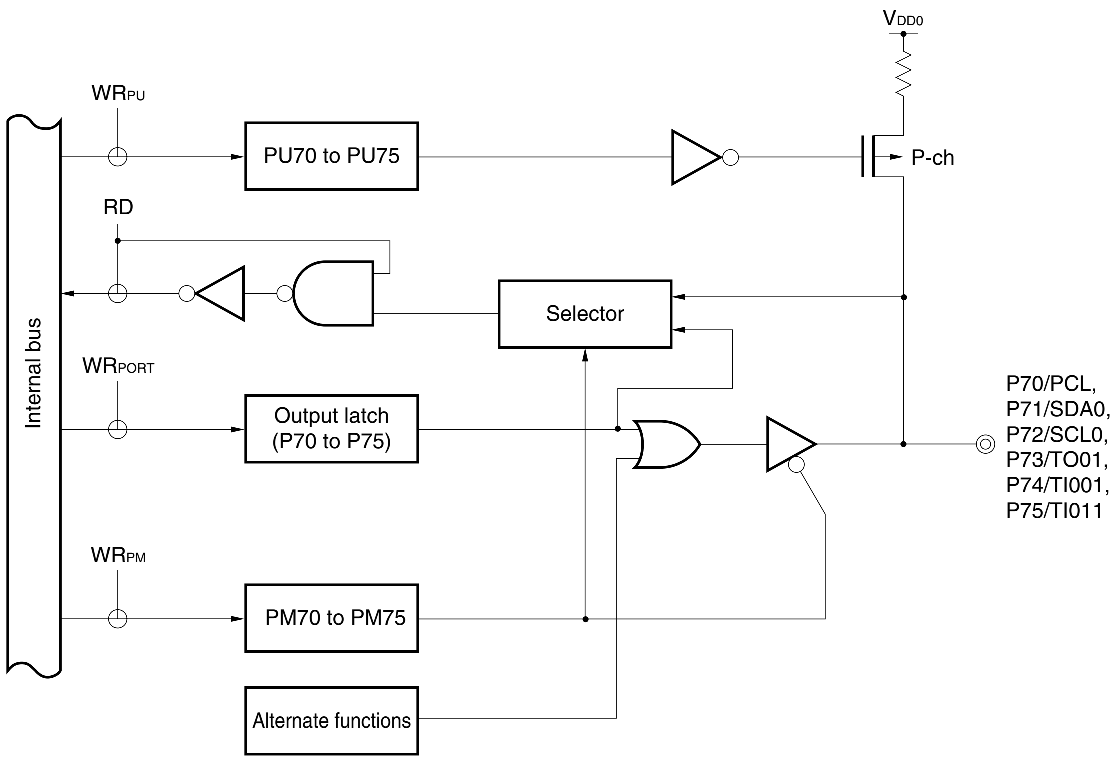
Port 7 is a 6-bit I/O port with an output latch. Input/output can be specified in 1-bit units using port mode register 7 (PM7). On-chip pull-up resistors can be connected to port 7 in 1-bit units by setting pull-up resistor option register 7 (PU7).

Port 7 can also be used for timer I/O, clock output, serial interface serial data I/O and serial clock I/O.

$\overline{\text{RESET}}$ input sets port 7 to input mode.

Figure 4-10 shows a block diagram of port 7.

Figure 4-10. Block Diagram of P70 to P75



PU: Pull-up resistor option register
PM: Port mode register
RD: Port 7 read signal
WR: Port 7 write signal

4.2.8 Port 8

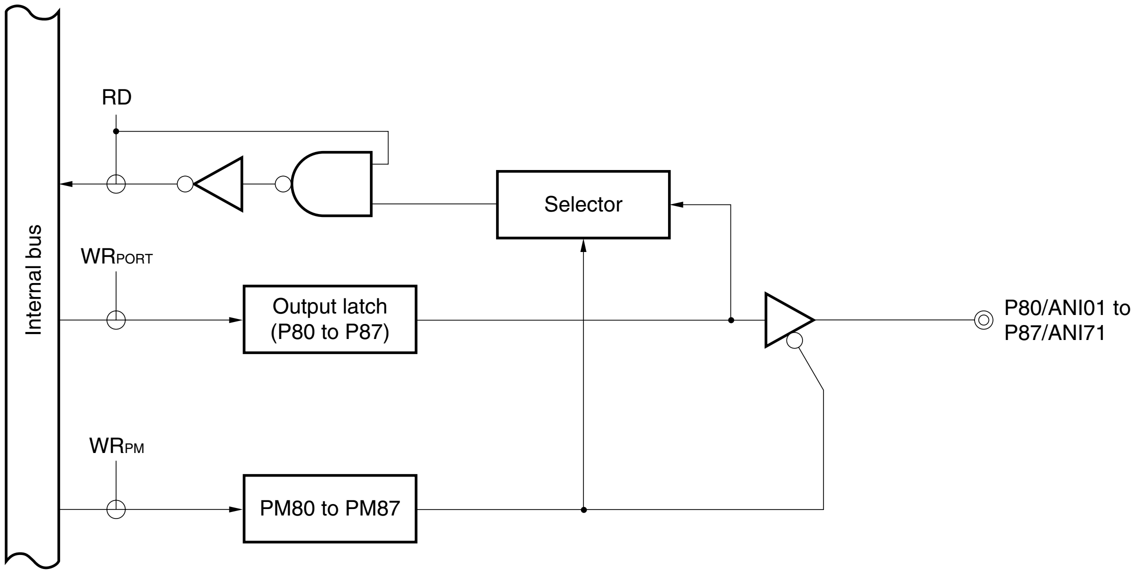
Port 8 is an 8-bit I/O port with an output latch. Input/output can be specified in 1-bit units using port mode register 8 (PM8).

Port 8 can also be used for A/D converter (AD01) analog input.

$\overline{\text{RESET}}$ input sets port 8 to input mode.

Figure 4-11 shows a block diagram of port 8.

Figure 4-11. Block Diagram of P80 to P87



PM: Port mode register
RD: Port 8 read signal
WR: Port 8 write signal

4.2.9 Port 9

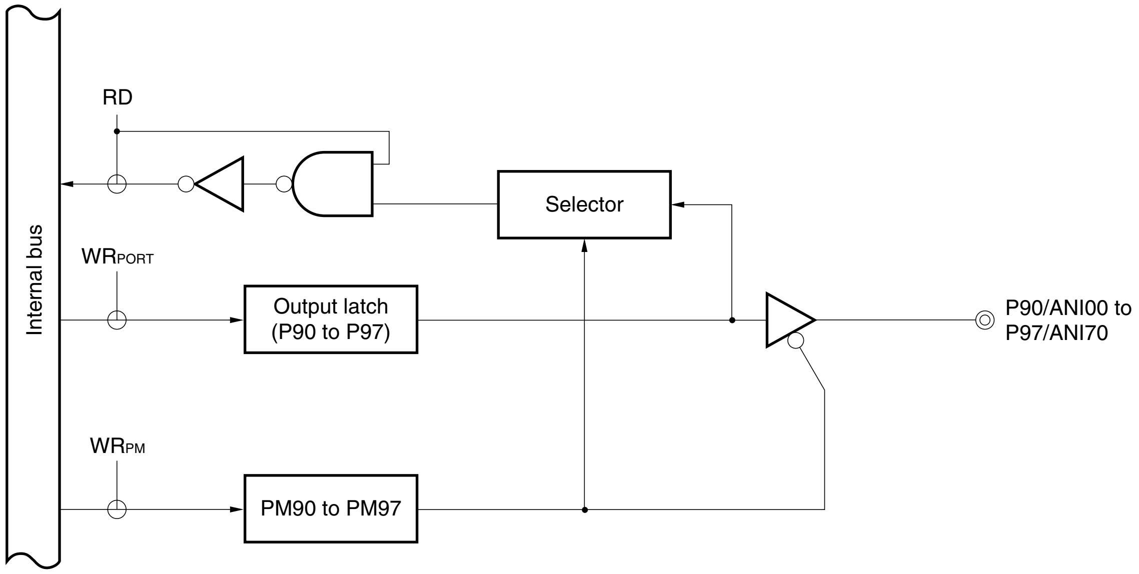
Port 9 is an 8-bit I/O port with an output latch. Input/output can be specified in 1-bit units with port mode register 9 (PM9).

Port 9 can also be used for A/D converter (AD00) analog input.

$\overline{\text{RESET}}$ input sets port 9 to input mode.

Figure 4-12 shows a block diagram of port 9.

Figure 4-12. Block Diagram of P90 to P97



PM: Port mode register
RD: Port 9 read signal
WR: Port 9 write signal

4.3 Registers Controlling Port Function

The following three types of registers are used to control the ports.

- Port mode registers (PM0, PM2 to PM9)
- Pull-up resistor option registers (PU0, PU2 to PU7)
- Memory expansion mode register (MEM)

(1) Port mode registers (PM0, PM2 to PM9)

These registers set port input/output in 1-bit units.
PM0 and PM2 to PM9 are independently set using a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets these registers to FFH.

Cautions 1. As port 0 has an alternate function as an external interrupt input, when the port function output mode is specified and the output level is changed, the interrupt request flag is set. When the output mode is used, therefore, the interrupt mask flag should be preset to 1.

★ **2. If a port has alternate function pins and it is used as an alternate output function, set the output latches (P2, P3, and P7) to 0.**

Figure 4-13. Format of Port Mode Registers (PM0, PM2 to PM9)

Address: FF20H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00

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: FF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	PM36	PM35	PM34	PM33	PM32	PM31	PM30

Address: FF24H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40

Address: FF25H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50

Address: FF26H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	PM67	PM66	PM65	PM64	1	1	1	1

Address: FF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	1	1	PM75	PM74	PM73	PM72	PM71	PM70

Address: FF28H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM8	PM87	PM86	PM85	PM84	PM83	PM82	PM81	PM80

Address: FF29H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM9	PM97	PM96	PM95	PM94	PM93	PM92	PM91	PM90

PMmn	Selection of Pmn pin I/O mode (m = 0, 2 to 7, n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

(2) Pull-up resistor option registers (PU0, PU2 to PU7)

These registers are used to set whether to use an internal pull-up resistor at each port or not. Setting PU0 and PU2 to PU7 enables the use of on-chip pull-up resistors at the corresponding port pins.
PU0 and PU2 to PU7 are set using a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets these registers to 00H.

Cautions 1. Pins P33, P71, P72, P80 to P87, and P90 to P97 do not incorporate pull-up resistors.

★ **2. When PUm is set to 1, an on-chip pull-up resistor is connected irrespective of the input/output mode. To use the register in output mode, therefore, set the bits of PUm to 0 (m = 0, 2 to 7).**

Figure 4-14. Format of Pull-Up Resistor Option Registers (PU0, PU2 to PU7)

Address: FF30H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU0	PU07	PU06	PU05	PU04	PU03	PU02	PU01	PU00

Address: FF32H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU2	PU27	PU26	PU25	PU24	PU23	PU22	PU21	PU20

Address: FF33H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU3	0	PU36	PU35	PU34	0	PU32	PU31	PU30

Address: FF34H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU4	PU47	PU46	PU45	PU44	PU43	PU42	PU41	PU40

Address: FF35H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU5	PU57	PU56	PU55	PU54	PU53	PU52	PU51	PU50

Address: FF36H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU6	PU67	PU66	PU65	PU64	0	0	0	0

Address: FF37H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU7	0	0	PU75	PU74	PU73	0	0	PU70

PUmn	Selection of internal pull-up resistor for Pmn pin (m = 0, 2 to 7, n = 0 to 7)
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

(3) Memory expansion mode register (MEM)

The memory expansion mode register (MEM) controls the detection of the falling edge at port 4.

The interrupt request flag (KRIF) can be set to 1 by falling edge detection.

MEM is set using an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets MEM to 00H.

Figure 4-15. Format of Memory Expansion Mode Register (MEM)

Address: FF47H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MEM	0	0	0	0	0	0	0	MM0

MM0	Falling edge detection control
0	Operation disabled
1	Operation enabled

Cautions 1. Be sure to set bits 1 to 7 to 0.

2. MEM detects the falling edge at any of pins P40 to P47. In addition, once a low level is input to any one of pins P40 to P47, the interrupt request signal (INTKR) is not generated even if the falling edge is input to the other pins.

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.

(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.

Caution In the case of a 1-bit memory manipulation instruction, although a single bit is manipulated, the port is accessed in 8-bit units. Therefore, in a port with a mixture of input and output pins, the output latch contents for pins specified as input are undefined except for the manipulated bit.

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.

(2) Input mode

The output latch contents are undefined, but since the output buffer is off, the pin status does not change.

Caution In the case of a 1-bit memory manipulation instruction, although a single bit is manipulated, the port is accessed in 8-bit units. Therefore, in a port with a mixture of input and output pins, the output latch contents for pins specified as input are undefined, even for bits other than the manipulated bit.

CHAPTER 5 CLOCK GENERATOR

5.1 Function of Clock Generator

The clock generator generates the clock to be supplied to the CPU and peripheral hardware.

★ The system clock oscillator oscillates at frequency of 4.0 to 8.4 MHz. The oscillation frequency varies depending on the type of resonator.

- Ceramic resonator: 4.0 to 8.4 MHz
- Crystal resonator: 4.0 to 4.2 MHz

Oscillation can be stopped by executing the STOP instruction.

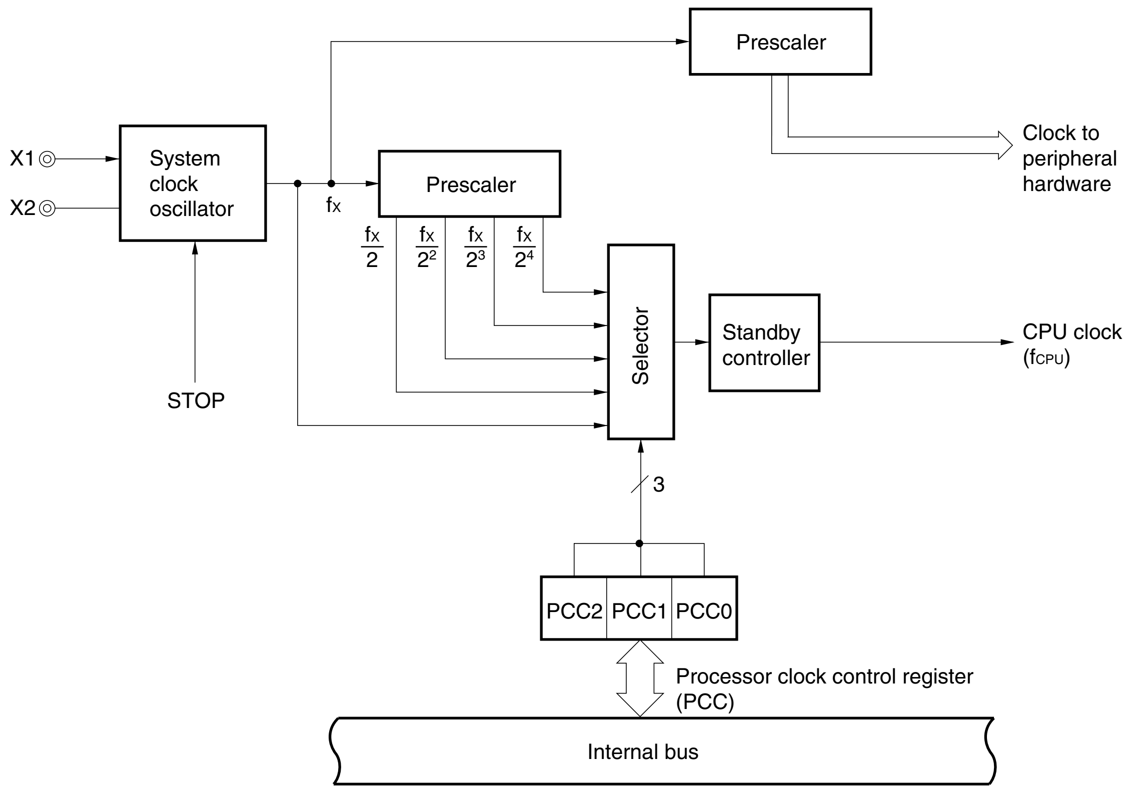
5.2 Configuration of Clock Generator

The clock generator includes the following hardware.

Table 5-1. Configuration of Clock Generator

Item	Configuration
Control register	Processor clock control register (PCC)
Oscillator	System clock oscillator

Figure 5-1. Block Diagram of Clock Generator



5.3 Register Controlling Clock Generator

The clock generator is controlled by the processor clock control register (PCC).
This register sets the CPU clock selection and the division ratio.
PCC is set using a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets PCC to 04H.

Figure 5-2. Format of Processor Clock Control Register (PCC)

Address: FFFBH After reset: 04H R/W

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

PCC2	PCC1	PCC0	Selection of CPU clock (f_{CPU})
0	0	0	f_X
0	0	1	$f_X/2$
0	1	0	$f_X/2^2$
0	1	1	$f_X/2^3$
1	0	0	$f_X/2^4$
Other than above			Setting prohibited

Caution Be sure to set bits 3 to 7 to 0.

Remark fx: System clock oscillation frequency

The fastest instructions of the μPD780833Y Subseries are carried out in 2 CPU clocks. The relationship of CPU clock (f_{CPU}) and minimum instruction execution time is shown in Table 5-2.

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

CPU Clock (f_{CPU})	Minimum Instruction Execution Time: 2/(f_{CPU})
f_X	0.48 μs
$f_X/2$	0.95 μs
$f_X/2^2$	1.91 μs
$f_X/2^3$	3.81 μs
$f_X/2^4$	7.68 μs

f_X = 4.19 MHz
fx: System clock oscillation frequency

5.4 System Clock Oscillator

5.4.1 System clock oscillator

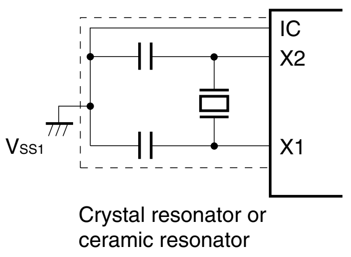
The system clock oscillator oscillates by means of a crystal resonator or a ceramic resonator (4.19 MHz TYP.) connected to the X1 and X2 pins.

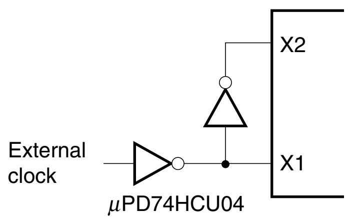
External clocks can be input to the system clock oscillator. In this case, input a clock signal to the X1 pin and a reversed-phase clock signal to the X2 pin.

Figure 5-3 shows the external circuit of the system clock oscillator.

Figure 5-3. External Circuit of System Clock Oscillator

(a) Crystal and ceramic oscillation **(b) External clock**





Cautions 1. Do not execute the STOP instruction while the external clock is being input. This is because the system clock operation will stop, and the X2 pin will be pulled up to V_{DD1} if the STOP instruction is input.

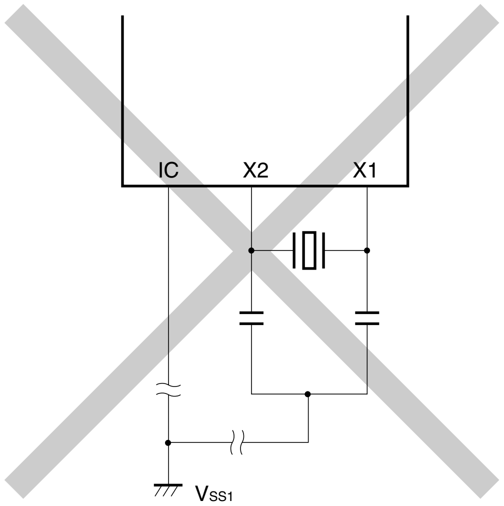
2. When using the system clock oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

- **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_{SS1}. Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from oscillator.**

Figure 5-4 shows examples of incorrect oscillator connection.

Figure 5-4. Examples of Incorrect Oscillator Connection (1/2)

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



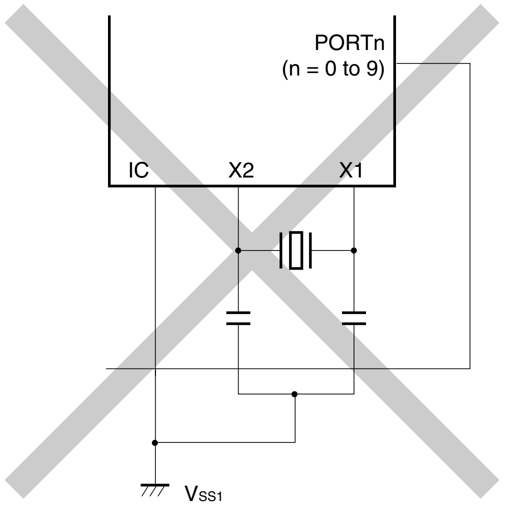
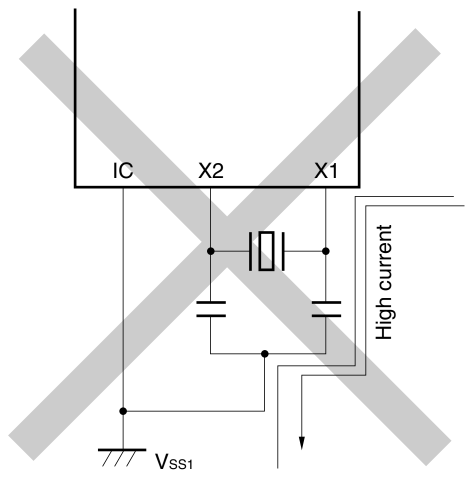
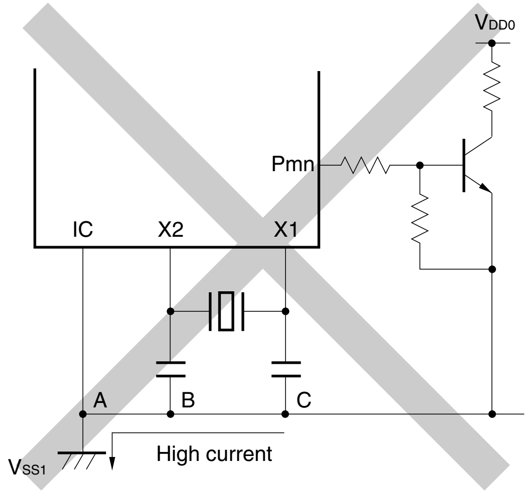


Figure 5-4. Examples of Incorrect Oscillator Connection (2/2)

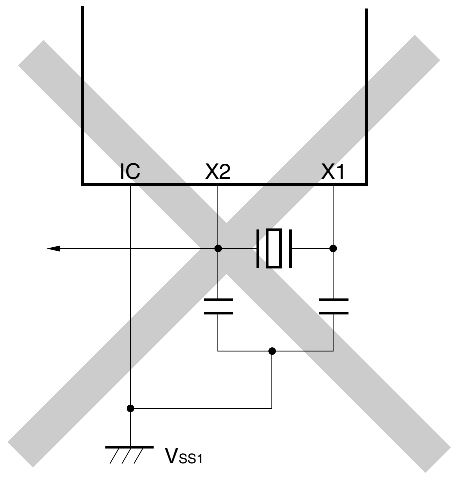
(c) Wiring near high alternating current



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



(e) Signals are fetched



5.4.2 Frequency divider

The frequency divider divides the system clock oscillator output (fx) and generates various clocks.

5.5 Operation of Clock Generator

The clock generator generates the following types of clocks and controls the CPU operation mode, including the standby mode.

- System clock fx
- CPU clock fCPU
- Clock to peripheral hardware

The following clock generator functions and operations are determined by the processor clock control register (PCC).

(a) Upon generation of the $\overline{\text{RESET}}$ signal, the lowest speed mode of the system clock (7.68 μs @ 4.19 MHz oscillation) is selected (PCC = 04H). System clock oscillation stops while a low level is being applied to the $\overline{\text{RESET}}$ pin.

(b) With the system clock selected, one of the five CPU clock types (0.48 μs, 0.96 μs, 1.92 μs, 3.84 μs @ 4.19 MHz operation) can be selected by setting PCC.

(c) Two standby modes, STOP and HALT, can be used.

(d) The system clock is divided and supplied to the peripheral hardware. Therefore, the peripheral hardware also stops if the system clock is stopped (except external input clock operation).

5.6 Changing CPU Clock Setting

5.6.1 Time required for switchover between CPU clocks

The CPU clock can be switched over using bits 0 to 2 (PCC0 to PCC2) of the processor clock control register (PCC).

The actual switchover operation is not performed immediately after writing to PCC; operation continues on the pre-switchover clock for several instructions (see **Table 5-3**).

Table 5-3. Maximum Time Required for CPU Clock Switchover

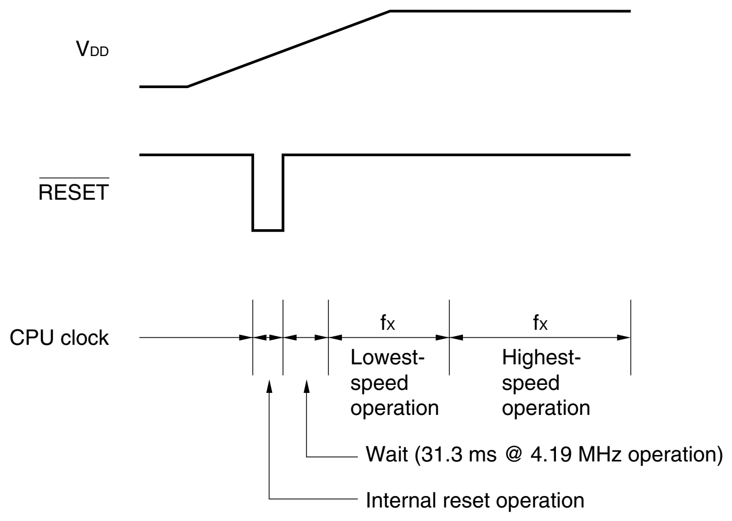
Set Value Before Switchover			Set Value After Switchover														
PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0
			0	0	0	0	0	1	0	1	0	0	1	1	1	0	0
0	0	0				16 instructions			16 instructions			16 instructions			16 instructions		
0	0	1	8 instructions						8 instructions			8 instructions			8 instructions		
0	1	0	4 instructions			4 instructions						4 instructions			4 instructions		
0	1	1	2 instructions			2 instructions			2 instructions						2 instructions		
1	0	0	1 instruction			1 instruction			1 instruction			1 instruction					

Remark One instruction is the minimum instruction execution time with the pre-switchover CPU clock.

5.6.2 CPU clock switching procedure

This section describes the procedure for switching between the CPU clocks.

Figure 5-5. CPU Clock Switching



<1> The CPU is reset by setting the $\overline{\text{RESET}}$ signal to low level after power-on. After that, when reset is released by setting the $\overline{\text{RESET}}$ signal to high level, the system clock starts oscillation. At this time, the oscillation stabilization time ($2^{17}/f_X$) is secured automatically.
After that, the CPU starts executing instructions at the minimum speed of the system clock (7.68 μs @ 4.19 MHz operation).

<2> After the lapse of a sufficient time for the V_{DD} voltage to increase to enable operation at maximum speed, PCC is rewritten and maximum-speed operation is performed.

CHAPTER 6 16-BIT TIMER/EVENT COUNTERS 00, 01

6.1 Outline of 16-Bit Timer/Event Counters 00, 01

A 16-bit timer/event counter can be used as an interval timer, for pulse width measurement (infrared ray remote control reception function), as an external event counter, for square wave output of any frequency, or for one-shot pulse output.

6.2 Function of 16-Bit Timer/Event Counters 00, 01

The 16-bit timer/event counters have the following functions.

- Interval timer
- PPG output
- Pulse width measurement
- External event counter
- Square-wave output
- One-shot pulse output

(1) Interval timer
The 16-bit timer/event counters generate an interrupt request at the preset time interval.

(2) PPG output
The 16-bit timer/event counters can output a rectangular wave whose frequency and output pulse can be set freely.

(3) Pulse width measurement
The 16-bit timer/event counters can measure the pulse width of an externally input signal.

(4) External event counter
The 16-bit timer/event counters can measure the number of pulses of an externally input signal.

(5) Square-wave output
The 16-bit timer/event counters can output a square wave with any selected frequency.

(6) One-shot pulse output
The 16-bit timer/event counters can output a one-shot pulse for which any output pulse width can be set.

6.3 Configuration of 16-Bit Timer/Event Counters 00, 01

The 16-bit timer/event counters include the following hardware.

Table 6-1. Configuration of 16-Bit Timer/Event Counters

Item	Configuration
Timer register	16-bits timer counter 0n (TM0n)
Registers	16-bit capture/compare register 00n, 01n (CR00n, CR01n)
Timer output	TO0n
Control registers	16-bit timer mode control register 0n (TMC0n) Capture/compare control register 0n (CRC0n) 16-bit timer output control register 0n (TOC0n) Prescaler mode register 0n (PRM0n) Port mode registers 3, 7 (PM3, PM7)Note

Note See **Figure 4-4 Block Diagram of P30 to P32 and P34 to P37** and **Figure 4-10 Block Diagram of P70 to 75.**

Figure 6-1. Block Diagram of 16-Bit Timer/Event Counter 00

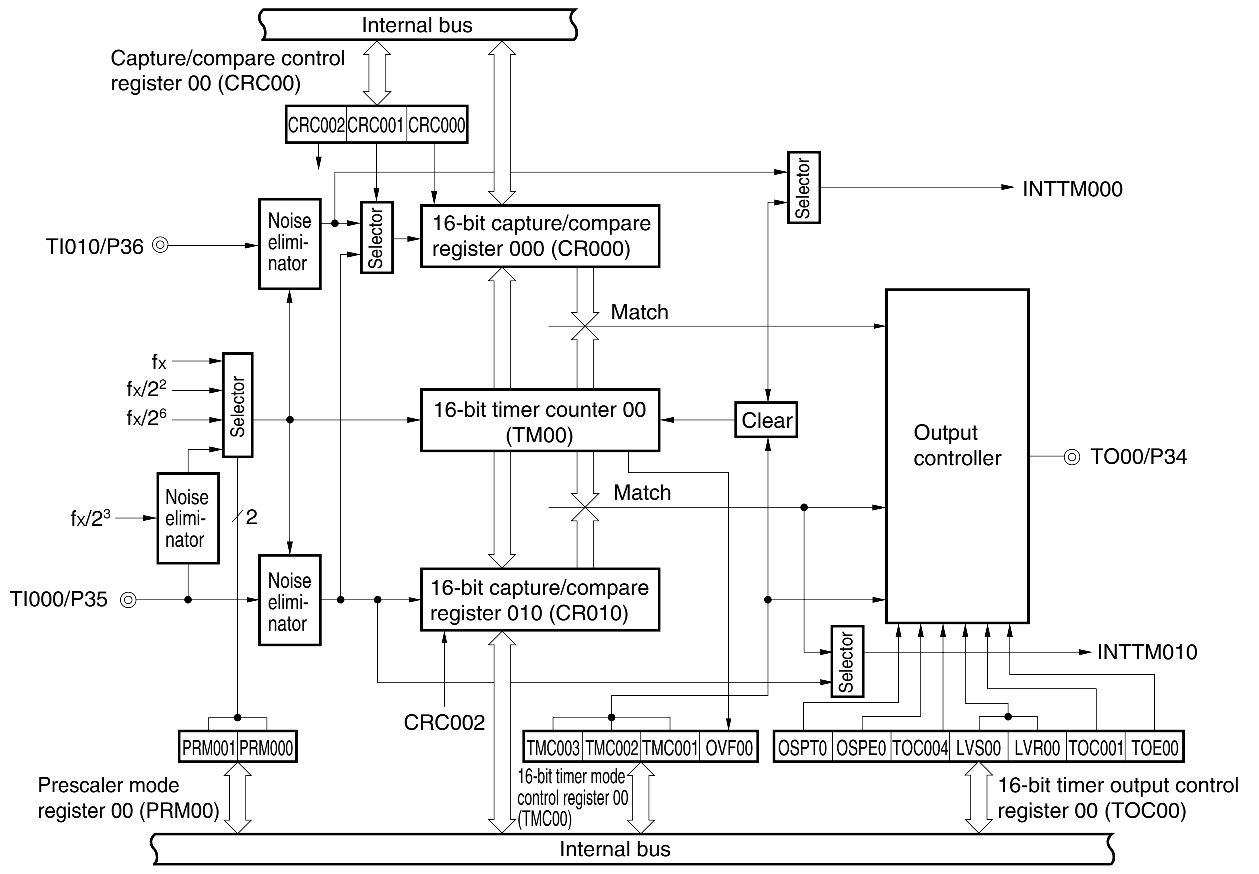
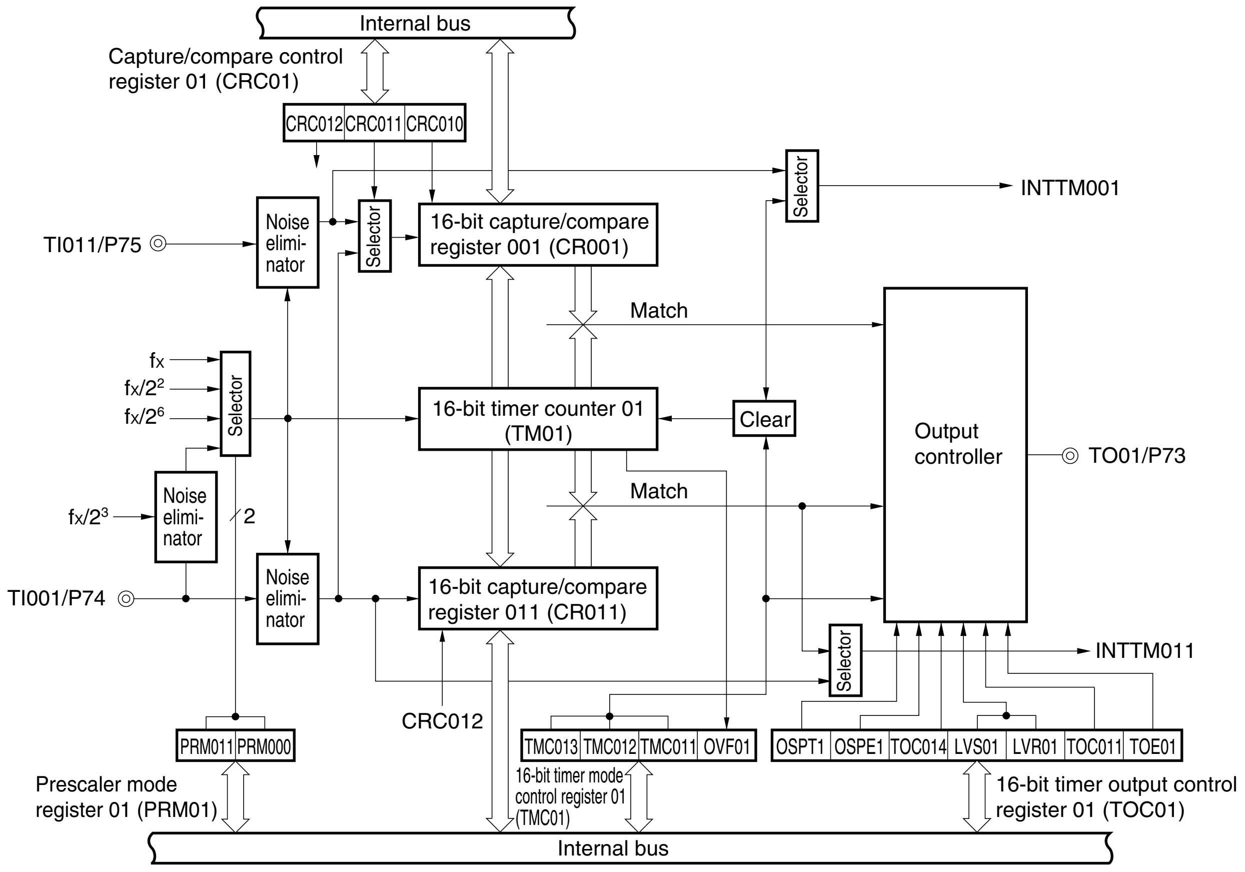


Figure 6-2. Block Diagram of 16-Bit Timer/Event Counter 01



(1) 16-bit timer counters 00, 01 (TM00, TM01)

TM00 and TM01 are 16-bit read-only registers that count the count pulses.

The counter is incremented in synchronization with the rising edge of the input clock. If the count value is read during operation, input of the count clock is temporarily stopped, and the count value at that point is read. The count value is reset to 0000H in the following cases:

<1> At $\overline{\text{RESET}}$ input
<2> If TMC0n3 and TMC0n2 are cleared
<3> If the valid edge of TI00n is input in the clear & start mode entered by inputting the valid edge of TI00n
<4> If TM0n and CR00n match in the clear & start mode entered on a match between TM0n and CR00n
<5> If OSPTn is set or if the valid edge of TI00n is input in the one-shot pulse output mode

Remark n = 0, 1

(2) 16-bit capture/compare registers 000, 001 (CR000, CR001)

CR000 and CR001 are 16-bit registers that can function as a capture register or a compare register. Whether the register is used as a capture register or as a compare register is set by bit 0 (CRC0n0) of capture/compare control register 0n (CR0n).

- **When CR00n is used as a compare register**
  The value set in CR00n is constantly compared with the 16-bit timer counter 0n (TM0n) count value, and an interrupt request (INTTM00n) is generated if they match. It can also be used as the register that holds the interval time when TM0n is set to interval timer operation.

- **When CR00n is used as a capture register**
  It is possible to select the valid edge of the TI00n pin or the TI01n pin as the capture trigger. Setting of the TI00n or TI01n valid edge is performed by prescaler mode register 0n (PRM0n).
  If CR00n is specified as a capture register and the capture trigger is specified to be the valid edge of the TI00n pin, the situation is as shown in Table 6-2. On the other hand, when the capture trigger is specified to be the valid edge of the TI01n pin, the situation is as shown in Table 6-3.

Table 6-2. TI00n Pin Valid Edge and Capture/Compare Register Capture Trigger

ES0n1	ES0n0	TI00n Pin Valid Edge	CR00n Capture Trigger	CR01n Capture Trigger
0	0	Falling edge	Rising edge	Falling edge
0	1	Rising edge	Falling edge	Rising edge
1	0	Setting prohibited	Setting prohibited	Setting prohibited
1	1	Both rising and falling edges	No capture operation	Both rising and falling edges

n = 0, 1

Table 6-3. TI01n Pin Valid Edge and Capture/Compare Register Capture Trigger

ES1n1	ES1n0	TI01n Pin Valid Edge	CR00n Capture Trigger
0	0	Falling edge	Falling edge
0	1	Rising edge	Rising edge
1	0	Setting prohibited	Setting prohibited
1	1	Both rising and falling edges	Both rising and falling edges

n = 0, 1

CR00n is set using a 16-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets CR00 to 0000H.

★ **Cautions 1. Set CR00n (n = 0, 1) to a value other than 0000H in a mode in which the timer is cleared and started on a match between TM0n (n = 0, 1) and CR00n (n = 0, 1). However, in the free-running mode and the clear mode of the valid edge of TI00n (n = 0, 1), if CR00n (n = 0, 1) is set to 0000H, an interrupt request (INTTM00n: n = 0, 1) is generated after the overflow (FFFFH).**

2. If the value of CR00n (n = 0, 1) after changing is smaller than the value of 16-bit timer counter 0n (TM0n: n = 0, 1), TM0n continues counting and overflows, then starts counting again from 0. Also, if the value of CR00n after changing is less than the value before changing, it is necessary to restart the timer after CR00n changes.

3. When P35 (P74) is used as the valid edge of TI000 (TI001), it cannot be used as a timer output (TO00 (TO01)). Also, if it is used as TO00 (TO01), it cannot be used as the valid edge of TI000 (TI001).

Remark n = 0, 1

(3) 16-bit capture/compare registers 010, 011 (CR010, CR011)

CR010 and CR011 are 16-bit registers which have the functions of both a capture register and a compare register. Whether it is used as a capture register or a compare register is set by bit 2 (CRC0n2) of capture/compare control register 0n (CRC0n).

- **When CR01n is used as a compare register**
  The value set in the CR01n is constantly compared with the 16-bit timer counter 0n (TM0n) count value, and an interrupt request (INTTM01n) is generated if they match.

- **When CR01n is used as a capture register**
  It is possible to select the valid edge of the TI00n pin as the capture trigger. The TI00n valid edge is set by means of prescaler mode register 0n (PRM0n).

CR01n is set using a 16-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets CR01 to 0000H.

★ **Caution Set CR01n (n = 0, 1) to a value other than 0000H in a mode in which the timer is cleared and started on a match between TM0n (n = 0, 1) and CR01n (n = 0, 1). However, in the free-running mode and the clear mode of the valid edge of TI01n (n = 0, 1), if CR01n (n = 0, 1) is set to 0000H, an interrupt request (INTTM01n: n = 0, 1) is generated after the overflow (FFFFH).**

Remark n = 0, 1

6.4 Registers Controlling 16-Bit Timer/Event Counters 00, 01

The following five types of registers are used to control 16-bit timer/event counters 00, 01.

- 16-bit timer mode control registers 00, 01 (TMC00, TMC01)
- Capture/compare control registers 00, 01 (CRC00, CRC01)
- 16-bit timer output control registers 00, 01 (TOC00, TOC01)
- Prescaler mode registers 00, 01 (PRM00, PRM01)
- Port mode registers 3, 7 (PM3, PM7)

(1) 16-bit timer mode control registers 00, 01 (TMC00, TMC01)

These registers set the 16-bit timer operation mode, 16-bit timer counter 00, 01 (TM00, TM01) clear mode, and the output timing, and detect an overflow.
TMC00 and TMC01 are set using a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets TMC00 and TMC01 to 00H.

Caution 16-bit timer counter 0n (TM0n) starts operating at the instant TMC0n2 and TMC0n3 (n = 0, 1) are set to a value other than 0 (operation stop mode). To stop operation, set TMC0n2 and TMC0n3 to 0.

Figure 6-3. Format of 16-Bit Timer Mode Control Register 00 (TMC00)

Address: FF60H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	$\boxed{0}$
TMC00	0	0	0	0	TMC003	TMC002	TMC001	OVF00

TMC003	TMC002	TMC001	Operation mode and clear mode selection	TO00 output timing selection	Interrupt request generation
0	0	0	Operation stopped (TM00 cleared to 0)	No change	Not generated
0	0	1			
0	1	0	Free-running mode	Match between TM00 and CR000 or match between TM00 and CR010	Generated on match between TM00 and CR000, or match between TM00 and CR010
0	1	1		Match between TM00 and CR000, match between TM00 and CR010 or TI000 valid edge	
1	0	0	Clear & start on TI000 valid edge	Match between TM00 and CR000 or match between TM00 and CR010	
1	0	1		Match between TM00 and CR000, match between TM00 and CR010 or TI000 valid edge	
1	1	0	Clear & start on match between TM00 and CR000	Match between TM00 and CR000 or match between TM00 and CR010	
1	1	1		Match between TM00 and CR000, match between TM00 and CR010 or TI000 valid edge	

OVF00	Detection of 16-bit timer counter 00 (TM00) overflow
0	Overflow not detected
1	Overflow detected

Cautions 1. Write to a bit other than the OVF00 flag after timer operation stops.

2. Set the valid edge of the TI000/P35 pin using prescaler mode register 00 (PRM00).

3. If clear & start mode entered on a match between TM00 and CR000 is selected, when the set value of CR000 is FFFFH and the TM00 value changes from FFFFH to 0000H, the OVF00 flag is set to 1.

Remark TO00: 16-bit timer/event counter 00 output pin
TI000: 16-bit timer/event counter 00 input pin
TM00: 16-bit timer counter 00
CR000: Compare register 000
CR010: Compare register 010

Figure 6-4. Format of 16-Bit Timer Mode Control Register 01 (TMC01)

Address: FF68H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	$\boxed{0}$
TMC01	0	0	0	0	TMC013	TMC012	TMC011	OVF01

TMC013	TMC012	TMC011	Operation mode and clear mode selection	TO01 output timing selection	Interrupt request generation
0	0	0	Operation stopped (TM01 cleared to 0)	No change	Not generated
0	0	1			
0	1	0	Free-running mode	Match between TM01 and CR001 or match between TM01 and CR011	Generated on match between TM01 and CR001, or match between TM01 and CR011
0	1	1		Match between TM01 and CR001, match between TM01 and CR011 or TI001 valid edge	
1	0	0	Clear & start on TI001 valid edge	Match between TM01 and CR001 or match between TM01 and CR011	
1	0	1		Match between TM01 and CR001, match between TM01 and CR011 or TI001 valid edge	
1	1	0	Clear & start on match between TM01 and CR001	Match between TM01 and CR001 or match between TM01 and CR011	
1	1	1		Match between TM01 and CR001, match between TM01 and CR011 or TI001 valid edge	

OVF01	Detection of 16-bit timer counter 01 (TM01) overflow
0	Overflow not detected
1	Overflow detected

Cautions 1. Write to a bit other than the OVF01 flag after timer operation stops.

2. Set the valid edge of the TI001/P74 pin using prescaler mode register 01 (PRM01).

3. If clear & start mode entered on a match between TM01 and CR001 is selected, when the set value of CR001 is FFFFH and the TM01 value changes from FFFFH to 0000H, the OVF01 flag is set to 1.

Remark TO01: 16-bit timer/event counter 01 output pin
TI001: 16-bit timer/event counter 01 input pin
TM01: 16-bit timer counter 01
CR001: Compare register 001
CR011: Compare register 011

(2) Capture/compare control registers 00, 01 (CRC00, CRC01)

These registers control the operation of the 16-bit capture/compare registers (CR000, CR010, CR001, CR011).
CRC00 and CRC01 are set using a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets CRC00 and CRC01 to 00H.

Figure 6-5. Format of Capture/Compare Control Register 00 (CRC00)

Address: FF62H 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 operation mode selection
0	Operates as compare register
1	Operates as capture register

CRC001	CR000 capture trigger selection
0	Captures on valid edge of TI010
1	Captures on reverse phase of valid edge of TI000

CRC000	CR000 operation mode selection
0	Operates as compare register
1	Operates as capture register

Cautions 1. Timer operation must be stopped before setting CRC00.

2. When clear & start mode entered on a match between TM00 and CR000 is selected using 16-bit timer mode control register 00 (TMC00), CR000 should not be specified as a capture register.

3. If both the rising and falling edges are selected as valid edges of TI000, capture is not performed.

Figure 6-6. Format of Capture/Compare Control Register 01 (CRC01)

Address: FF6AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CRC01	0	0	0	0	0	CRC012	CRC011	CRC010

CRC012	CR011 operation mode selection
0	Operates as compare register
1	Operates as capture register

CRC011	CR001 capture trigger selection
0	Captures on valid edge of TI011
1	Captures on reverse phase of valid edge of TI001

CRC010	CR001 operation mode selection
0	Operates as compare register
1	Operates as capture register

Cautions 1. Timer operation must be stopped before setting CRC01.

2. When clear & start mode entered on a match between TM01 and CR001 is selected using 16-bit timer mode control register 01 (TMC01), CR001 should not be specified as a capture register.

3. If both the rising and falling edges are selected as valid edges of TI001, capture is not performed.

(3) 16-bit timer output control registers 00, 01 (TOC00, TOC01)

These registers control the operation of the 16-bit timer/event counters 00, 01 output controller. These registers set/reset the R-S type flip-flop (LV0), enable/disable output inversion, enable/disable 16-bit timer/event counter 00, 01 timer output, enable/disable a one-shot pulse output operation, and output trigger for a one-shot pulse by software.

TOC00 and TOC01 are set using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TOC00 and TOC01 to 00H.

Figure 6-7. Format of 16-Bit Timer Output Control Register 00 (TOC00)

Address: FF63H After reset: 00H R/W

Symbol	7	$\boxed{6}$	$\boxed{5}$	4	$\boxed{3}$	$\boxed{2}$	1	$\boxed{0}$
TOC00	0	OSPT0	OSPE0	TOC004	LVS00	LVR00	TOC001	TOE00

OSPT0	Control of one-shot pulse output trigger by software
0	One-shot pulse trigger not used
1	One-shot pulse trigger used

OSPE0	One-shot pulse output control
0	Continuous pulse output
1	One-shot pulse output[Note]

TOC004	Timer output F/F control by match of CR010 and TM00
0	Inversion operation disabled
1	Inversion operation enabled

LVS00	LVR00	16-bit timer/event counter 00 timer output F/F status setting
0	0	No change
0	1	Timer output F/F reset (0)
1	0	Timer output F/F set (1)
1	1	Setting prohibited

TOC001	Timer output F/F control by match of CR000 and TM00
0	Inversion operation disabled
1	Inversion operation enabled

TOE00	16-bit timer/event counter 00 timer output control
0	Output disabled (output set to level 0)
1	Output enabled

Note The one-shot pulse output operates normally only in the free-running mode and in the clear and start mode at the valid edge of TI000.

Caution Timer operation must be stopped before setting TOC00.

Remarks 1. If LVS00 and LVR00 are read after data is set, they will be 0.

2. OSPT0 is cleared automatically after data is set, so when read it is 0.

Figure 6-8. Format of 16-Bit Timer Output Control Register 01 (TOC01)

Address: FF6BH After reset: 00H R/W

Symbol	7	$\boxed{6}$	$\boxed{5}$	4	$\boxed{3}$	$\boxed{2}$	1	$\boxed{0}$
TOC01	0	OSPT1	OSPE1	TOC014	LVS01	LVR01	TOC011	TOE01

OSPT1	Control of one-shot pulse output trigger by software
0	One-shot pulse trigger not used
1	One-shot pulse trigger used

OSPE1	One-shot pulse output control
0	Continuous pulse output
1	One-shot pulse output[Note]

TOC014	Timer output F/F control by match of CR011 and TM01
0	Inversion operation disabled
1	Inversion operation enabled

LVS01	LVR01	16-bit timer/event counter 01 timer output F/F status setting
0	0	No change
0	1	Timer output F/F reset (0)
1	0	Timer output F/F set (1)
1	1	Setting prohibited

TOC011	Timer output F/F control by match of CR001 and TM01
0	Inversion operation disabled
1	Inversion operation enabled

TOE01	16-bit timer/event counter 01 timer output control
0	Output disabled (output set to level 0)
1	Output enabled

Note The one-shot pulse output operates normally only in the free-running mode and in the clear and start mode at the valid edge of TI000.

Caution Timer operation must be stopped before setting TOC01.

Remarks 1. If LVS01 and LVR01 are read after data is set, they will be 0.

2. OSPT1 is cleared automatically after data is set, so when read it is 0.

(4) Prescaler mode registers 00, 01 (PRM00, PRM01)

These registers set the 16-bit timer counter 00, 01 (TM00, TM01) count clock and TI000, TI001 input valid edges.

PRM00 and PRM01 are set using an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PRM00 and PRM01 to 00H.

Figure 6-9. Format of Prescaler Mode Register 00 (PRM00)

Address: FF61H 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 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 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
0	0	fx (4.19 MHz)
0	1	fx/2^2 (1.05 MHz)
1	0	fx/2^6 (65.5 kHz)
1	1	TI000 valid edge[Note]

Note A pulse that is longer than 2 clock cycles of the internal clock is required for an external clock.

Cautions 1. If the valid edge of TI000 is set for the count clock, do not set it for the clear and start mode or the capture trigger.

2. PRM00 should be set only after timer operation has stopped.

3. For a capture trigger to capture with certainty, a pulse with a length of more than 2 clock cycles of the selected count clock is required. In addition, in cases where TI000 or TI010 is high level immediately after system reset, the rising edge is detected immediately after TM00 operation is enabled, so exercise caution in the case of pull-up, etc.

Remarks 1. fx: System clock oscillation frequency

2. TI000, TI010: Input pins of 16-bit timer/event counter 00

3. Values in parentheses ( ) apply to operation when fx = 4.19 MHz.

Figure 6-10. Format of Prescaler Mode Register 01 (PRM01)

Address: FF69H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PRM01	ES111	ES110	ES011	ES010	0	0	PRM011	PRM010

ES111	ES110	TI011 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES011	ES010	TI001 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

PRM011	PRM010	Count clock selection
0	0	fx (4.19 MHz)
0	1	fx/2^2 (1.05 MHz)
1	0	fx/2^6 (65.5 kHz)
1	1	TI001 valid edge^Note^

Note A pulse that is longer than 2 clock cycles of the internal clock is required for an external clock.

Cautions 1. If the valid edge of TI001 is set for the count clock, do not set it for the clear and start mode or the capture trigger.

2. PRM01 should be set only after timer operation has stopped.

3. For a capture trigger to capture with certainty, a pulse with a length of more than 2 clock cycles of the selected count clock is required. In addition, in cases where TI001 or TI011 is high level immediately after system reset, the rising edge is detected immediately after TM01 operation is enabled, so exercise caution in the case of pull-up, etc.

Remarks 1. fx: System clock oscillation frequency

2. TI001, TI011: Input pins of 16-bit timer/event counter 01

3. Values in parentheses ( ) apply to operation when fx = 4.19 MHz.

(5) Port mode registers 3, 7 (PM3, PM7)

These registers set ports 3 and 7 to input/output in 1-bit units.

When using the P34/TO00 pin and P73/TO01 pin for timer output, set PM34 and PM73, and the output latch of P34 and P73 to 0.

PM3 and PM7 are set using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PM3 and PM7 to FFH.

Figure 6-11. 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	PM36	PM35	PM34	PM33	PM32	PM31	PM30

PM3n	P3n pin I/O mode selection (n = 0 to 6)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Figure 6-12. Format of Port Mode Register 7 (PM7)

Address: FF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	1	1	PM75	PM74	PM73	PM72	PM71	PM70

PM7n	P7n pin I/O mode selection (n = 0 to 5)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

6.5 Operation of 16-Bit Timer/Event Counters 00, 01

6.5.1 Interval timer operation

Setting 16-bit timer mode control register 0n (TMC0n) and capture/compare control register 0n (CRC0n) as shown in Figure 6-13 allows operation as an interval timer. An interrupt request is generated repeatedly using the count value preset in 16-bit capture/compare register 00n (CR00n) as the interval.

When the count value of 16-bit timer counter 0n (TM0n) matches the value set to CR00n, counting continues with the TM0n value cleared to 0, and the interrupt request signal (INTTM00n) is generated.

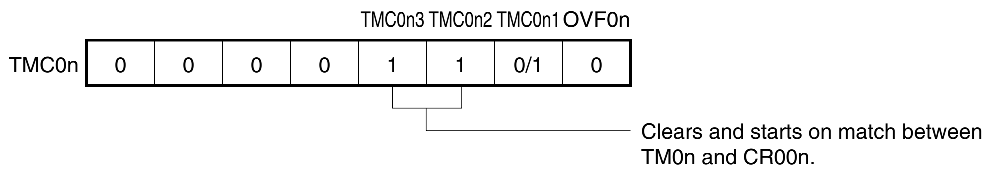
The count clock of TM0n can be selected using bits 0 and 1 (PRM0n0, PRM0n1) of prescaler mode register 0n (PRM0n).

See **6.6 Notes on 16-Bit Timer/Event Counter (3) Operation after compare register change during timer count operation** about the operation when the compare register value is changed during timer count operation.

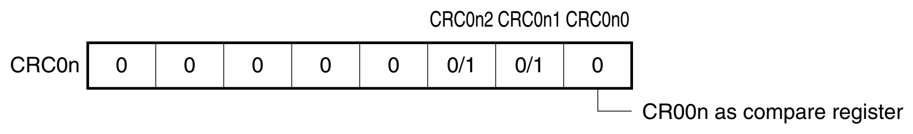
Remark n = 0, 1

Figure 6-13. Control Register Settings for Interval Timer Operation

(a) 16-bit timer mode control register 0n (TMC0n)



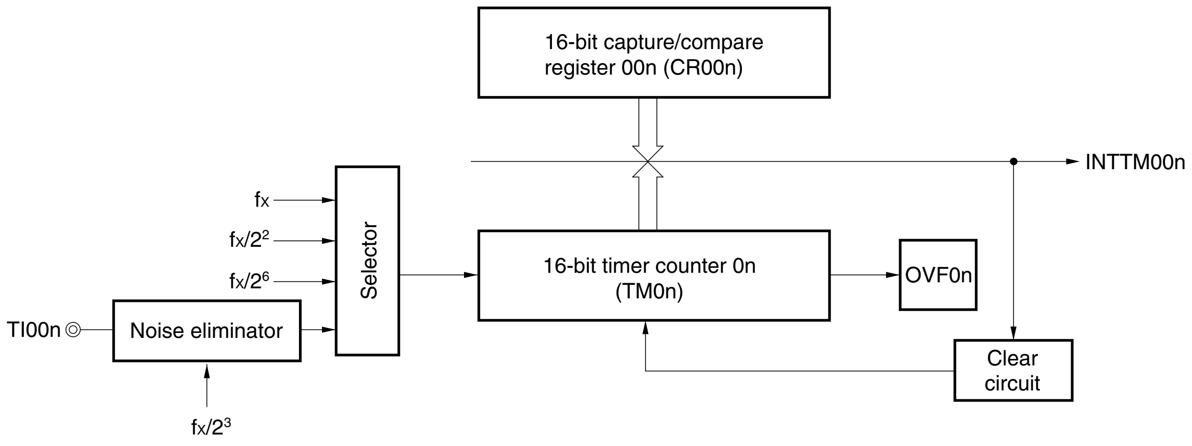
(b) Capture/compare control register 0n (CRC0n)



Remarks 1. 0/1: Setting 0 or 1 allows another function to be used simultaneously with the interval timer. See Figures 6-3 to 6-6.

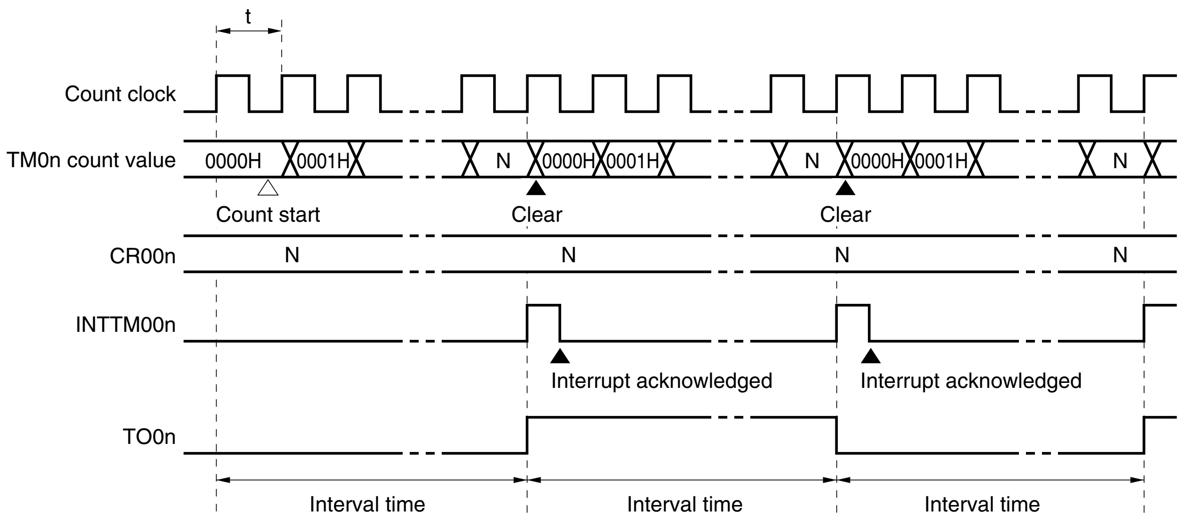
2. n = 0, 1

Figure 6-14. Configuration Diagram of Interval Timer


16-bit capture/compare register 00n (CR00n)
INTTM00n
fx
fx/2²
fx/2⁶
Selector
16-bit timer counter 0n (TM0n)
OVF0n
TI00n
Noise eliminator
Clear circuit
fx/2³

Remark n = 0, 1

Figure 6-15. Timing of Interval Timer Operation



Remarks 1. Interval time = (N + 1) × t: N = 0001H to FFFFH

2. n = 0, 1

6.5.2 PPG output operation

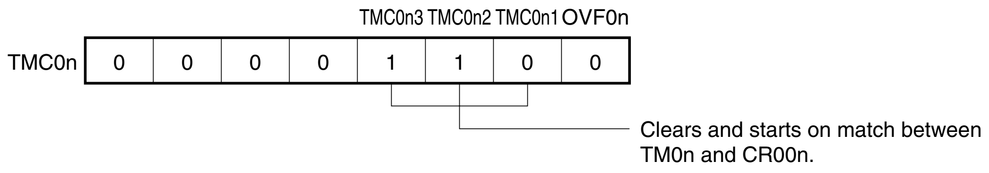
Setting 16-bit timer mode control register 0n (TMC0n) and capture/compare control register 0n (CRC0n) as shown in Figure 6-16 allows operation as PPG (Programmable Pulse Generator) output.

In the PPG output operation, rectangular waves are output from the TO0n pin with the pulse width and the cycle that correspond to the count values preset in 16-bit capture/compare register 01n (CR01n) and in 16-bit capture/compare register 00n (CR00n).

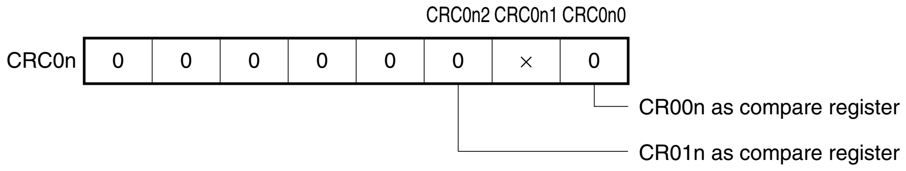
Remark n = 0, 1

Figure 6-16. Control Register Settings for PPG Output Operation

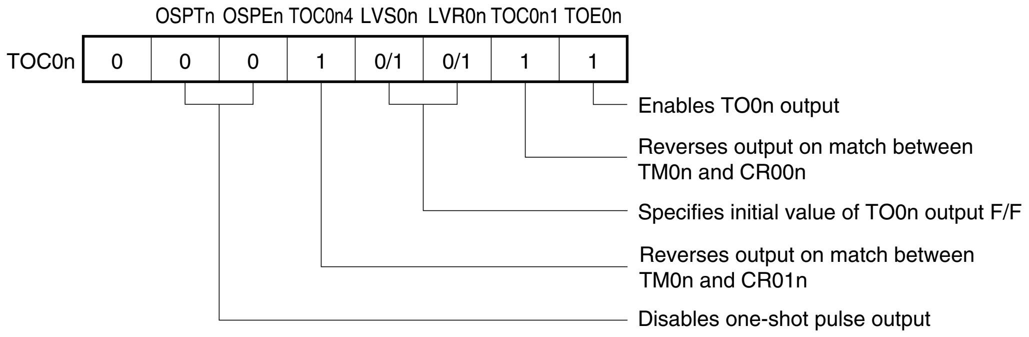
(a) 16-bit timer mode control register 0n (TMC0n)


TMC0n3 TMC0n2 TMC0n1 OVF0n
TMC0n
0 0 0 0 1 1 0 0
Clears and starts on match between TM0n and CR00n.

(b) Capture/compare control register 0n (CRC0n)



(c) Timer output control register 0n (TOC0n)


OSPTn OSPEn TOC0n4 LVS0n LVR0n TOC0n1 TOE0n
TOC0n
0 0 0 1 0/1 0/1 1 1
Enables TO0n output
Reverses output on match between TM0n and CR00n
Specifies initial value of TO0n output F/F
Reverses output on match between TM0n and CR01n
Disables one-shot pulse output

Cautions 1. CR00n and CR01n should be set to values in the following range:
0000H < CR01n < CR00n ≤ FFFFH

2. The cycle of the pulse generated by PPG output becomes (CR00n setting + 1), and the duty becomes (CR01n setting + 1)/(CR00n setting + 1).

Remark ×: Don't care
n = 0, 1

6.5.3 Pulse width measurement operation

It is possible to measure the pulse width of the signals input to the TI00n and TI01n pins using 16-bit timer counter 0n (TM0n).

There are two measurement methods: measuring with TM0n used in free-running mode, and measuring by restarting the timer in synchronization with the edge of the signal input to the TI00n pin.

(1) Pulse width measurement with free-running counter and one capture register

When 16-bit timer counter 0n (TM0n) is operated in free-running mode (see register settings in **Figure 6-17**) and the edge specified by prescaler mode register 0n (PRM0n) is input to the TI00n pin, the value of TM0n is taken into 16-bit capture/compare register 01n (CR01n) and an external interrupt request signal (INTTM01n) is set.

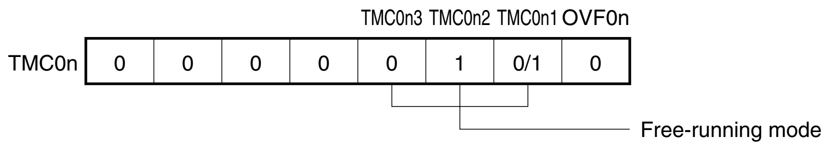
The valid edge of the TI00n pin is specified by bits 4 and 5 (ES0n0, ES0n1) of PRM0n, and the rising edge, falling edge or both edges can be selected.

Detection of the valid edge of the TI00n pin performs sampling of the count clock selected by PRM0n, and when a valid level is detected two times, the first capture operation is performed, making it possible to eliminate noise with a short pulse width.

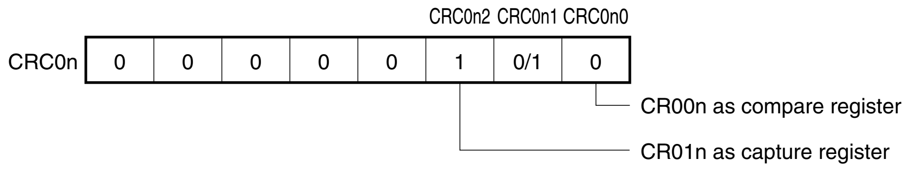
Remark n = 0, 1

Figure 6-17. Control Register Settings for Pulse Width Measurement with Free-Running Counter and One Capture Register

(a) 16-bit timer mode control register 0n (TMC0n)



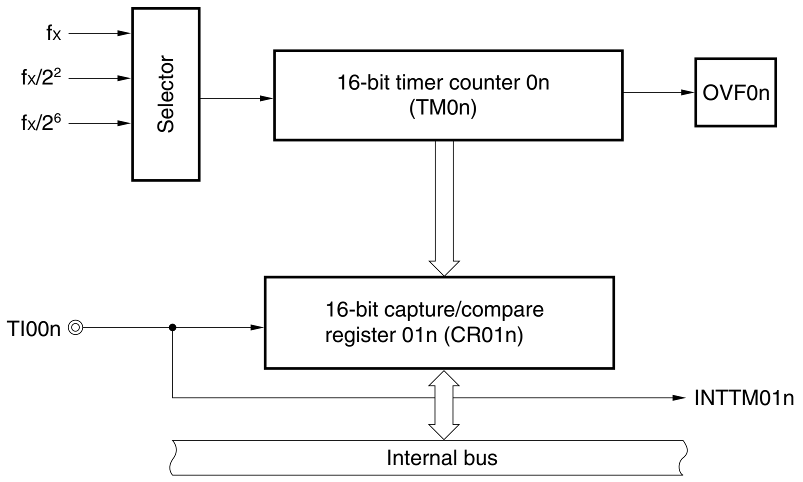
(b) Capture/compare control register 0n (CRC0n)



Remarks 1. 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. See Figures 6-3 to 6-6.

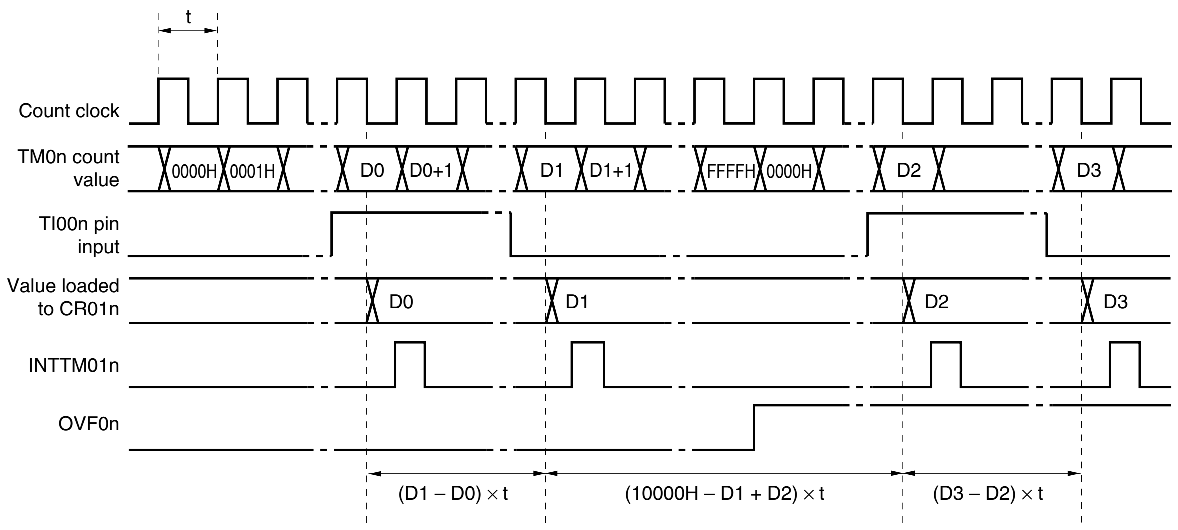
2. n = 0, 1

Figure 6-18. Configuration Diagram for Pulse Width Measurement by Free-Running Counter


fx
fx/2²
fx/2⁶
Selector
16-bit timer counter 0n (TM0n)
OVF0n
16-bit capture/compare register 01n (CR01n)
TI00n
INTTM01n
Internal bus

Remark n = 0, 1

★ **Figure 6-19. Timing of Pulse Width Measurement Operation by Free-Running Counter and One Capture Register (with Both Edges Specified)**



Remark n = 0, 1

(2) Measurement of two pulse widths with free-running counter

When 16-bit timer counter 0n (TM0n) is operated in free-running mode (see register settings in **Figure 6-20**), it is possible to simultaneously measure the pulse widths of the two signals input to the TI00n pin and the TI01n pin.

When the edge specified by bits 4 and 5 (ES0n0, ES0n1) of prescaler mode register 0n (PRM0n) is input to the TI00n pin, the value of TM0n is taken into 16-bit capture/compare register 01n (CR01n) and an external interrupt request signal (INTTM01n) is set.

Also, when the edge specified by bits 6 and 7 (ES1n0, ES1n1) of PRM0n is input to the TI01n pin, the value of TM0n is taken into 16-bit capture/compare register 00n (CR00n) and an external interrupt request signal (INTTM00n) is set.

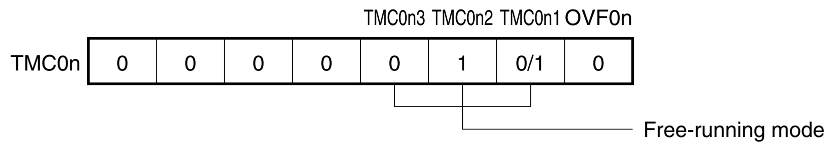
The valid edges of the TI00n and TI01n pins are specified by bits 4 and 5 (ES0n0, ES0n1), and bits 6 and 7 (ES1n0, ES1n1) of PRM0n, respectively. It is possible to select the rising edge, falling edge or both edges as the valid edge.

Detection of the valid edge of the TI00n pin performs sampling according to the cycle of the count clock selected by PRM0n, and when the valid level is detected two times, the first capture operation is performed, making it possible to eliminate noise with a short pulse width.

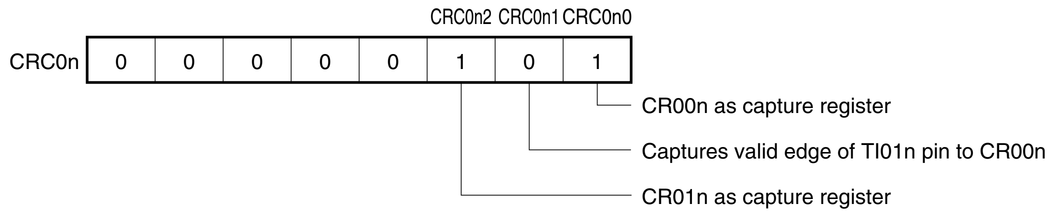
Remark n = 0, 1

Figure 6-20. Control Register Settings for Measurement of Two Pulse Widths with Free-Running Counter

(a) 16-bit timer mode control register 0n (TMC0n)



(b) Capture/compare control register 0n (CRC0n)


CRC0n2
CRC0n1
CRC0n0
CRC0n
0
0
0
0
0
1
0
1
CR00n as capture register
Captures valid edge of TI01n pin to CR00n
CR01n as capture register

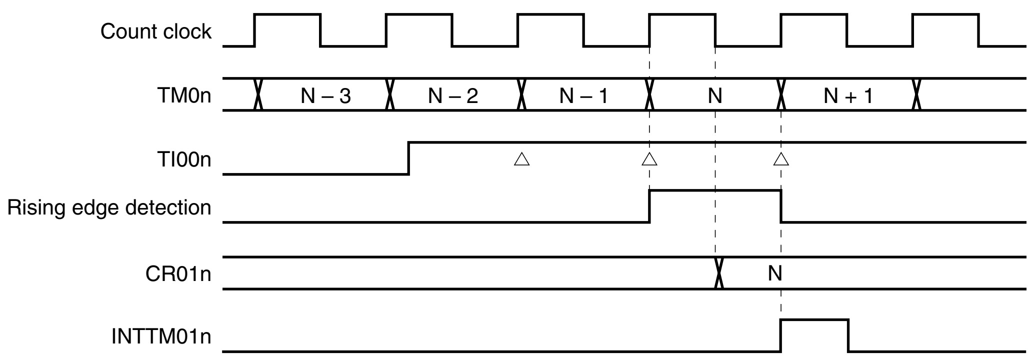
Remarks 1. 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. See Figures 6-3 to 6-6.

2. n = 0, 1

- **Capture operation (free-running mode)**

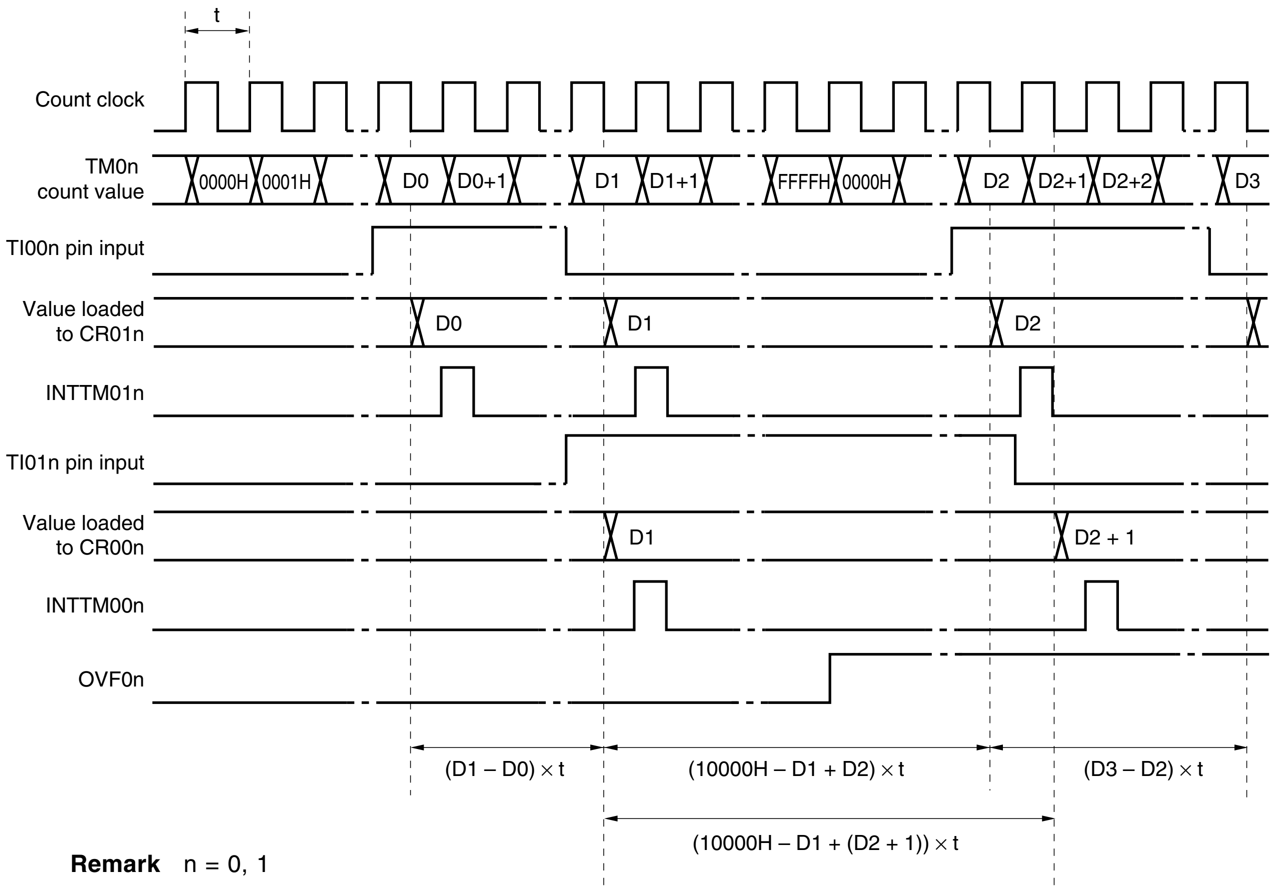
  The capture register operation when a capture trigger is input is shown below.

★ **Figure 6-21. CR01n Capture Operation with Rising Edge Specified**



Remark n = 0, 1

★ **Figure 6-22. Timing of Pulse Width Measurement Operation by Free-Running Counter (with Both Edges Specified)**



Remark n = 0, 1

(3) Pulse width measurement with free-running counter and two capture registers

When 16-bit timer counter 0n (TM0n) is operated in free-running mode (see register settings in **Figure 6-23**), it is possible to measure the pulse width of the signal input to the TI00n pin.

When the edge specified by bits 4 and 5 (ES0n0, ES0n1) of prescaler mode register 0n (PRM0n) is input to the TI00n pin, the value of TM0n is taken into 16-bit capture/compare register 01n (CR01n) and an external interrupt request signal (INTTM01n) is set.

Also, when the inverse edge of that of the capture operation is input to CR01n, the value of TM0n is taken into 16-bit capture/compare register 00n (CR00n).

The valid edge of TI00n pin is specified by bits 4 and 5 (ES0n0, ES0n1) of PRM0n, and it is possible to select the rising edge or falling edge.

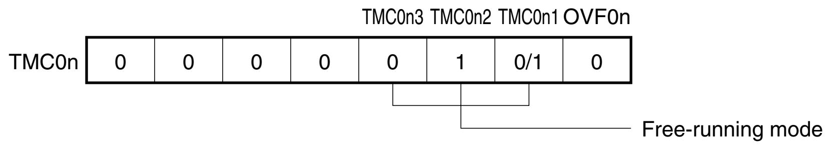
Detection of the valid edge of TI00n pin performs sampling according to the count clock cycle selected by PRM0n, and when a valid level is detected two times, the first capture operation is performed, making it possible to eliminate noise with a short pulse width.

Caution If the valid edge of TI00n pin is specified to be both the rising and falling edges, capture/compare register 00n (CR00n) cannot perform the capture operation.

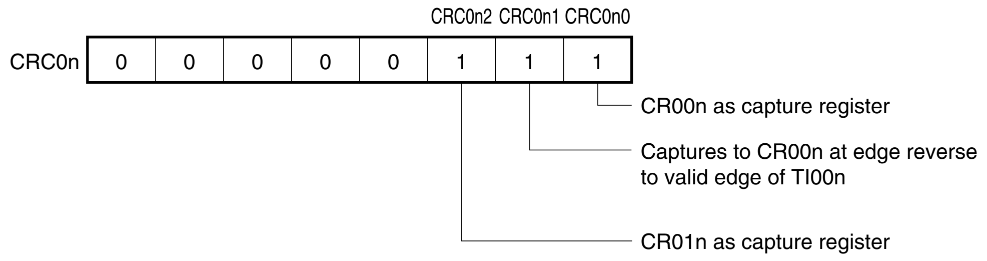
Remark n = 0, 1

Figure 6-23. Control Register Settings for Pulse Width Measurement with Free-Running Counter and Two Capture Registers

(a) 16-bit timer mode control register 0n (TMC0n)



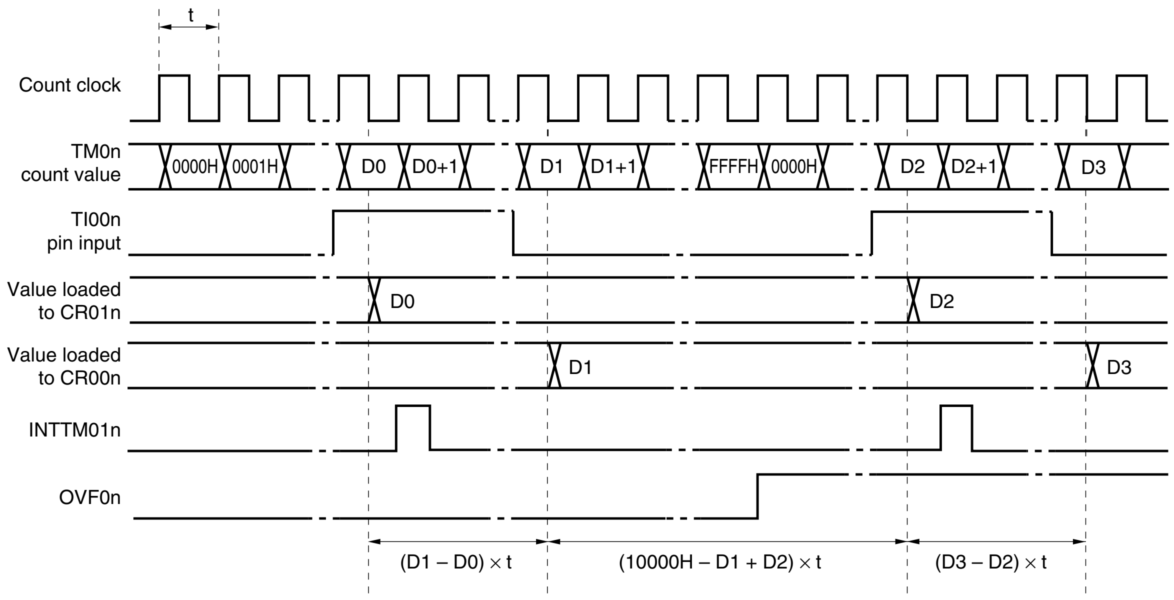
(b) Capture/compare control register 0n (CRC0n)


CRC0n2
CRC0n1
CRC0n0
CRC0n
0
0
0
0
0
1
1
1
CR00n as capture register
Captures to CR00n at edge reverse to valid edge of TI00n
CR01n as capture register

Remarks 1. 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. See Figures 6-3 to 6-6.

2. n = 0, 1

★ **Figure 6-24. Timing of Pulse Width Measurement Operation by Free-Running Counter and Two Capture Registers (with Rising Edge Specified)**



Remark n = 0, 1

(4) Pulse width measurement by means of restart

When the input of a valid edge to the TI00n pin is detected, the count value of 16-bit timer counter 0n (TM0n) is taken into 16-bit capture/compare register 01n (CR01n), and then the pulse width of the signal input to the TI00n pin is measured by clearing TM0n and restarting the count (see register settings in **Figure 6-25**).

The valid edge of TI00n pin is specified by bits 4 and 5 (ES0n0, ES0n1) of prescaler mode register 0n (PRM0n), and it is possible to select either the rising edge or falling edge.

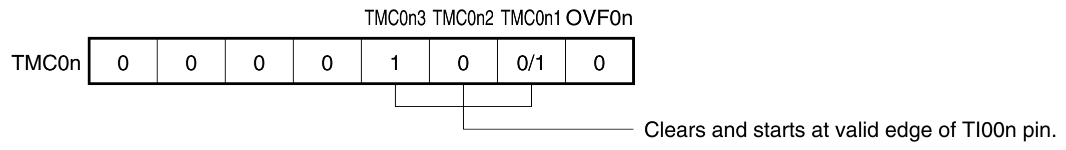
Detection of the valid edge of the TI00n pin performs sampling according to the count clock cycle selected by PRM0n, and when valid level is detected two times, the first capture operation is performed, making it possible to eliminate noise with a short pulse width.

Caution **If the valid edge of the TI00n pin is specified to be both the rising and falling edges, capture/compare register 00n (CR00n) cannot perform the capture operation.**

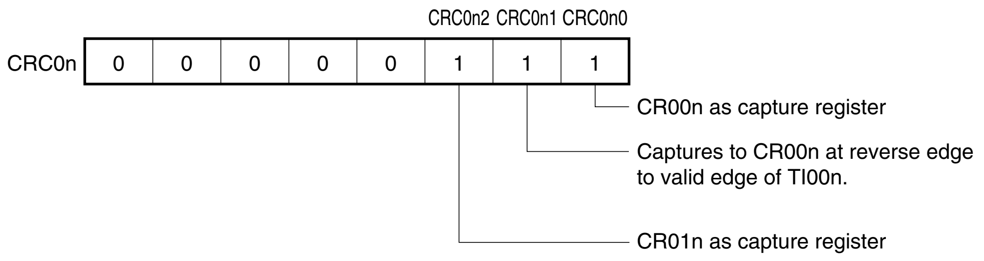
Remark n = 0, 1

Figure 6-25. Control Register Settings for Pulse Width Measurement by Means of Restart

(a) 16-bit timer mode control register 0n (TMC0n)


TMC0n3 TMC0n2 TMC0n1 OVF0n
TMC0n
0 0 0 0 1 0 0/1 0
Clears and starts at valid edge of TI00n pin.

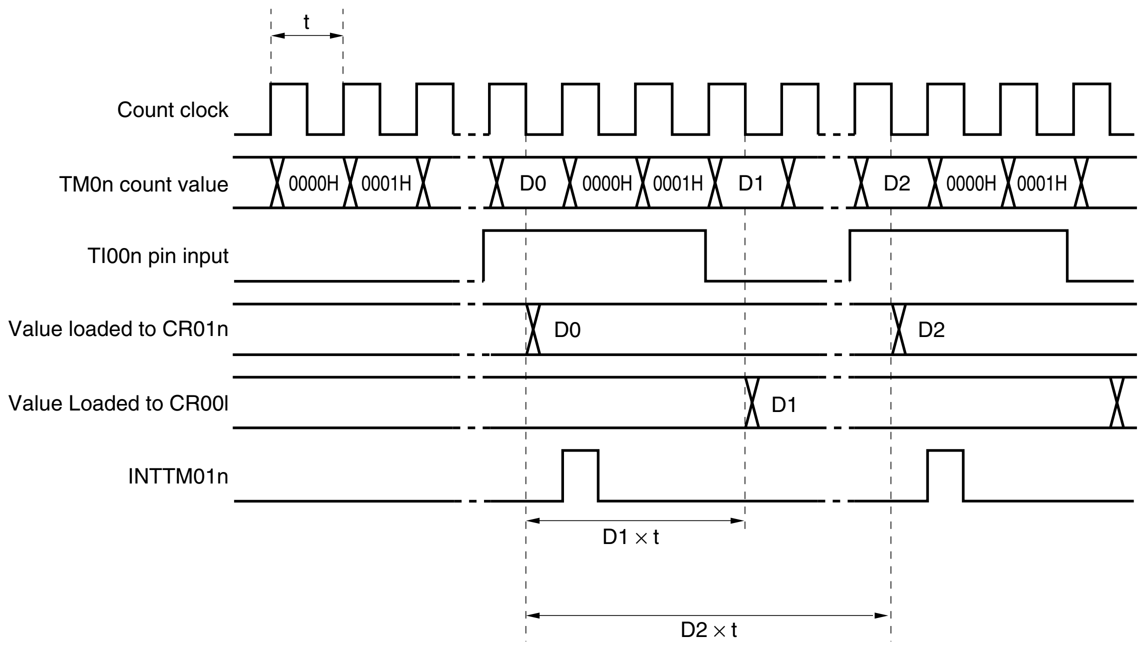
(b) Capture/compare control register 0n (CRC0n)


CRC0n2 CRC0n1 CRC0n0
CRC0n
0 0 0 0 0 1 1 1
CR00n as capture register
Captures to CR00n at reverse edge to valid edge of TI00n.
CR01n as capture register

Remarks 1. 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. See Figures 6-3 to 6-6.

2. n = 0, 1

★ **Figure 6-26. Timing of Pulse Width Measurement Operation by Means of Restart (with Rising Edge Specified)**



Remark n = 0, 1

6.5.4 External event counter operation

The external event counter counts the number of external clock pulses to be input to the TI00n pin by 16-bit timer counter 0n (TM0n).

TM0n is incremented each time the valid edge specified by prescaler mode register 0n (PRM0n) is input.

When the TM0n count value matches the 16-bit capture/compare register 00n (CR00n) value, TM0n is cleared to 0 and the interrupt request signal (INTTM00n) is generated.

CR00n should be set to a value other than 0000H (a 1-pulse count operation is not possible).

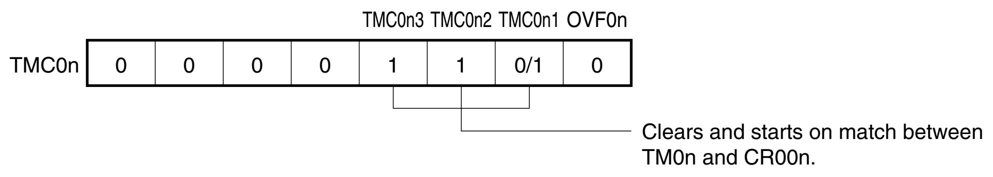
Specify the valid edge of the TI00n pin using bits 4 and 5 (ES0n0, ES0n1) of PRM0n. It is possible to select the rising edge, falling edge or both edges.

Detection of the valid edge of the TI00n pin performs sampling of the fx/2^3 clock cycle, and when the valid level is detected two times, the first operation is performed, making it possible to eliminate noise with a short pulse width.

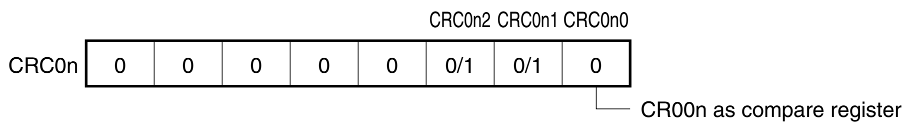
Remark n = 0, 1

Figure 6-27. Control Register Settings in External Event Counter Mode

(a) 16-bit timer mode control register 0n (TMC0n)


TMC0n3 TMC0n2 TMC0n1 OVF0n
TMC0n
0 0 0 0 1 1 0/1 0
Clears and starts on match between TM0n and CR00n.

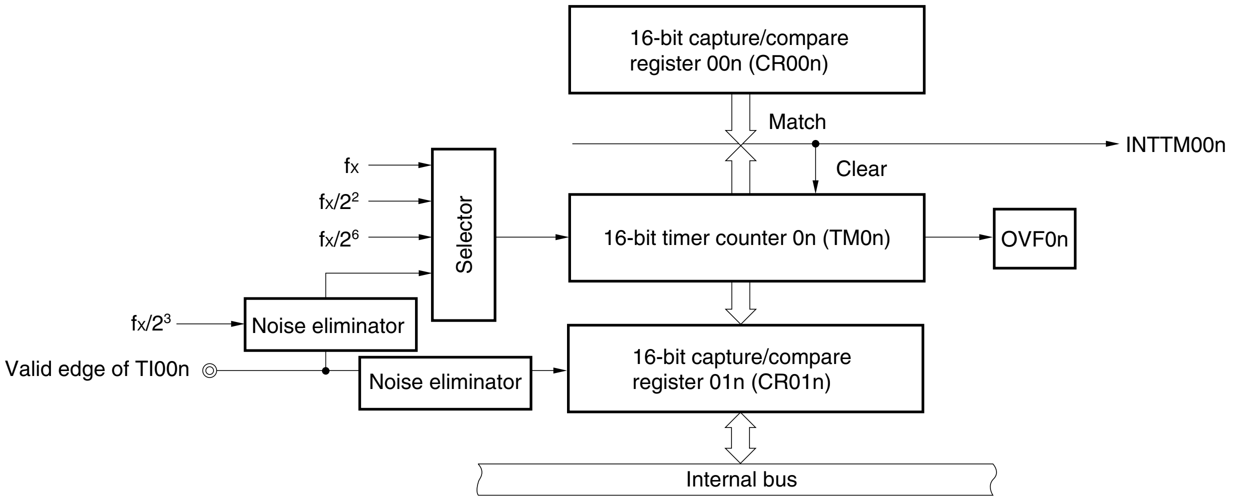
(b) Capture/compare control register 0n (CRC0n)



Remarks 1. 0/1: Setting 0 or 1 allows another function to be used simultaneously with the external event counter. See Figures 6-3 to 6-6.

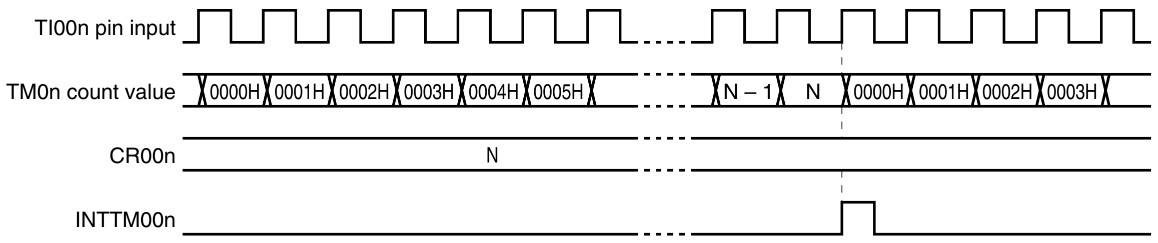
2. n = 0, 1

Figure 6-28. Configuration Diagram of External Event Counter


16-bit capture/compare register 00n (CR00n)
Match
INTTM00n
Clear
fx
fx/2²
fx/2⁶
Selector
16-bit timer counter 0n (TM0n)
OVF0n
fx/2³
Noise eliminator
Valid edge of TI00n
Noise eliminator
16-bit capture/compare register 01n (CR01n)
Internal bus

Remark n = 0, 1

Figure 6-29. External Event Counter Operation Timing (with Rising Edge Specified)



Caution When reading the external event counter count value, TM0n (n = 0, 1) should be read.

Remark n = 0, 1

6.5.5 Square-wave output operation

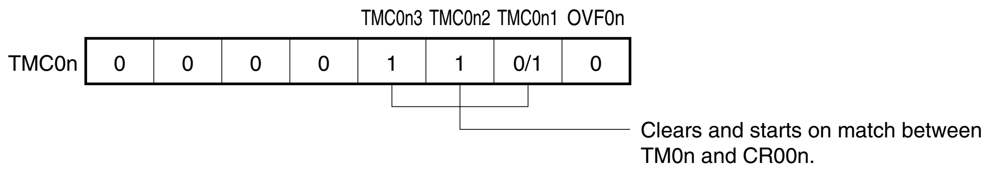
A square wave with any selected frequency can be output at intervals of the count value preset to 16-bit capture/ compare register 00n (CR00n).

The TO0n pin output status is reversed at intervals of the count value preset to CR00n by setting bit 0 (TOE0n) and bit 1 (TOC0n1) of 16-bit timer output control register 0n (TOC0n) to 1. This enables a square wave with any selected frequency to be output.

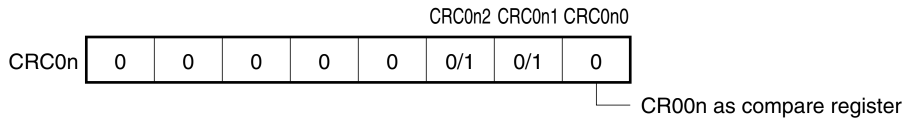
Remark n = 0, 1

Figure 6-30. Control Register Settings in Square-Wave Output Mode

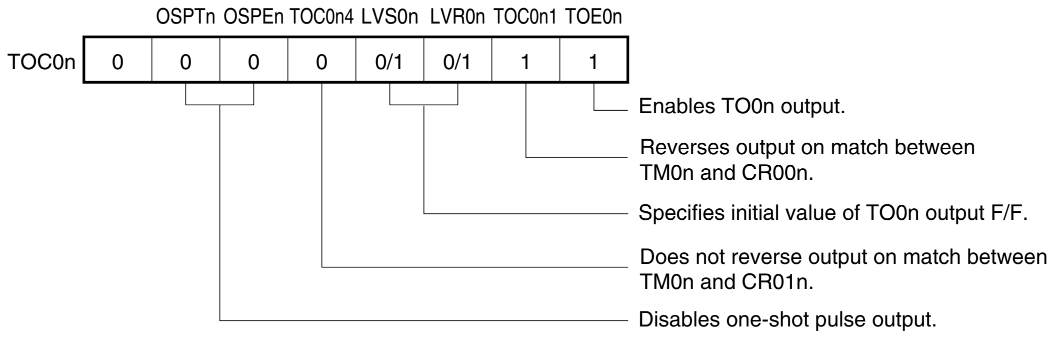
(a) 16-bit timer mode control register 0n (TMC0n)



(b) Capture/compare control register 0n (CRC0n)



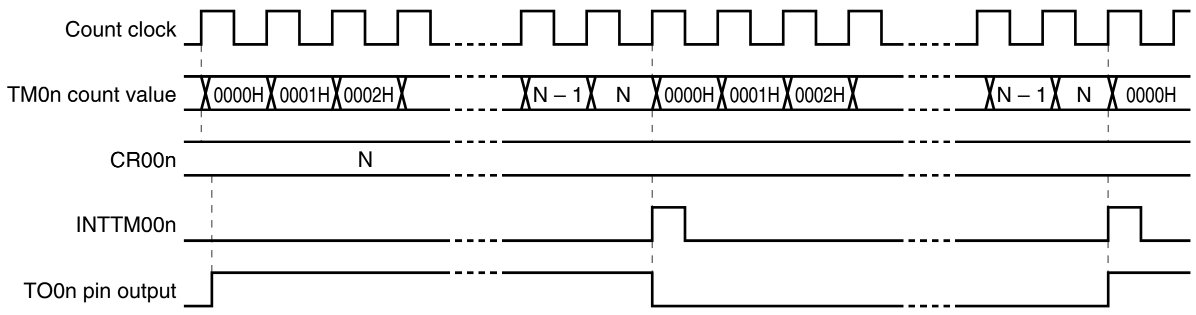
(c) 16-bit timer output control register 0n (TOC0n)


OSPTn
OSPEn
TOC0n4
LVS0n
LVR0n
TOC0n1
TOE0n
TOC0n
0
0
0
0
0/1
0/1
1
1
Enables TO0n output.
Reverses output on match between TM0n and CR00n.
Specifies initial value of TO0n output F/F.
Does not reverse output on match between TM0n and CR01n.
Disables one-shot pulse output.

Remarks 1. 0/1: Setting 0 or 1 allows another function to be used simultaneously with square-wave output. See Figures 6-3 to 6-8.

2. n = 0, 1

Figure 6-31. Square-Wave Output Operation Timing



Remark n = 0, 1

6.5.6 One-shot pulse output operation

A one-shot pulse synchronized with a software trigger or external trigger (pin TI00n input) can be output.

(1) One-shot pulse output by software trigger

If 16-bit timer mode control register 0n (TMC0n), capture/compare control register 0n (CRC0n), and 16-bit timer output control register 0n (TOC0n) are set as shown in Figure 6-32, and bit 6 (OSPTn) of TOC0n is set to 1 by software, a one-shot pulse is output from the TO0n pin.

By setting OSPTn to 1, 16-bit timer/event counter 0n is cleared and started, and the output becomes active at the count value preset (N) in 16-bit capture/compare register 01n (CR01n). Thereafter, the output becomes inactive[Note] by the count value preset (M) in 16-bit capture/compare register 00n (CR00n).

TM0n continues to operate after a one-shot pulse is output. To stop TM0n, TMC0n must be set to 00H.

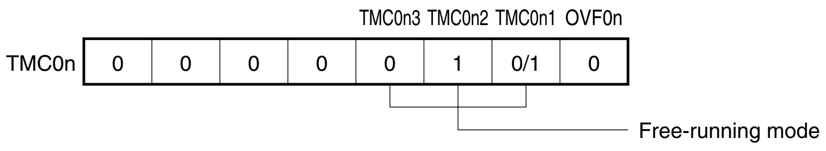
★ **Note** This is an example where N < M. If N > M, output is activated by the count value set in CR00n and inactivated by the count value set in CR01n.

Caution While a one-shot pulse is being output, do not set OSPTn (bit 6 of 16-bit timer output control register 0n (TOC0n)) to 1. To output a one-shot pulse again, wait until the current one-shot pulse output ends. ★

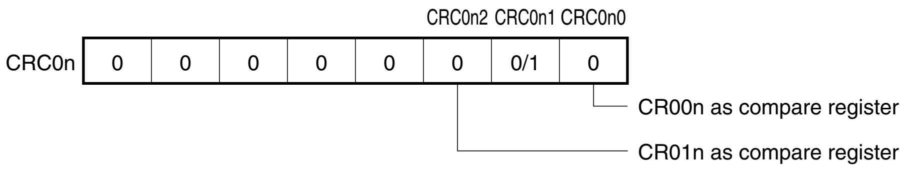
Remark n = 0, 1

Figure 6-32. Control Register Settings for One-Shot Pulse Output Operation Using Software Trigger

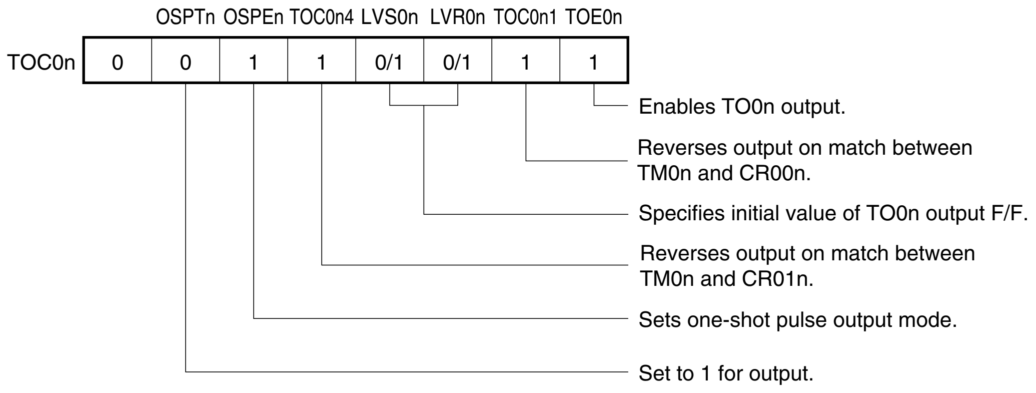
(a) 16-bit timer mode control register 0n (TMC0n)



(b) Capture/compare control register 0n (CRC0n)



(c) 16-bit timer output control register 0n (TOC0n)

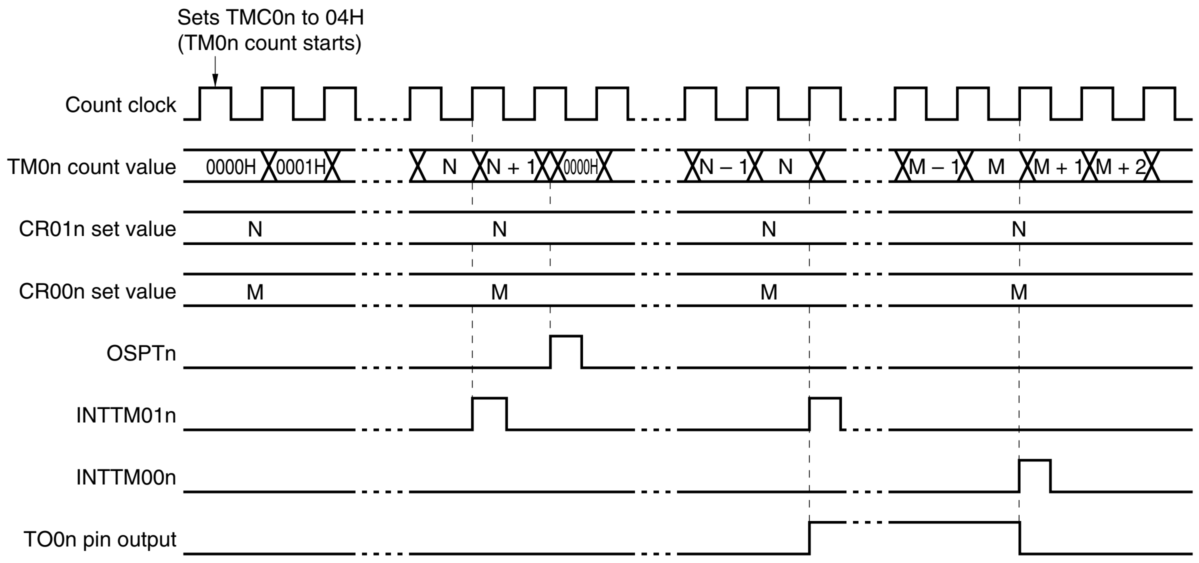

OSPTn
OSPEn
TOC0n4
LVS0n
LVR0n
TOC0n1
TOE0n
TOC0n
0
0
1
1
0/1
0/1
1
1
Enables TO0n output.
Reverses output on match between TM0n and CR00n.
Specifies initial value of TO0n output F/F.
Reverses output on match between TM0n and CR01n.
Sets one-shot pulse output mode.
Set to 1 for output.

★ **Caution Do not set CR00n and CR01n to 0000H.**

Remarks 1. 0/1: Setting 0 or 1 allows another function to be used simultaneously with one-shot pulse output. See Figures 6-3 to 6-8.

2. n = 0, 1

Figure 6-33. Timing of One-Shot Pulse Output Operation Using Software Trigger



Caution **16-bit timer counter 0n (TM0n) starts operating the moment a value other than 0, 0 (operation stop mode) is set to TMC0n2 and TMC0n3, respectively.**

Remark n = 0, 1
N < M

(2) One-shot pulse output by external trigger

This sets 16-bit timer mode control register 0n (TMC0n), capture/compare control register 0n (CRC0n) and 16-bit timer output control register 0n (TOC0n) as shown in Figure 6-34 and outputs the valid edge of the TI00n pin as an external trigger in a one-shot pulse from the TO0n pin.

The TI00n pin's valid edge is specified in bits 4 and 5 (ES00n, ES01n) of prescaler mode register 0n (PRM0n) and 3 types, rising edge, falling edge or both edges, can be selected.

At the valid edge to the TI00n pin, the 16-bit timer/event counter clears and starts and the output becomes active at the count value preset (N) in 16-bit capture/compare register 01n (CR01n). After that, the output becomes inactive[Note] at the count value preset (M) in 16-bit capture/compare register 00n (CR00n).

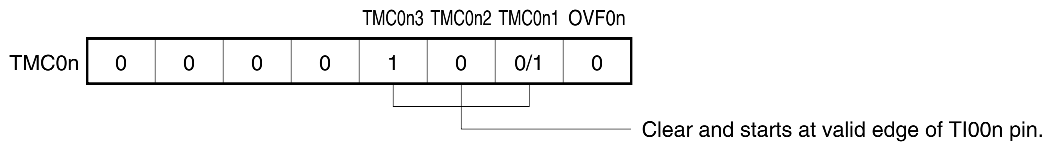
★ **Note** This is an example where N < M. If N > M, output is activated by the count value set in CR00n and inactivated by the count value set in CR01n.

Caution **When a one-shot pulse is being output, even if another external trigger is generated, it is disregarded.**

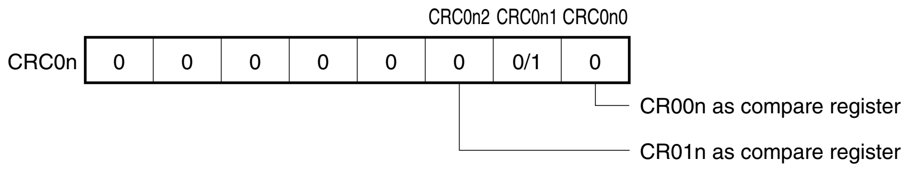
Remark n = 0, 1

Figure 6-34. Control Register Settings for One-Shot Pulse Output Operation Using External Trigger

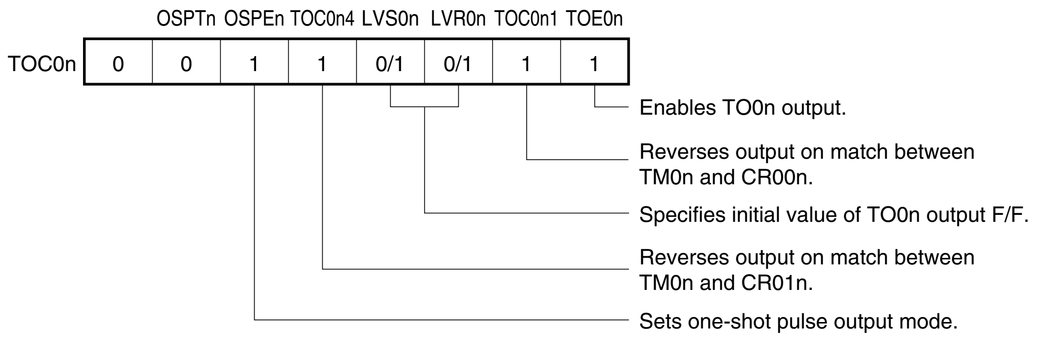
(a) 16-bit timer mode control register 0n (TMC0n)



(b) Capture/compare control register 0n (CRC0n)



(c) 16-bit timer output control register 0n (TOC0n)

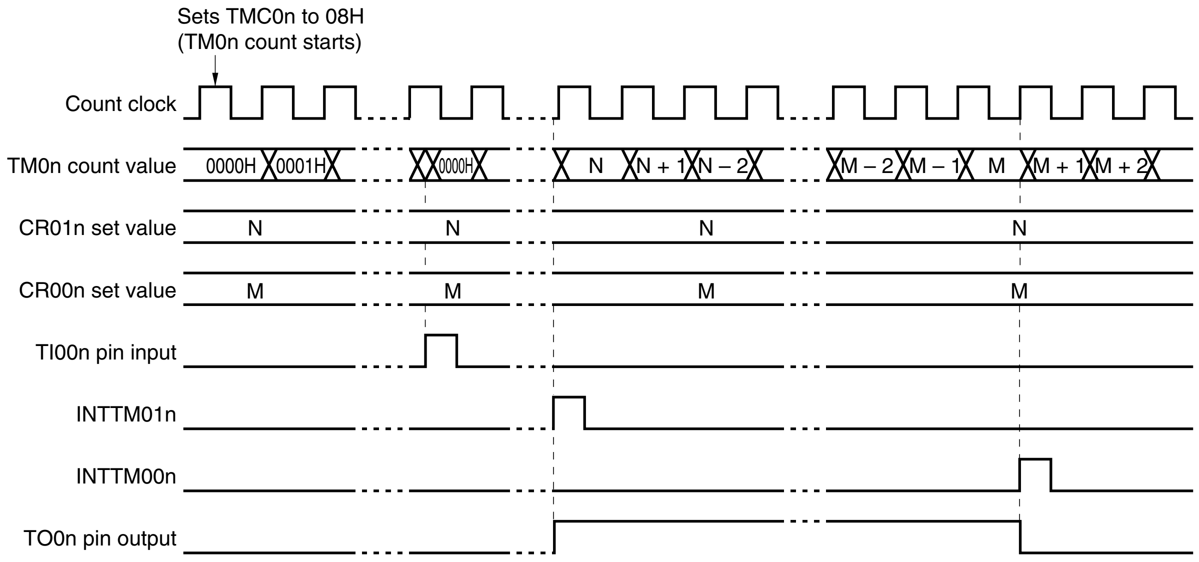

OSPTn
OSPEn
TOC0n4
LVS0n
LVR0n
TOC0n1
TOE0n
TOC0n
0
0
1
1
0/1
0/1
1
1
Enables TO0n output.
Reverses output on match between TM0n and CR00n.
Specifies initial value of TO0n output F/F.
Reverses output on match between TM0n and CR01n.
Sets one-shot pulse output mode.

★ **Caution Do not set CR00n and CR01n to 0000H.**

Remarks 1. 0/1: Setting 0 or 1 allows another function to be used simultaneously with one-shot pulse output. See Figures 6-3 to 6-8.

2. n = 0, 1

Figure 6-35. Timing of One-Shot Pulse Output Operation Using External Trigger (Clear and Start by Valid Edge of TI00n, When Rising Edge Is Specified)



Caution **16-bit timer counter 0n (TM0n) starts operating the moment a value other than 0, 0 (operation stop mode) is set to TMC0n2 and TMC0n3, respectively.**

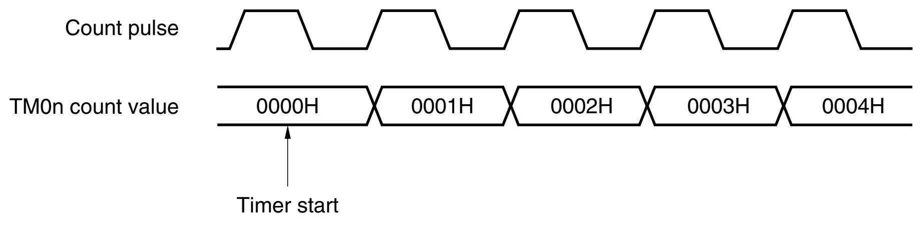
Remark n = 0, 1
N < M

6.6 Notes on 16-Bit Timer/Event Counter

(1) Timer start errors

An error of up to one clock occurs after the timer is started until a match signal is generated. This is because 16-bit timer counter 0n (TM0n: n = 0, 1) is started asynchronously to the count pulse.

Figure 6-36. Start Timing for 16-Bit Timer Counter 0n



Remark n = 0, 1

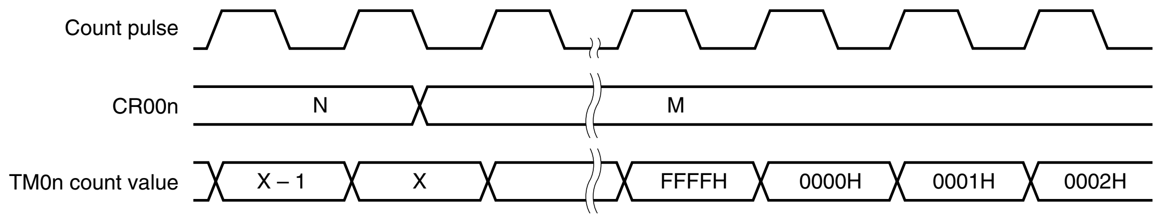
(2) 16-bit compare register setting (when a match between TM0n and CR00n triggers clear & start mode)

Set 16-bit capture/compare registers 00n to other than 00H, 01n (CR00n, CR01n: n = 0, 1) to any value other than 0000H. This means a 1-pulse count operation cannot be performed when the 16-bit timer/event counter is used as an event counter.

(3) Operation after compare register change during timer count operation

If the value after 16-bit capture/compare register 00n (CR00n: n = 0, 1) is changed is smaller than that of 16-bit timer counter 0n (TM0n: n = 0, 1), TM0n continues counting, overflows and then restarts counting from 0. Thus, if the value (M) after CR00n changes is smaller than that (N) before the change, it is necessary to reset and restart the timer after changing CR00n.

Figure 6-37. Timing After Change of Compare Register During Timer Count Operation



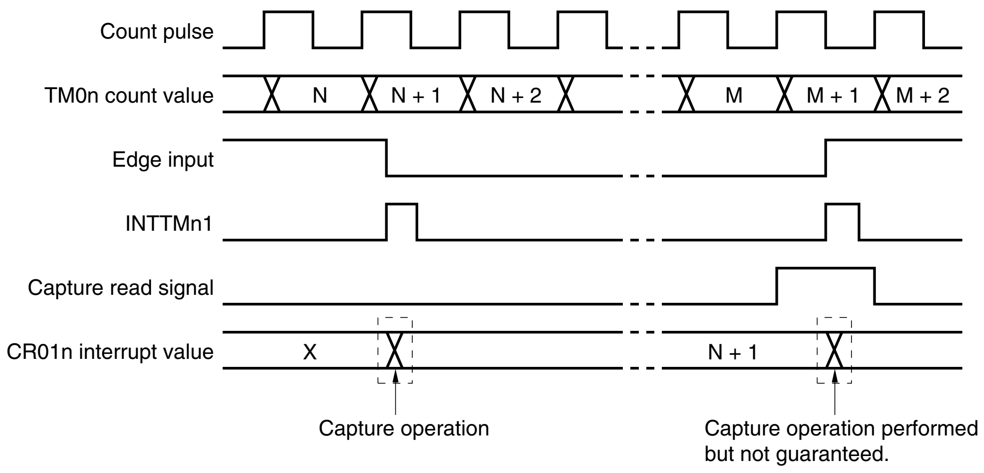
Remark N > X > M
n = 0, 1

(4) Capture register data retention timing

If the valid edge of the TI00n pin is input during 16-bit capture/compare register 01n (CR01n) read, CR01n performs a capture operation, but the capture value at this time is not guaranteed. However, the interrupt request flag (TMIF01n) is set upon detection of the valid edge.

Remark n = 0, 1

★ **Figure 6-38. Capture Register Data Retention Timing**



Remark n = 0, 1

(5) Valid edge setting

Set the valid edge of the TI00n pin after setting bits 2 and 3 (TMC0n2 and TMC0n3) of 16-bit timer mode control register 0n (TMC0n) to 0, 0, respectively, and then stopping timer operation. The valid edge is set by bits 4 and 5 (ES0n0 and ES0n1) of prescaler mode register 0n (PRM0n).

Remark n = 0, 1

(6) Re-trigger of one-shot pulse

(a) One-shot pulse output by software

While a one-shot pulse is being output, do not set OSPTn (bit 6 of 16-bit timer output control register 0n (TOC0n)) to 1. When outputting a one-shot pulse again, output it after the current one-shot pulse output ends.

(b) One-shot pulse output by external trigger

If the external trigger is generated while the one-shot pulse is being output, the counter is cleared and restarted, and the one-shot pulse is output again.

★ **(c) One-shot pulse output function**

When the one-shot pulse output function is used with a software trigger, the level of the TI00n pin or its alternate-function port pin cannot be changed. Because the external trigger is valid in this case, the timer is cleared and started by the level of these pins. Consequently a pulse is output at an unexpected timing.

Remark n = 0, 1

(7) Operation of OVF0n flag

<1> The OVF0n flag (bit 6 of 16-bit timer mode control register 0n (TMC0n)) is set to 1 under the following conditions.

Either the mode on a match between TM0 and CR00, the mode in which the 16-bit timer/event counter is cleared and started on the valid edge of TI00n, or free-running mode is selected.

↓

CR00n is set to FFFFH.

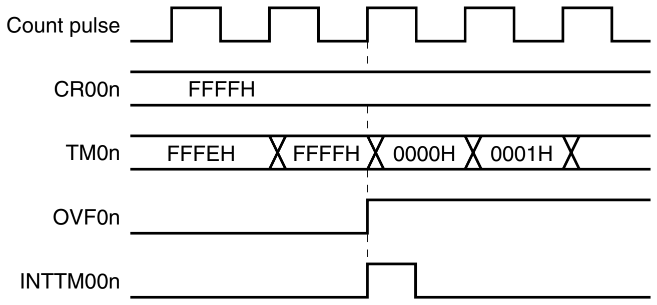
↓

TM0n is counted up from FFFFH to 0000H.

Remark n = 0, 1

★

Figure 6-39. Operation Timing of OVF0n Flag



Remark n = 0, 1

<2> After TM0n overflows, it is reset and the clear instruction becomes invalid even though the OVF0n flag is cleared before the next count clock (before TM0n becomes 0001H).

Remark n = 0, 1

(8) Contention operations

<1> Contention operation between read time of 16-bit capture/compare register 00n, 01n (CR00n, CR01n) and capture trigger input (CR00n and CR01n used as capture register)

Capture trigger input is takes precedence. The data read from CR00n and CR01n is undefined.

<2> Match timing of contention operation between write period of 16-bit capture/compare register 00n, 01n (CR00n, CR01n) and 16-bit timer counter 0n (TM0n) (CR00n and CR01n used as compare register)

Match detection is not performed normally. Do not write any data to CR00n and CR01n near the match timing.

Remark n = 0, 1

(9) Timer operation

<1> Even if 16-bit timer counter 0n (TM0n) is read, the value is not captured in 16-bit capture/compare register 01n (CR01n).

<2> The sampling clock differs between the count clock TI00n and the capture trigger TI00n. The former is fx/2^3 and the latter is the count clock (see **Figures 6-9** and **6-10**.)

<3> Regardless of the operation mode of the CPU, if the timer is stopped, the noise of the external interrupt request input is not eliminated.

<4> One-shot pulse output operates normally only in the free-running mode or in the clear and start mode entered at the TI00n valid edge. In the clear and start mode entered on a match of TM0n and CR00n, there is no overflow, so one-shot pulse output is not possible.

Remark n = 0, 1

(10) Capture operation

<1> When the valid edge of TI00n (n = 0, 1) is specified for the count clock, the capture register that specified TI00n as the trigger cannot perform the capture operation normally.

<2> The capture operation is performed at the rise of the count clock, but the external interrupt signal (INTTM00n, INTTM01n) is set at the rise of the next count clock.

(11) Compare operation

<1> When 16-bit capture/compare registers 00n and 01n (CR00n, CR01n) are rewritten during timer operation, if the values are close to or greater than the timer values, there is a possibility that the match interrupt will not be generated, or the clear operation will not be performed properly.

<2> CR00n and CR01n set in the compare mode cannot perform a capture operation even if the capture trigger is input.

Remark n = 0, 1

(12) Edge detection

When the TI00n pin or TI01n pin is high level immediately after system reset and if the rising edge or both edges are specified as the valid edge of the TI00n pin or TI01n pin, the rising edge is detected immediately after operation of 16-bit timer counter 0n (TM0n) is enabled. Be careful when the TI00n pin or TI01n pin are pulled up. When operation is enabled again after once being stopped, the rising edge cannot be detected.

Remark n = 0, 1

CHAPTER 7 8-BIT TIMER/EVENT COUNTERS 50, 51, 52

★ 7.1 Outline of 8-Bit Timer/Event Counters 50, 51, 52

These counters can be used as an interval timer or external event counter and for square-wave output of any frequency, or PWM output. Moreover, two 8-bit timer/event counters can be used as one 16-bit timer/event counter.

7.2 Function of 8-Bit Timer/Event Counters 50, 51, 52

8-bit timer/event counters 50, 51, 52 (TM50, TM51, and TM52) have the following two modes.

- Mode in which one 8-bit timer/event counter is used alone (single mode)
- Mode in which two or more 8-bit timer/event counters are connected in cascade (16-bit resolution: cascade mode)

These two modes are explained below.

(1) Mode in which one 8-bit timer/event counter is used alone (single mode)

In this mode, the counter can be used for the following functions.

- Interval timer
- External event counter
- Square-wave output
- PWM output

(2) Mode in which TM50 and TM51 are connected in cascade (16-bit resolution: cascade mode)

By connecting 8-bit timer/event counters in cascade, they can be used as a 16-bit timer/event counter.
In this mode, the counters can be used for the following functions.

- 16-bit resolution interval timer
- 16-bit resolution external event counter
- 16-bit resolution square-wave output

7.3 Configuration of 8-Bit Timer/Event Counters 50, 51, 52

The 8-bit timer/event counters include the following hardware.

Table 7-1. Configuration of 8-Bit Timer/Event Counter

Item	Configuration
Timer register	8-bit timer counter 5n (TM5n)
Register	8-bit timer compare register 5n (CR5n)
Timer output	TO5n
Control registers	8-bit timer mode control register 5n (TMC5n) Timer clock selection register 5n (TCL5n) Port mode register 2 (PM2)**Note**

Note Refer to **Figure 4-3 Block Diagram of P20 to P27**.

Remark n = 0 to 2

Figure 7-1. Block Diagram of 8-Bit Timer/Event Counter 50

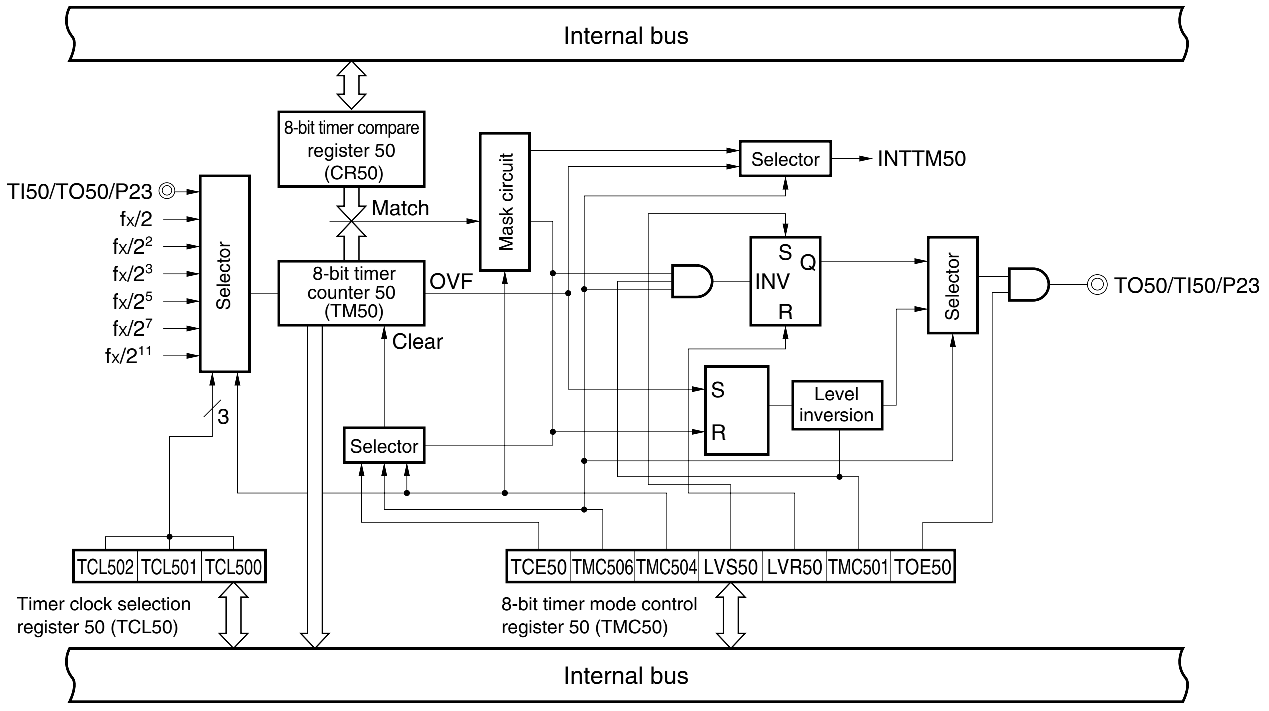


Figure 7-2. Block Diagram of 8-Bit Timer/Event Counter 51

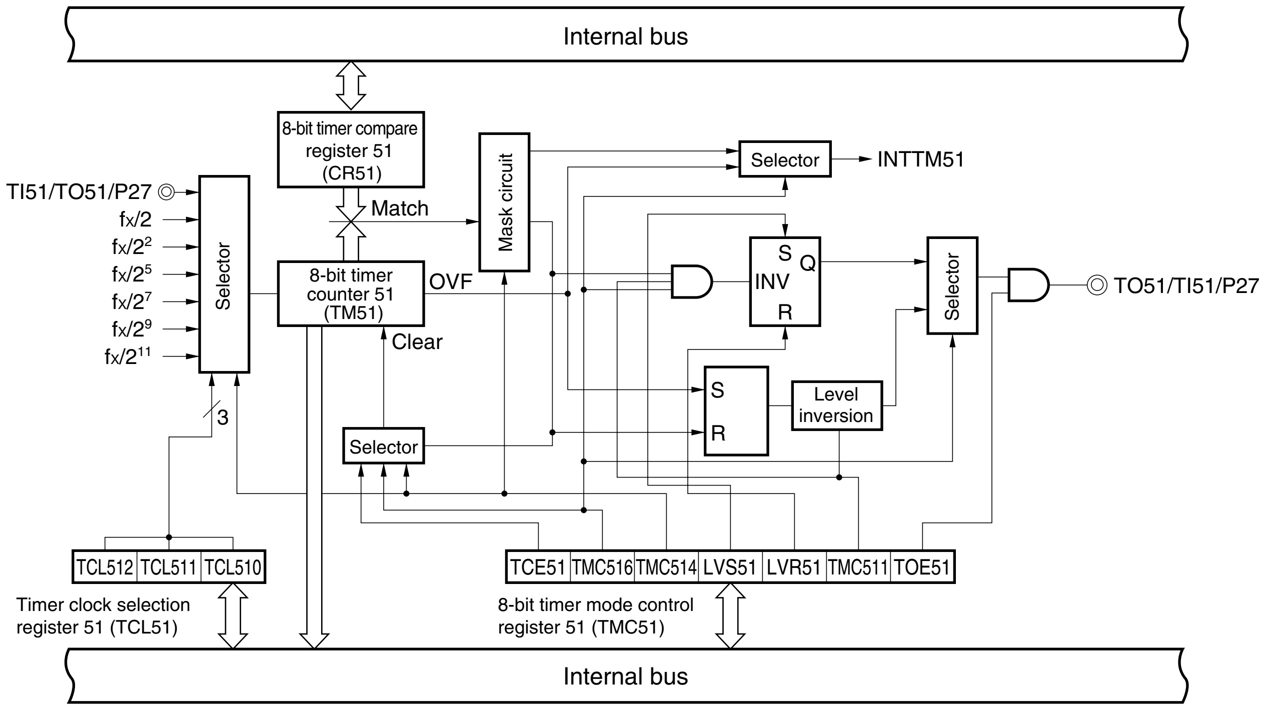
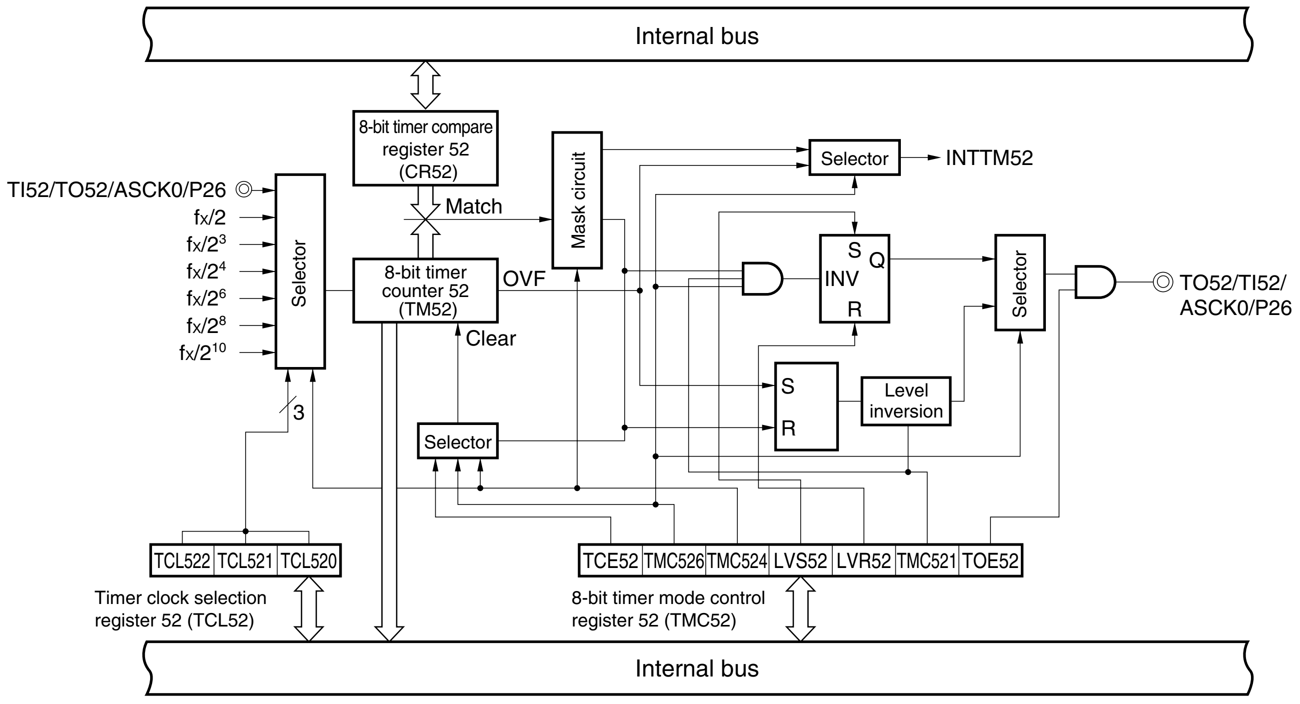


Figure 7-3. Block Diagram of 8-Bit Timer/Event Counter 52



(1) 8-bit timer counters 50, 51, and 52 (TM50, TM51, and TM52)

TM50, TM51, and TM52 are 8-bit read-only registers that count count pulses.

These counters are incremented in synchronization with the rising edge of the count clock.

TM50 and TM51 can be connected in cascade and used as a 16-bit timer.

When TM50 and TM51 are connected in cascade and used as a 16-bit timer, the values of these timers/counters can be read by using a 16-bit manipulation instruction. TM50 and TM51 are connected by an internal 8-bit bus, which means that TM50 and TM51 are read one at a time. Therefore because the value of one may change while the other is being read, for accuracy be sure to read TM50 and TM51 twice and compare their first and second values.

If a count value is read during operation, input of the count clock is temporarily stopped, and the count value at that point is read. The count value is cleared to 00H in the following cases.

<1> When $\overline{\text{RESET}}$ is input

<2> When TCE5n is cleared

<3> When there is a match between TM5n and CR5n in clear & start mode

Caution In the cascade mode, the 16-bit timer is cleared to 00H regardless of whether bit TCE51 of TM51 is cleared.

Remark n = 0 to 2

(2) 8-bit timer compare registers 50, 51, and 52 (CR50, CR51, and CR52)

The value set to CR5n is always compared with the count value of 8-bit timer counter 5n (TM5n). When the value of the compare register matches the value of the timer counter, an interrupt request (INTTM5n) is generated (in a mode other than the PWM mode).

The value of CR5n can be set in a range of 00H to FFH and can be rewritten during counting.

If TM50 and TM51 are connected in cascade and used as a 16-bit timer, CR50 and CR51 operate as a 16-bit compare register. Therefore, the count value and register value are compared in 16-bit units, and if the two values matches, an interrupt request (INTTM50) is generated. At this time, interrupt request INTTM51 is also generated. When connecting TM50 and TM51 in cascade, therefore, mask the INTTM51 interrupt request.

★ **Caution Stop the timer operation of 8-bit timer counter 5n (TM5n) connected in cascade before setting data to 8-bit compare register 5n (CR5n) in the cascade mode.**

Remark n = 0 to 2

7.4 Registers Controlling 8-Bit Timer/Event Counters 50, 51, 52

The following three types of registers are used to control 8-bit timer/event counters 50, 51, and 52.

- 8-bit timer mode control registers 50, 51, 52 (TMC50, TMC51, TMC52)
- Timer clock selection registers 50, 51, 52 (TCL50, TCL51, TCL52)
- Port mode register 2 (PM2)

(1) 8-bit timer mode control registers 50, 51, 52 (TMC50, TMC51, TMC52)

TMC50, TMC51, and TMC52 perform the following six operations.

<1> Control the count operation of 8-bit timer counters 50, 51, and 52 (TM50, TM51, and TM52)
<2> Select the operation mode of 8-bit timer counters 50, 51, and 52 (TM50, TM51, and TM52)
<3> Select the single mode or cascade mode
<4> Set the status of the timer output F/F (flip-flop)
<5> Control the timer F/F or select the active level in PWM (free-running) mode
<6> Control the timer output

TMC50, TMC51, and TMC52 are set using a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets these registers to 04H.
Figures 7-4 to 7-6 show the formats of TMC50, TMC51, and TMC52.

Figure 7-4. Format of 8-Bit Timer Mode Control Register 50 (TMC50)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TMC50	TCE50	TMC506	0	TMC504	LVS50	LVR50	TMC501	TOE50	FF79H	04H	R/W

TCE50	TM50 count operation control
0	Disables count operation after clearing counter to 0 (disables prescaler)
1	Starts counting

TMC506	TM50 operation mode selection
0	Clears and starts on match between TM50 and CR50
1	PWM (free-running) mode

TMC504	Single mode/cascade mode selection
0	Single mode
1	Cascade mode (setting prohibited)

LVS50	LVR50	Timer output F/F status setting of 8-bit timer/event counter 50
0	0	Not affected
0	1	Resets timer output F/F (to 0).
1	0	Sets timer output F/F (to 1).
1	1	Setting prohibited

TMC501	Other than PWM mode (TMC506 = 0)	PWM mode (TMC506 = 1)
	Timer F/F control	Active level selection
0	Disables inverted operation	Active high
1	Enables inverted operation	Active low

TOE50	Timer output control of 8-bit timer/event counter 50
0	Disables output (port mode)
1	Enables output

★ **Cautions 1. Before clearing TCE50 to 0, set the interrupt mask flag (TMMK50) to 1. This is because an interrupt may occur after TCE50 has been cleared. Clear TCE50 using the following procedure:**

```
TMMK50 = 1 ; Sets mask.
TCE50 = 0  ; Clears timer.
TMIF50 = 0 ; Clears interrupt request flag.
TMMK50 = 0 ; Clears mask.
               ⋮
TCE50 = 1  ; Starts timer.
               ⋮
```

2. Do not set TMC504 to 1. When performing the TM50 and TM51 cascade connection, set TMC514 (bit 4 of 8-bit timer mode control register 51 (TMC51)) to 1.

Remarks 1. PWM output is at the inactive level in the PWM mode because TCE50 = 0.

2. If LVS50 and LVR50 are read immediately after data has been set, these bits are 0.

Figure 7-5. Format of 8-Bit Timer Mode Control Register 51 (TMC51)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TMC51	TCE51	TMC516	0	TMC514	LVS51	LVR51	TMC511	TOE51	FF7BH	04H	R/W

TCE51	TM51 count operation control
0	Disables count operation after clearing counter to 0 (disables prescaler)
1	Starts counting.

TMC516	TM51 operation mode selection
0	Clears and starts on match between TM51 and CR51
1	PWM (free-running) mode

TMC514	Single mode/cascade mode selection
0	Single mode
1	Cascade mode (connected to TM50)

LVS51	LVR51	Timer output F/F status setting of 8-bit timer/event counter 51
0	0	Not affected
0	1	Resets timer output F/F (to 0).
1	0	Sets timer output F/F (to 1).
1	1	Setting prohibited

TMC511	Other than PWM mode (TMC516 = 0)	PWM mode (TMC516 = 1)
	Timer F/F control	Active level selection
0	Disables inverted operation	Active high
1	Enables inverted operation	Active low

TOE51	Timer output control of 8-bit timer/event counter 51
0	Disables output (port mode)
1	Enables output

★ **Caution Before clearing TCE51 to 0, set the interrupt mask flag (TMMK51) to 1. This is because an interrupt may occur after TCE51 has been cleared. Clear TCE51 using the following procedure:**

```
TMMK51 = 1 ; Sets mask.
TCE51 = 0  ; Clears timer.
TMIF51 = 0 ; Clears interrupt request flag.
TMMK51 = 0 ; Clears mask.
           :
TCE51 = 1  ; Starts timer.
           :
```

Remarks 1. PWM output is at the inactive level in the PWM mode because TCE51 = 0.

2. If LVS51 and LVR51 are read immediately after data has been set, these bits are 0.

Figure 7-6. Format of 8-Bit Timer Mode Control Register 52 (TMC52)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TMC52	TCE52	TMC526	0	TMC524	LVS52	LVR52	TMC521	TOE52	FF7DH	04H	R/W

TCE52	TM52 count operation control
0	Disables count operation after clearing counter to 0 (disables prescaler)
1	Starts counting

TMC526	TM52 operation mode selection
0	Clears and starts on match between TM52 and CR52
1	PWM (free-running) mode

TMC524	Single mode/cascade mode selection
0	Single mode
1	Cascade mode (setting prohibited)

LVS52	LVR52	Timer output F/F status setting of 8-bit timer/event counter 52
0	0	Not affected
0	1	Resets timer output F/F (to 0).
1	0	Sets timer output F/F (to 1).
1	1	Setting prohibited

TMC521	Other than PWM mode (TMC526 = 0)	PWM mode (TMC526 = 1)
	Timer F/F control	Active level selection
0	Disables inverted operation	Active high
1	Enables inverted operation	Active low

TOE52	Timer output control of 8-bit timer/event counter 52
0	Disables output (port mode)
1	Enables output

★ **Cautions 1. Before clearing TCE52 to 0, set the interrupt mask flag (TMMK52) to 1. This is because an interrupt may occur after TCE52 has been cleared. Clear TCE52 using the following procedure:**

```
TMMK52 = 1 ; Sets mask.
TCE52 = 0  ; Clears timer.
TMIF52 = 0 ; Clears interrupt request flag.
TMMK52 = 0 ; Clears mask.
           :
TCE52 = 1  ; Starts timer.
           :
```

2. Do not set TMC524 to 1. The cascade connection mode cannot be used in TM52.

Remarks 1. PWM output is at the inactive level in the PWM mode because TCE52 = 0.

2. If LVS52 and LVR52 are read immediately after data has been set, these bits are 0.

(2) Timer clock selection registers 50, 51, 52 (TCL50, TCL51, TCL52)

These registers specify the count clock of 8-bit timer/event counters 50, 51, and 52 (TM50, TM51, and TM52) and the valid edges of TI50, TI51, and TI52 inputs.

TCL50, TCL51, and TCL52 are set using an 8-bit memory manipulation instruction.

RESET input sets these registers to 00H.

Figures 7-7 to 7-9 show the formats of TCL50, TCL51, and TCL52.

Figure 7-7. Format of Timer Clock Selection Register 50 (TCL50)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TCL50	0	0	0	0	0	TCL502	TCL501	TCL500	FF78H	00H	R/W

TCL502	TCL501	TCL500	Count clock selection
0	0	0	Falling edge of TI50
0	0	1	Rising edge of TI50
0	1	0	fx/2 (2.10 MHz)
0	1	1	fx/2^2 (1.05 MHz)
1	0	0	fx/2^3 (524 kHz)
1	0	1	fx/2^5 (131 kHz)
1	1	0	fx/2^7 (32.8 kHz)
1	1	1	fx/2^{11} (2.05 kHz)

Cautions 1. Before rewriting the data of TCL50, stop the timer operation once.

2. Be sure to set bits 3 to 7 of TCL50 to 0.

Remarks 1. fx: System clock oscillation frequency

2. ( ): @ fx = 4.19 MHz operation

Figure 7-8. Format of Timer Clock Selection Register 51 (TLC51)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TCL51	0	0	0	0	0	TCL512	TCL511	TCL510	FF7AH	00H	R/W

TCL512	TCL511	TCL510	Count clock selection
0	0	0	Falling edge of TI51
0	0	1	Rising edge of TI51
0	1	0	fx/2 (2.10 MHz)
0	1	1	fx/2^2 (1.05 MHz)
1	0	0	fx/2^5 (131 kHz)
1	0	1	fx/2^7 (32.8 kHz)
1	1	0	fx/2^9 (8.19 kHz)
1	1	1	fx/2^{11} (2.05 kHz)

Cautions 1. Before rewriting the data of TCL51, stop the timer operation once.

2. Be sure to set bits 3 to 7 of TCL51 to 0.

Remarks 1. fx: System clock oscillation frequency

2. ( ): @ fx = 4.19 MHz operation

3. The setting of TCL510 to TCL512 is invalid when TM50 and TM51 are connected in cascade.

Figure 7-9. Format of Timer Clock Selection Register 52 (TCL52)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TCL52	0	0	0	0	0	TCL522	TCL521	TCL520	FF7CH	00H	R/W

TCL522	TCL521	TCL520	Count clock selection
0	0	0	Falling edge of TI52
0	0	1	Rising edge of TI52
0	1	0	fx/2 (2.10 MHz)
0	1	1	fx/2^3 (262 kHz)
1	0	0	fx/2^4 (131 kHz)
1	0	1	fx/2^6 (65.5 kHz)
1	1	0	fx/2^8 (16.4 kHz)
1	1	1	fx/2^{10} (4.10 kHz)

Cautions 1. Before rewriting the data of TCL52, stop the timer operation once.

2. Be sure to set bits 3 to 7 of TCL52 to 0.

Remarks 1. fx: System clock oscillation frequency

2. ( ): @ fx = 4.19 MHz operation

(3) Port mode register 2 (PM2)

This register sets port 2 in the input or output mode in 1-bit units.

When pins P23/TI50/TO50, P26/TI52/TO52/ASCK0, and P27/TI51/TO51 are used for timer output, clear PM23, PM26, and PM27 and the output latches of P23, P26, and P27.

PM2 is set using a 1-bit or 8-bit memory manipulation instruction.

RESET input sets PM2 to FFH.

Figure 7-10. Format of Port Mode Register 2 (PM2)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FF22H	FFH	R/W

PM2n	P2n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

7.5 Operation of 8-Bit Timer/Event Counters 50, 51, 52

7.5.1 Interval timer (8-bit) operation

The 8-bit timer/event counters operate as interval timers that repeatedly generate an interrupt request at time intervals specified by the count values preset to 8-bit timer compare register 5n (CR5n).

When the count values of 8-bit timer counter 5n (TM5n) match the values set to the compare register CR5n, the value of TM5n is cleared to 0, TM5n continues counting, and at the same time on an interrupt request signal (INTTM5n) is generated.

The count clock of TM5n can be selected by bits 0 to 2 (TCL5n0 to TCL5n2) of timer clock selection register 5n (TCL5n).

For details of the operation when the value of the compare register is changed during timer count operation, refer to **7.6 Notes on 8-Bit Timer/Event Counters (2)**.

<Setting>

<1> Set each register.
- TCL5n: Selects count clock.
- CR5n: Compare value
- TMC5n: Selects clear & start mode entered on match between TM5n and CR5n (TMC5n = 0000×××0B × = Don’t care).

<2> The count operation is started when TCE5n is set to 1.

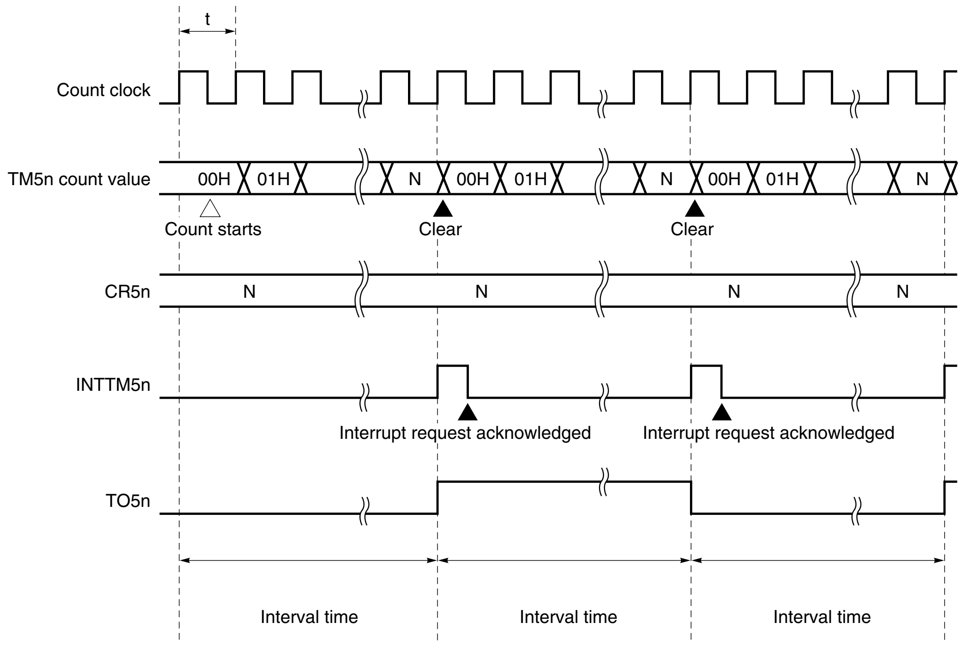
<3> INTTM5n occurs when the values of TM5n and CR5n match (TM5n is cleared to 00H).

<4> After that, INTTM5n repeatedly occurs at the same interval. Clear TCE5n to 0 to stop the count operation.

Remark n = 0 to 2

Figure 7-11. Interval Timer Operation Timing (1/3)

(a) Basic operation

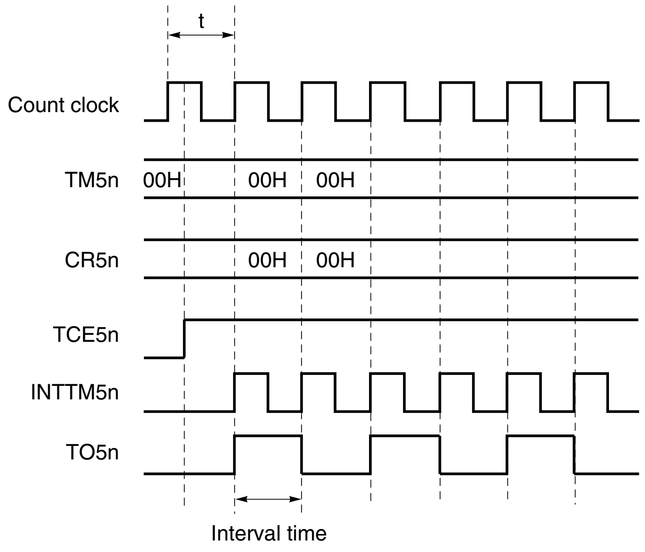


Remarks 1. Interval time = (N + 1) × t
N = 00H to FFH

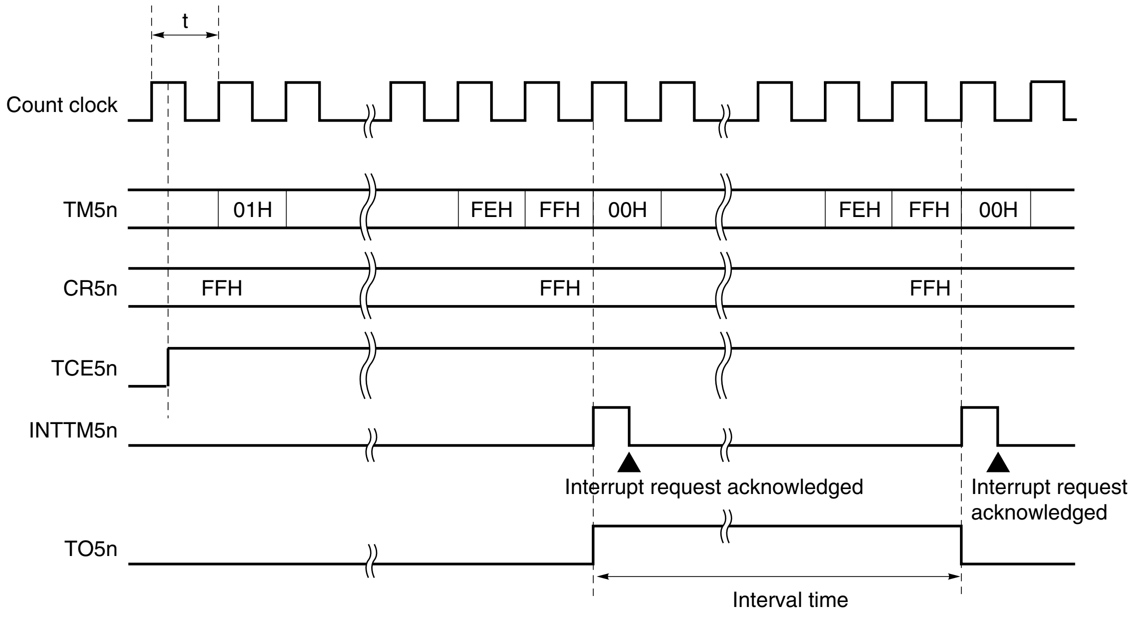
2. n = 0 to 2

Figure 7-11. Interval Timer Operation Timing (2/3)

(b) When CR5n = 00H



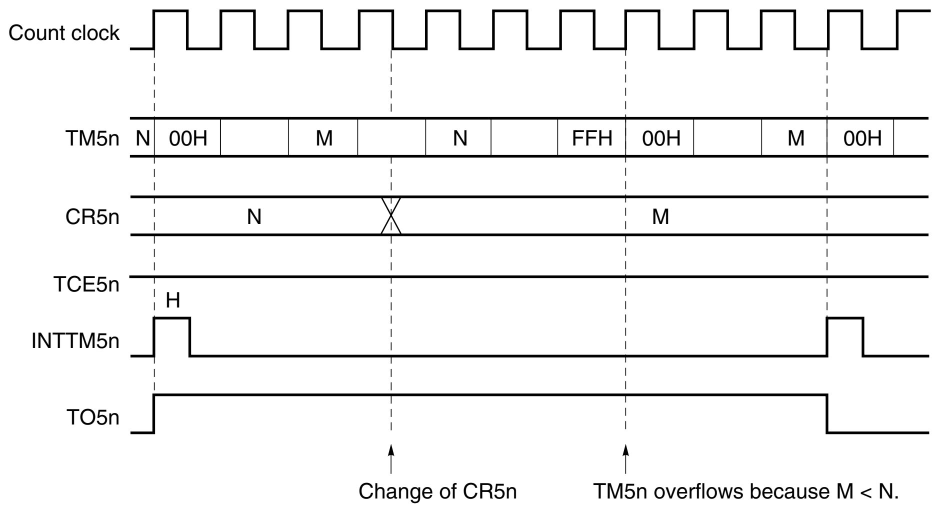
(c) When CR5n = FFH



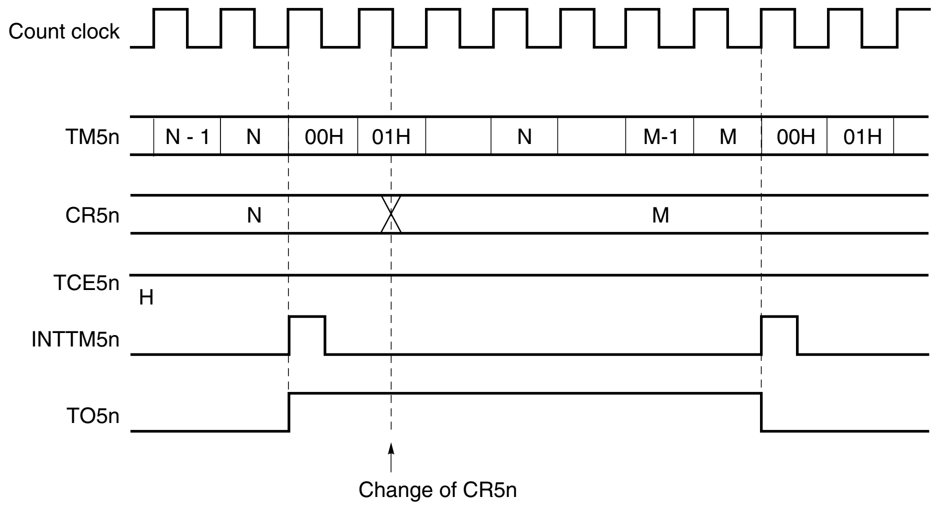
Remark n = 0 to 2

Figure 7-11. Interval Timer Operation Timing (3/3)

(d) Operation when CR5n is changed (M < N)



(e) Operation when CR5n is changed (M > N)



Remark n = 0 to 2

7.5.2 External event counter operation

The external event counter counts the number of clock pulses externally input to pins TI50/TO50/P23, TI51/TO51/ P27, and TI52/TO52/ASCK0/P26 by using 8-bit timer counter 5n (TM5n).

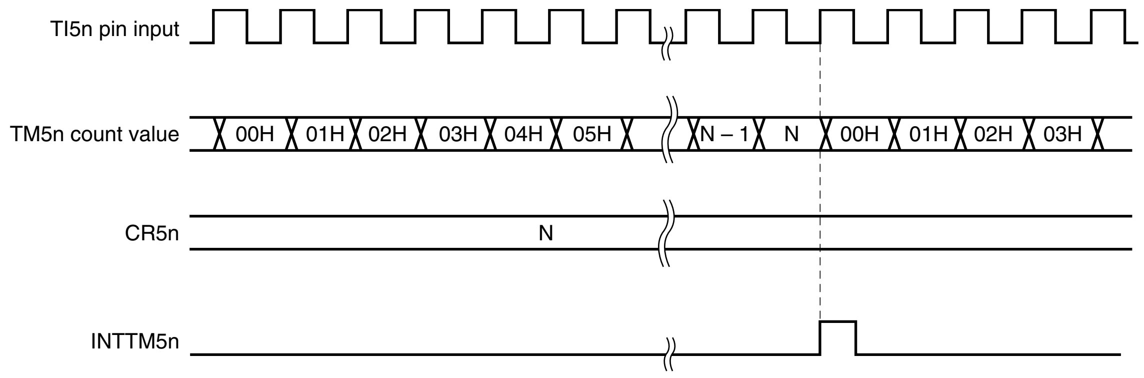
Each time the valid edge specified by timer clock selection register 5n (TCL5n) is input, the value of TM5n is incremented. Either the rising edge or falling edge can be specified as the valid edge.

When the count value of TM5n matches the values of 8-bit timer compare register 5n (CR5n), TM5n is cleared to 0 and an interrupt request signal (INTTM5n) is generated.

INTTM5n is generated whenever the TM5n value matches the value of CR5n.

Remark n = 0 to 2

Figure 7-12. External Event Counter Operation Timing (with Rising Edge Specified)



Remark N = 00H to FFH
n = 0 to 2

7.5.3 Square-wave output (8-bit resolution) operation

The 8-bit timer/event counters operate as a square-wave output at an interval preset to 8-bit timer compare register 5n (CR5n).

When bit 0 (TOE5n) of 8-bit timer mode control register 5n (TMC5n) is set to 1, the output status of TO5n is inverted at the interval time specified by the count value preset to CR5n. In this way, a square wave of any frequency (duty factor = 50%) can be output.

<Setting>

<1> Set each register.

- Clear the port latch and port mode register 2 (PM2) to 0.
- TCL5n: Selects count clock
- CR5n: Compare value
- TMC5n: Clear and start mode entered on match between TM5n and CR5n

LVS5n	LVR5n	Timer Output F/F Status Setting
1	0	High-level output
0	1	Low-level output

Enables inversion of timer output F/F

Timer output enabled → TOE5n = 1

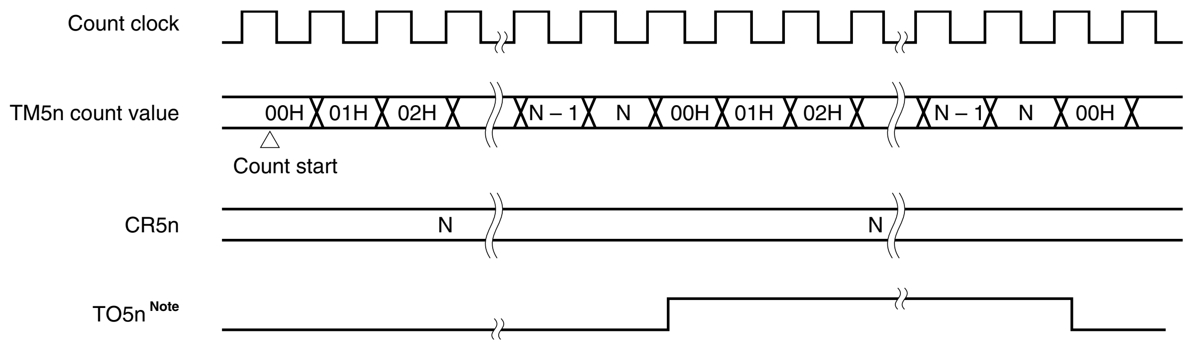
<2> The count operation is started if TCE5n is set to 1.

<3> The timer output F/F is inverted if the values of TM5n and CR5n match.
INTTM5n is generated and TM5n is cleared to 00H.

<4> After that, the timer output F/F is inverted at the same interval, and a square wave is output from TO5n.

Remark n = 0 to 2

Figure 7-13. Square-Wave Output Operation Timing



Note The initial value of the TO5n output can be set using bits 2 and 3 (LVR5n and LVS5n) of 8-bit timer mode control register 5n (TMC5n).

Remark n = 0 to 2

7.5.4 8-bit PWM output operation

The PWM output operation is performed when bit 6 (TMC5n6) of 8-bit timer mode control register 5n (TMC5n) is set to 1.

A pulse with a duty factor determined by the value set to 8-bit timer compare register 5n (CR5n) is output from TO5n.

Set CR5n to the width of the active level of the PWM pulse. The active level can be selected using bit 1 (TMC5n1).

The count clock can be selected by bits 0 to 2 (TCL5n0 to TCL5n2) of timer clock selection register 5n (TCL5n).

The PWM output can be enabled or disabled by bit 0 (TOE5n) of TMC5n.

★ **Caution CR5n can only be rewritten once per cycle in PWM mode.**

(1) Basic operation of PWM output

<Setting>

<1> Clear the port latch (P23, P26, P27) and port mode register 2 (PM23, PM26, PM27) to 0.
<2> Set an active level width using 8-bit timer compare register 5n (CR5n).
<3> Select a count clock using timer clock selection register 5n (TCL5n).
<4> Set an active level using bit 1 (TMC5n1) of TMC5n.
<5> The count operation is started when bit 7 (TCE5n) of TMC5n is set to 1.
To stop the count operation, clear TCE5n to 0.

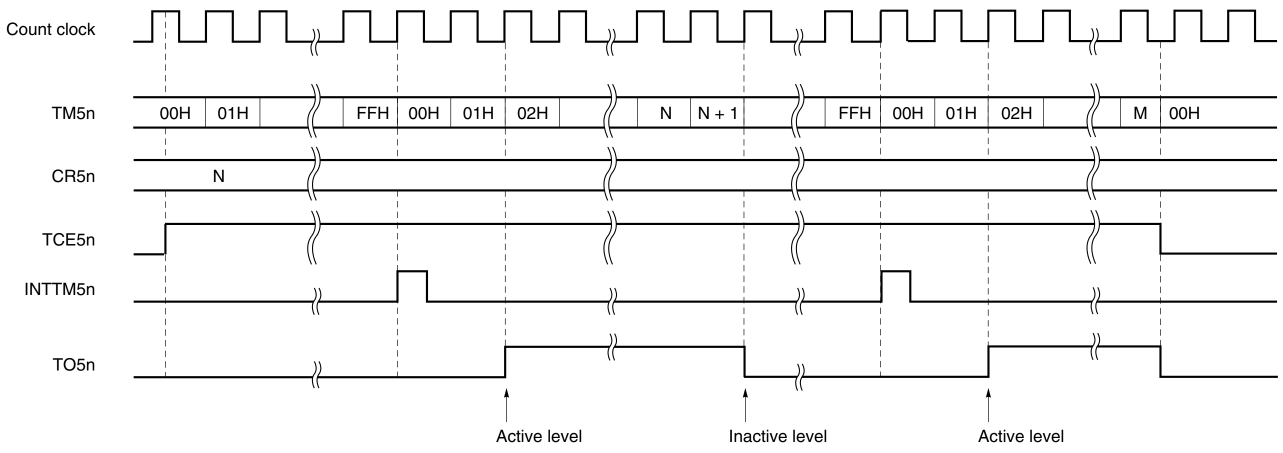
<PWM output operation>

<1> When the count operation is started, an inactive level is output as the PWM output (output from TO5n) until an overflow occurs.
<2> When an overflow occurs, the active level set in step <1> above is output. This active level is continuously output until the value of CR5n matches the count value of 8-bit timer counter 5n (TM5n).
<3> An inactive level is output as the PWM output after the value of CR5n has matched the count value of TM5n, until an overflow occurs again.
<4> After that, <2> and <3> are repeated until the count operation is stopped.
<5> When the count operation is stopped by clearing TCE5n to 0, the PWM output becomes inactive.

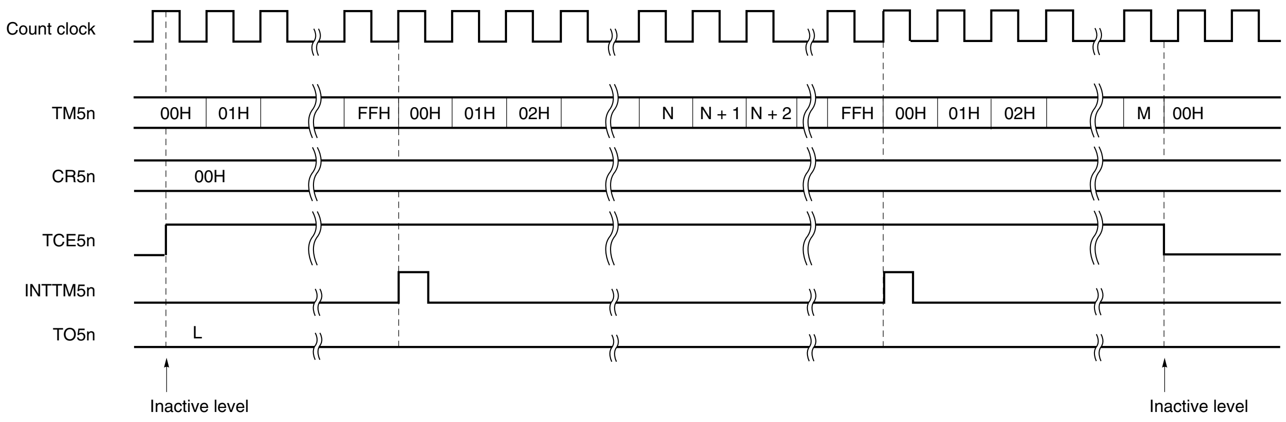
Remark n = 0 to 2

Figure 7-14. PWM Output Operation Timing

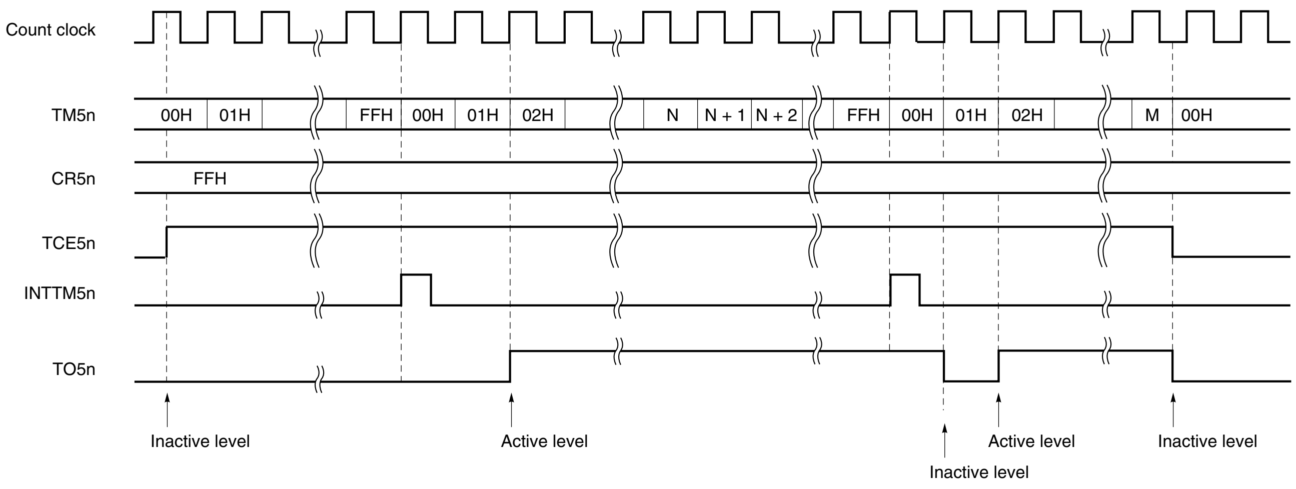
(a) Basic operation (when active level = H)



(b) When CR5n = 0



(c) When CR5n = FFH

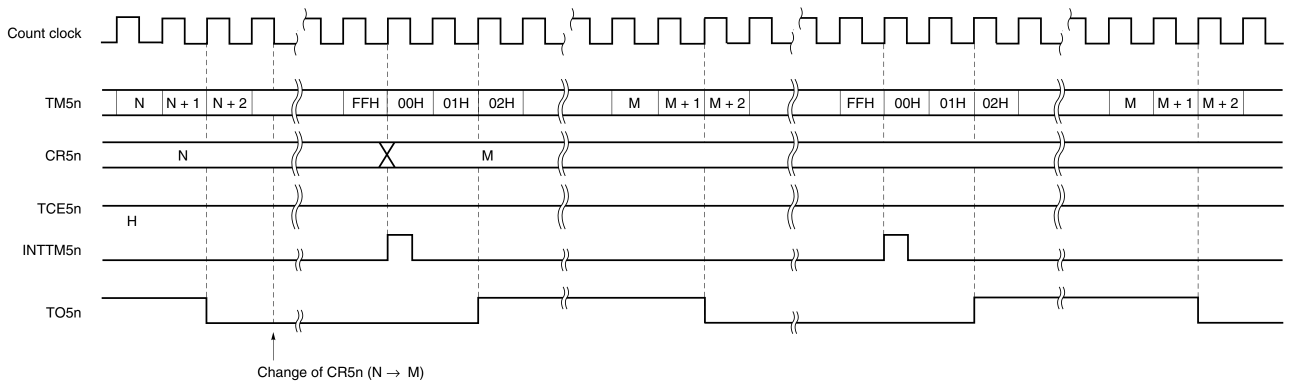


Remark n = 0 to 2

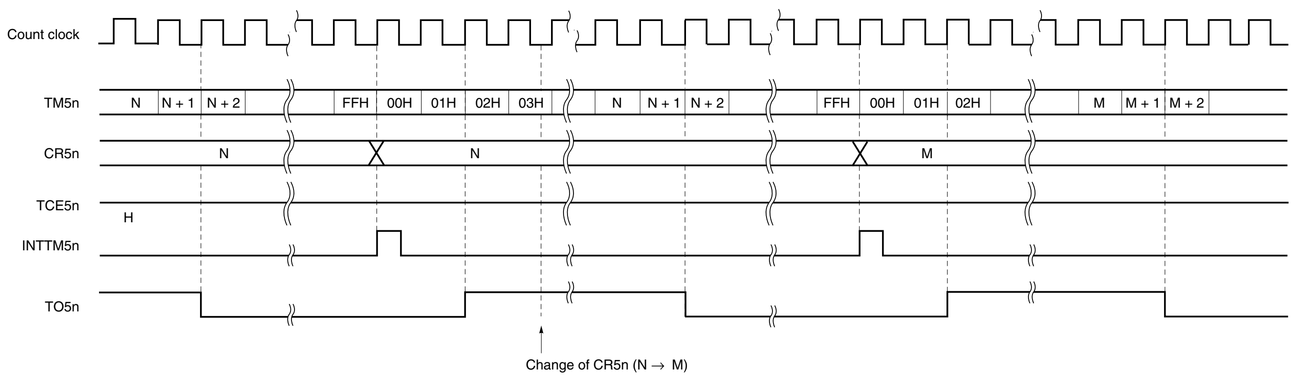
(2) Operation when CR5n is changed

Figure 7-15. Operation Timing When CR5n Is Changed

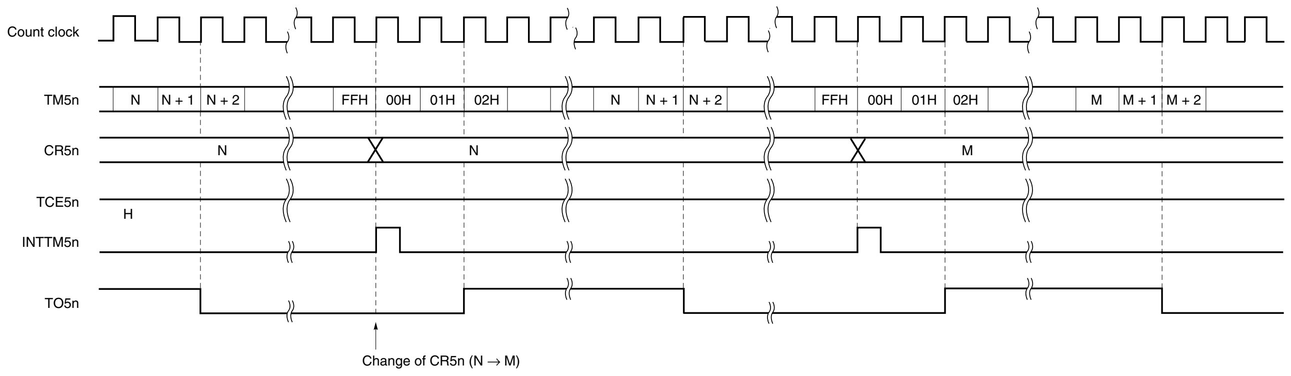
(a) If value of CR5n is changed from N to M before TM5n overflows



(b) If value of CR5n is changed from N to M after TM5n overflows



(c) If value of CR5n is changed from N to M during 2 clocks (00H, 01H) immediately after TM5n overflows



Remark n = 0 to 2

7.5.5 Interval timer (16-bit) operation

- **Cascade (16-bit timer) mode**

  The 16-bit resolution timer/counter mode is set by setting bit 4 (TMC514) of 8-bit timer mode control register 51 (TMC51) to 1.

  In this mode, TM50 and TM51 operate as a 16-bit interval timer that repeatedly generates an interrupt request at intervals specified by the count value preset to 8-bit timer compare registers 50 and 51 (CR50 and CR51).

<Setting>

<1> Set each register.

• TCL50:	TM50 selects a count clock. TM51, which is connected in cascade, does not have to be set.
• CR50 and CR51:	Compare values (each compare value can be set in a range of 00H to FFH).
• TMC50 and TMC51:	Select the mode that clears and starts the timer on a match between TM50 and CR50 (TM51 and CR51).

$$\begin{bmatrix} \text{TM50} \rightarrow \text{TMC50} = 0000\times\times\times0\text{B} \quad \times\text{: Don't care} \\ \text{TM51} \rightarrow \text{TMC51} = 0001\times\times\times0\text{B} \quad \times\text{: Don't care} \end{bmatrix}$$

<2> By setting TMC51 to TCE51 = 1 first, and then setting TMC50 to TCE50 = 1 for the count operation is started.

<3> When the value of TM50 connected in cascade matches the value of CR50, TM50 generates INTTM50 (TM50 and TM51 are cleared to 00H).

<4> After that, INTTM50 is repeatedly generated at the same interval.

Cautions 1. Be sure to set the compare registers (CR50 and CR51) after stopping the timer operation.

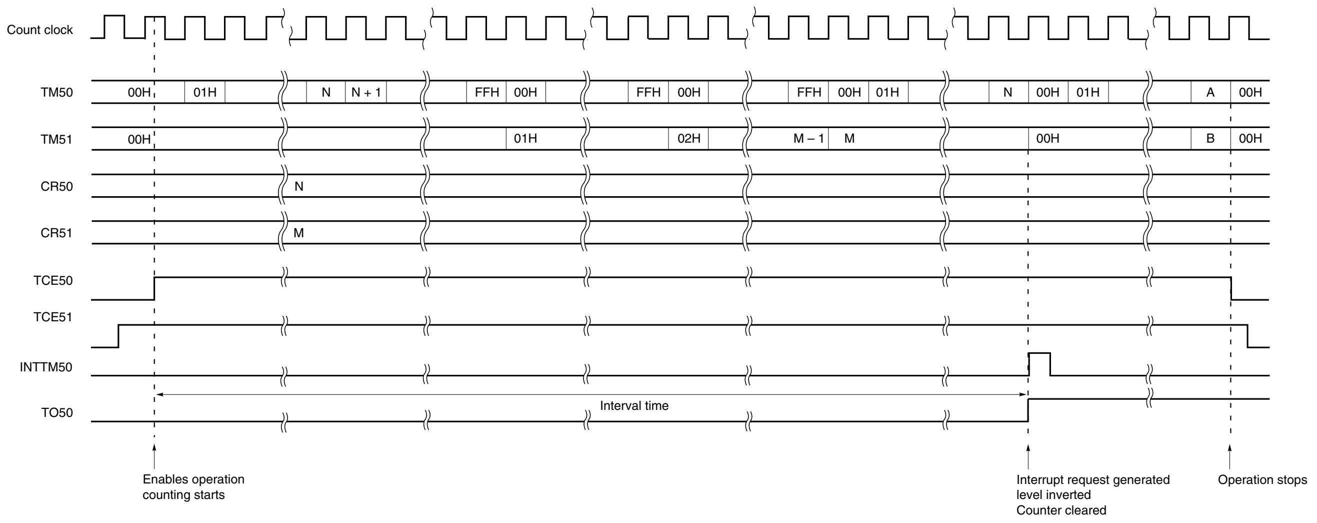
2. Even if the timers are connected in cascade, TM51 generates INTTM51 when the count value of TM51 matches the value of CR51. Be sure to mask TM51 to disable the generation of an interrupt.

3. Set TCE50 and TCE51 in the order of TM51, then TM50.

4. Counting can be started or stopped by setting or clearing only the TCE50 bit of TM50 to 1 or 0.

Figure 7-16 shows an example of the timing in the 16-bit resolution cascade mode.

Figure 7-16. 16-Bit Resolution Cascade Mode

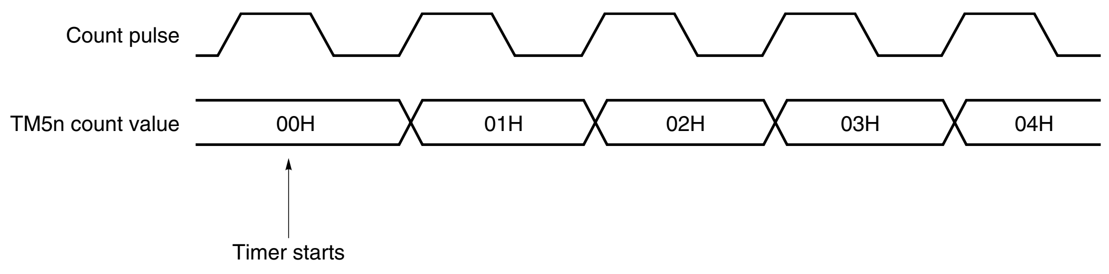


7.6 Notes on 8-Bit Timer/Event Counters

(1) Error on starting timer

An error of up to 1 clock occurs after the timer is started until a match signal is generated. This is because 8-bit timer counter 5n (TM5n: n = 0 to 2) is started asynchronously to the count pulse.

Figure 7-17. Start Timing of 8-Bit Timer/Counter

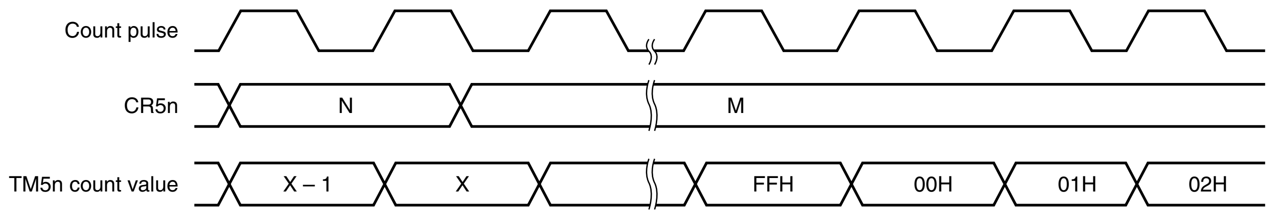


Remark n = 0 to 2

(2) Operation after changing value of compare register during timer count operation

If a new value of 8-bit timer compare register 5n (CR5n: n = 0 to 2) is less than the value of the corresponding 8-bit timer counter 5n (TM5n: n = 0 to 2), TM5n continues counting, overflows, and restarts counting from 0. Therefore, if the new value of CR5n (M) is less than its old value (N), it is necessary to restart the timer after changing the value of CR5n.

Figure 7-18. Timing After Changing Values of Compare Registers During Timer Count Operation



Caution Except when TI5n input is selected, be sure to clear TCE5n to 0 before setting the STOP mode.

Remark N > X > M
n = 0 to 2

(3) Reading TM5n during timer operation

Because the count clock is stopped when TM5n is read during operation, select a count clock with a waveform whose high-/low-level is longer than two CPU clock cycles. For example, in the case of a CPU clock (f_{CPU}) equal to f_X, TM5n can be read as long as the selected count clock is $f_X/4$ or lower.

Remark n = 0 to 2
f_X: System clock oscillation frequency

CHAPTER 8 WATCH TIMER

★ 8.1 Outline of Watch Timer

This timer can set a flag every 0.5 or 0.25 seconds and simultaneously generate an interrupt request at a preset time interval.

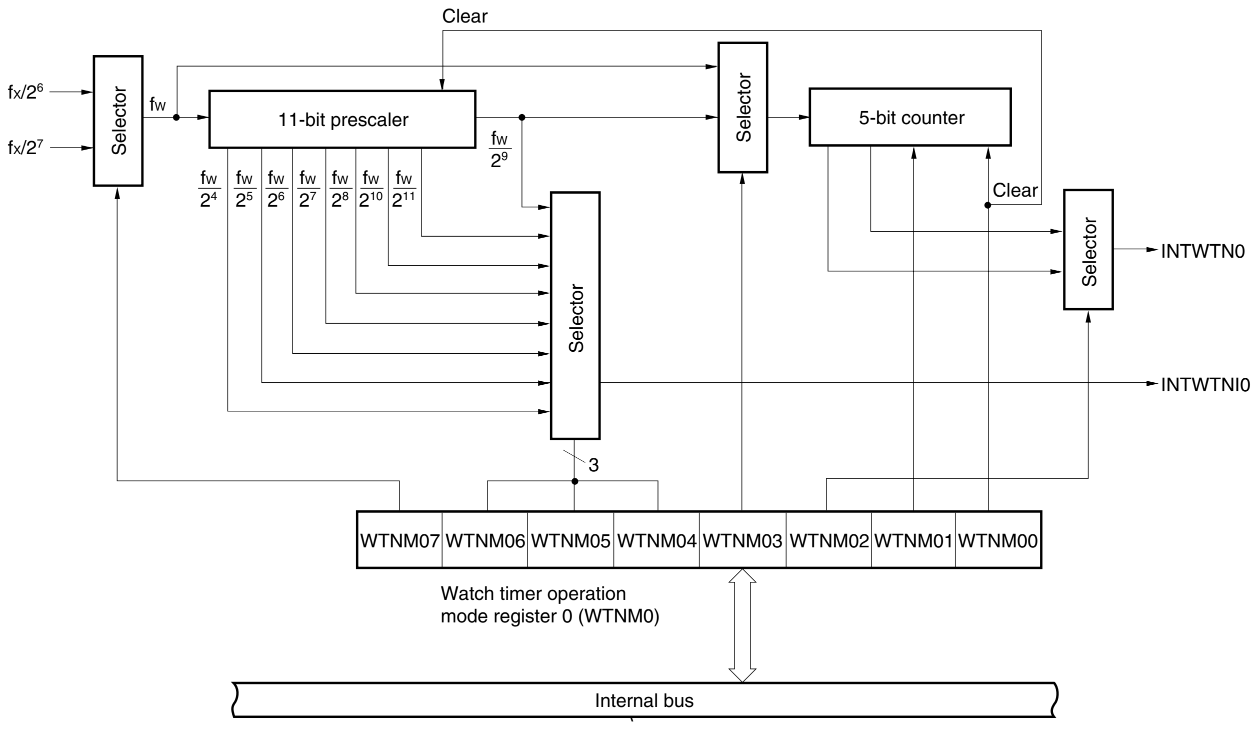
8.2 Function of Watch Timer

The watch timer has the following functions.

- Watch timer
- Interval timer

The watch timer and the interval timer can be used simultaneously.
Figure 8-1 shows a block diagram of the watch timer.

Figure 8-1. Block Diagram of Watch Timer


Clear
fx/2^6
fx/2^7
Selector
fw
11-bit prescaler
fw/2^9
fw/2^4 fw/2^5 fw/2^6 fw/2^7 fw/2^8 fw/2^10 fw/2^11
Selector
Selector
5-bit counter
Clear
Selector
INTWTN0
INTWTNI0
3
WTNM07 WTNM06 WTNM05 WTNM04 WTNM03 WTNM02 WTNM01 WTNM00
Watch timer operation mode register 0 (WTNM0)
Internal bus

Remark fx: System clock oscillation frequency
fw: Watch timer clock frequency

(1) Watch timer

The watch timer generates an interrupt request (INTWTN0) at a preset interval, such as 0.5 seconds or 0.25 seconds, using the system clock.

(2) Interval timer

The watch timer generates an interrupt request (INTWTNI0) at a preset time interval.

Table 8-1. Interval Time of Interval Timer

Interval Time	When Operated at fx = 4.194304 MHz
$2^{10} \times 1/f_X$	244 μs
$2^{11} \times 1/f_X$	488 μs
$2^{12} \times 1/f_X$	977 μs
$2^{13} \times 1/f_X$	1.95 ms
$2^{14} \times 1/f_X$	3.91 ms
$2^{15} \times 1/f_X$	7.81 ms
$2^{16} \times 1/f_X$	15.6 ms
$2^{17} \times 1/f_X$	31.3 ms
$2^{18} \times 1/f_X$	62.6 ms

Remark fx: System clock oscillation frequency

8.3 Configuration of Watch Timer

The watch timer includes the following hardware.

Table 8-2. Configuration of Watch Timer

Item	Configuration
Prescaler	11 bits × 1, 5 bits × 1
Control register	Watch timer operation mode register 0 (WTNM0)

8.4 Registers Controlling Watch Timer

Watch timer operation mode register 0 (WTNM0) is a register used to control the watch timer.

- **Watch timer operation mode register 0 (WTNM0)**
  This register enables/disables the watch timer operation, and sets the 11-bit prescaler interval time and 5-bit counter operation control. WTNM0 is set using a 1-bit or 8-bit memory manipulation instruction.
  $\overline{\text{RESET}}$ input sets WTNM0 to 00H.

Figure 8-2. Format of Watch Timer Operation Mode Register 0 (WTNM0)

Address: FF41H After reset: 00H R/W

Symbol	7	6	5	4	3	2	$\boxed{1}$	$\boxed{0}$
WTNM0	WTNM07	WTNM06	WTNM05	WTNM04	WTNM03	WTNM02	WTNM01	WTNM00

WTNM07	Watch timer count clock selection
0	fx/2^7 (32.7 kHz)
1	fx/2^6 (65.5 kHz)

WTNM06	WTNM05	WTNM04	11-bit prescaler interval time selection
0	0	0	2^4/fw
0	0	1	2^5/fw
0	1	0	2^6/fw
0	1	1	2^7/fw
1	0	0	2^8/fw
1	0	1	2^9/fw
1	1	0	2^{10}/fw
1	1	1	2^{11}/fw

WTNM03	WTNM02	Selection of watch timer interrupt request timing		
			WTNM07 = 0	WTNM07 = 1
0	0	2^{14}/fw	2^{21}/fx (0.5 s)	2^{20}/fx (0.25 s)
0	1	2^{13}/fw	2^{20}/fx (0.25 s)	2^{19}/fx (0.125 s)
1	0	2^5/fw	2^{12}/fx (977 μs)	2^{11}/fx (488 μs)
1	1	2^4/fw	2^{11}/fx (488 μs)	2^{10}/fx (244 μs)

WTNM01	5-bit counter operation control
0	Clear after operation stop
1	Start

WTNM00	Watch timer enable operation
0	Operation stopped (both 11-bit prescaler and 5-bit counter cleared)
1	Operation enabled

Remarks 1. fw: Watch timer clock frequency (fx/2^6 or fx/2^7)
2. fx: System clock oscillation frequency
3. Figures in parentheses apply to operation with fx = 4.194304 MHz.

8.5 Operation of Watch Timer

8.5.1 Watch timer operation

The watch timer operates with preset time intervals such as 0.5 seconds using the system clock (4.194304 MHz).

Bits 2, 3, and 7 (WTNM02, WTNM03, WTNM07) of watch timer operation mode register 0 (WTNM0) are used to select the watch timer time.

The watch timer generates an interrupt request (INTWTN0) at a constant time interval.

If bit 0 (WTNM00) and bit 1 (WTNM01) of watch timer operation mode register 0 (WTNM0) are set to 1, the count operation starts. If set to 0, the 5-bit counter is cleared and the count operation stops.

For simultaneous operation of the interval timer, a zero-second start is possible by setting WTNM01 to 0.

However, in this case, since the 11-bit prescaler is not cleared, at the first overflow (INTWTN0) after the clock timer's zero second start, an error of up to $2^{11} \times 1/f_W$ seconds is generated.

8.5.2 Interval timer operation

The watch timer operates as interval timer that repeatedly generates an interrupt request (INTWTNI0) at an interval specified by a preset count value.

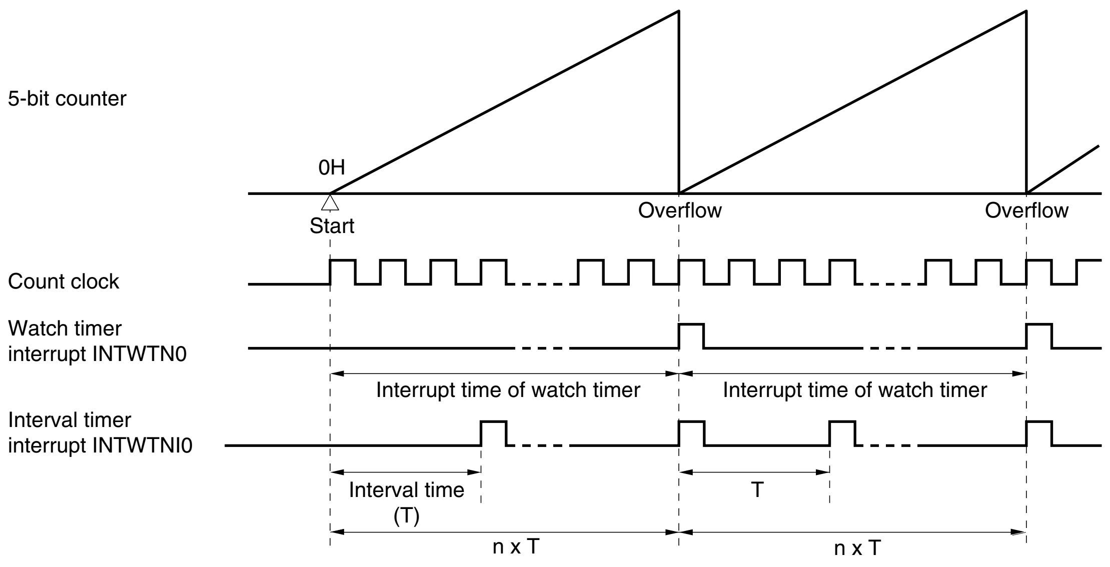
The interval time can be selected using bits 4 to 6 and 7 (WTNM04 to WTNM06 and WTNM07) of watch timer operation mode register 0 (WTNM0).

Table 8-3. Interval Time of Interval Timer

WTNM06	WTNM05	WTNM04	Interval Time	When Operated at f_X = 4.194304 MHz	
				WTNM07 = 0	WTNM07 = 1
0	0	0	$2^4 \times 1/f_W$	488 μs	244 μs
0	0	1	$2^5 \times 1/f_W$	977 μs	488 μs
0	1	0	$2^6 \times 1/f_W$	1.95 ms	977 μs
0	1	1	$2^7 \times 1/f_W$	3.91 ms	1.95 ms
1	0	0	$2^8 \times 1/f_W$	7.81 ms	3.91 ms
1	0	1	$2^9 \times 1/f_W$	15.6 ms	7.81 ms
1	1	0	$2^{10} \times 1/f_W$	31.3 ms	15.6 ms
1	1	1	$2^{11} \times 1/f_W$	62.6 ms	31.3 ms

Remark f_X: System clock oscillation frequency
f_W: Watch timer clock frequency

Figure 8-3. Operation Timing of Watch Timer/Interval Timer


5-bit counter
0H
Start
Overflow
Overflow
Count clock
Watch timer interrupt INTWTN0
Interrupt time of watch timer
Interrupt time of watch timer
Interval timer interrupt INTWTNI0
Interval time (T)
T
n x T
n x T

★ **Caution** **If the watch timer and 5-bit counter are enabled by watch timer mode control register 0 (WTNM0) (by setting bits 0 (WTNM00) and 1 (WTNM01) of WTNM0 to 1), the time from this setting to the occurrence of the first interrupt request (INTWTN0) is not exactly the value set by bit 3 (WTNM03) of WTNM0. This is because the 5-bit counter starts counting one cycle later than the output of the 11-bit prescaler. The second INTWTN0 signal and those that follow are generated at exactly the set time.**

Remark n: The number of interval timer operations

CHAPTER 9 WATCHDOG TIMER

★ 9.1 Outline of Watchdog Timer

The watchdog timer can also be used to generate a non-maskable interrupt request, maskable interrupt request, or $\overline{\text{RESET}}$ signal at a preset time interval.

9.2 Function of Watchdog Timer

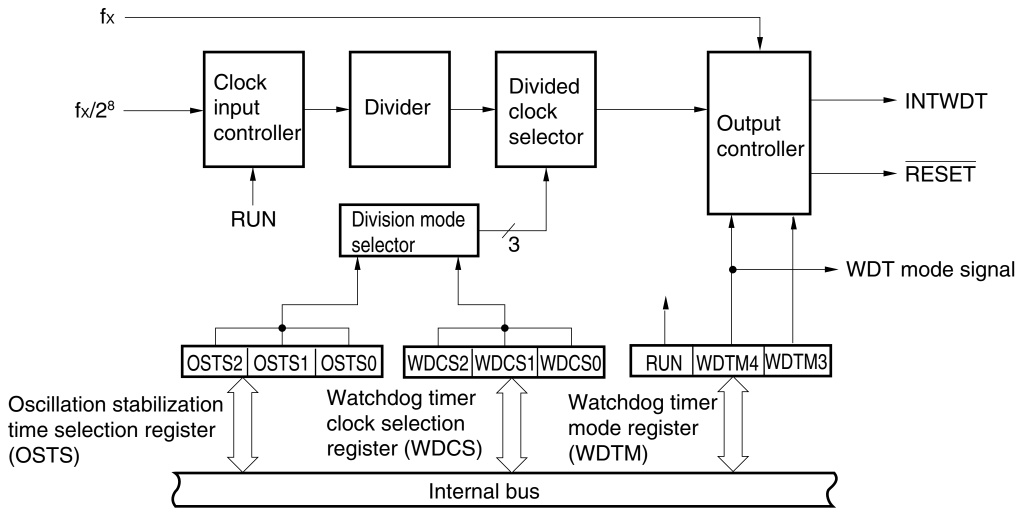
The watchdog timer has the following functions.

- Watchdog timer
- Interval timer
- Oscillation stabilization time selection

Caution Select the watchdog timer mode or the interval timer mode using the watchdog timer mode register (WDTM). (The watchdog timer and the interval timer cannot be used simultaneously.)

Figure 9-1 shows a block diagram of the watchdog timer.

★ **Figure 9-1. Block Diagram of Watchdog Timer**



Remark fx: System clock oscillation frequency

(1) Watchdog timer mode

In this node, the watchdog timer detects inadvertent program loops. Upon detection of the program loop, a non-maskable interrupt request or $\overline{\text{RESET}}$ can be generated.

Table 9-1. Watchdog Timer Program Loop Detection Time

Program Loop Detection Times
$2^{12} \times 1/f_X$ (977 μs)
$2^{13} \times 1/f_X$ (1.95 ms)
$2^{14} \times 1/f_X$ (3.91 ms)
$2^{15} \times 1/f_X$ (7.81 ms)
$2^{16} \times 1/f_X$ (15.6 ms)
$2^{17} \times 1/f_X$ (31.3 ms)
$2^{18} \times 1/f_X$ (62.6 ms)
$2^{20} \times 1/f_X$ (250 ms)

Remarks 1. f_X: System clock oscillation frequency

2. Figures in parentheses apply to operation with f_X = 4.19 MHz

(2) Interval timer mode

The watchdog timer generates an interrupt request at a preset time interval.

Table 9-2. Interval Time

Interval Time
$2^{12} \times 1/f_X$ (977 μs)
$2^{13} \times 1/f_X$ (1.95 ms)
$2^{14} \times 1/f_X$ (3.91 ms)
$2^{15} \times 1/f_X$ (7.81 ms)
$2^{16} \times 1/f_X$ (15.6 ms)
$2^{17} \times 1/f_X$ (31.3 ms)
$2^{18} \times 1/f_X$ (62.6 ms)
$2^{20} \times 1/f_X$ (250 ms)

Remarks 1. f_X: System clock oscillation frequency

2. Figures in parentheses apply to operation with f_X = 4.19 MHz

9.3 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

Table 9-3. Configuration of Watchdog Timer

Item	Configuration
Control registers	Watchdog timer clock selection register (WDCS) Watchdog timer mode register (WDTM) Oscillation stabilization time selection register (OSTS)

9.4 Registers Controlling Watchdog Timer

The following three registers are used to control the watchdog timer.

- Watchdog timer clock selection register (WDCS)
- Watchdog timer mode register (WDTM)
- Oscillation stabilization time selection register (OSTS)

(1) Watchdog timer clock selection register (WDCS)

This register sets the overflow time of the watchdog timer and the interval timer.
WDCS is set using an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets WDCS to 00H.

Figure 9-2. Format of Watchdog Timer Clock Selection Register (WDCS)

Address: FF42H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
WDCS	0	0	0	0	0	WDCS2	WDCS1	WDCS0

WDCS2	WDCS1	WDCS0	Overflow time of watchdog timer/interval timer
0	0	0	2^{12}/fx (977 μs)
0	0	1	2^{13}/fx (1.95 ms)
0	1	0	2^{14}/fx (3.91 ms)
0	1	1	2^{15}/fx (7.81 ms)
1	0	0	2^{16}/fx (15.6 ms)
1	0	1	2^{17}/fx (31.3 ms)
1	1	0	2^{18}/fx (62.6 ms)
1	1	1	2^{20}/fx (250 ms)

Remarks 1. fx: System clock oscillation frequency

2. Figures in parentheses apply to operation with fx = 4.19 MHz

(2) Watchdog timer mode register (WDTM)

This register sets the watchdog timer operation mode and enables/disables counting.

WDTM is set using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets WDTM to 00H.

Figure 9-3. Format of Watchdog Timer Mode Register (WDTM)

Address: FFF9H After reset: 00H R/W

Symbol	$\boxed{7}$	6	5	4	3	2	1	0
WDTM	RUN	0	0	WDTM4	WDTM3	0	0	0

RUN	Watchdog timer operation mode selection[Note 1]
0	Count stopped
1	Counter is cleared and counting starts

WDTM4	WDTM3	Watchdog timer operation mode selection[Note 2]
0	×	Interval timer mode[Note 3] (Maskable interrupt request occurs upon generation of overflow)
1	0	Watchdog timer mode 1 (Non-maskable interrupt request occurs upon generation of overflow)
1	1	Watchdog timer mode 2 (Reset operation is activated upon generation of overflow)

Notes 1. Once set to 1, RUN cannot be cleared to 0 by software.
Thus, once counting starts, it can only be stopped by $\overline{\text{RESET}}$ input.

2. Once set to 1, WDTM3 and WDTM4 cannot be cleared to 0 by software.

3. The watchdog timer starts operating as an interval timer when RUN is set to 1.

Caution When RUN is set to 1 so that the watchdog timer is cleared, the actual overflow time is up to
★ **2^8/fx seconds shorter than the time set by the watchdog timer clock selection register (WDCS).**

Remark ×: Don't care

(3) Oscillation stabilization time selection register (OSTS)

This register selects the oscillation stabilization time from when a reset is applied or STOP mode released until oscillation is stabilized.

OSTS is set using an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets OSTS to 04H. Thus, the time required to release the STOP mode by $\overline{\text{RESET}}$ input is 2^{17}/fx.

Figure 9-4. Format of Oscillation Stabilization Time Selection Register (OSTS)

Address: FFFAH After reset: 04H R/W

Symbol	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

OSTS2	OSTS1	OSTS0	Selection of oscillation stabilization time
0	0	0	2^{12}/fx (977 μs)
0	0	1	2^{14}/fx (3.91 ms)
0	1	0	2^{15}/fx (7.81 ms)
0	1	1	2^{16}/fx (15.6 ms)
1	0	0	2^{17}/fx (31.3 ms)
Other than above			Setting prohibited

Remarks 1. fx: System clock oscillation frequency

2. Figures in parentheses apply to operation with fx = 4.19 MHz

9.5 Operation of Watchdog Timer

9.5.1 Watchdog timer operation

When bit 4 (WDTM4) of the watchdog timer mode register (WDTM) is set to 1, the watchdog timer operates to detect any program loops.

The program loop detection time interval is selected using bits 0 to 2 (WDCS0 to WDCS2) of the watchdog timer clock selection register (WDCS).

The watchdog timer is started by setting bit 7 (RUN) of WDTM to 1. After the watchdog timer is started, set RUN to 1 within the set program loop detection time interval. The watchdog timer can be cleared and counting started by setting RUN to 1. If RUN is not set to 1 and the program loop detection time is exceeded, a system reset or non-maskable interrupt request is generated according to the value of the WDTM bit 3 (WDTM3).

The watchdog timer continues operating in the HALT mode but stops in the STOP mode. Thus, set RUN to 1 before the STOP mode is set to clear the watchdog timer, and then execute the STOP instruction.

Caution The actual program loop detection time may be shorter than the set time by up to of 0.5%.

Table 9-4. Watchdog Timer Program Loop Detection Time

Program Loop Detection Time
$2^{12} \times 1/f_X$ (977 μs)
$2^{13} \times 1/f_X$ (1.95 ms)
$2^{14} \times 1/f_X$ (3.91 ms)
$2^{15} \times 1/f_X$ (7.81 ms)
$2^{16} \times 1/f_X$ (15.6 ms)
$2^{17} \times 1/f_X$ (31.3 ms)
$2^{18} \times 1/f_X$ (62.6 ms)
$2^{20} \times 1/f_X$ (250 ms)

Remarks 1. f_X: System clock oscillation frequency

2. Figures in parentheses apply to operation with f_X = 4.19 MHz.

9.5.2 Interval timer operation

The watchdog timer operates as an interval timer that generates interrupt requests repeatedly at an interval specified by a preset count value when bit 4 (WDTM4) of the watchdog timer mode register (WDTM) is set to 0.

The interval time of interval timer is selected using bits 0 to 2 (WDCS0 to WDCS2) of the watchdog timer clock selection register (WDCS). When bit 7 (RUN) of WDTM is set to 1, the watchdog timer operates as an interval timer.

When the watchdog timer operates as an interval timer, the interrupt mask flag (WDTMK) and priority specification flag (WDTPR) are enabled and the maskable interrupt request (INTWDT) can be generated. Among maskable interrupts, INTWDT has the highest priority by default.

The interval timer continues operating in the HALT mode but stops in STOP mode. Thus, set RUN to 1 before the STOP mode is set to clear the interval timer, and then execute the STOP instruction.

Cautions 1. Once bit 4 (WDTM4) of WDTM is set to 1 (this selects the watchdog timer mode), the interval timer mode is not set unless $\overline{\text{RESET}}$ is input.

2. The interval time just after being set by WDTM may be shorter than the set time by up to 0.5%.

Table 9-5. Interval Time of Interval Timer

Interval Time
$2^{12} \times 1/f_X$ (977 μs)
$2^{13} \times 1/f_X$ (1.95 ms)
$2^{14} \times 1/f_X$ (3.91 ms)
$2^{15} \times 1/f_X$ (7.81 ms)
$2^{16} \times 1/f_X$ (15.6 ms)
$2^{17} \times 1/f_X$ (31.3 ms)
$2^{18} \times 1/f_X$ (62.6 ms)
$2^{20} \times 1/f_X$ (250 ms)

Remarks 1. f_X: System clock oscillation frequency

2. Figures in parentheses apply to operation with f_X = 4.19 MHz.

CHAPTER 10 CLOCK OUTPUT CONTROLLER

⋆ 10.1 Outline of Clock Output Controller

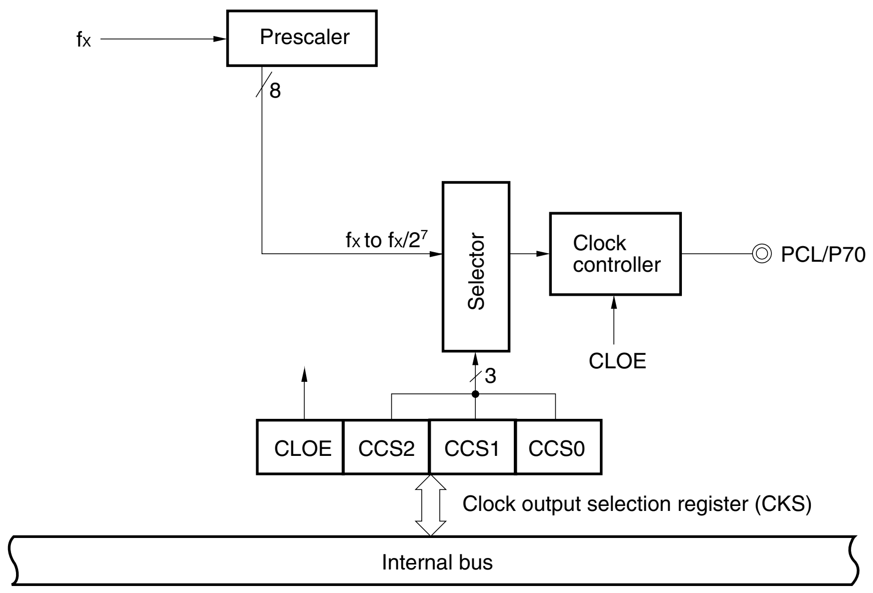
The clock output controller supplies other devices with the divided system clock.

10.2 Function of Clock Output Controller

The clock output controller is used for carrier output during remote control transmission and clock output for supplying peripheral LSIs. The clock selected with the clock output selection register (CKS) is output.

Figure 10-1 shows a block diagram of the clock controller.

Figure 10-1. Block Diagram of Clock Output Controller



10.3 Configuration of Clock Output Controller

The clock output controller includes the following hardware.

Table 10-1. Configuration of Clock Output Controller

Item	Configuration
Control registers	Clock output selection register (CKS) Port mode register 7 (PM7)Note

Note See **Figure 4-11 Block Diagram of P70 to P75**.

10.4 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 7 (PM7)

(1) Clock output selection register (CKS)

This register enables/disables output for clock output (PCL), and sets the output clock.

CKS is set using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CKS to 00H.

Figure 10-2. Format of Clock Output Selection Register (CKS)

Address: FF40H After reset: 00H R/W

Symbol	7	6	5	$\boxed{4}$	3	2	1	0
CKS	0	0	0	CLOE	0	CCS2	CCS1	CCS0

CLOE	PCL output enable/disable
0	Clock divider operation stopped. PCL fixed to low level
1	Clock divider operation enabled. PCL output enabled.

CCS2	CCS1	CCS0	PCL output clock selection
0	0	0	fx (4.19 MHz)
0	0	1	fx/2 (2.10 MHz)
0	1	0	fx/2^2 (1.05 MHz)
0	1	1	fx/2^3 (524 KHz)
1	0	0	fx/2^4 (262 kHz)
1	0	1	fx/2^5 (131 kHz)
1	1	0	fx/2^6 (65.6 kHz)
1	1	1	fx/2^7 (32.8 kHz)
Other than above			Setting prohibited

Caution Be sure to set bits 3 and 5 to 7 of CKS to 0.

Remarks 1. fx : System clock oscillation frequency

2. Figures in parentheses apply to operation with fx = 4.19 MHz.

(2) Port mode register 7 (PM7)

This register sets port 7 to input/output in 1-bit units.

When using the P70/PCL pin for clock output, set PM70 and the output latch of P70 to 0.

PM7 is set using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PM7 to FFH.

Figure 10-3. Format of Port Mode Register 7 (PM7)

Address: FF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	1	1	PM75	PM74	PM73	PM72	PM71	PM70

PM7n	P7n pin I/O mode selection (n = 0 to 5)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

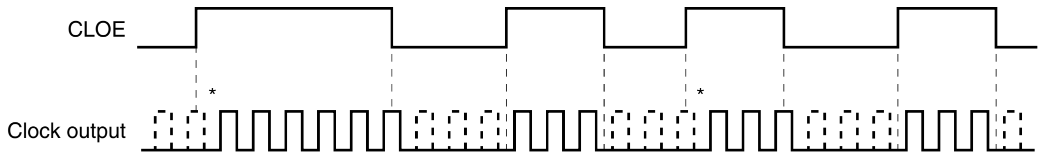
10.5 Operation of Clock Output Controller

The clock pulse is output by the following procedure.

<1> Select the clock pulse output frequency using bits 0 to 2 (CCS0 to CCS2) of the clock output selection register (CKS) (clock pulse output in disabled state).
<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 when clock output is switched between enable and disable. As shown in Figure 10-4, be sure to start output from the low period of the clock (marked with ***** in the figure). When stopping output, do so after securing the high level of the clock.

Figure 10-4. Remote Control Output Application Example



CHAPTER 11 A/D CONVERTER

Two A/D converters (AD00 and AD01) with 8-bit resolution are incorporated in the μPD780833Y Subseries. These two converters have exactly the same functions. Therefore, unless otherwise specified, AD00 is used throughout this chapter to describe the functions of both AD00 and AD01. If using AD01, substitute the register, bit and pin names shown in Table 11-1.

Table 11-1. AD00 and AD01 Name Differences

Item	AD00	AD01
Pin name	ANI00 to ANI70 AV_{SS0} AV_{REF0} AV_{DD0}	ANI01 to ANI71 AV_{SS1} AV_{REF1} **Note**
A/D converter mode register 0n	ADM00	ADM01
A/D converter mode register 0n address	FF80H	FF88H
A/D converter mode register 0n internal bit names	ADCS00 FR020 FR010 FR000	ADCS01 FR021 FR011 FR001
Analog input channel specification register 0n	ADS00	ADS01
Analog input channel specification register 0n address	FF81H	FF89H
Analog input channel specification register 0n internal bit names	ADS002 ADS001 ADS000	ADS012 ADS011 ADS010
A/D conversion results register 0n	ADCR00	ADCR01
A/D conversion results register 0n address	FF17H	FF8BH
Interrupt request name	INTAD00	INTAD01

Note The AV_{REF1} pin is also used for the AD01 analog power supply (see **Figure 11-2**).

Remark n = 0, 1

11.1 Function of A/D Converter

The A/D converter is an 8-bit resolution converter that converts analog inputs into digital values. It can control up to 8 analog input channels (ANI00 to ANI70).

An A/D conversion operation can only be started by software. Conversion is started by setting A/D converter mode register 00 (ADM00).

Select one channel for analog input from ANI0 to ANI7 to perform A/D conversion. The A/D conversion operation is executed repeatedly. Each time an A/D conversion operation ends, an interrupt request (INTAD00) is generated.

Figure 11-1. Block Diagram of A/D Converter (AD00)

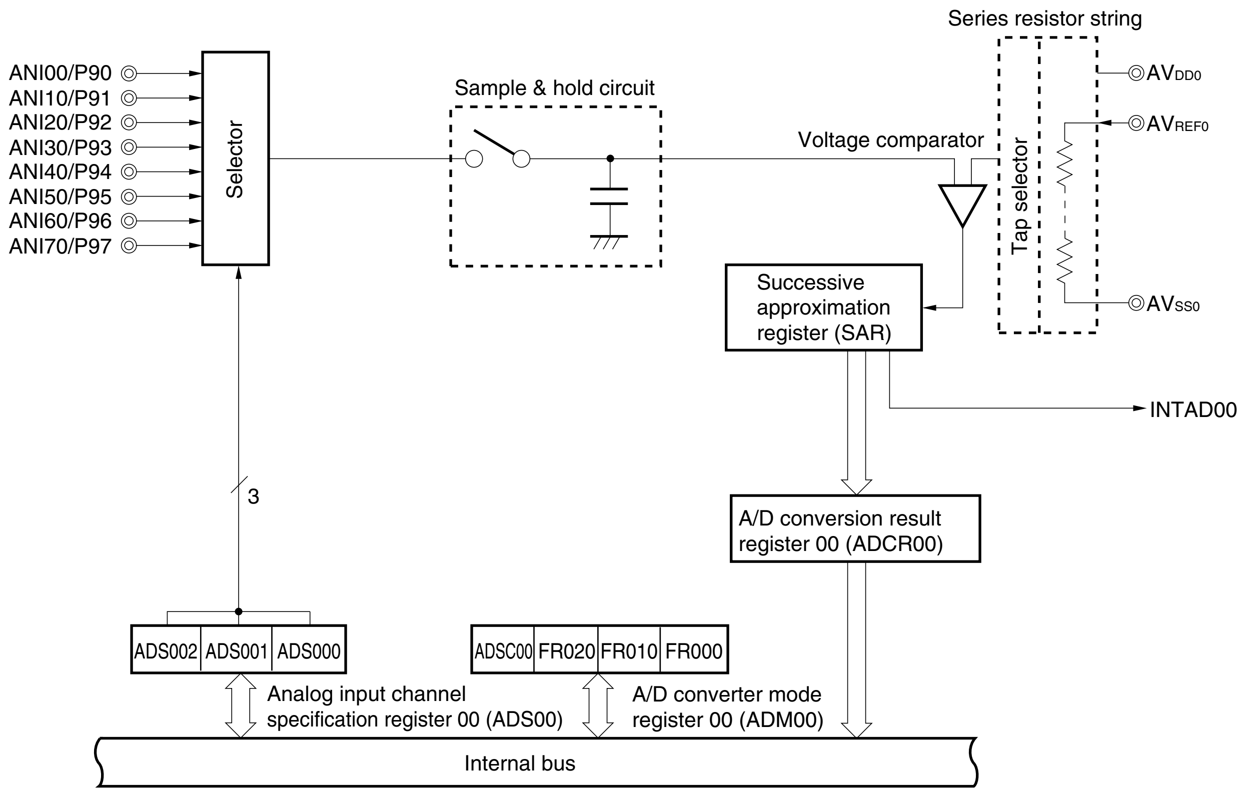
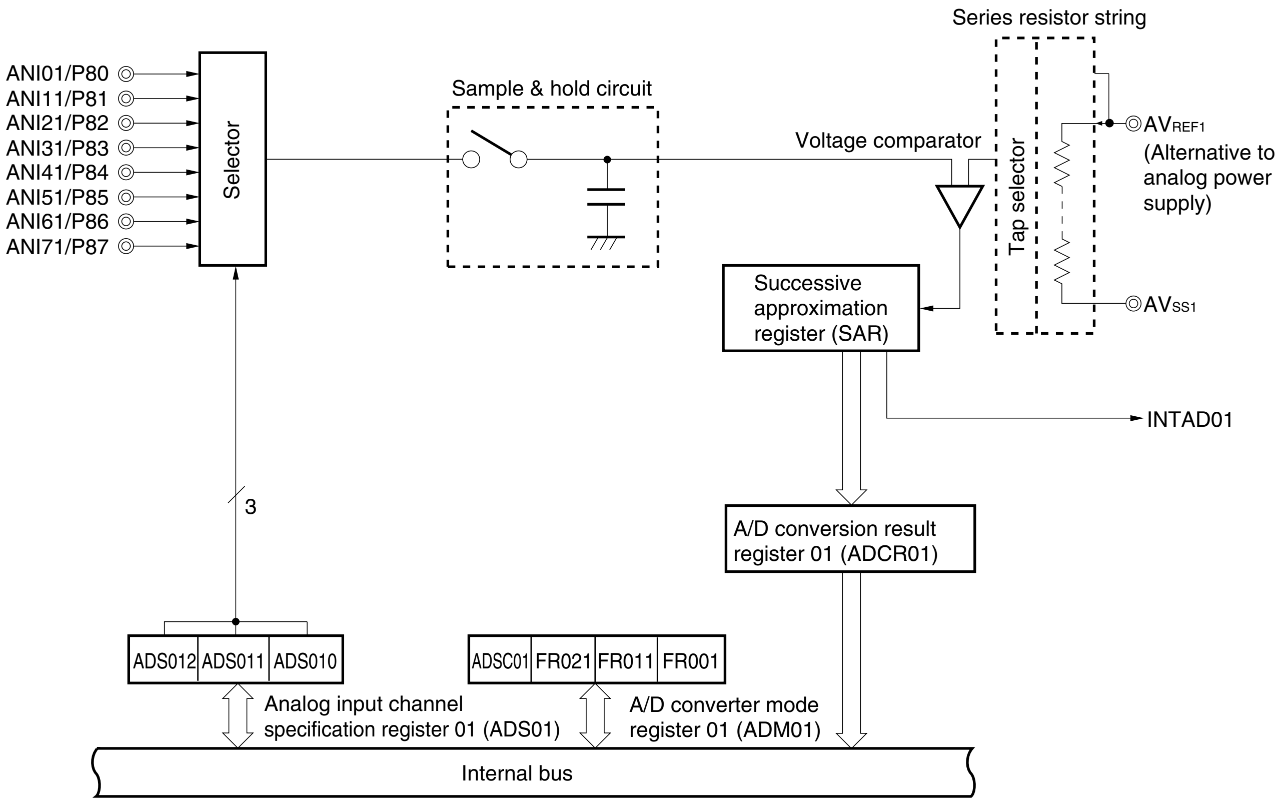


Figure 11-2. Block Diagram of A/D Converter (AD01)



11.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

Table 11-2. Configuration of A/D Converter

Item	Configuration
Analog input	8 channels (ANI00 to ANI70)
Registers	Successive approximation register (SAR) A/D conversion result register 00 (ADCR00)
Control registers	A/D converter mode register 00 (ADM00) Analog input channel specification register 00 (ADS00)

(1) Successive approximation register (SAR)

This register compares the analog input voltage value to the voltage tap (compare voltage) value applied from the series resistor string, and holds the result from the most significant bit (MSB).

When up to the least significant bit (LSB) is held (end of A/D conversion), the SAR contents are transferred to the A/D conversion result register.

(2) A/D conversion result register 00 (ADCR00)

ADCR00 is an 8-bit register that stores the A/D conversion result. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register.

ADCR00 is read by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes ADCR00 undefined.

Caution When A/D converter mode register 00 (ADM00) and analog input channel specification register 00 (ADS00) are written, the contents of ADCR00 may become undefined. Read the conversion result following conversion completion before writing to ADM00 and ADS00. Using a timing other than the above may cause an incorrect conversion result to be read.

(3) Sample & hold circuit

The sample & hold circuit samples each analog input signal sequentially applied from the input circuit, and sends it to the voltage comparator. This circuit holds the sampled analog input voltage value during A/D conversion.

(4) Voltage comparator

The voltage comparator compares the analog input to the series resistor string output voltage.

(5) Series resistor string

The series resistor string is connected between AV_{REF0} and AV_{SS0}, and generates a voltage to be compared to the analog input.

(6) ANI00 to ANI70 pins

These are eight analog input pins at which analog signals to undergo A/D conversion are input to the A/D converter. ANI00 to ANI70 are alternate-function pins that can also be used for digital input.

Cautions 1. Use the ANI00 to ANI70 input voltages within the specified range. If a voltage higher than AV_{REF0} or lower than AV_{SS0} is applied (even if within the absolute maximum rating range), the conversion value of that channel will be undefined and the conversion values of other channels may also be affected.

2. Analog input (ANI00 to ANI70) pins are alternate-function pins that can also be used as input port pins (P90 to P97). When A/D conversion is performed by selecting any one of ANI00 to ANI70, do not execute input instructions to port 9 during conversion, as this may cause the lower conversion resolution.

When a digital pulse is applied to a pin adjacent to the pin undergoing A/D conversion, A/ D conversion values may not be obtained as expected due to coupling noise. Thus, do not apply pulses to a pin adjacent to the pin undergoing A/D conversion.

(7) AV_{REF0}, AV_{REF1} pins

The AV_{REF0} pin inputs the reference voltage for A/D converter AD00.

This pin converts signals input to ANI00 to ANI70 into digital signals according to the voltage applied between AV_{REF0} and AV_{SS0}.

The AV_{REF1} pin inputs the reference voltage for A/D converter AD01.

This pin is also used for the analog power supply.

AV_{REF1} converts signals input to ANI01 to ANI71 into digital signals according to the voltage applied between AV_{REF1} and AV_{SS1}.

Even when A/D converter AD01 is not used, be sure to use the AV_{REF1} pin at the same potential as the V_{DD0} pin.

Caution A series resistor string of several tens of kΩ is connected between the AV_{REF0} pin and AV_{SS0} pin, and between the AV_{REF1} pin and AV_{SS1} pin. Therefore, if the output impedance of the reference voltage source is high, a series connection is created with the series resistor string between the AV_{REF0} pin and AV_{SS0} pin, and between the AV_{REF1} pin and AV_{SS1} pin, and the reference voltage error will become large.

(8) AV_{SS0}, AV_{SS1} pins

The AV_{SS0} pin is the GND potential pin for A/D converter AD00.

The AV_{SS1} pin is the GND potential pin for A/D converter AD01.

Be sure to use the AV_{SS0} and AV_{SS1} pins at the same potential as the V_{SS0} pin even when the A/D converter is not used.

(9) AV_{DD0} pin

This is the analog power supply pin for A/D converter AD00. Be sure to use this pin at the same potential as the V_{DD0} pin even when the A/D converter is not used.

11.3 Registers Controlling A/D Converter

The following two registers are used to control the A/D converter.

- A/D converter mode register 00 (ADM00)
- Analog input channel specification register 00 (ADS00)

(1) A/D converter mode register 00 (ADM00)

This register sets the conversion time for the analog input to be A/D converted and starts and stops conversion.

ADM00 is set using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ADM00 to 00H.

Figure 11-3. Format of A/D Converter Mode Register 00 (ADM00)

Address: FF80H After reset: 00H R/W

Symbol	$\boxed{7}$	6	5	4	3	2	1	0
ADM00	ADCS00	0	FR020	FR010	FR000	0	0	0

ADCS00	A/D conversion operation control
0	Conversion operation stopped.
1	Conversion operation enabled.

FR020	FR010	FR000	Conversion time selection[Note 1]
0	0	0	144/fx (34.3 μs)
0	0	1	120/fx (28.6 μs)
0	1	0	96/fx (22.9 μs)
1	0	0	72/fx (17.2 μs)
1	0	1	60/fx (14.3 μs)
1	1	0	48/fx (setting prohibited[Note 2])
Other than above			Setting prohibited

Notes 1. Set so that the A/D conversion time is 14 μs or more.

2. This setting is prohibited because the A/D conversion time is less than 14 μs.

Cautions 1. Be sure to set bits 0 to 2 and 6 of ADM00 to 0.

2. When rewriting FR000 to FR020 to other than the same data, stop A/D conversion operations once prior to rewriting.

Remarks 1. fx: System clock oscillation frequency

2. Figures in parentheses apply to operation with fx = 4.19 MHz.

(2) Analog input channel specification register 00 (ADS00)

This register specifies the analog voltage input port for A/D conversion.

ADS00 is set using an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ADS00 to 00H.

Figure 11-4. Format of Analog Input Channel Specification Register 00 (ADS00)

Address: FF81H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS00	0	0	0	0	0	ADS002	ADS001	ADS000

ADS002	ADS001	ADS000	Analog input channel specification
0	0	0	ANI00
0	0	1	ANI10
0	1	0	ANI20
0	1	1	ANI30
1	0	0	ANI40
1	0	1	ANI50
1	1	0	ANI60
1	1	1	ANI70

Caution Be sure to set bits 3 to 7 of ADS00 to 0.

11.4 Operation of A/D Converter

11.4.1 Basic operations of A/D converter

<1> Select one channel for A/D conversion using analog input channel specification register 00 (ADS00).

<2> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<3> When sampling has been performed for a certain time, the sample & hold circuit is placed in the hold state and the input analog voltage is held until the A/D conversion operation ends.

<4> Bit 7 of the successive approximation register (SAR) is set. The series resistor string voltage tap is set to (1/2) AV_{REF0} by the tap selector.

<5> The voltage difference between the series resistor string voltage tap and analog input is compared by the voltage comparator. If the analog input is greater than (1/2) AV_{REF0}, the MSB of SAR remains set. If the analog input is smaller than (1/2) AV_{REF0}, the MSB is reset.

<6> Next, bit 6 of SAR is automatically set, and the operation proceeds to the next comparison. The series resistor string voltage tap is selected according to the preset value of bit 7, as described below.

- Bit 7 = 1: (3/4) AV_{REF0}
- Bit 7 = 0: (1/4) AF_{REF0}

The voltage tap and analog input voltage are compared and bit 6 of SAR is manipulated as follows.

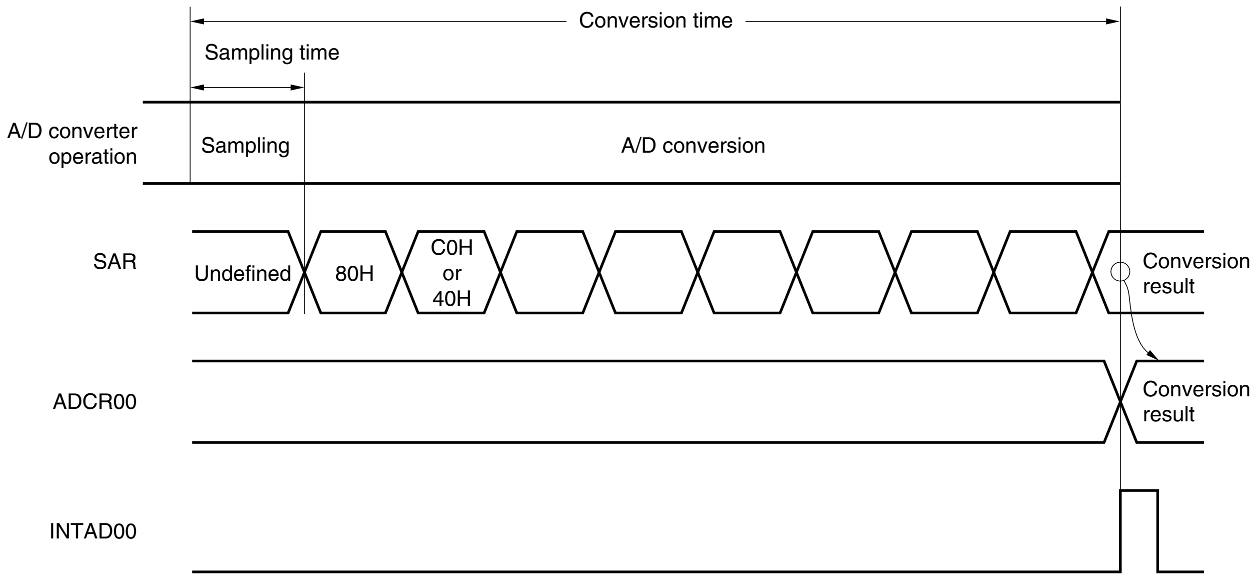
- Analog input voltage ≥ Voltage tap: Bit 6 = 1
- Analog input voltage < Voltage tap: Bit 6 = 0

<7> Comparison is continued in this way up to bit 0 of SAR.

<8> Upon completion of the comparison of 8 bits, a valid result value remains in SAR, and the result value is transferred to and latched in A/D conversion result register 00 (ADCR00).
At the same time, the A/D conversion end interrupt request (INTAD00) can also be generated.

Caution The first A/D conversion value just after A/D conversion operations start may not satisfy the ratings.

Figure 11-5. Basic Operation of A/D Converter



A/D conversion operations are performed continuously until bit 7 (ADCS00) of A/D converter mode register 00 (ADM00) is reset (0) by software.

If a write operation is performed to ADM00 or analog input channel specification register 00 (ADS00) during an A/D conversion operation, the conversion operation is initialized, and if the ADCS00 bit is set (1), conversion starts again from the beginning.

$\overline{\text{RESET}}$ input makes A/D conversion result register 00 (ADCR00) to undefined.

11.4.2 Input voltage and conversion results

The relationship between the analog input voltage input to the analog input pins (ANI00 to ANI70) and the A/D conversion result (stored in A/D conversion result register 00 (ADCR00)) is shown by the following expression.

$$\text{ADCR00} = \text{INT}\left(\frac{V_{IN}}{AV_{REF0}} \times 256 + 0.5\right)$$

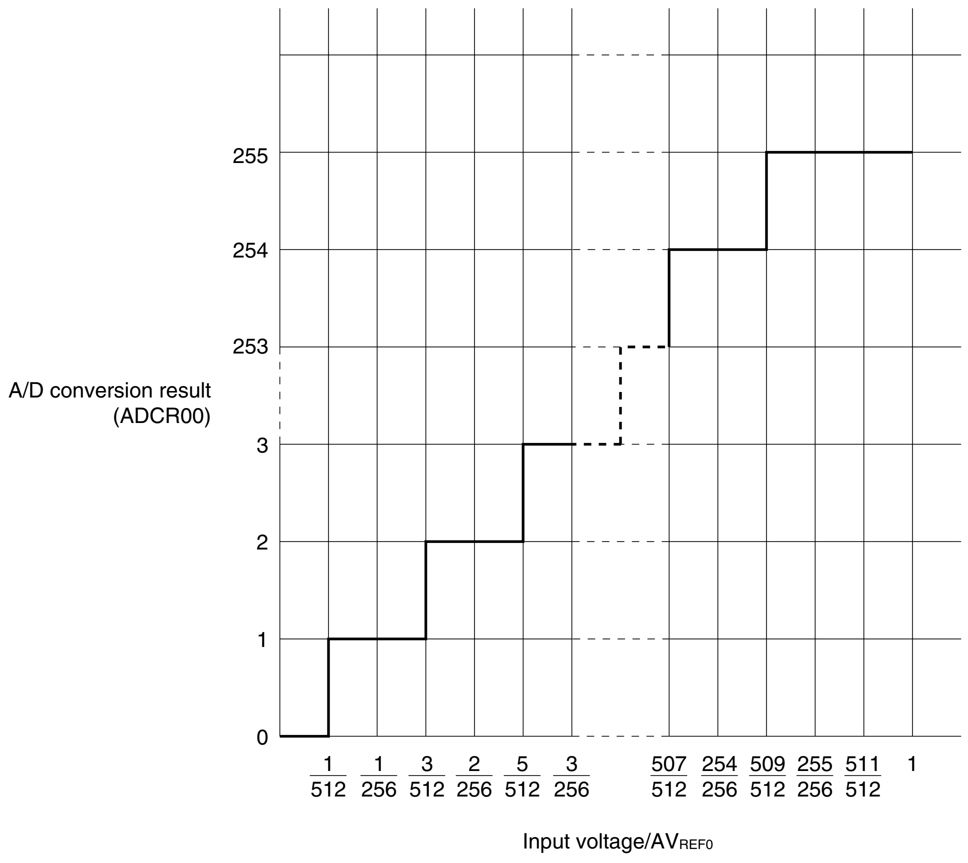
or

$$(\text{ADCR00} - 0.5) \times \frac{AV_{REF0}}{256} \leq V_{IN} < (\text{ADCR00} + 0.5) \times \frac{AV_{REF0}}{256}$$

where,	INT( ):	Function that returns integer part of value in parentheses
	V_{IN}:	Analog input voltage
	AV_{REF0}:	AV_{REF0} pin voltage
	ADCR0:	A/D conversion result register 00 (ADCR00) value

Figure 11-6 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 11-6. Relationship Between Analog Input Voltage and A/D Conversion Result



11.4.3 A/D converter operation mode

Select one analog input channel from among ANI00 to ANI70 using analog input channel specification register 00 (ADS00) and start A/D conversion.

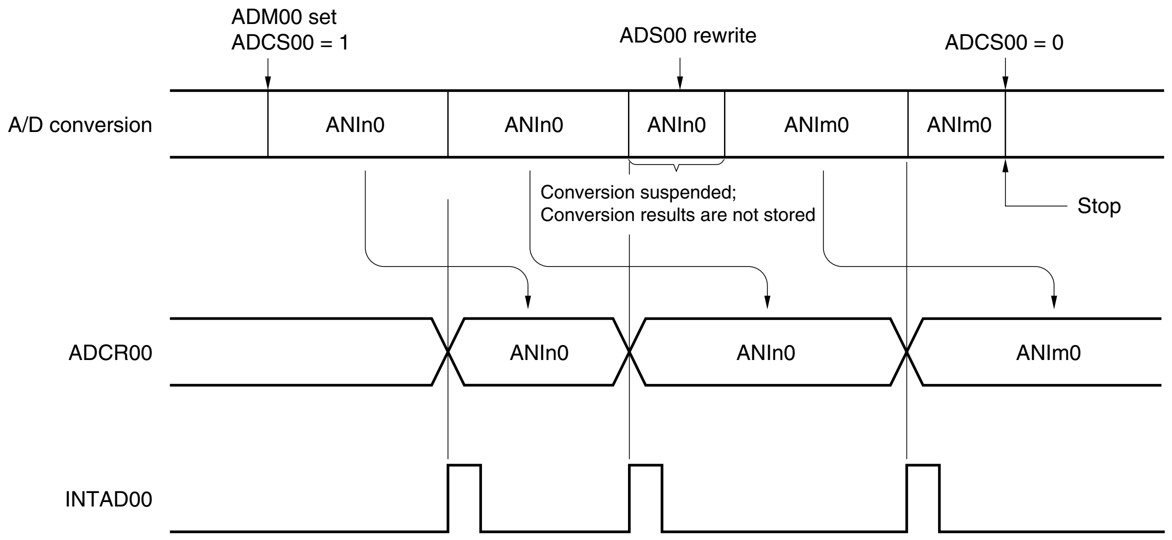
Setting bit 7 (ADCS00) of A/D converter mode register 00 (ADM00) to 1 starts A/D conversion of the voltage applied to the analog input pin specified by analog input channel specification register 00 (ADS00).

Upon the end of the A/D conversion, the conversion result is stored in A/D conversion result register 00 (ADCR00), and the interrupt request signal (INTAD00) is generated. After one A/D conversion operation is started and ended, the next conversion operation is immediately started. A/D conversion operations are repeated until new data is written to ADS00.

If ADS00 is rewritten during A/D conversion, the converter suspends A/D conversion and A/D conversion of the newly selected analog input channel is started.

If data in which ADCS00 is 0 is written to ADM00 during A/D conversion, the A/D conversion operation stops immediately.

Figure 11-7. A/D Conversion by Software Start



Remarks 1. n = 0, 1, ..., 7

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

★ ## 11.5 How to Read A/D Converter Characteristics Table

Terms used specifically for the A/D converter are defined below.

(1) Resolution

The lowest identifiable analog input voltage, or the ratio of the analog input voltage to one bit of a digital output, is known as 1LSB (Least Significant Bit). The ratio to the full scale of 1LSB is expressed as %FSR (full-scale range).

In the case of 10-bit resolution:

$$\begin{aligned} 1\text{LSB} &= 1/2^{10} = 1/1024 \\ &= 0.098\%\text{FSR} \end{aligned}$$

The accuracy is unrelated to the resolution, and is determined by the overall error.

(2) Overall error

This indicates the maximum difference between the actual and theoretical measurement values.
Zero-scale error, full-scale error, integral linearity error, differential linearity error, and combinations of these errors are expressed as the overall error.
Note that the quantization error is not included in the overall error.

(3) Quantization error

When analog values are converted to digital values, an error of ±1/2 LSB inevitably occurs. In an A/D converter, because analog input voltages within a range of ± 1/2 LSB are converted into the same digital code, this 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 values shown in the characteristics table.

Figure 11-8. Overall Error

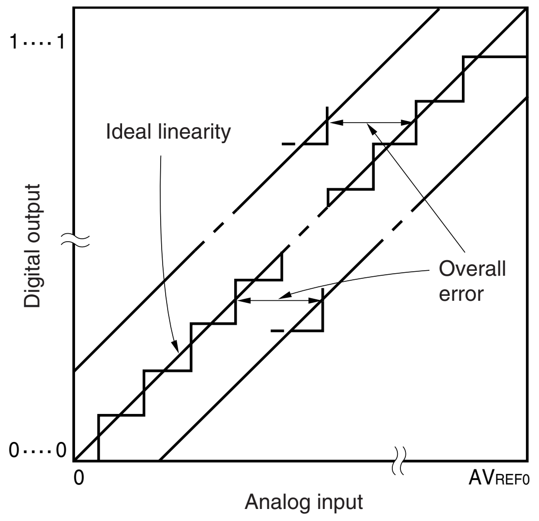
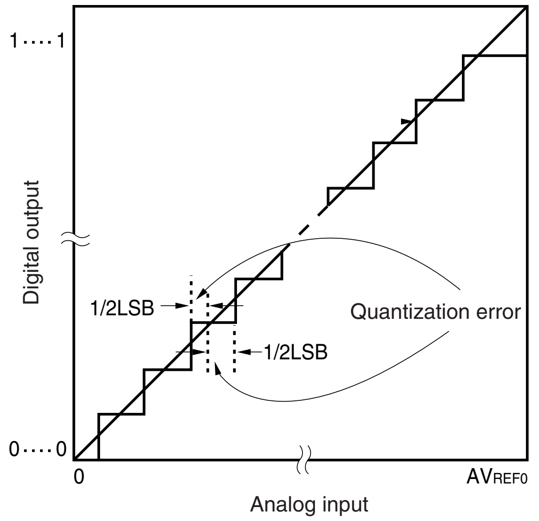


Figure 11-9. Quantization Error

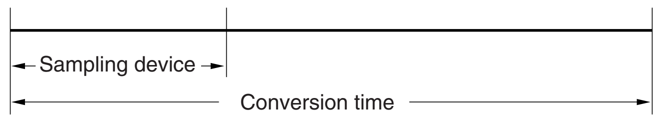


(4) Conversion time

This expresses the time from when the analog input voltage is applied to when the digital output is obtained. The sampling time is included in the conversion time value shown in the characteristics table.

(5) Sampling time

This is the time during which the analog switch is on to allow the analog voltage to be fetched in the sample & hold circuit.



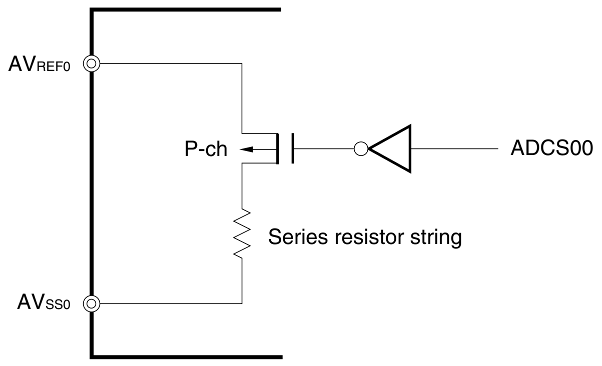
11.6 Notes on A/D Converter

(1) Current consumption in standby mode

The A/D converter stops operating in the standby mode (bit 7 (ADCS00) of A/D converter mode register 00 (ADM00) is set to 0), which enabled a reduction in the current consumption.

Figure 11-10 shows how to reduce the current consumption in the standby mode.

Figure 11-10. Example of Reducing Current Consumption in Standby Mode



(2) Input range of ANI00 to ANI70

The input voltages of ANI00 to ANI70 should be within the specified range. In particular, if a voltage higher than AV_{REF0} or lower than AV_{SS0} is input (even if within the absolute maximum rating range), the conversion value of that channel will be undefined and the conversion values of other channels may also be affected.

(3) Contention operations

<1> Contention between A/D conversion result register 00 (ADCR00) write and ADCR00 read by instruction at the end of conversion
ADCR00 read is given priority. After the read operation, the new conversion result is written to ADCR00.

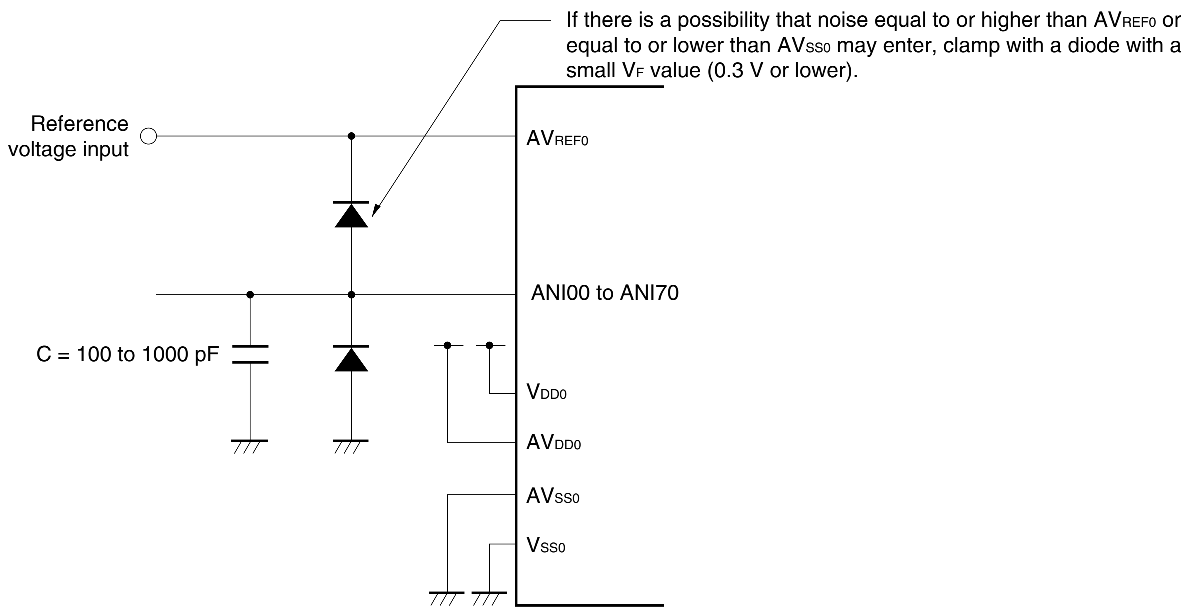
★ <2> Contention between ADCR00 write and external trigger signal input at the end of conversion
The external trigger signal is not acknowledged during A/D conversion. Therefore, the external trigger signal is not acknowledged during ADCR00 write.

<3> Contention between ADCR00 write and A/D converter mode register 00 (ADM00) write or analog input channel specification register 00 (ADS00) write
ADM00 or ADS00 write is given priority. ADCR00 write is not performed, nor is the conversion end interrupt request signal (INTAD00) generated.

(4) Noise countermeasures

To maintain the 8-bit resolution, attention must be paid to noise input to the AV_{REF0} and ANI00 to ANI70 pins. Because the effect increases in proportion to the output impedance of the analog input source, it is recommended that a capacitor be connected externally as shown in Figure 11-11 to reduce noise.

Figure 11-11. Analog Input Pin Connection


If there is a possibility that noise equal to or higher than AVREF0 or equal to or lower than AVSS0 may enter, clamp with a diode with a small VF value (0.3 V or lower).
Reference voltage input
AVREF0
ANI00 to ANI70
C = 100 to 1000 pF
VDD0
AVDD0
AVSS0
VSS0

(5) ANI00 to ANI70

The analog input pins (ANI00 to ANI70) also function as input port pins (P90 to P97).

When A/D conversion is performed with any of pins ANI00 to ANI70 selected, do not execute an input instruction to port 1 while conversion is in progress, as this may reduce the conversion resolution.

Also, if digital pulses are applied to a pin adjacent to the pin undergoing A/D conversion, the expected A/D conversion value may not be obtainable due to coupling noise. Therefore, avoid applying pulses to pins adjacent to the pin undergoing A/D conversion.

(6) AV_{REF0} pin input impedance

A series resistor string of several tens of kΩ is connected between the AV_{REF0} pin and the AV_{SS0} pin.

Therefore, when the output impedance of the reference voltage is too high, it seems as if the AV_{REF0} pin and the AV_{SS0} pin series resistor string are connected in series. This may cause a greater reference voltage error.

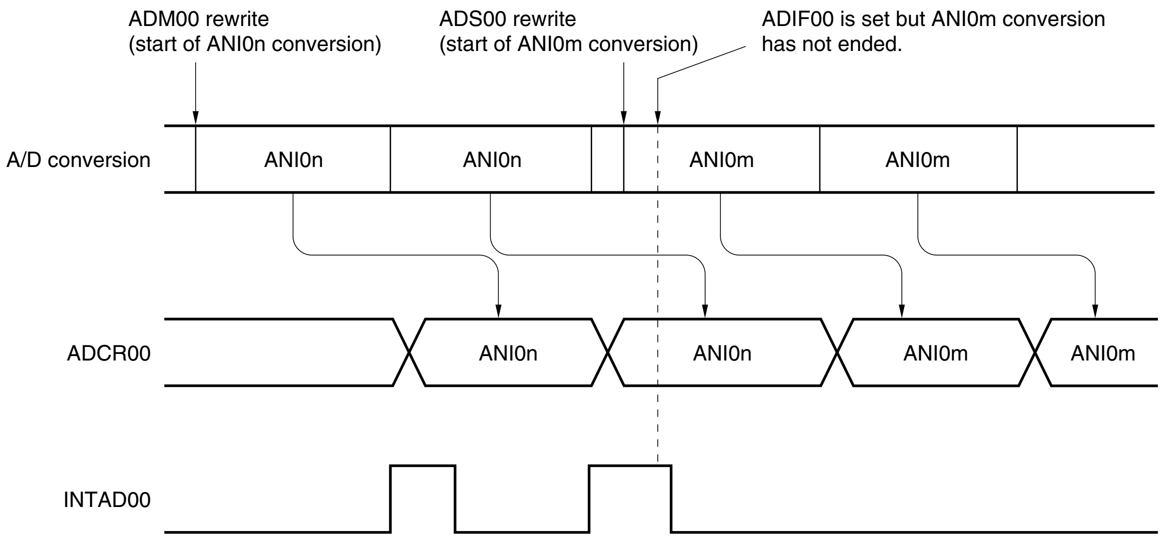
(7) Interrupt request flag (ADIF00)

The interrupt request flag (ADIF00) is not cleared even if analog input channel specification register 00 (ADS00) is changed.

Therefore, if an analog input pin is changed during A/D conversion, the A/D conversion result and conversion end interrupt request flag for the pre-change analog input may be set just before ADS00 is rewritten. Caution is therefore required since, at this time, when ADIF00 is read immediately just after ADS00 is rewritten, ADIF00 is set despite the fact that A/D conversion for the post-change analog input has not ended.

When A/D conversion is stopped and then resumed, clear ADIF00 before the A/D conversion operation is resumed.

Figure 11-12. A/D Conversion End Interrupt Request Generation Timing


ADM00 rewrite
(start of ANI0n conversion)
ADS00 rewrite
(start of ANI0m conversion)
ADIF00 is set but ANI0m conversion has not ended.
A/D conversion
ANI0n
ANI0n
ANI0m
ANI0m
ADCR00
ANI0n
ANI0n
ANI0m
ANI0m
INTAD00

Remarks 1. n = 0, 1, ..., 7

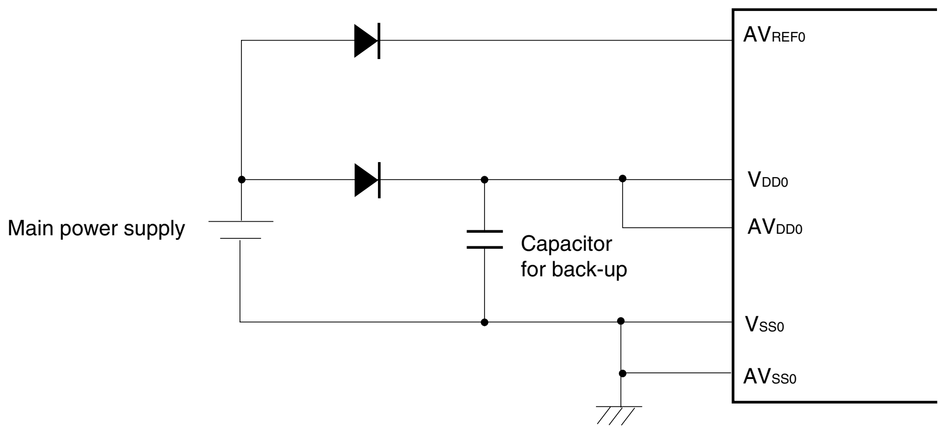
2. m = 0, 1, ..., 7

(8) AV$_{DD0}$ pin

The AV$_{DD0}$ pin is the analog circuit power supply pin. It supplies power to the input circuits of the ANI00 to ANI70 pins.

Therefore, be sure to apply the same voltage as V$_{DD0}$ to this pin even when the application circuit is designed so as to switch its power supply to a backup battery.

Figure 11-13. AV$_{DD0}$ Pin Connection



(9) Conversion results just after A/D conversion start

The first A/D conversion value just after A/D conversion operations start may not satisfy the ratings. Poll the A/D conversion end interrupt request (INTAD00) and take measures such as discarding the first conversion result.

(10) A/D conversion result register 00 (ADCR00) read operation

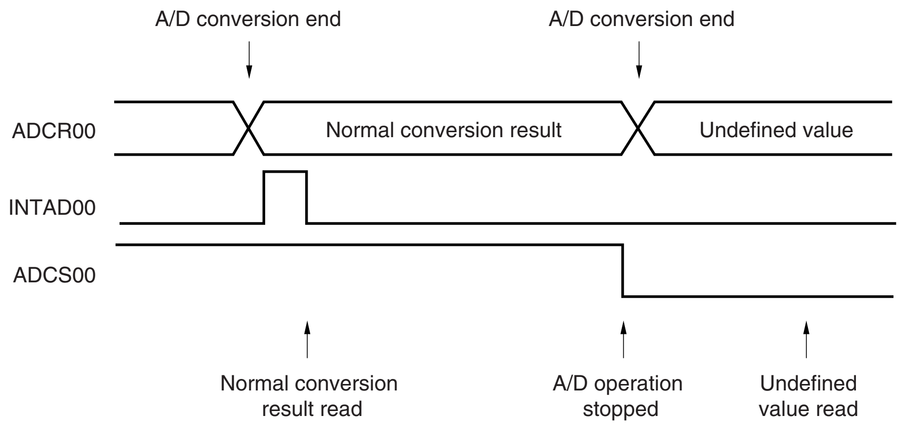
When A/D converter mode register 00 (ADM00) and analog input channel specification register 00 (ADS00) are written, the contents of ADCR00 may become undefined. Read the conversion result following conversion completion before writing to ADM00 and ADS00. Using a timing other than the above may cause an incorrect conversion result to be read.

★ **(11) Timing that makes the A/D conversion result undefined**

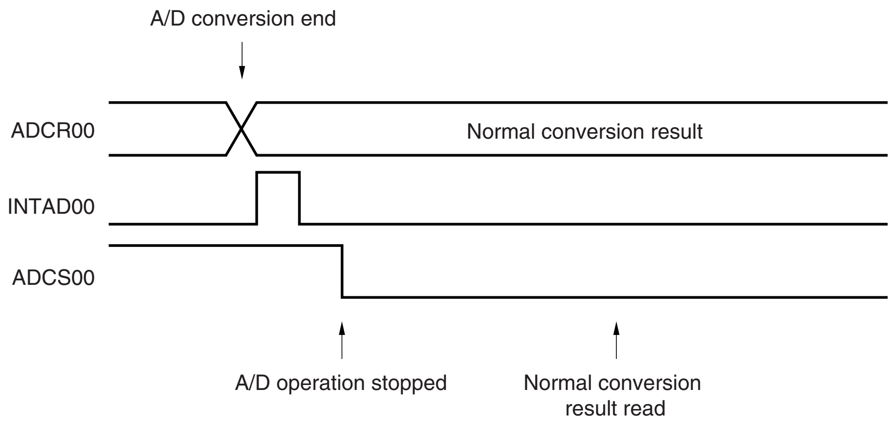
If the timing of the end of A/D conversion and the timing of the stop of operation of the A/C converter conflict, the A/D conversion value may be undefined. Because of this, be sure to read the A/D conversion result while the A/D converter is in operation. Furthermore, when reading an A/D conversion result after the A/D converter operation has stopped, be sure to have done so by the time the next conversion result is complete.

The conversion result read timing is shown in Figures 11-14 and 11-15 below.

★ **Figure 11-14. Conversion Result Read Timing (When Conversion Result Is Undefined)**



★ **Figure 11-15. Conversion Result Read Timing (When Conversion Result Is Normal)**



★ **(12) Cautions on board design**

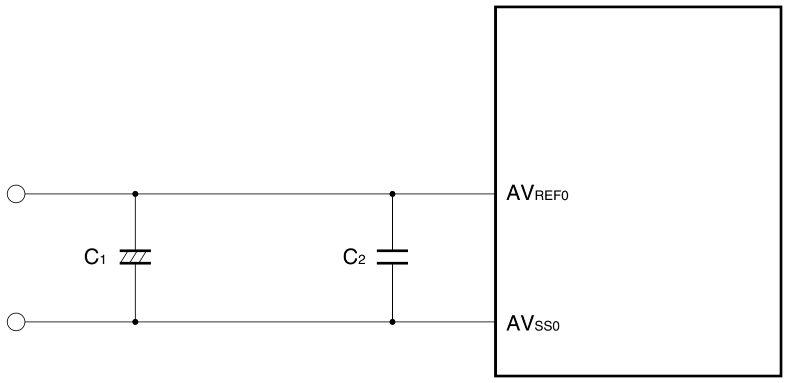
In order to avoid negative effects from digital circuit noise on the board, analog circuits must be placed as far away as possible from digital circuits. It is particularly important to prevent analog and digital signal lines from crossing or coming into close proximity, as A/D conversion characteristics are vulnerable to degradation from the induction of noise or other such factors.

Connect AV_{SS0} and V_{SS0} to a single stable location on the board.

★ **(13) AV_{REF0} pin**

Connect a capacitor to the AV_{REF0} pin to minimize conversion error caused by noise. Note also that the voltage applied to the AV_{REF0} pin following the resumption of A/D conversion after it was stopped may be unstable, causing a degradation in the accuracy of the A/D conversion. Be sure to connect a capacitor AV_{REF0} pin in this case also. An example of capacitor connection is shown in Figure 11-16 below.

★ **Figure 11-16. Example of Capacitor Connection to AV_{REF0} Pin**



Remark C1: 4.7 μF to 10 μF (reference value)
C2: 0.01 μF to 0.1 μF (reference value)
Connect C2 as close to the pin as possible.

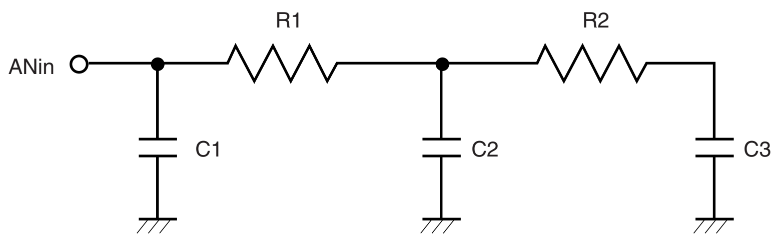
★ **(14) Internal equivalence circuit and allowable signal source impedance of ANI00 to ANI70**

In order to complete sampling within the sampling time and obtain a high enough A/D conversion accuracy, it is necessary to sufficiently reduce the impedance of the sensor and other signal sources. Figure 11-17 shows the internal equivalence circuit of the ANI00 to ANI70 pins in the microcontroller.

If the impedance of the signal source is high, it can be made to seem smaller by connecting a large capacitance to the ANI00 to ANI70 pins. A circuit example is shown in Figure 11-18. In this case, because a low pass filter is configured in the circuit, impedance will no longer be able to follow analog signals with large differential coefficients.

When converting high-speed analog signals or performing conversion in scan mode, be sure to insert a low-impedance buffer.

★ **Figure 11-17. Internal Equivalence Circuit of ANI00 to ANI70 Pins**



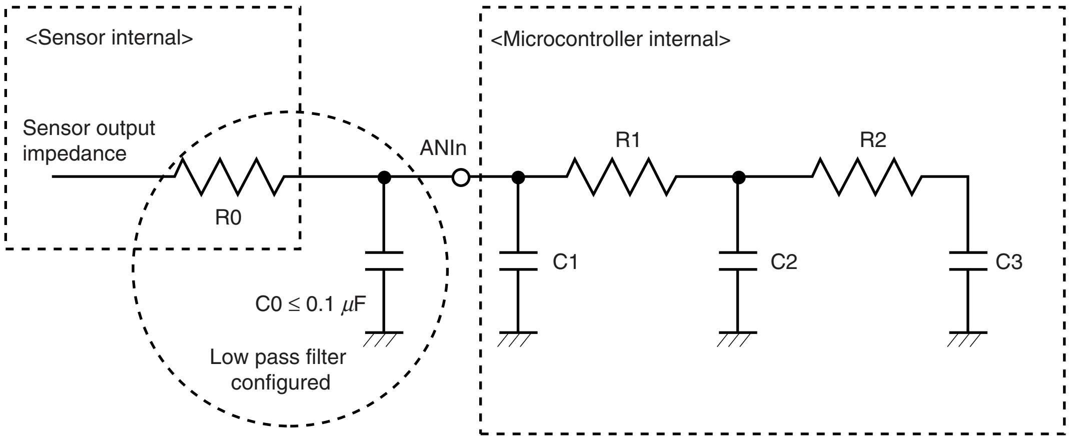
Remark n = 00 to 70

★ **Table 11-3. Resistance and Capacitance Values for Equivalence Circuits (Reference Values)**

AV_{REF0}	R1	R2	C1	C2	C3
1.8 V	75 kΩ	30 kΩ	8 pF	4 pF	2 pF
2.7 V	12 kΩ	8 kΩ	8 pF	3 pF	2 pF
4.5 V	4 kΩ	2.7 kΩ	8 pF	1.4 pF	2 pF

Caution The resistance and capacitance values in Table 11-3 cannot be guaranteed.

★ **Figure 11-18. Example of Circuit When Signal Source Impedance Is High**



Remark n = 00 to 70

CHAPTER 12 J1850 (Class 2)

The μPD780833Y Subseries includes a bus controller conforming to J1850 (Class 2).

12.1 Protocol Overview

The protocol of J1850 (Class 2) is the variable pulse width modulation (VPW) method. The protocol conforms to the rules stated below.

1) The potential level is changed at each bit.
2) A logical value, 1 or 0, is determined by the potential level and pulse width.
3) A logical value of 0 takes precedence.

A message begins with an SOF (start of frame) symbol and contains one to 11 bytes of data except in block mode, in which the data length is not limited. Data is transmitted serially in descending order of bits, that is, bit 7 is transmitted first, and bit 0 is transmitted last. The last bit, bit 0, is followed by the cyclic redundancy check (CRC) field. A message is concluded with and EOF (end of frame). A break signal is a single pulse.

A message example is shown below.

Figure 12-1. Example of Message

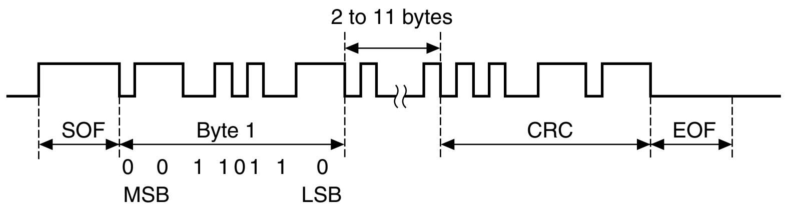
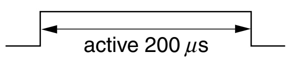
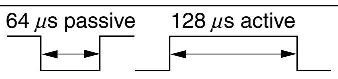
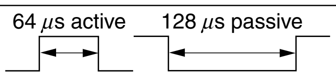
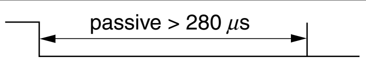
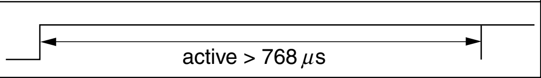


Table 12-1. Examples of Symbols

Symbol Name	Symbol (Example of Symbols in Normal Transmission Mode)
SOF (Start Of Frame)	
Logical state 0	
Logical state 1	
EOF (End Of Frame)	
Break signal	

12.2 Class 2 Internal Register Configuration

Class 2 includes the following nine registers.

- Class 2 control register 1 (C2CT1)
- Class 2 control register 2 (C2CT2)
- Class 2 status register 1 (C2ST1)
- Class 2 status register 2 (C2ST2)
- Class 2 transmit FIFO register (C2TXFIFO)
- Class 2 receive FIFO register (C2RXFIFO)
- Class 2 rise time propagation delay correction register (C2PDR)
- Class 2 rise time propagation delay correction register (C2PDF)
- Class 2 clock selection register (C2CLK)

(1) Class 2 control register 1 (C2CT1)

This is one of the two registers that control communication using the Class 2 protocol. C2CT1 is set using a 1-bit or 8-bit memory manipulation instruction. The C2BRK, C2ABTD, C2TXS, C2ECL, and C2ABRD bits are reset automatically based on the internal clock. The auto reset bit is reset automatically (0) approximately 1.2 μs (system clock @ 4.19 MHz operation[Note]) after the bit manipulation (setting (1), then processing of each bit). $\overline{\text{RESET}}$ input sets C2CT1 to 00H.

Note 4.19 MHz = 4.19304 MHz (the clock is written as 4.19 MHz, but actually operates at 4.19304 MHz).

Caution Do not write to C2CT1 more than once within approximately 1.2 μs (system clock @ 4.19 MHz operation). The following malfunction may occur when bits 0, 2 to 4, and 7 of C2CT1 (C2BRK, C2ABTD, C2TXS, C2ECL and C2ABRD) are automatically reset (0).

- **If C2CT1 has been written to twice or more within 0.48 μs (system clock @ 4.19 MHz operation), unexecuted data is overwritten and the command will become invalid.**
- **If C2BRK, C2ABTD, C2TXS, C2ECL, and C2ABRD are written to within 1.2 μs (system clock @ 4.19 MHz operation), during the asynchronous reset period, writing may become invalid.**

Figure 12-2. Format of Class 2 Control Register 1 (C2CT1) (1/2)

Address: FFC0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
C2CT1	C2ABRD	C2IE	C2NCRC	C2ECL	C2TXS	C2ABTD	C2TAR	C2BRK

C2ABRD	Receive data reset
0	The receive data is not reset.
1	The receive data is reset. • The address pointer for receive FIFO access control is reset. • The receive flag (C2RXF2 and C2RXF1) to 11B is reset If the C2ABRD is set (1) during reception: • Reception will not be resumed until the next SOF symbol • No reception error will occur Do not set the C2ABRD bit to 1 when transmission is requested (when C2TXRQ = 1). After C2ABRD is set (1), it is automatically reset (0) after approximately 1.2 μsNote.

C2IE	Interrupt request signal enable/disable
0	Reception error, reception completion, and sleep enable interrupts are disabled. The wake-up interrupt is not disabled. The interrupt request flag (C2INT1 and C2INT2) is valid even when C2IE = 0.
1	Interrupt request signals are enabled.

C2NCRC	CRC usage setting
0	CRC used.
1	CRC not used.
Switch the C2NCRC bit during the following period: • Between a hardware reset and the C2IDL bit being set • When C2IDL = 1	

C2ECL	Error flag clearance
0	The error flags are not cleared.
1	The error flags (C2SHTE, C2OVE, C2CRCE, C2BYTE, C2SYME, and C2BRKE) are cleared. After C2ECL is set (1), it is automatically reset (0) after approximately 1.2 μsNote.

Note System clock @ 4.19 MHz operation

Remark

C2RXF2, C2RXF1:	Bits 7 and 6 of class 2 status register 1 (C2ST1)
C2INT2, C2INT1:	Bits 5 and 4 of class 2 status register 1 (C2ST1)
C2IDL:	Bit 7 of class 2 status register 2 (C2ST2)
C2SHTE, C2OVE, C2CRCE, C2BYTE, C2SYME, C2BRKE:	Bits 5 to 0 of class 2 status register 2 (C2ST2)

Figure 12-2. Format of Class 2 Control Register 1 (C2CT1) (2/2)

C2TXS	Transmission start
0	Transmission is not started
1	Transmission is started. The C2TxRQ bit is set. Transmission is not performed in the following cases even if C2TXS is set (1): • The transmit FIFO is empty • This bit is set (1) simultaneously with C2ABTD After C2TXS is set (1), it is automatically reset (0) after approximately 1.2 μs[Note].

C2ABTD	Transmit data reset
0	The transmit data is not reset.
1	The transmit data is reset. • The C2XFUL, C2TXRQ, and C2TXON bits are reset (0) • The address pointer for TxRAM access control is reset The C2TXS bit is not set (1) if specified simultaneously with C2TXS. If transmission is under way when the C2ABTD bit is set (1), transmission is terminated immediately. After C2ABTD is set (1), it is automatically reset (0) after approximately 1.2 μs[Note].

C2TAR	Automatic retransmission enable/disable
0	Automatic retransmission is enabled. Retransmission is repeated until transmission is completed normally. However, in the following cases, retransmission is not executed. • A short error is detected • A break signal is received • The C2ABTD bit is set (1)
1	Automatic retransmission is disabled. If a break signal is received, C2TAR is automatically reset (0).
Switch the C2TAR bit during the following period: • Between a hardware reset and C2IDL being set (1) • When C2IDL = 1	

C2BRK	Break signal transmission
0	A break signal is not transmitted.
1	A break signal is transmitted. • A break signal is transmitted immediately regardless of the transmission or reception state • A break signal in normal mode is transmitted even in the ×4 mode • The C2X4 bit is reset (0) when the C2BRK bit is set (1) C2BRK is automatically reset (0) immediately after it is set (1) (break signal transmitted). **[Reference]** When a break signal is received Class 2 performs the following operations. • C2X4 is reset (0) • C2TAR is reset (0) • The TxRAM access control address pointer is reset. • C2XFUL, C2TXRQ, and C2TXON are reset (0).

Note System clock @ 4.19 MHz operation

Remark	C2XFUL, C2TXRQ, C2TXON:	Bits 2 to 0 of class 2 status register 1 (C2ST1)
	C2IDL:	Bit 7 of class 2 status register 2 (C2ST2)
	C2X4:	Bit 2 of class 2 control register 2 (C2ST2)

(2) Class 2 control register 2 (C2CT2)

This is one of the two registers that control communication according to the Class 2 protocol. C2CT2 is set using a 1-bit or 8-bit memory manipulation instruction. Bits 3 and 4 (C2SLP, C2WAK) of C2CT2 are reset automatically. The auto reset bit is automatically reset (0) approximately 1.2 μs (system clock @ 4.19 MHz operation) after a bit manipulation (setting (1), then processing of each bit).

$\overline{\text{RESET}}$ input sets C2CT2 to 00H.

Caution Do not write to more than once within approximately 1.2 μs (system clock @ 4.19 MHz operation). The following malfunction may occur when bits 0 and 3 (C2WAK, C2SLP) of C2CT2 are reset automatically (0).

- **If C2CT2 has been written to more than once within 0.48 μs (system clock @ 4.19 MHz operation), unexecuted data is overwritten and the command will become invalid.**
- **If C2WAK and C2SLP are written to within 1.2 μs (system clock @ 4.19 MHz operation), writing may become invalid during the asynchronous reset period.**

Figure 12-3. Format of Class 2 Control Register 2 (C2CT2)

Address: FFC1H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
C2CT2	C2WAK	0	0	C2BLK	C2SLP	C2X4	0	0

C2WAK	Wake-up flag
0	Wake-up is not enabled.
1	Wake-up is enabled. In Class 2, the input of the C2RX pin is not nade an interrupt request signal directly. It is used when changing to the wake-up state from the sleep state without waiting for a bus signal. After C2WAK is set (1), it is automatically reset (0) after approximately 1.2 μs[Note].

C2BLK	Block mode set/release
0	Block mode is released. • It is impossible to read receive data before an EOF symbol
1	Block mode is set. • It is possible to read received data sequentially in byte units, even before an EOF symbol

C2SLP	Sleep request
0	No sleep request.
1	A sleep status is requested. • The C2SLPRQ bit is set • A sleep enable interrupt request is generated when the bus is idle (C2IDL = 1) • After a sleep enable interrupt request is issued, a rising edge input signal to the C2RX pin is taken directly as an interrupt request signal After C2SLP is set (1), it is automatically reset (0) after approximately 1.2 μs[Note].

C2X4	×4 mode set/release
0	×4 mode is released.
1	×4 mode is set. • If a break signal is received, C2X4 is automatically reset (0) • At the point when a break signal is transmitted, C2X4 is automatically reset (0)

Note System clock @ 4.19 MHz operation

Remark C2SLPRQ: Bit 6 of class 2 status register 2 (C2ST2)
C2BRK: Bit 0 of class 2 control register 1 (C2CT1)

(3) Class 2 status register 1 (C2ST1)

This is one of the two registers that indicate the status of Class 2. It can be used to monitor the transmission/reception status. It also indicates the interrupt signal type.

C2ST1 is read using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets C2ST1 to C0H.

Figure 12-4. Format of Class 2 Status Register 1 (C2ST1)

Address: FFC2H After reset: C0H R

Symbol	7	6	5	4	3	2	1	0
C2ST1	C2RXF2	C2RXF1	C2INT2	C2INT1	C2XNOW	C2XFUL	C2TXRQ	C2TXON

C2RXF2	C2RXF1	Receive status flag
0	0	Receive data is being prepared in C2RXFIFO (period during which reading is disabled).
0	1	C2RXFIFO contains valid data.
1	0	The data in C2RXFIFO is the last data in the message frame.
1	1	C2RXFIFO contains no valid data.

C2INT2	C2INT1	Interrupt request flag
0	0	Wake-up request.
0	1	Completion of reception.
1	0	Sleep enable.
1	1	Reception error.

C2XNOW	Timing flag for disabling writing to the transmit FIFO
0	Writing to the transmit FIFO is enabled.
1	Writing to the transmit FIFO is disabled.

C2XFUL	Transmit FIFO full flag
0	The transmit FIFO is not full. Data can be written to C2TXFIFO.
1	The transmit FIFO is full. Any attempt to write data to C2TXFIFO is ignored.

C2TXRQ	C2TXON	Transmission status flag
0	0	No transmission request has been issued/transmission has been completed.
0	1	Undefined.
1	0	Transmission standby.
1	1	Data is being transmitted from the C2TX pin.

Cautions 1. If the C2RXF2 and C2RXF1 bits are set to 00B, access to the receive FIFO is prohibited. Any attempt to read from the FIFO in this status results in the previous data being read again, because the next data is not ready to be read. The period during which reading from the FIFO is prohibited (while the C2RXF2 and C2RXF1 bits are set to 00B) consists of about 3.8 μs (system clock @ 4.19 MHz operation) (maximum).

2. If the C2XNOW bit is set to 1, writing to the transmit FIFO is prohibited. If an attempt is made to write to the FIFO in this status, the previous write data is overwritten and becomes invalid. The period during which writing to the FIFO is prohibited (while the C2XNOW bit is set to 1) consists of about 3.8 μs (system clock @ 4.19 MHz operation) (maximum).

★ **3. The contents of the C2XNOW bit may not be reflected immediately after writing to the transmit FIFO when the system clock is operating at 8 MHz. It is necessary to wait at least 2.5 internal clocks to confirm the contents are properly reflected.**

★ **4. The contents of the C2XFUL bit may not be reflected immediately after writing to the transmit FIFO. It is necessary to wait at least 12 internal clocks to confirm the contents are reflected properly.**

Remark		
	C2TXFIFO:	Class 2 transmit FIFO register
	C2RXFIFO:	Class 2 receive FIFO register

(4) Class 2 status register 2 (C2ST2)

This is one of the two registers that indicate the status of Class 2. It can be used to monitor the error, bus, and sleep request status.

C2ST2 is read using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets C2ST2 to 00H.

Figure 12-5. Format of Class 2 Status Register 2 (C2ST2) (1/2)

Address: FFC3H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
C2ST2	C2IDL	C2SLPRQ	C2SHTE	C2OVE	C2CRCE	C2BYTE	C2SYME	C2BRKE

C2IDL	Bus idle flag
0	The bus is not idle (communication is in progress).
1	The bus is idle.

C2SLPRQ	Sleep request flag
0	A sleep request is not being issued (one has been acknowledged).
1	A sleep request has been issued (the system is waiting for the bus to become idle).

C2SHTE	Short error flag
0	No short errors.
1	A short error occurred (a bus GND short circuit was recognized during transmission).

C2OVE	Overrun error flag
0	No overrun errors.
1	An overrun error occurred (data was received which exceeded the receive FIFO capacity).

C2CRCE	CRC error flag
0	No CRC errors.
1	The CRC check result is abnormal.

C2BYTE	Byte error flag
0	No byte errors.
1	A byte error occurred (the received data was not in 1-byte units when the EOF was recognized).

C2SYME	Symbol error flag
0	No symbol errors.
1	A symbol error occurred (the received symbol was not of the specified pulse width).

Figure 12-5. Format of Class 2 Status Register 2 (C2ST2) (2/2)

C2BRKE	Break error flag
0	No break errors.
1	A break error occurred (a break signal was received).

Caution The error flags (C2SHTE, C2OVE, C2CRCE, C2BYTE, C2SYME, C2BRKE) are cleared by setting bit 4 (C2ECL) (1) of class 2 control register 1 (C2CT1). The next reception is not performed until the error flags have been cleared.

(5) Class 2 transmit FIFO register (C2TXFIFO)

The class 2 transmit FIFO register is a “window” through which transmit data is written to the transmit FIFO. It can be write-accessed only by using an 8-bit memory manipulation instruction. $\overline{\text{RESET}}$ input sets C2TXFIFO to 00H. Transmit data is written to the transmit FIFO sequentially, according to the address pointer incorporated in Class 2. While writing to the transmit FIFO, a write disable timing is generated with a maximum length of approximately 3.8 μs (system clock @ 4.19 MHz operation). During the write disable time, bit 3 (C2XNOW) of class 2 status register 1 (C2ST1) becomes 1. If the C2XNOW bit is set to 1, any attempt to write data results in the previous data being overwritten, which makes it invalid.

Eleven bytes of memory are assigned to the transmit FIFO. When the transmit FIFO contains 11 bytes of data to be transmitted, bit 2 (C2XFUL) of C2ST1 is set (1). Any attempt to write more than 11 bytes of data to the transmit FIFO is ignored, and the existing data in the FIFO is retained. The transmit FIFO is automatically reset when:

- A break signal is detected (C2BRKE = 1)
- A short error is detected (C2SHTE = 1)
- Bit 2 (C2ABTD) of class 2 control register 1 (C2CT1) is set to 1

Caution C2TXFIFO has the same address assigned to it as the class 2 receive FIFO register (C2RXFIFO). If C2TXFIFO is read, therefore, the receive FIFO’s value is read.

Remark C2BRKE: Bit 0 of class 2 status register 2 (C2ST2)
C2SHTE: Bit 5 of class 2 status register 2 (C2ST2)

(6) Class 2 receive FIFO register (C2RXFIFO)

The class 2 receive FIFO register is a “window” through which receive data is read from the receive FIFO. It can be read-accessed only by using an 8-bit memory manipulation instruction. $\overline{\text{RESET}}$ input sets C2RXFIFO to 00H. Receive data is read from the receive FIFO sequentially, according to the address pointer incorporated in Class 2. While reading the receive FIFO, a read disable timing is generated with a maximum length of approximately 3.8 μs (system clock @ 4.19 MHz operation). During the read disable time, bits 6 and 7 (C2RXF1 and C2RXF2) of class 2 status register 1 (C2ST1) become 00B. If the C2RXF2 and C2RXF1 bits are set to 00B, any attempt to read data results in the previous data being read, because the FIFO is not ready to be read.

Twenty-one bytes of memory are assigned to the receive FIFO. When the receive FIFO contains 20 bytes of receive data to be read, and an attempt is made to write one more byte to the receive FIFO, an overrun error occurs (in this case, one byte is overwritten). When this error occurs, reception is not performed.

The receive FIFO is automatically reset when:

- Bit 2 (C2ABTD) of class 2 control register 1 (C2CT1) is set to 1

If the macro detects an error, it discards all the receive data received subsequent to the SOF symbol. Data received before the error occurred is not discarded.

Caution C2RXFIFO has the same address assigned to it as the class 2 transmit FIFO register (C2TXFIFO). If C2RXFIFO is read, therefore, the transmit FIFO’s value is read.

(7) Class 2 rise time propagation delay correction register (C2PDR)

This register specifies the rise time propagation delay for a class 2 transceiver (analog) block for which compensation for distortion on the low-level side of the transmission pulse width, due to delay, is to be applied. Compensation is performed using an internal circuit. After hardware reset, during the interval until bit 7 (C2IDL) of class 2 status register 2 (C2ST2) becomes 1, writing is enabled.

Writing to this register is prohibited after a hardware reset, until the C2IDL bit is set. C2PDR is set using an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets C2PDR to 00H.

Figure 12-6. Format of Class 2 Rise Time Propagation Delay Correction Register (C2PDR)

Address: FFC5H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
C2PDR	PDRX2	PDRX1	PDRX0	PDR4	PDR3	PDR2	PDR1	PDR0

PDRX2 to PDRX0	×4 mode rise time propagation delay (System clock @ 4.19 MHz operation)
	Set the ×4 mode rise time propagation delay of the Class 2 transceiver. PDRX setting value: t_{PDRX}/0.48 [μs]/2 Class 2 corrects the transmit low pulse width by shortening it by t_{PDRX} (= PDRX setting × 0.48 [μs] × 2). The maximum correction value is about 3.8 μs (= 4 × 0.48 [μs] × 2). PDRX must not be set to a value that satisfies 16 [μs] – (2 × PDRX + 20) × 0.48 [μs] ≤ 2.

PDR4 to PDR0	Normal mode rise time propagation delay (System clock @ 4.19 MHz operation)
	Set the normal mode rise time propagation delay of the Class 2 transceiver. PDR setting value: t_{PDR}/0.48 [μs]/2 Class 2 corrects the transmit low pulse width by shortening it by t_{PDR} (= PDR setting × 0.48 [μs] × 2). The maximum correction value is about 29.8 μs (= 31 × 0.48 [μs] × 2).

Caution Do not set PDRX to a value that satisfies 16 [μs] – (2 × PDRX + 20) × 0.48 [μs] ≤ 2.
The reason is that when transmission is performed in ×4 mode, the minimum pulse width is 16 μs, whereas twenty internal clock pulses are required after a signal is input to the C2RX pin until the internal timer is reset.
The required correction is made in advance. The corrected standard transmit time is as follows:
16 [μs] – (2 × PDRX + 20) × 0.48 [μs]
Normal transmission is disabled if this standard time is two or less.
For example, when the system clock is operating at 4.19 MHz and PDRX is set to 5:
16 [μs] – ((2 × 5 + 20) × 0.48) = 1.6 [μs] ≤ 2
This results in a transmission malfunction.

(8) Class 2 fall time propagation delay correction register (C2PDF)

This register specifies the fall time propagation delay for a class 2 transceiver (analog) block for which compensation for distortion on the high-level side or the transmission pulse width, due to delay, is to be applied. Compensation is performed using an internal circuit. After hardware reset, during the interval until bit 7 (C2IDL) of class 2 status register 2 (C2ST2) becomes 1, writing is enabled.

Writing to this register is prohibited after a hardware reset, until the C2IDL bit is set.

C2PDF is set using an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets C2PDF to 00H.

Figure 12-7. Format of Class 2 Fall Time Propagation Delay Correction Register (C2PDF)

Address: FFC6H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
C2PDF	PDFX2	PDFX1	PDFX0	PDF4	PDF3	PDF2	PDF1	PDF0

PDFX2 to PDFX0	×4 mode fall time propagation delay (System clock @ 4.19 MHz operation)
	Set the ×4 mode fall time propagation delay of the Class 2 transceiver. PDFX setting value: t_{PDFX}/0.48 [μs]/2 Class 2 corrects the transmit high pulse width by shortening it by t_{PDFX} (= PDFX setting × 0.48 [μs] × 2). The maximum correction value is about 3.8 μs (= 4 × 0.48 [μs] × 2). PDFX may not be set to a value that satisfies 16 [μs] – (2 × PDFX + 20) × 0.48 [μs] ≤ 2.

PDF4 to PDF0	Normal mode fall time propagation delay (System clock @ 4.19 MHz operation)
	Set the normal mode fall time propagation delay of the Class 2 transceiver. PDF setting value: t_{PDF}/0.48 [μs]/2 Class 2 corrects the transmit high pulse width by shortening it by t_{PDF} (= PDF setting × 0.48 [μs] × 2). The maximum correction value is about 29.8 μs (= 31 × 0.48 [μs] × 2).

Caution Do not set PDFX to a value that satisfies 16 [μs] – (2 × PDFX + 20) × 0.48 [μs] ≤ 2.
The reason is that when transmission is being performed in ×4 mode, the minimum pulse width is 16 μs, whereas twenty internal clock pulses are required after a signal is input to the C2RX pin until the internal timer is reset.
The required correction is made in advance. The corrected standard transmit time is as follows:
16 [μs] – (2 × PDFX + 20) × 0.48 [μs]
Normal transmission is disabled if this standard time is two or less.
For example, when the system clock is operating at 4.19 MHz and if PDFX is 5:
16 [μs] – ((2 × 5 + 20) × 0.48) = 1.6 [μs] ≤ 2
This results in a transmission malfunction.

(9) Class 2 clock selection register (C2CLK)

This register enables/disables class 2, and selects the class 2 internal clock and decoder. After hardware reset, during the interval until bit 7 (C2IDL) of class 2 status register 2 (C2ST2) becomes 1, overwrite operations are enabled.

No other interval may be used for overwriting. C2CLK is set using a 1-bit or 8-bit memory manipulation instruction. $\overline{\text{RESET}}$ input sets C2CLK to 00H.

Figure 12-8. Format of Class 2 Clock Selection Register (C2CLK)

Address: FFC7H After reset: 00H R/W

Symbol	$\boxed{7}$	6	5	4	3	2	1	0
C2CLK	C2EN	0	C2SC1	C2SC0	0	C2DEC2	C2DEC1	C2DEC0

C2EN	Class 2 enable
0	Everything other than the C2PDR and C2PDF registers are reset. The internal clock is stopped.
1	Class 2 is enabled. The C2DEC2 to C2DEC0, C2CS1, C2CS0, C2PDR, and C2PDF registers cannot be overwritten.

C2SC1	C2SC0	Class 2 internal clock selection			
		Selected clock	fx = 4.194304 MHz	fx = 4.19 MHz	fx = 4.0 MHz
0	0	$f_X/2^6$	Setting prohibited (65.536 kHz)	Setting prohibited (65.46875 kHz)	Setting prohibited (62.5 kHz)
0	1	$f_X/2$	2.097 MHz[Note 1]	2.095 MHz[Note 2]	2.0 MHz[Note 3]
1	0	$f_X/2^2$	Setting prohibited (1.049 MHz)	Setting prohibited (1.047 MHz)	Setting prohibited (1.0 MHz)
1	1	$f_X/3$	Setting prohibited (1.398 MHz)	Setting prohibited (1.397 MHz)	Setting prohibited (1.3 MHz)

C2DEC2	C2DEC1	C2DEC0	Decoder selection
0	0	0	Decoder 0 (for 2.0 MHz internal clock)
0	0	1	Decoder 1 (for 2.095 MHz internal clock)
0	1	0	Decoder 2 (for 2.097 MHz internal clock)
0	1	1	Decoder 3 (for 2.1 MHz internal clock)
1	0	0	Decoder 4 (for 2.1168 MHz internal clock)
1	0	1	Decoder 5 (for 2.25 MHz internal clock)
1	1	0	Decoder 6 (for 2.455 MHz internal clock)
1	1	1	Decoder 7 (for 2.5 MHz internal clock)

Notes 1. Decoder 2 should be selected (C2DEC2 to C2DEC0 = 010B).
2. Decoder 1 should be selected (C2DEC2 to C2DEC0 = 001B).
3. Decoder 0 should be selected (C2DEC2 to C2DEC0 = 000B).

Caution Be sure to set bits 3 and 6 to 0.

★ **Remarks 1.** fx: System clock oscillator frequency

2. Examples of the system clock frequency and C2DEC0 to C2DEC2, C2SC1, and C2SC0 settings are shown below.

C2DEC2	C2DEC1	C2DEC0	C2SC1	C2SC0	C2SC1	C2SC0	C2SC1	C2SC0	C2SC1	C2SC0
			0	0	0	1	1	0	1	1
0	0	0	Setting prohibited		4 MHz		8 MHz		6 MHz	
0	0	1			4.19 MHz		8.38 MHz		6.285 MHz	
0	1	0			4.194 MHz		8.388 MHz		6.29 MHz	
0	1	1			4.2 MHz		8.4 MHz		6.3 MHz	
1	0	0			4.2336 MHz		Setting prohibited		6.3504 MHz	
1	0	1			4.5 MHz				6.75 MHz	
1	1	0			4.91 MHz				7.365 MHz	
1	1	1			5 MHz				Setting prohibited	

12.3 Operation Timing

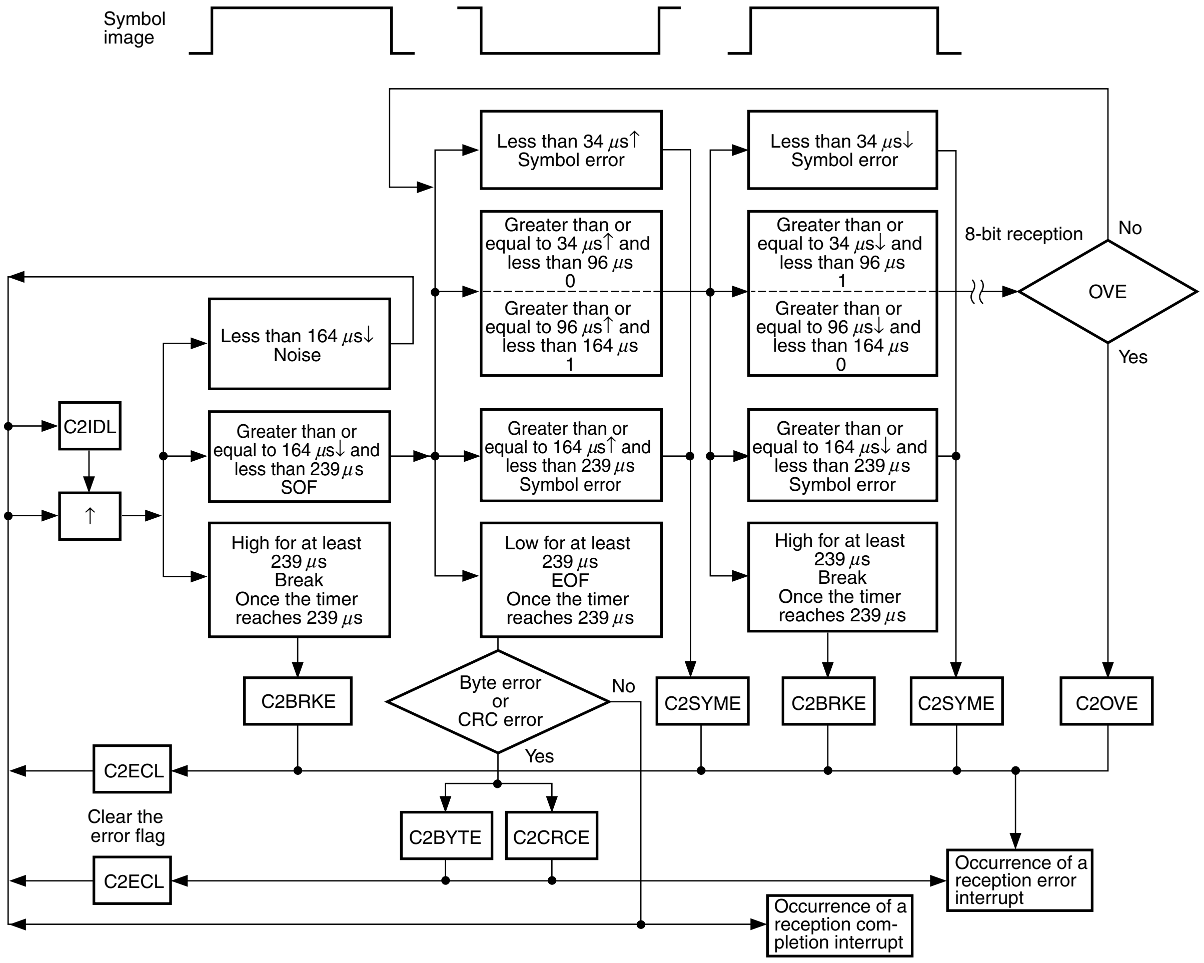
12.3.1 Reception

Class 2 converts the pulse width and potential of a received waveform to a bit stream. Reception begins once an SOF (start of frame) symbol is detected, and ends with the detection of an EOF (end of frame) symbol. Up to 21 bytes of data can be stored, regardless of the number of message frames. If a reception error occurs, or when reception ends, an interrupt request signal (INTC2) is issued. A reception error causes reception to stop. Reception is not resumed until the error flag is reset (bit 4 (C2ECL) of class 2 control register 1 (C2CT1) is set (1)).

Class 2 starts reception at the rising edge of the CDRX pin. The symbols that must be judged before the rising edge, such as a break signal, EOF, and IDLE, are judged at the respective standard reception timing.

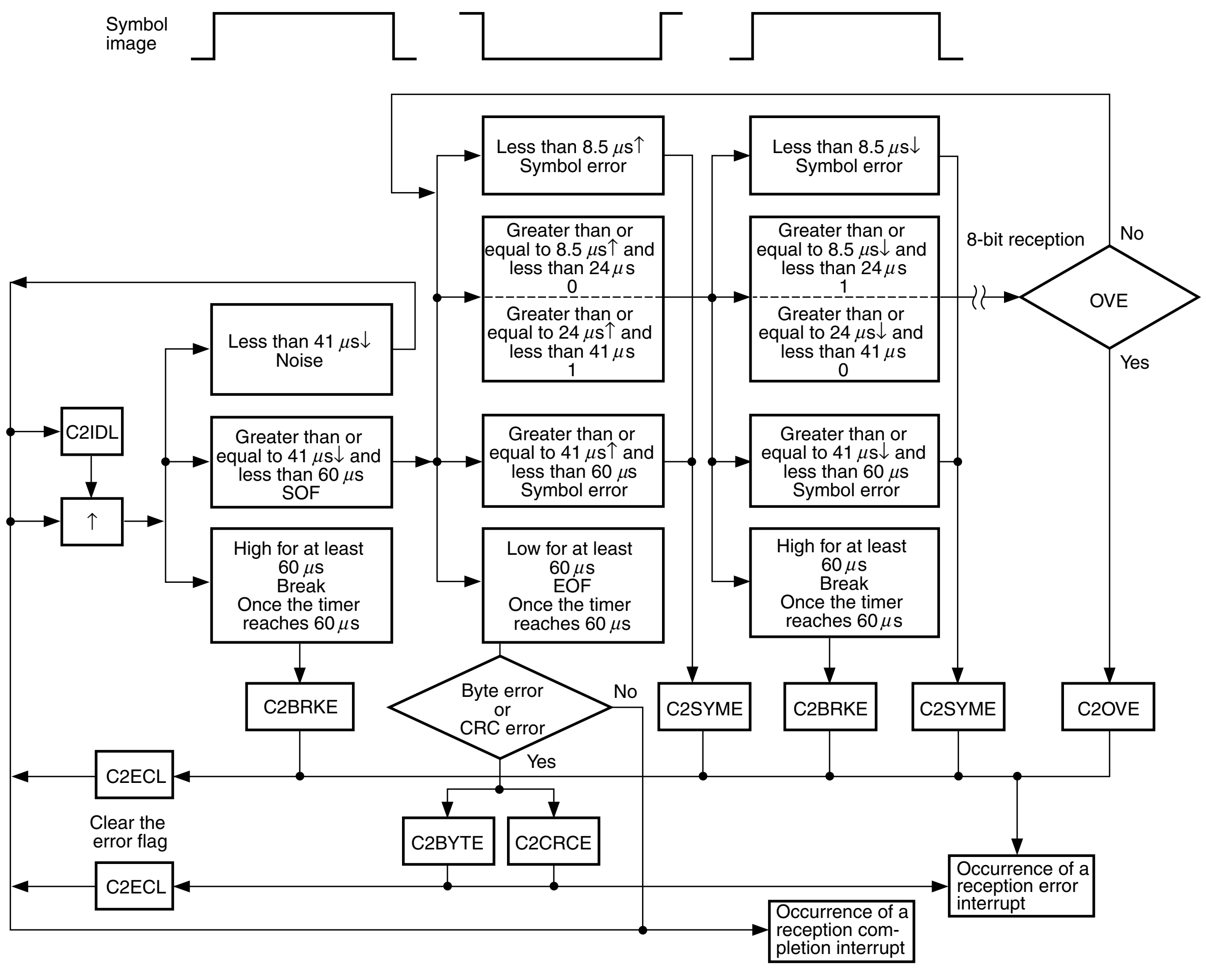
Data being received cannot be read until the completion of reception. When bit 4 (C2BLK) of class 2 control register 2 (C2CT2) is set (1), received data is read sequentially, in byte units.

Figure 12-9. Reception Status Transition Diagram (Normal Mode)


Symbol image
C2IDL
↑
Less than 164 μs↓ Noise
Greater than or equal to 164 μs↓ and less than 239 μs SOF
High for at least 239 μs Break Once the timer reaches 239 μs
Less than 34 μs↑ Symbol error
Greater than or equal to 34 μs↑ and less than 96 μs 0
Greater than or equal to 96 μs↑ and less than 164 μs 1
Greater than or equal to 164 μs↑ and less than 239 μs Symbol error
Low for at least 239 μs EOF Once the timer reaches 239 μs
Less than 34 μs↓ Symbol error
Greater than or equal to 34 μs↓ and less than 96 μs 1
Greater than or equal to 96 μs↓ and less than 164 μs 0
Greater than or equal to 164 μs↓ and less than 239 μs Symbol error
High for at least 239 μs Break Once the timer reaches 239 μs
8-bit reception
OVE
No
Yes
Byte error or CRC error
No
Yes
C2BRKE
C2SYME
C2BRKE
C2SYME
C2OVE
C2ECL
Clear the error flag
C2ECL
C2BYTE
C2CRCE
Occurrence of a reception error interrupt
Occurrence of a reception com-pletion interrupt

Remark	C2IDL:	Bus idle flag (bit 7 of class 2 status register 2 (C2ST2))
	C2BRKE:	Break error flag (bit 0 of class 2 status register 2 (C2ST2))
	C2SYME:	Symbol error flag (bit 1 of class 2 status register 2 (C2ST2))
	C2BYTE:	Byte error flag (bit 2 of class 2 status register 2 (C2ST2))
	C2CRCE:	CRC error flag (bit 3 of class 2 status register 2 (C2ST2))
	C2OVE:	Overrun error flag (bit 4 of class 2 status register 2 (C2ST2))
	C2SHTE:	Short error flag (bit 5 of class 2 status register 2 (C2ST2))
	C2ECL:	Error flag reset (bit 4 of class 2 control register 1 (C2CT1))

Figure 12-10. Reception Status Transition Diagram (×4 Mode)

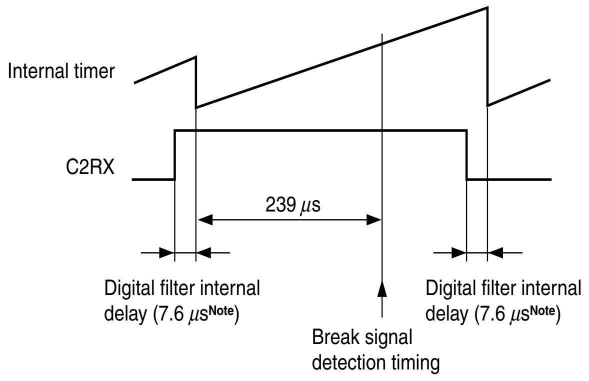

Symbol image
Less than 8.5 μs↑ Symbol error
Less than 8.5 μs↓ Symbol error
Greater than or equal to 8.5 μs↑ and less than 24 μs 0
Greater than or equal to 24 μs↑ and less than 41 μs 1
Greater than or equal to 8.5 μs↓ and less than 24 μs 1
Greater than or equal to 24 μs↓ and less than 41 μs 0
8-bit reception
No
OVE
Yes
Less than 41 μs↓ Noise
C2IDL
↑
Greater than or equal to 41 μs↓ and less than 60 μs SOF
Greater than or equal to 41 μs↑ and less than 60 μs Symbol error
Greater than or equal to 41 μs↓ and less than 60 μs Symbol error
High for at least 60 μs Break Once the timer reaches 60 μs
Low for at least 60 μs EOF Once the timer reaches 60 μs
High for at least 60 μs Break Once the timer reaches 60 μs
C2BRKE
Byte error or CRC error
No
Yes
C2SYME
C2BRKE
C2SYME
C2OVE
C2ECL
Clear the error flag
C2BYTE
C2CRCE
C2ECL
Occurrence of a reception error interrupt
Occurrence of a reception com-pletion interrupt

Remark	C2IDL:	Bus idle flag (bit 7 of class 2 status register 2 (C2ST2))
	C2BRKE:	Break error flag (bit 0 of class 2 status register 2 (C2ST2))
	C2SYME:	Symbol error flag (bit 1 of class 2 status register 2 (C2ST2))
	C2BYTE:	Byte error flag (bit 2 of class 2 status register 2 (C2ST2))
	C2CRCE:	CRC error flag (bit 3 of class 2 status register 2 (C2ST2))
	C2OVE:	Overrun error flag (bit 4 of class 2 status register 2 (C2ST2))
	C2SHTE:	Short error flag (bit 5 of class 2 status register 2 (C2ST2))
	C2ECL:	Error flag reset (bit 4 of class 2 control register 1 (C2CT1))

Figure 12-11. Standard Reception Timing (Normal Mode) (1/2)

(1) Standard break signal time

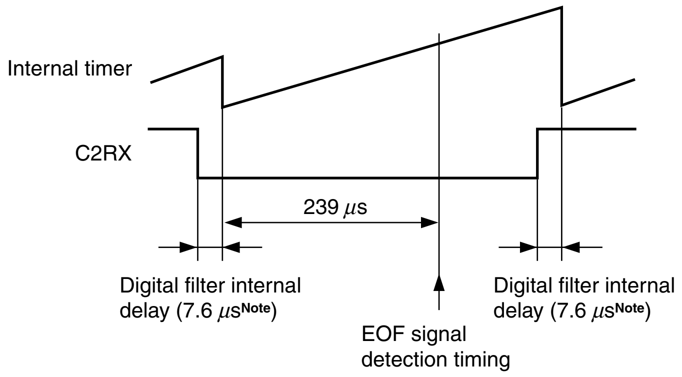
When 239 μs has elapsed since the start of internal timer counting


Internal timer
C2RX
239 μs
Digital filter internal delay (7.6 μsNote)
Break signal detection timing
Digital filter internal delay (7.6 μsNote)

Note System clock @ 4.19 MHz operation

(2) Standard EOF (End Of Frame) signal time

When 239 μs has elapsed since the start of internal timer counting

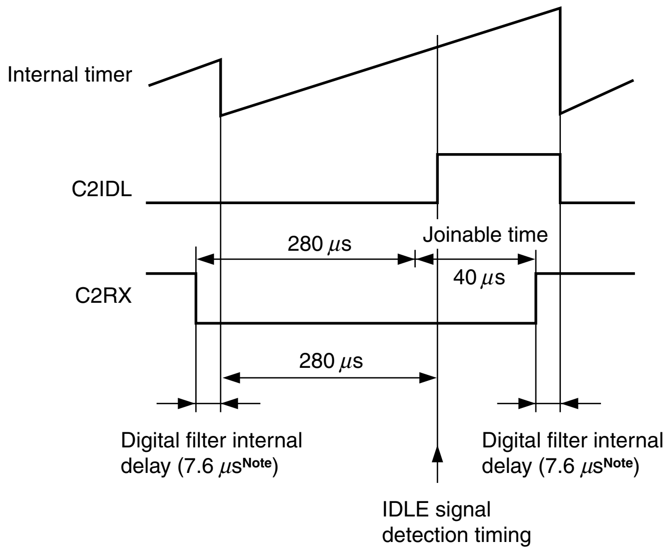

Internal timer
C2RX
239 μs
Digital filter internal delay (7.6 μsNote)
EOF signal detection timing
Digital filter internal delay (7.6 μsNote)

Note System clock @ 4.19 MHz operation

Figure 12-11. Standard Reception Timing (Normal Mode) (2/2)

(3) Standard IDLE (bus idle) time

When 280 μs has elapsed since the start of internal timer counting


Internal timer
C2IDL
C2RX
280 μs
Joinable time
40 μs
280 μs
Digital filter internal delay (7.6 μsNote)
Digital filter internal delay (7.6 μsNote)
IDLE signal detection timing

Note System clock @ 4.19 MHz operation

Remark If a receive signal rises within the period between 280 and 320 μs, that is 40 μs, after the receive signal falls during transmission standby (IDLE), "SOF join" occurs. This is called "Joinable Time."

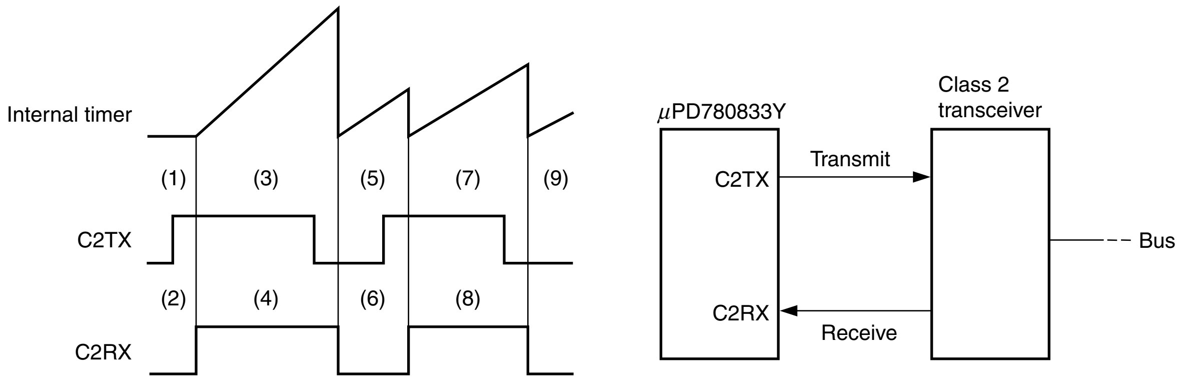
12.3.2 Transmission

Class 2 performs transmission while monitoring the receive data.

The receive (C2RX pin) signal usually lags behind the transmit (C2TX pin) signal by 14 to 25 μs as a result of the receiver characteristics. If Class 2 fails to obtain the right to transmit, it stops transmission.

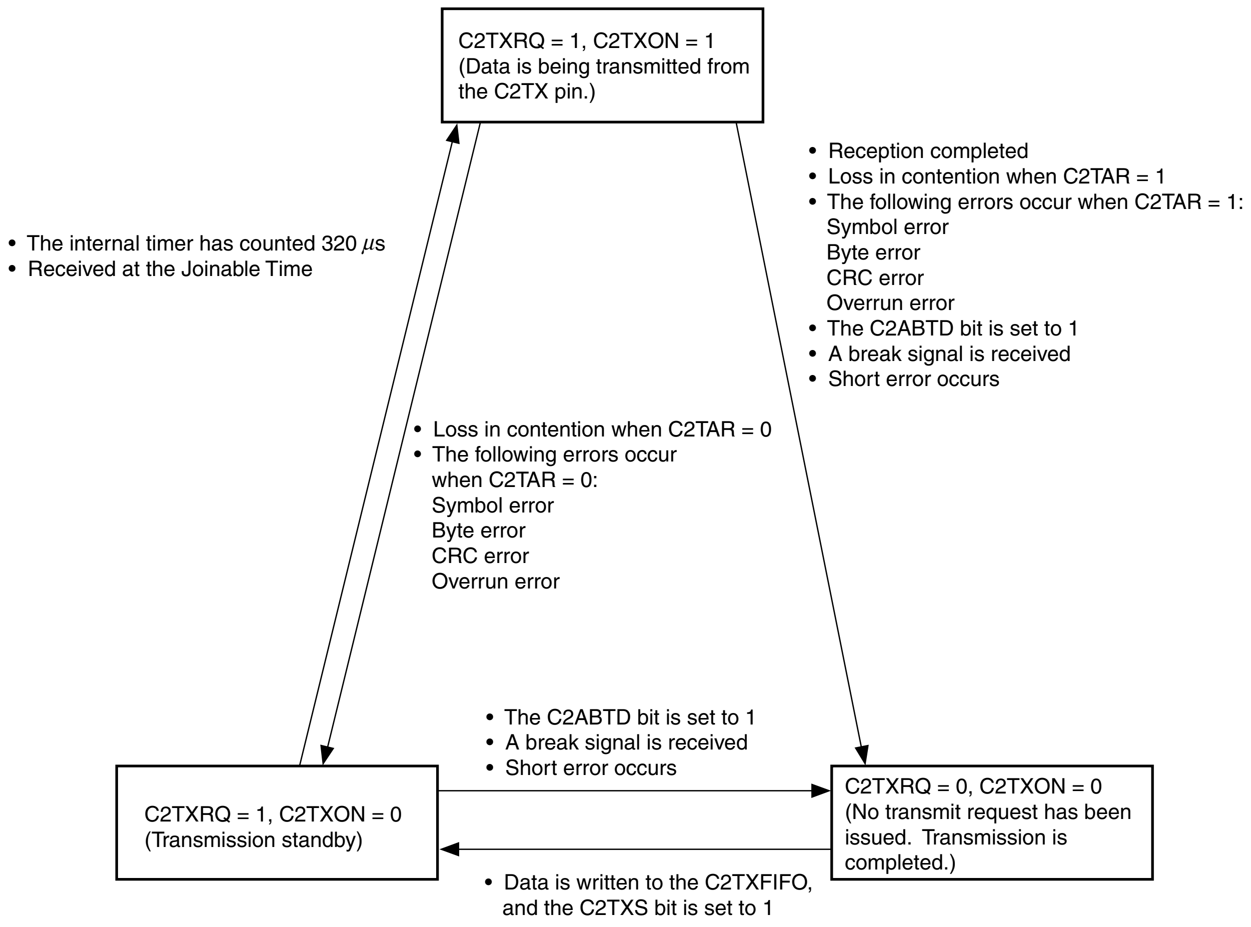
First, Class 2 writes the transmit data into the transmit FIFO (C2TXFIFO) in preparation for transmission. Next, it sets bit 3 (C2TXS) of class 2 control register 1 (C2CT1). If the transmit FIFO is empty, setting the C2TXS bit results in nothing being transmitted.

Figure 12-12. Transmission



1) When the C2TXS bit is set (1), the C2TX pin becomes high level. Simultaneously, the transmitting node will enter reception mode.
2) The internal timer starts at the rising edge of the C2RX pin.
   If a short occurs, a reception error interrupt request (INTC2) is generated to stop transmission.
3) The C2TX pin becomes low level when the SOF symbol on the bus reaches the pulse width. (SOF transmission)
4) The potential of the bus is dropped by the use of an external bus pull-down resistor.
   An SOF symbol is detected at the falling edge of the C2RX pin. At the same time, the internal timer is reset. (SOF reception)
5) The level of the output is low. The MSB of byte 1 is transmitted. If its level is low, the C2TX pin becomes high level when the bit data symbol is output to the bus (the first bit is transmitted).
6) Upon receiving a symbol, Class 2 compares it with the transmit symbol. If a conflict with another node has occurred (if the current node has a lower priority), Class 2 stops transmission (the first bit is received).
7) The level of the output is high. Bit 6 of byte 1 is transmitted. If its level is high, the C2TX pin becomes low level when the bit data symbol is output to the bus (the next bit is transmitted).
8) Upon receiving a receive symbol, Class 2 compares it with its transmit symbol. If a conflict with another node has occurred (if the current node has a lower priority), Class 2 stops transmission (the next bit is received).
   If a short occurs, Class 2 stops transmission.
9) Steps 5) to 8) are repeated until all transmit data has been transmitted.
   A CRC field is added as required. Transmission ends upon the detection of an EOF symbol.

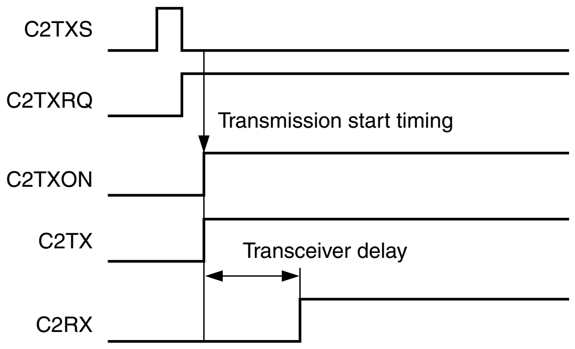
Figure 12-13. Transmission Status Transition Diagram (Transmission Status Flag)


C2TXRQ = 1, C2TXON = 1
(Data is being transmitted from the C2TX pin.)
• The internal timer has counted 320 μs
• Received at the Joinable Time
• Reception completed
• Loss in contention when C2TAR = 1
• The following errors occur when C2TAR = 1:
Symbol error
Byte error
CRC error
Overrun error
• The C2ABTD bit is set to 1
• A break signal is received
• Short error occurs
• Loss in contention when C2TAR = 0
• The following errors occur when C2TAR = 0:
Symbol error
Byte error
CRC error
Overrun error
• The C2ABTD bit is set to 1
• A break signal is received
• Short error occurs
C2TXRQ = 1, C2TXON = 0
(Transmission standby)
C2TXRQ = 0, C2TXON = 0
(No transmit request has been issued. Transmission is completed.)
• Data is written to the C2TXFIFO, and the C2TXS bit is set to 1

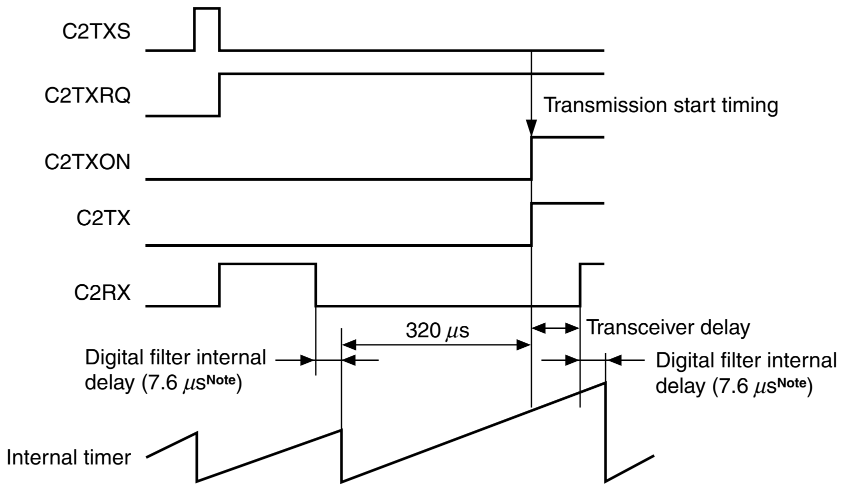
Remark	C2TXON, C2TXRQ:	Transmission status flag (bits 0 and 1 of class 2 status register 1 (C2ST1)
	C2TAR:	Release automatic retransmission (bit 1 of class 2 control register 1 (C2CT1)
	C2ABTD:	Transmission data reset (bit 2 of class 2 control register 1 (C2CT1)
	C2TXS:	Transmission start (bit 3 of class 2 control register 1 (C2CT1)

Figure 12-14. Transmission Start Timing (1/2)

(1) If 320 μs has elapsed since the last signal (the falling edge): Transmission begins immediately.



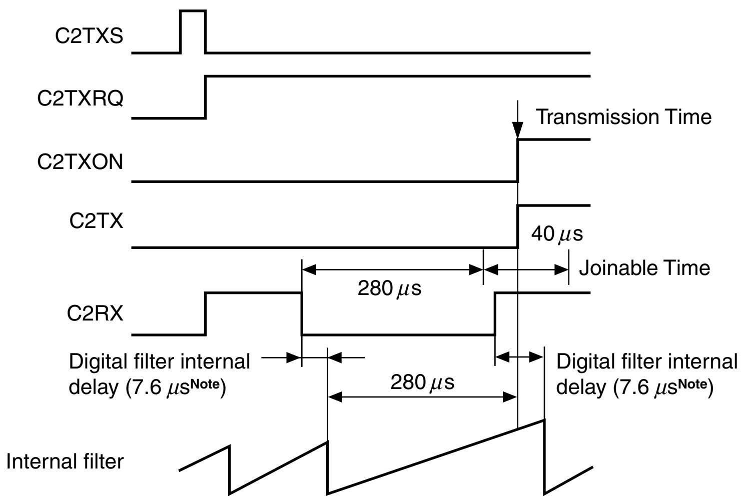
(2) If the bus is busy: After 320 μs has elapsed since the last signal (a falling edge), transmission begins. (The signal interval at the C2RX pin is the sum of the internal delay, external delay, and 320 μs).


C2TXS
C2TXRQ
Transmission start timing
C2TXON
C2TX
C2RX
320 μs
Transceiver delay
Digital filter internal delay (7.6 μsNote)
Digital filter internal delay (7.6 μsNote)
Internal timer

Note System clock @ 4.19 MHz operation

Figure 12-14. Transmission Start Timing (2/2)

(3) When a rising edge to the C2RX pin arrives at the Joinable Time during transmission standby: If a rising edge to the C2RX pin arrives after 280 μs has elapsed since the last signal (a falling edge), the SOF symbol is joined.

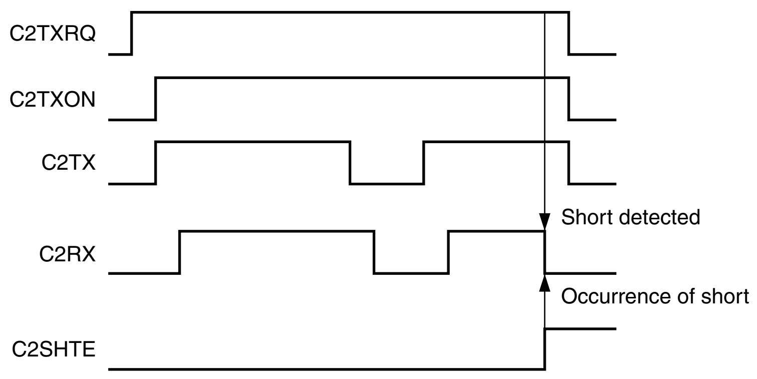


Note System clock @ 4.19 MHz operation

Remark If a receive signal rises within the period between 280 and 320 μs, that is 40μs, after the receive signal filling during transmission standby (IDLE), “SOF join” occurs. This is called “Joinable Time.”

Figure 12-15. Short Detection (at Transmission Only) (1/2)

(1) If a falling edge is detected when the C2TX pin outputs a high level:



(2) If no signal is returned to the C2RX pin when the C2TX pin outputs an SOF symbol:

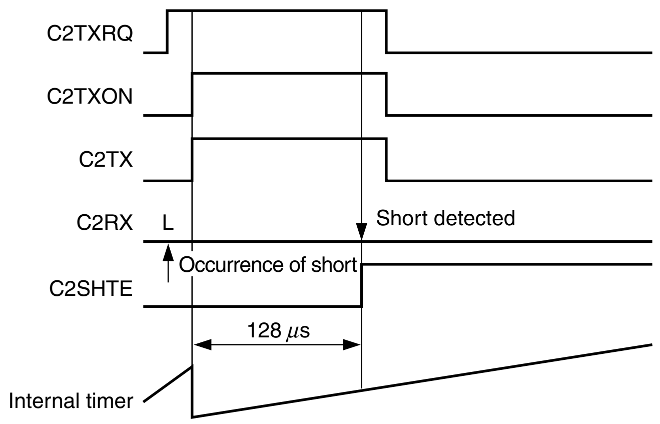


Figure 12-15. Short Detection (at Transmission Only) (2/2)

(3) If no signal is returned to the C2RX pin when the C2TX pin outputs data:
CRC and byte errors occur. A short is detected at 2) in the previous page upon retransmission after the error flag is cleared (C2ECL = 1).

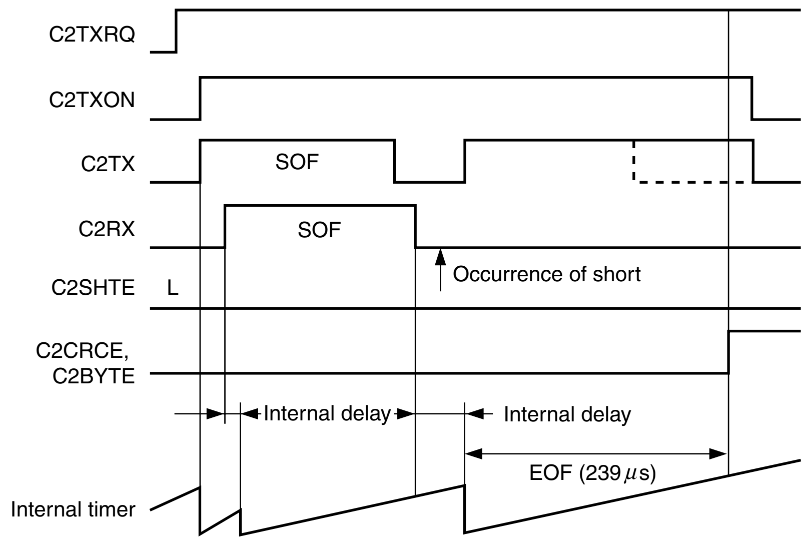
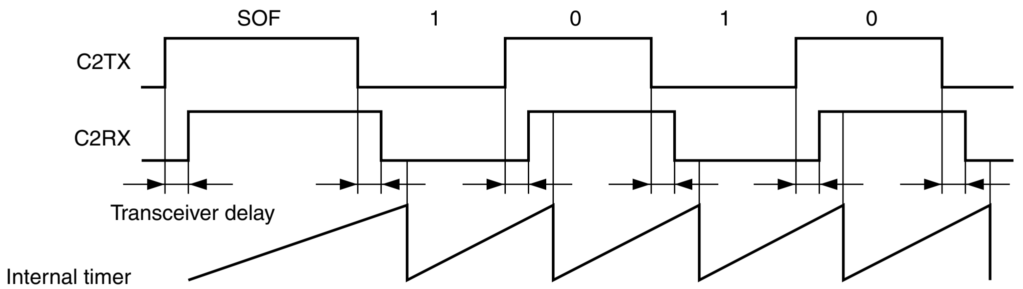
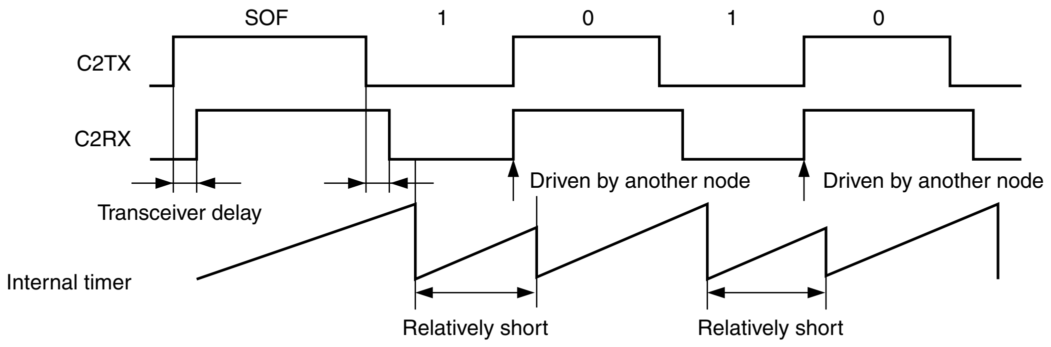


Figure 12-16. Collision 1 (If Clock Speed Varies for Same Symbol)

(1) Normal



(2) Collision with any other fast node



(3) Collision with any other slow node

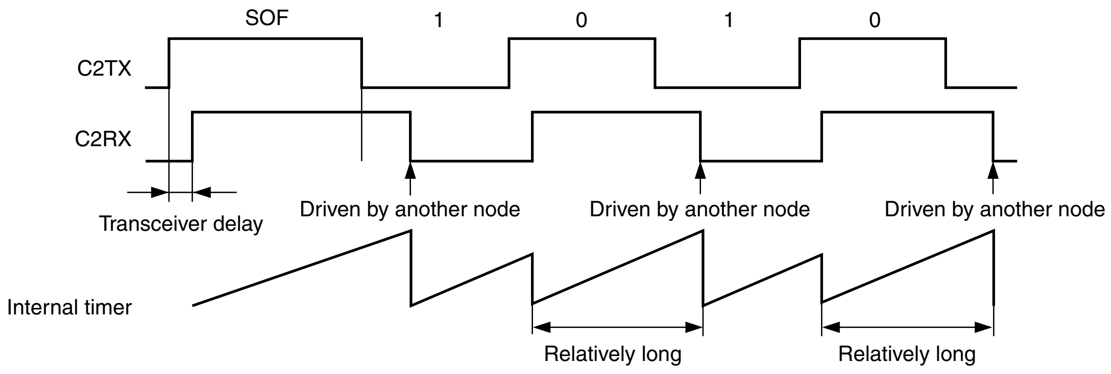
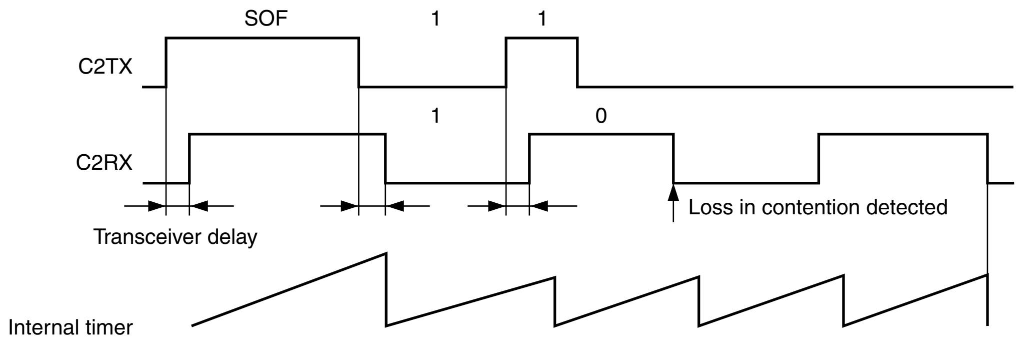
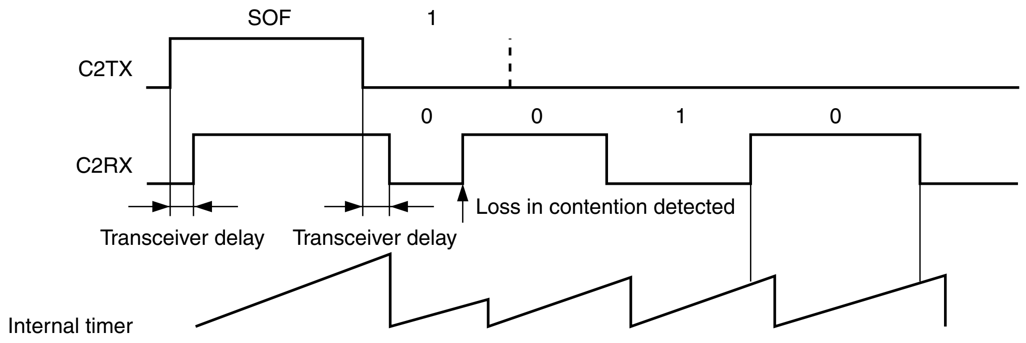


Figure 12-17. Collision 2 (Loss in Contention with Different Symbols)

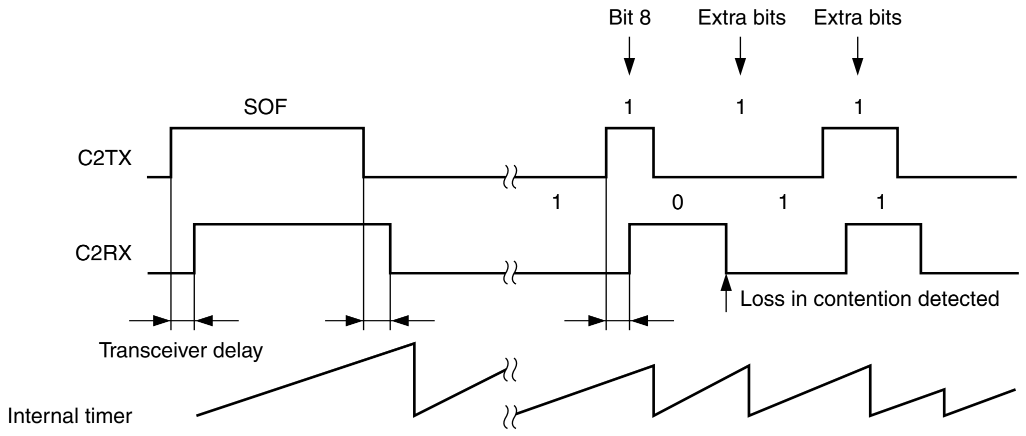
(1) Loss in contention for active (high) symbol



(2) Loss in contention for passive (low) symbol



(3) Loss in contention for a symbol between bytes (extra bits = +11B)



12.3.3 Error detection

Class 2 provides the following 6 error detection functions.

Table 12-2. Types of Errors and Flags

Type of Error	Error Flag
Break signal (break error)	C2BRKE (bit 0 of class 2 status register 2 (C2ST2))
Symbol error	C2SYME (bit 1 of class 2 status register 2 (C2ST2))
Byte error	C2BYTE (bit 2 of class 2 status register 2 (C2ST2))
CRC error	C2CRCE (bit 3 of class 2 status register 2 (C2ST2))
Overrun error	C2OVE (bit 4 of class 2 status register 2 (C2ST2))
Short error	C2SHTE (bit 5 of class 2 status register 2 (C2ST2))

If an error is detected, the corresponding error flag is set, and interrupt request signal (INTC2) is generated. If an error flag is set (1), reception is not performed. An error flag is cleared if bit 4 (C2ECL) of class 2 control register 1 (C2CT1) is set (1).

If a short error or break signal is detected, Class 2 resets the transmit section.

If an error other than a short error or break signal is detected, output of a transmit signal is terminated.

When automatic retransmission is set (bit 1 (C2TAR) of C2CT1 is set to 0), a transmission request is continued. Automatic retransmission starts in the bus idle status after the error flag is cleared (C2ECL = 1).

(1) Break signal (C2BRKE) (detection upon reception)

If a high level continues for at least 239 μs upon reception, it is recognized as a break signal. When a break signal is detected, Class 2 operates as follows.

- Sets automatic retransmission (C2TAR = 0)
- Releases ×4 mode (C2X4 = 1)
- Resets the address pointer of the transmission system.
- Status flag of transmission system (C2TXON, C2TXRQ, and C2XFUL = 000B)

Remark		
	C2TAR:	Bit 1 of class 2 control register 1 (C2CT1)
	C2X4:	Bit 1 of class 2 control register 2 (C2CT2)
	C2TXON, C2TXRQ, and C2XFUL:	Bits 0 to 2 of class 2 status register 1 (C2ST1)

(2) Symbol error (C2SYME) (detection upon reception)

Once an SOF symbol has been detected, a symbol error is detected if the following waveforms, for which no symbol is specified, are received.

Table 12-3. Waveforms Causing Symbol Errors

Symbol Error		Remark
Pulse width of less than 34 μs	High level	Shorter than the receive spec for active 1
	Low level	Shorter than the receive spec for passive 0
163 μs ≤ pulse width < 239 μs	High level	Longer than active 0 and shorter than Active Break
	Low level	Longer than passive 1 and shorter than an EOF symbol

(3) Byte error (C2BYTE) (detected at reception)

When an EOF symbol is detected, a byte error is detected if the received data is not in byte units.

(4) CRC error (C2CRCE) (detected at reception)

When a CRC is used (bit 5 (C2NCRC) of class 2 control register 1 (C2CT1) is 0), a CRC check is performed. When an EOF symbol is detected, a CRC error is detected if the result of checking the relationship between the received data and CRC field using the CRC check circuit is abnormal.

(5) Overrun error (C2OVE) (detected at reception)

An overrun error is detected if receive data that has not been read exceeds the capacity of the receive FIFO (21 bytes).

(6) Short error (detected at transmission: The flag is C2SHTE.)

A short is detected if it is likely that the bus has encountered a shorted GND. A short error is recognized in the following cases. No V_{DD} short will be detected.

- No receive edge is recognized even if about 128 μs has elapsed after the transmission starts.
- A falling edge of reception is recognized during the transmission of a high-level symbol.

12.3.4 Interrupt operation

Class 2 generates interrupt request signals (INTC2) from to the following four interrupt sources.

- Wake-up request
- Reception completion
- Sleep enable
- Reception error

If INTC2 is generated, the interrupt request flags (bits 4 and 5 (C2INT1 and C2INT2) of class 2 status register 1 (C2ST1)) are set in accordance with the interrupt source (See **Table 12-4**).

Table 12-4. Interrupt Request Flag

C2INT2	C2INT1	Interrupt Request Flag
0	0	Wake-up request
0	1	Reception completion
1	0	Sleep enable
1	1	Reception error

If interrupt requests occur continuously, the previous interrupt request flag is overwritten. Therefore, an interval of interrupt request generation must be considered at programming.

A wake-up request signal is generated during the sleep state.

Even if interrupt request signal generation is prohibited (C2IE = 0), a wake-up request is not masked.

During the normal wake-up state, a wake-up request interrupt signal is masked.

(1) Wake-up request

In the sleep state, if a rising edge signal of the C2RX pin is detected, INTC2 is generated (INTC2 is generated even in STOP mode).

(2) Reception completion

When reception has been completed normally, INTC2 is generated.

(3) Sleep enable

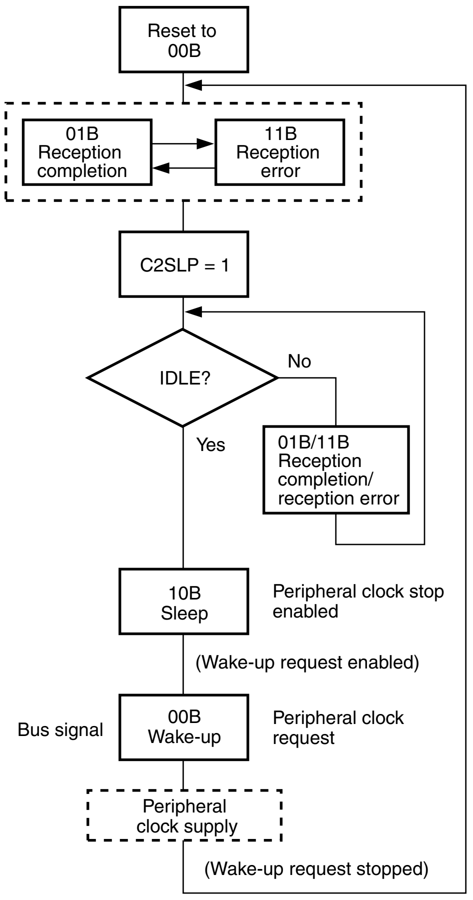
If a sleep request (C2SLP = 1) is issued, INTC2 is generated when the bus becomes idle.

(4) Reception error

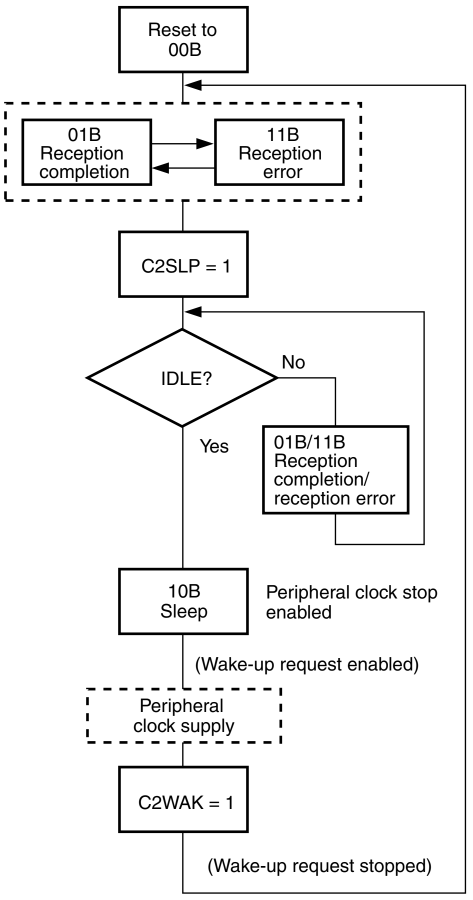
INTC2 is generated upon the detection of a reception error.

Figure 12-18. Interrupt Flag Status Transition Diagram

(1) Sleep state released by bus input



(2) Sleep state released by C2WAK


Reset to 00B
01B Reception completion
11B Reception error
C2SLP = 1
IDLE?
No
Yes
01B/11B Reception completion/ reception error
10B Sleep
Peripheral clock stop enabled
(Wake-up request enabled)
Peripheral clock supply
C2WAK = 1
(Wake-up request stopped)

Remark	00B:	C2INT2 = 0, C2INT1 = 0 (wake-up request)
	01B:	C2INT2 = 0, C2INT1 = 1 (reception completion)
	10B:	C2INT2 = 1, C2INT1 = 0 (sleep enable)
	11B:	C2INT2 = 1, C2INT1 = 1 (reception error)

Table 12-5. Minimum Time for Flag Change

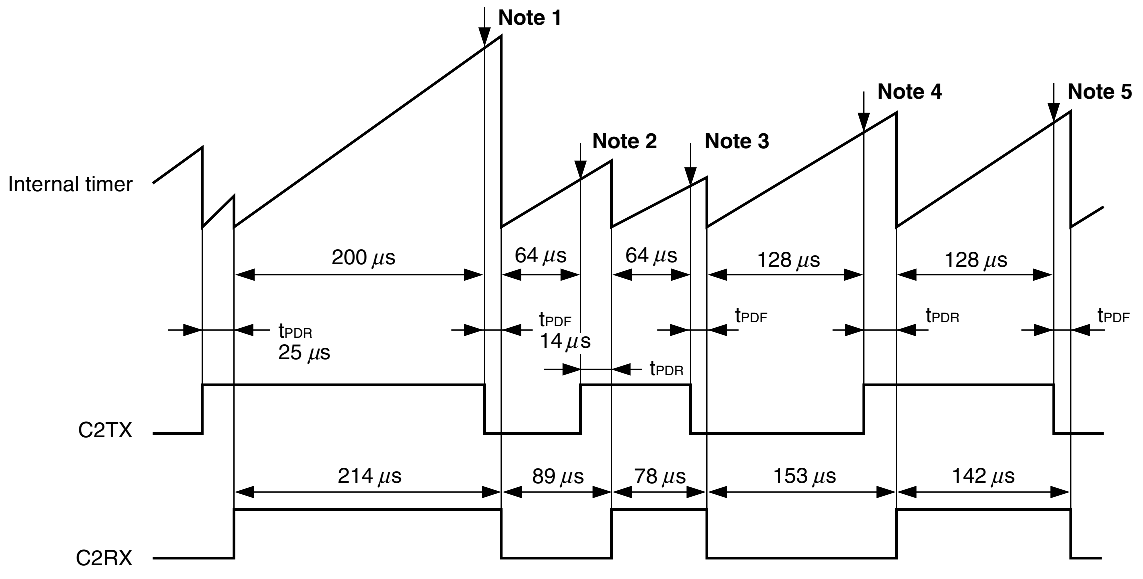
C2INT2, C2INT1	Flag Status	Minimum Time for Flag Change
00B	Wake/reset	Immediately after sleep enable
01B	Reception completion	675 μs (SOF, DATA, EOF)
10B	Sleep enable	41 μs (sleep request upon reception)
11B	Reception error	239 μs (break signal)

12.3.5 Propagation delay correction (for correcting the standard transmit time)

(1) Normal mode

If the rise time propagation delay (t_{PDR}) of the Class 2 transceiver is 25 μs, and the fall time propagation delay (t_{PDF}) is 14 μs, for example, setting PDR0 to PDR4 and PDF0 to PDF4 to 0 results in the C2TX and C2RX waveforms having the following appearance.

Figure 12-19. Waveforms of C2TX and C2RX Pins (Normal Mode: PDR0 to PDR4 = 0 and PDF0 to PDF4 = 0)



Notes 1. Transmit SOF standard transmission time in normal mode

2. Transmit Passive 0 standard transmission time in normal mode

3. Transmit Active 1 standard transmission time in normal mode

4. Transmit Passive 1 standard transmission time in normal mode

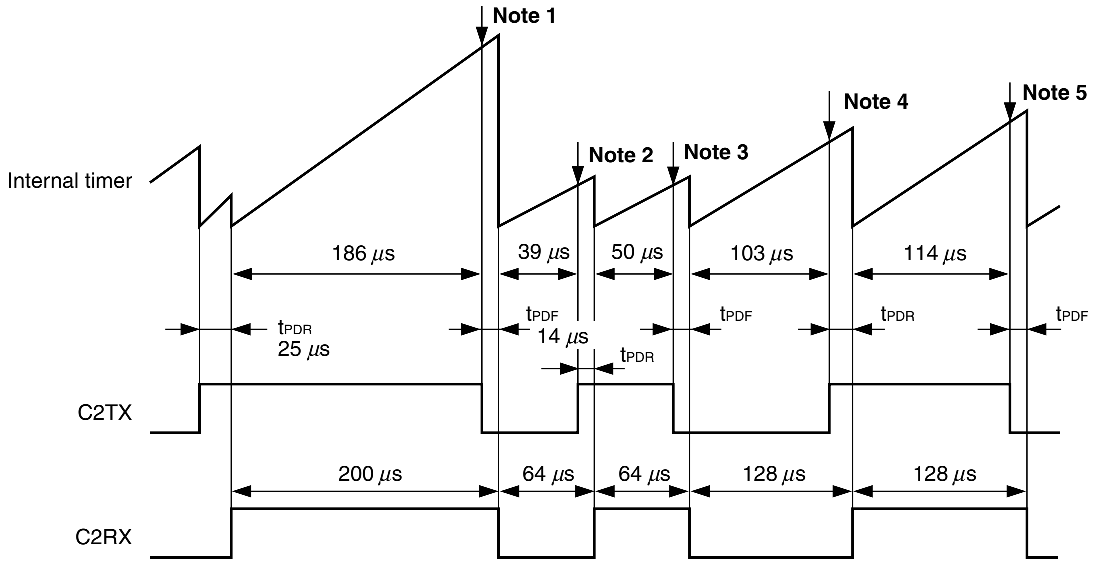
5. Transmit Active 0 standard transmission time in normal mode

If the C2RX pin is recognized as a signal on the bus, each symbol is distorted by the propagation delay characteristics of the transceiver. If the system clock frequency is 4.19 MHz, for example, and the internal clock is at 0.48 µs per clock pulse, the propagation delay correction registers will be set as follows.

- PDR = 25 μs/0.48 μs/2 ≒ 26
- PDF = 14 μs/0.48 μs/2 ≒ 15

Setting PDR0 to PDR4 = 26 and PDF0 to PDF4 = 15, respectively, causes Class 2 to correct the corresponding standard transmission, and the results in waveforms shown in Figure 12-20.

Figure 12-20. Waveforms of C2TX and C2RX Pins (Normal Mode: PDR0 to PDR4 = 26 and PDF0 to PDF4 = 15)

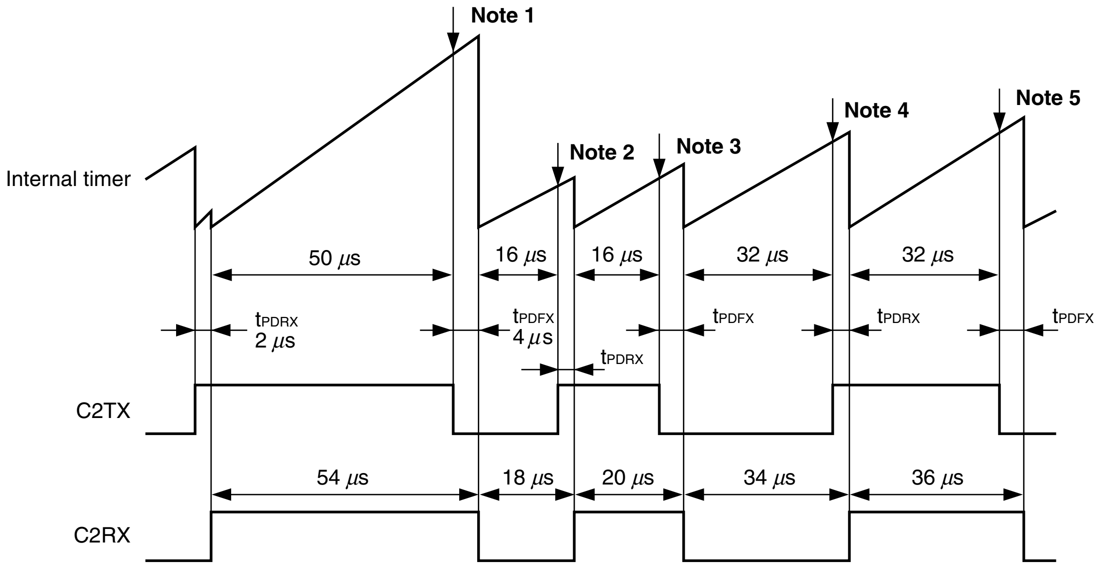


Notes
1. Transmit SOF standard transmission time in normal mode
2. Transmit Passive 0 standard transmission time in normal mode
3. Transmit Active 1 standard transmission time in normal mode
4. Transmit Passive 1 standard transmission time in normal mode
5. Transmit Active 0 standard transmission time in normal mode

(2) ×4 mode

In 4× mode, if the rise time propagation delay (t_{PDRX}) of the Class 2 transceiver is 2 μs, and the fall time propagation delay (t_{PDRX}) is 4 μs, for example, setting PDRX0 to PDRX2 and PDFX0 to PDFX2 to 0 results in the C2TX and C2RX waveforms having the following appearance.

Figure 12-21. Waveforms of C2TX and C2RX Pins
(×4 Mode: PDRX0 to PDRX2 = 0 and PDFX0 to PDFX2 = 0)



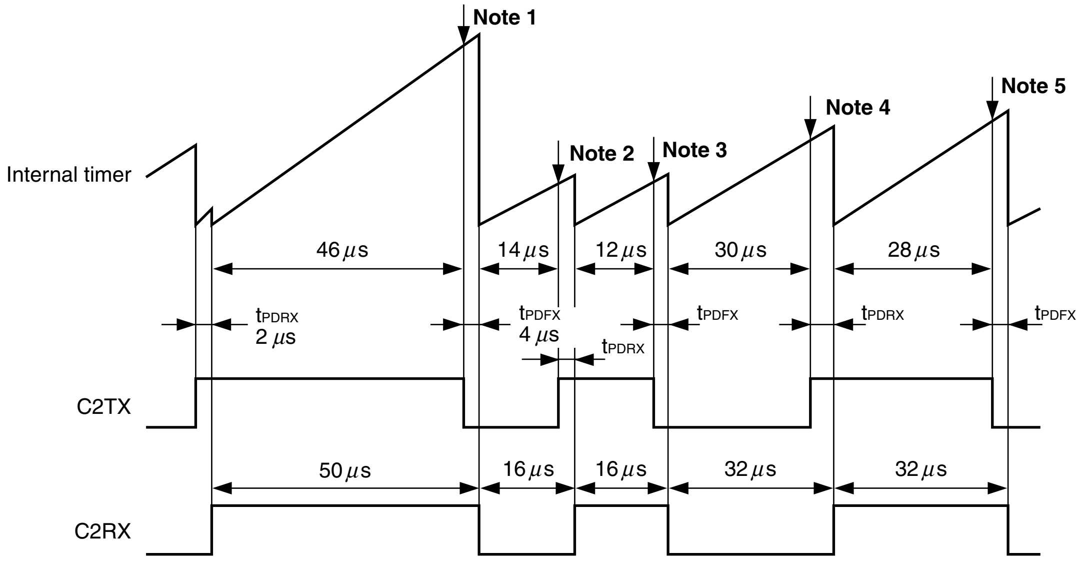
Notes 1. Transmit SOF standard transmission time in ×4 mode
2. Transmit Passive 0 standard transmission time in ×4 mode
3. Transmit Active 1 standard transmission time in ×4 mode
4. Transmit Passive 1 standard transmission time in ×4 mode
5. Transmit Active 0 standard transmission time in ×4 mode

If the C2RX pin is recognized as a signal on the bus, each symbol is distorted by the propagation delay characteristics of the transceiver. If the system clock frequency is 4.19 MHz, for example, and the internal clock is at 0.48 μs per clock pulse, the propagation delay correction registers will be set as follows.

- PDRX = 2 μs/0.48 μs/2 ≒ 2
- PDFX = 4 μs/0.48 μs/2 ≒ 4

Setting PDRX0 to PDRX2 = 2 and PDFX0 to PDFX2 = 4, respectively, causes Class 2 to correct the corresponding standard transmission, and the results in waveforms shown in Figure 12-22.

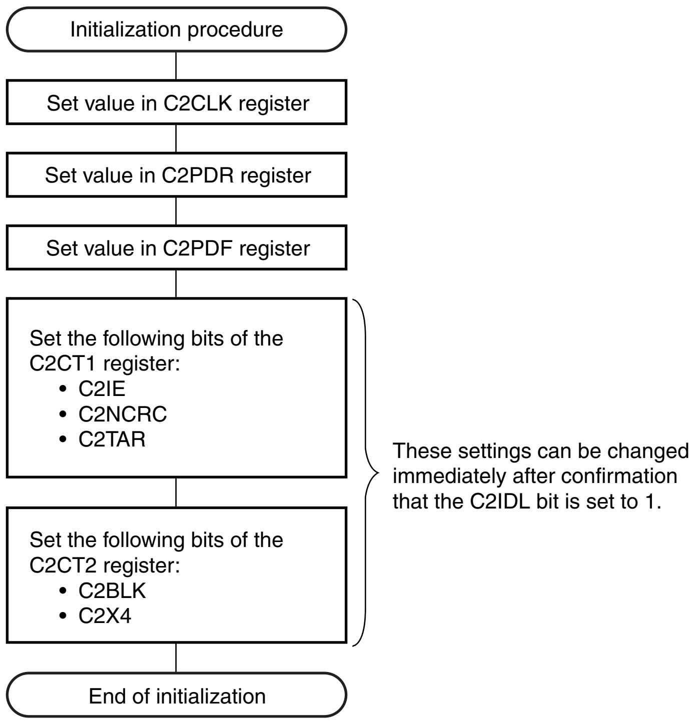
Figure 12-22. Waveforms of C2TX and C2RX Pins
(×4 Mode: PDRX0 to PDRX2 = 2 and PDFX0 to PDFX2 = 4)



Notes 1. Transmit SOF standard transmission time in ×4 mode
2. Transmit Passive 0 standard transmission time in ×4 mode
3. Transmit Active 1 standard transmission time in ×4 mode
4. Transmit Passive 1 standard transmission time in ×4 mode
5. Transmit Active 0 standard transmission time in ×4 mode

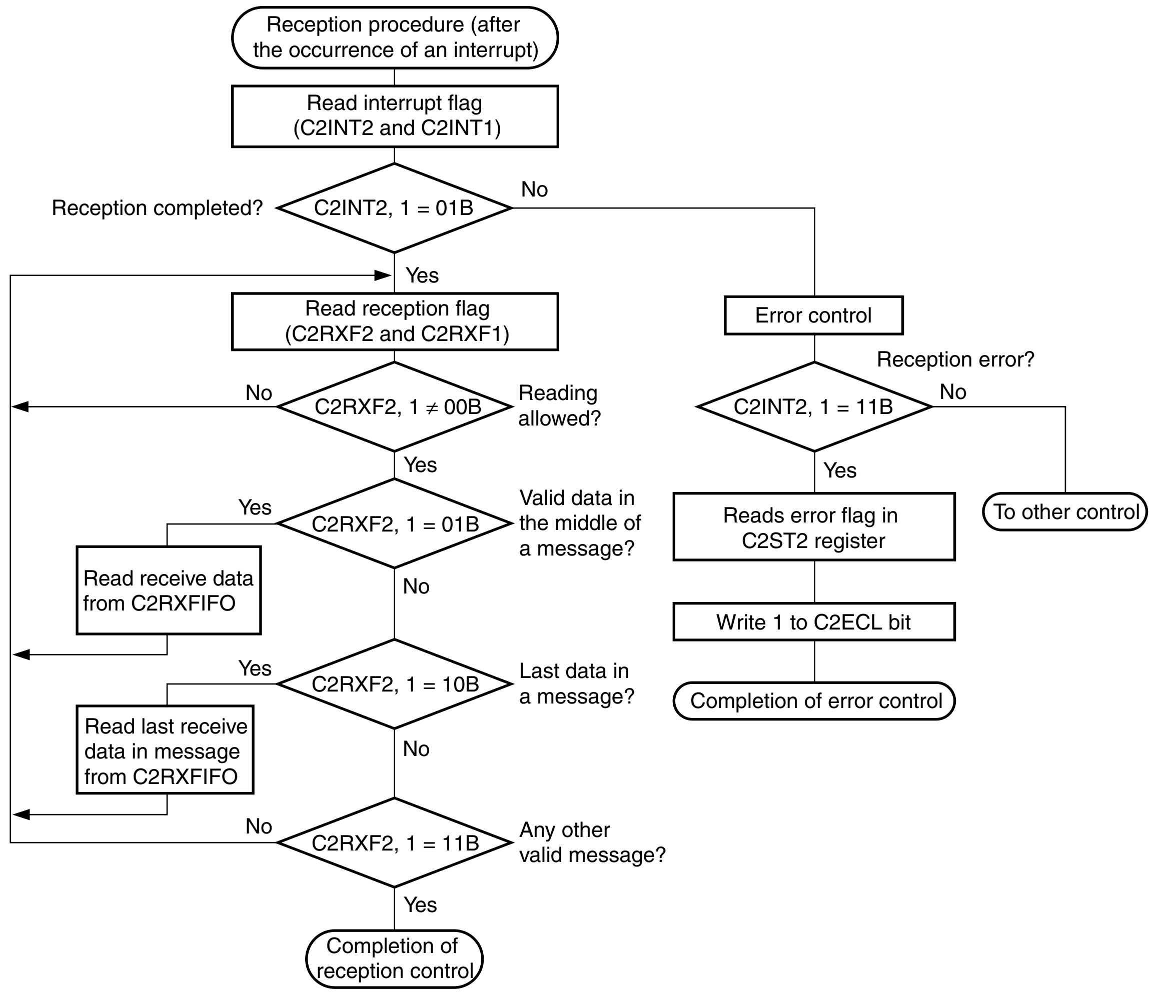
12.3.6 Communication control based on CPU

(1) Example of initialization


Initialization procedure
Set value in C2CLK register
Set value in C2PDR register
Set value in C2PDF register
Set the following bits of the C2CT1 register:
• C2IE
• C2NCRC
• C2TAR
Set the following bits of the C2CT2 register:
• C2BLK
• C2X4
These settings can be changed immediately after confirmation that the C2IDL bit is set to 1.
End of initialization

(2) Reception control

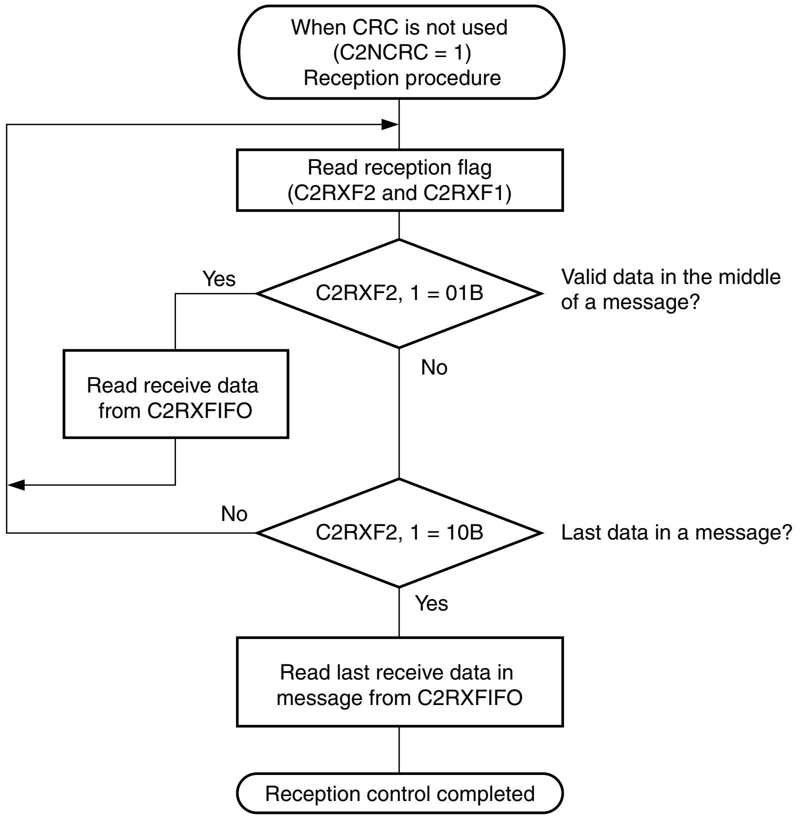
(a) Example of reception control in normal mode


Reception procedure (after the occurrence of an interrupt)
Read interrupt flag (C2INT2 and C2INT1)
Reception completed?
C2INT2, 1 = 01B
No
Yes
Read reception flag (C2RXF2 and C2RXF1)
No
C2RXF2, 1 ≠ 00B
Reading allowed?
Yes
Yes
C2RXF2, 1 = 01B
Valid data in the middle of a message?
Read receive data from C2RXFIFO
No
Yes
C2RXF2, 1 = 10B
Last data in a message?
Read last receive data in message from C2RXFIFO
No
No
C2RXF2, 1 = 11B
Any other valid message?
Yes
Completion of reception control
Error control
Reception error?
C2INT2, 1 = 11B
No
Yes
Reads error flag in C2ST2 register
To other control
Write 1 to C2ECL bit
Completion of error control

Caution On occurrence of a reception error, transmission or reception is not performed until the error flag is cleared.

(b) Example of reception control in block mode (1/2)

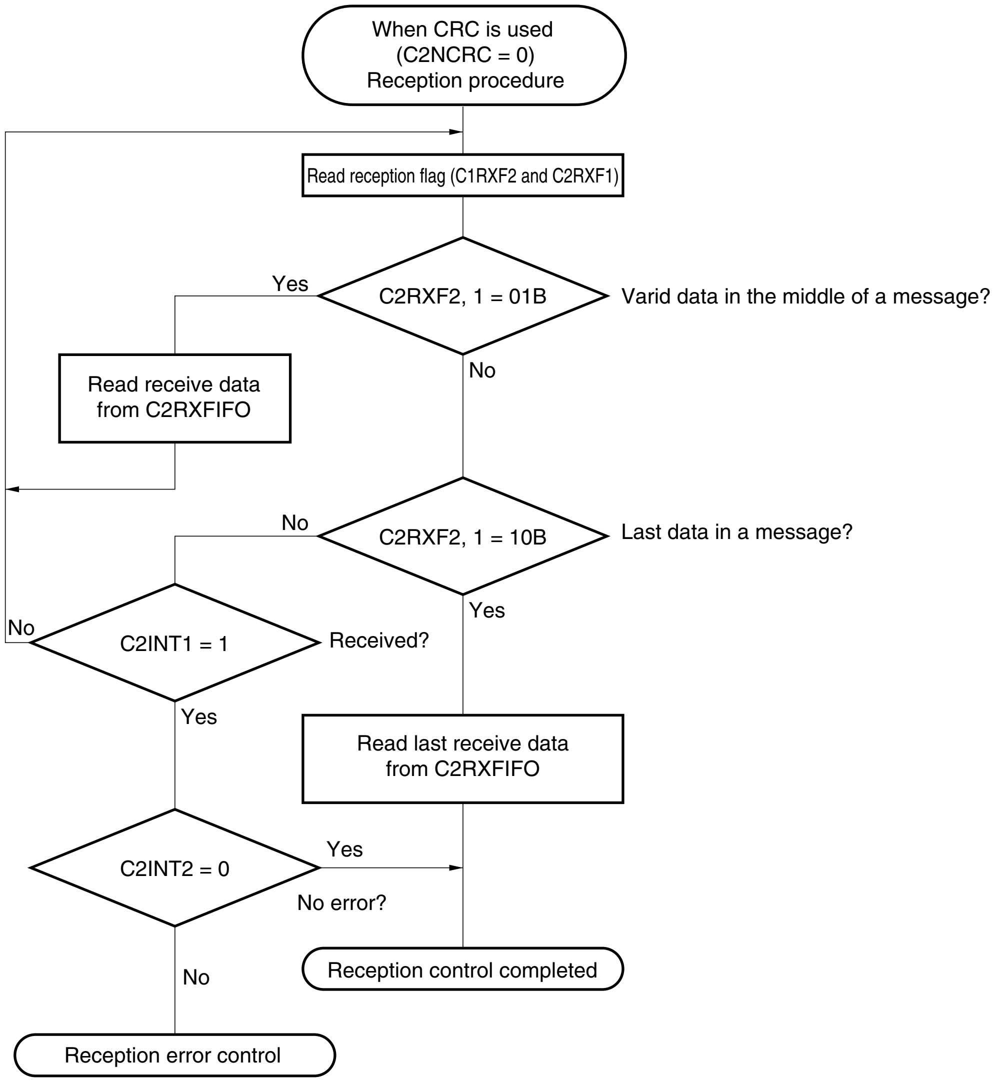
<1> When CRC is not used (C2NCRC = 1)


When CRC is not used (C2NCRC = 1) Reception procedure
Read reception flag (C2RXF2 and C2RXF1)
C2RXF2, 1 = 01B
Valid data in the middle of a message?
Yes
No
Read receive data from C2RXFIFO
C2RXF2, 1 = 10B
Last data in a message?
No
Yes
Read last receive data in message from C2RXFIFO
Reception control completed

★ **Caution Setting C2IE to 1 enables the generation of an interrupt request signal when reception is completed or a reception error occurs. To perform self-writing, set C21E to 0 to prevent generation of an interrupt request signal.**

★ **(b) Example of reception control in block mode (2/2)**

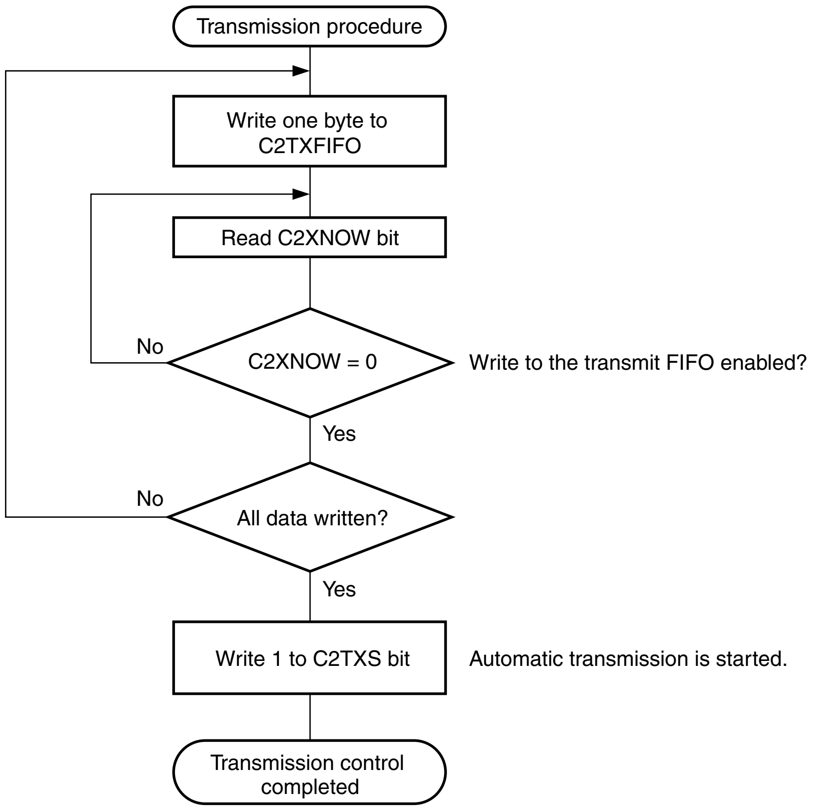
<2> When CRC is used (C2NCRC = 0)


When CRC is used (C2NCRC = 0) Reception procedure
Read reception flag (C1RXF2 and C2RXF1)
C2RXF2, 1 = 01B
Varid data in the middle of a message?
Yes
No
Read receive data from C2RXFIFO
C2RXF2, 1 = 10B
Last data in a message?
No
Yes
C2INT1 = 1
Received?
No
Yes
Read last receive data from C2RXFIFO
C2INT2 = 0
Yes
No error?
No
Reception control completed
Reception error control

★ **Caution Setting C2IE to 1 enables the generation of an interrupt request signal when reception is completed or a reception error occurs. To perform self-writing, set C21E to 0 to prevent generation of an interrupt request signal.**

(3) Transmission control

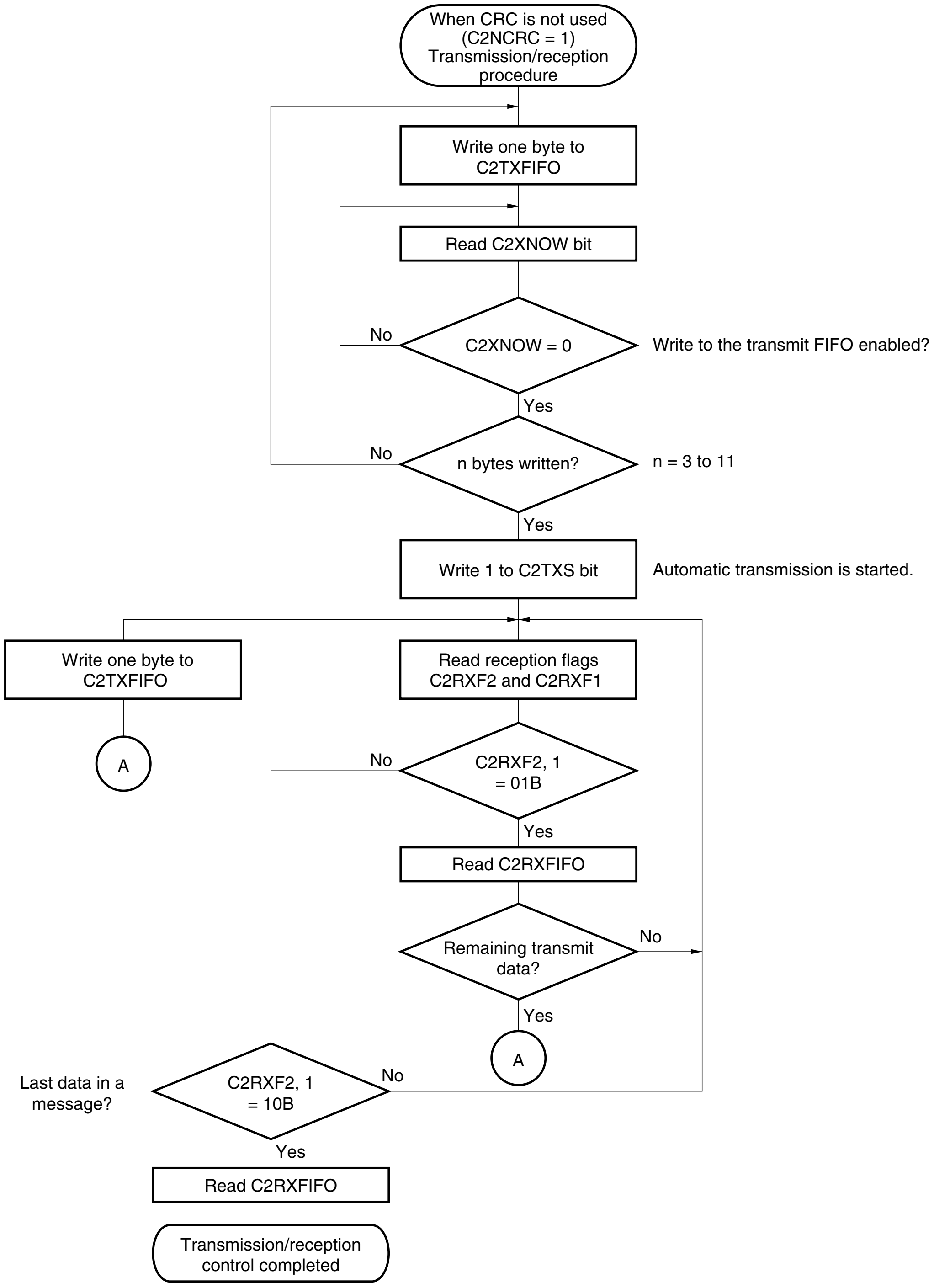
(a) Example of transmission control in normal mode (a message with 11 bytes or less is transmitted)


Transmission procedure
Write one byte to C2TXFIFO
Read C2XNOW bit
No
C2XNOW = 0
Write to the transmit FIFO enabled?
Yes
No
All data written?
Yes
Write 1 to C2TXS bit
Automatic transmission is started.
Transmission control completed

★ **Caution It is necessary to wait at least 2.5 internal clocks to read the contents of the C2XNOW bit after data is written to C2TXFIFO when operating with a system clock oscillation frequency of 8 MHz.**

★ **(b) Example of transmission/reception control in block mode (a message consisting of 12 bytes or more is transmitted/received) (1/2)**

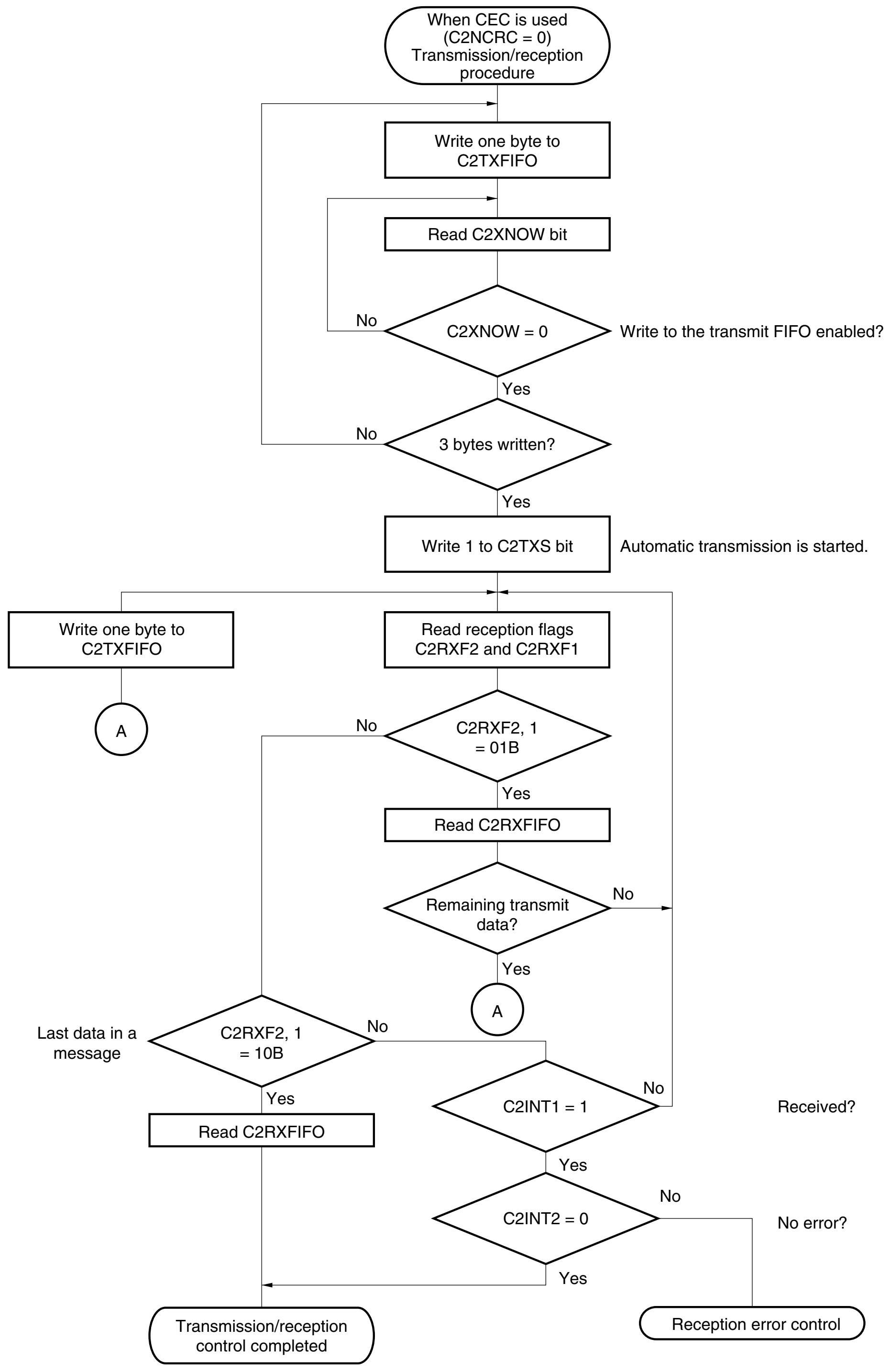
★ <1> When CRC is not used (C2NCRC = 1)


When CRC is not used (C2NCRC = 1) Transmission/reception procedure
Write one byte to C2TXFIFO
Read C2XNOW bit
C2XNOW = 0
No
Yes
Write to the transmit FIFO enabled?
n bytes written?
No
Yes
n = 3 to 11
Write 1 to C2TXS bit
Automatic transmission is started.
Write one byte to C2TXFIFO
A
Read reception flags C2RXF2 and C2RXF1
C2RXF2, 1 = 01B
No
Yes
Read C2RXFIFO
Remaining transmit data?
No
Yes
A
Last data in a message?
C2RXF2, 1 = 10B
No
Yes
Read C2RXFIFO
Transmission/reception control completed

Caution It is necessary to wait at least 2.5 internal clocks to read the contents of the C2XNOW bit after data is written to C2TXFIFO when operating with a system clock oscillation frequency of 8 MHz.

★ **(b) Example of transmission/reception control in normal mode (a message with 11 bytes or less is transmitted/received) (2/2)**

<2> When CRC is used (C2NCRC = 0)


When CEC is used (C2NCRC = 0) Transmission/reception procedure
Write one byte to C2TXFIFO
Read C2XNOW bit
C2XNOW = 0
Write to the transmit FIFO enabled?
No
Yes
3 bytes written?
No
Yes
Write 1 to C2TXS bit
Automatic transmission is started.
Write one byte to C2TXFIFO
A
Read reception flags C2RXF2 and C2RXF1
C2RXF2, 1 = 01B
No
Yes
Read C2RXFIFO
Remaining transmit data?
No
Yes
A
Last data in a message
C2RXF2, 1 = 10B
No
Yes
Read C2RXFIFO
C2INT1 = 1
Received?
No
Yes
C2INT2 = 0
No error?
No
Yes
Transmission/reception control completed
Reception error control

Caution It is necessary to wait at least 2.5 internal clocks to read the contents of the C2XNOW bit after data is written to C2TXFIFO when operating with a system clock oscillation frequency of 8 MHz.

12.4 Standby Function

The standby function enables the following operations.

(1) HALT mode

If Class 2 is in the wake-up state, normal operation (automatic transmission/reception) is performed. When in the sleep state, normal operation (automatic transmission and reception) starts after the rising edge of the C2RX pin is detected.

(2) STOP mode

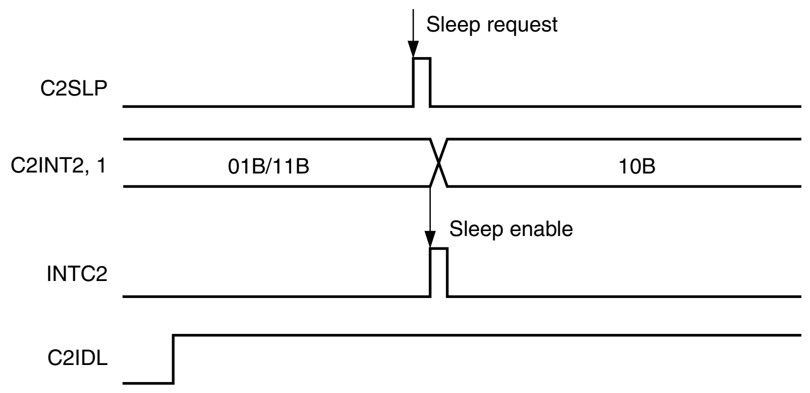
If Class 2 is in the wake-up state, operation stops with each signal level retained. When in the sleep state, operation stops with each signal level retained after the rising edge of the C2RX pin is detected.

The following section explains the sleep and wake-up states.

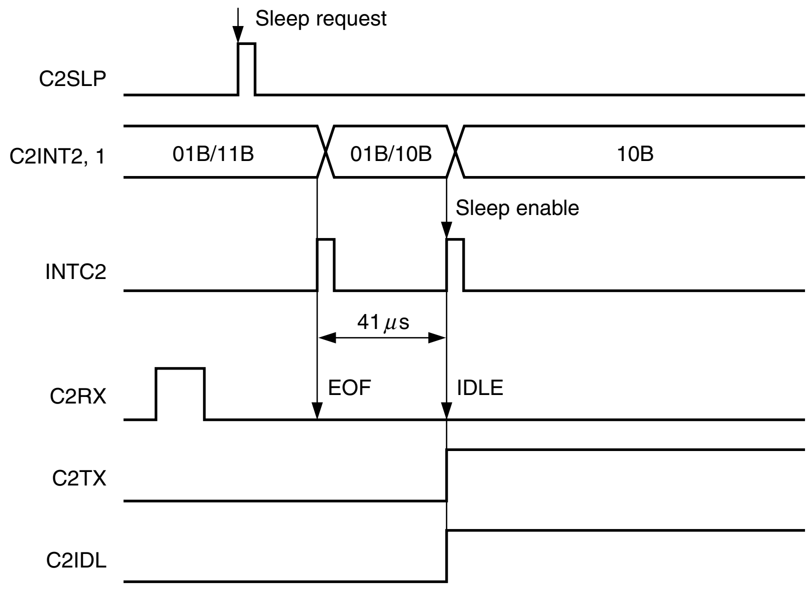
12.4.1 Sleep state

The sleep state is a state in which stopping of the peripheral clock is enabled and the rising edge of the C2RX pin becomes a direct interrupt request signal. By setting C2SLP (1), the sleep state can be requested. When sleep preparations have been made (= when the bus is in the idle state), the interrupt request flags are set to sleep enable (C2INT2, C2INT1 = 10B) and an interrupt request signal (INTC2) is generated.

(1) If a sleep request is issued while the bus is idle

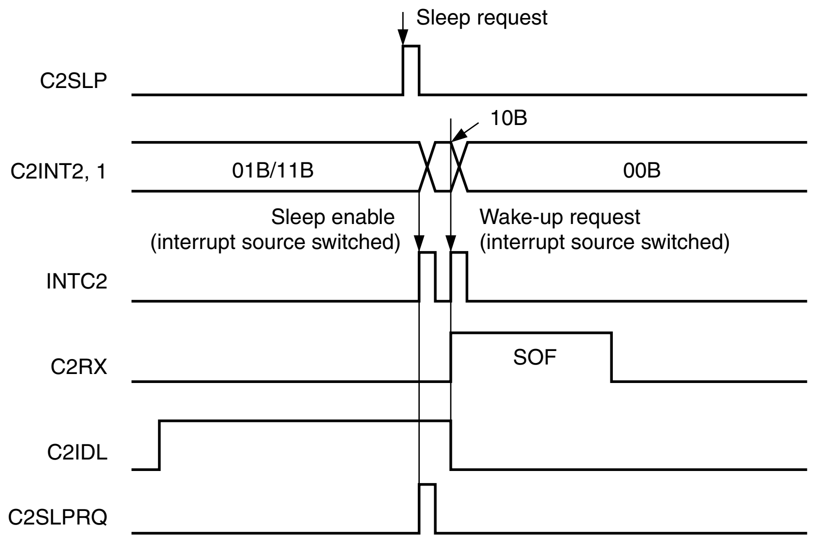


(2) If a sleep request is issued when the bus is busy



Remark	C2INT2, 1 = 01B:	Reception completion
	C2INT2, 1 = 11B:	Reception error
	C2INT2, 1 = 10B:	Sleep enable

(3) If the sleep enable flag cannot be confirmed



Remark	C2INT2, 1 = 01B:	Reception completion
	C2INT2, 1 = 11B:	Reception error
	C2INT2, 1 = 10B:	Sleep enable

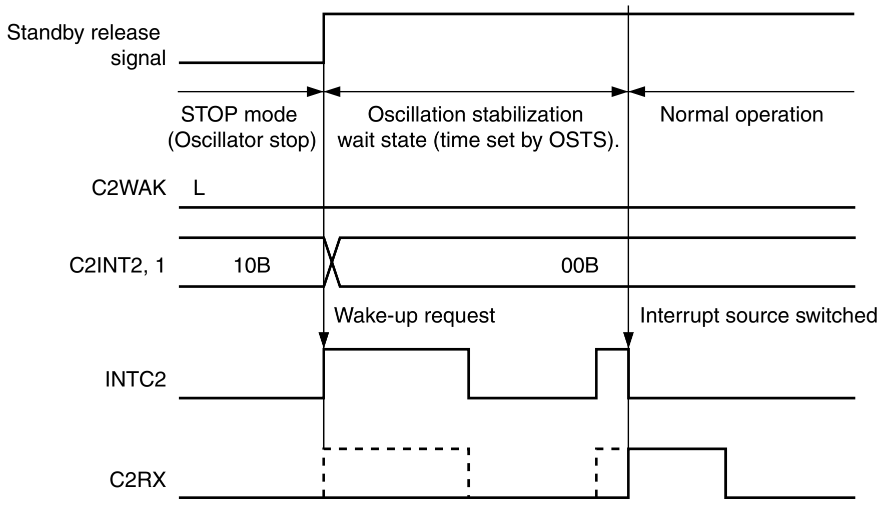
If the bus becomes active immediately after the sleep enable interrupt, it is likely that, when the CPU reads the interrupt request flag, it is already 00B (wake-up requested). In this case, a message can be received normally, but Class 2 will not enter the sleep state, because it has already entered that state once. During sleep request issuance C2SLPRQ becomes 1.

12.4.2 Wake-up state

In the wake-up state, operations are performed normally. In the sleep state, the interrupt request signal (INTC2) is generated by the rising edge of the C2RX pin (the bus signal becomes an interrupt request signal which directly requests wake-up). Even in the sleep state, except in the STOP mode, receiving operations are executed and interrupt request signals (reception completion, reception error, sleep enable) are generated.

If sleep is released by the C2RX pin in the STOP mode, receiving operations cannot be executed during the oscillation stabilization wait state (the time set in the OSTS register) after the standby is canceled by the rising edge of the C2RX pin.

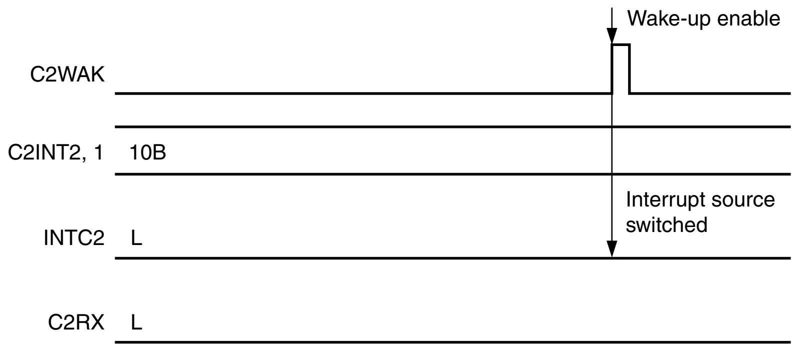
(1) Sleep release operation by the C2RX pin when in the STOP mode


Standby release signal
STOP mode (Oscillator stop)
Oscillation stabilization wait state (time set by OSTS).
Normal operation
C2WAK
L
C2INT2, 1
10B
00B
Wake-up request
Interrupt source switched
INTC2
C2RX

Remark The broken lines indicate the actual bus signal.

OSTS:	Oscillation stabilization time selection register
C2INT2, 1 = 10B:	Sleep enable
C2INT2, 1 = 00B:	Wake-up request

(2) Sleep released by the CPU (when in the HALT mode, normal operation mode)



Remark C2INT2, 1 = 10B: Sleep enable

12.4.3 Differences between the wake-up and sleep states

The difference between the wake-up status and the sleep status is shown below.

Table 12-6. Differences Between Wake-Up and Sleep States

	Wake-Up State	Sleep State
Interrupt request	Reception completion Reception error Sleep enable	Reception completion (except stop mode) Reception error (except stop mode) Sleep enable (except stop mode) Wake-up request (rising edge of C2RX pin)
Reception	Normal reception	Normal reception (except STOP mode)
Transmission	Normal transmission	Normal transmission (except STOP mode)

12.5 Class 2 Cautions

(1) Access restrictions to class 2 control register 1 (C2CT1).

Do not write to C2CT1 more than once within approximately 1.2 μs (system clock @ 4.19 MHz operation). The following malfunction may occur when bits 0, 2 to 4, and 7 of C2CT1 (C2BRK, C2ABTD, C2TXS, C2ECL and C2ABRD) are automatically reset (0).

- If C2CT1 has been written to more than once within 0.48 μs (system clock @ 4.19 MHz operation), unexecuted data is overwritten and the command will become invalid.
- If the auto reset bit is written to within 1.2 μs (system clock @ 4.19 MHz operation), writing may become invalid during the asynchronous reset period.

(2) Access restrictions to class 2 control register 2 (C2CT2)

Do not write to C2CT2 more than once within approximately 1.2 μs (system clock @ 4.19 MHz operation). The following malfunction may occur when bits 0 and 3 (C2WAK, C2SLP) of C2CT2 are reset automatically (0).

- If C2CT2 has been written to more than once within 0.48 μs (system clock @ 4.19 MHz operation), unexecuted data is overwritten and the command will become invalid.
- If the auto reset bit is written to within 1.2 μs (system clock @ 4.19 MHz operation), writing may become invalid during the asynchronous reset period.

(3) Access restrictions to the class 2 transmit FIFO register (C2TXFIFO)

While writing to the transmit FIFO, a write disable timing is generated with a maximum length of approximately 3.8 μs (system clock @ 4.19 MHz operation). During the write disable time, bit 3 (C2XNOW) of class 2 status register 1 (C2ST1) becomes 1. If the C2XNOW bit is set to 1, any attempt to write data results in the previous data being overwritten, which makes it invalid.

(4) Access restrictions to the class 2 receive FIFO register (C2RXFIFO)

Whie reading the receive FIFO, a read disable timing is generated with a maximum length of approximately 3.8 μs (system clock @ 4.19 MHz operation). During the read disable time, bits 6 and 7 (C2RXF1 and C2RXF2) of class 2 status register 1 (C2ST1) become 00B. If the C2RXF2 and C2RXF1 bits are set to 00B, any attempt to read data results in the previous data being read, because the FIFO is not ready to be read.

13.1 Function of Serial Interface

Serial interface UART0 has the following two modes.

(1) Operation stop mode

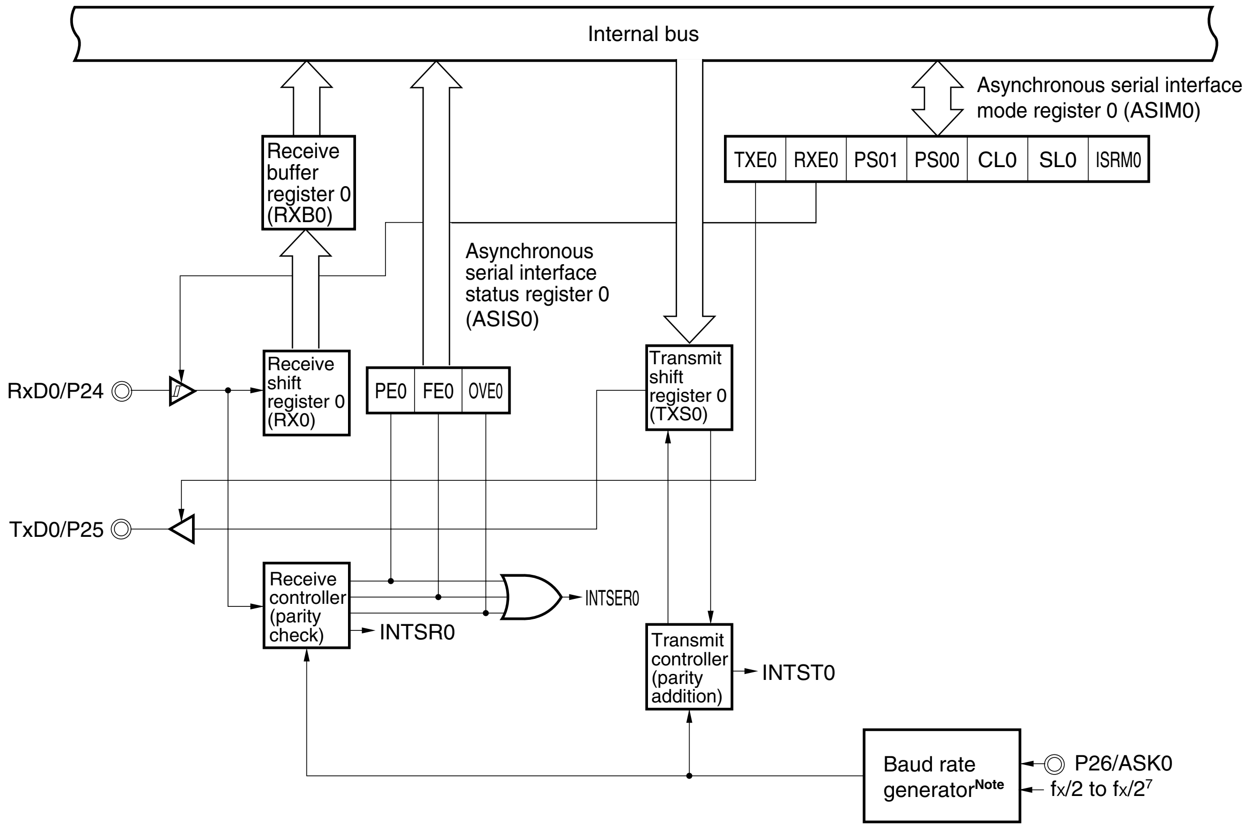
This mode is used to reduce power consumption when serial transfers are not performed.
For details, see **13.4.1 Operation stop mode**.

(2) Asynchronous serial interface (UART) mode

This mode enables full-duplex operation wherein one byte of data after the start bit is transmitted and received.
The on-chip baud rate generator dedicated to UART enables communication using a wide range of selectable baud rates. In addition, a baud rate can also be defined by dividing clocks input to the ASCK0 pin.
The UART baud rate generator can also be used to generate a MIDI-standard baud rate (31.25 Kbps).
For details, see **13.4.2 Asynchronous serial interface (UART) mode**.

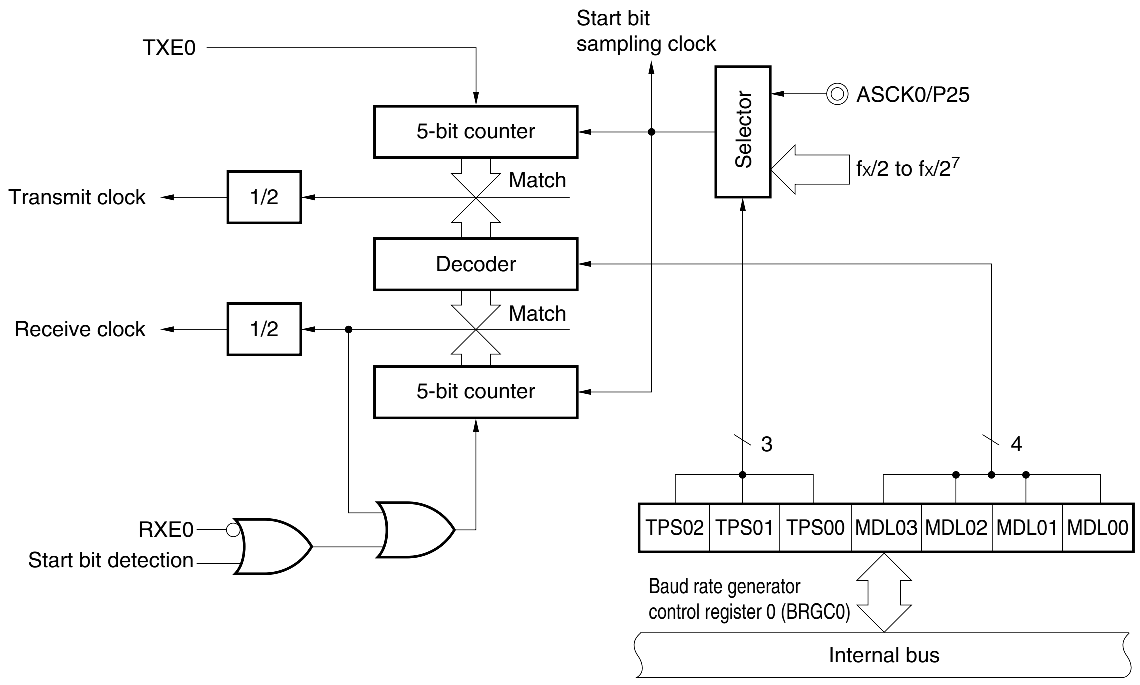
Figure 13-1 shows a block diagram of serial interface UART0.

Figure 13-1. Block Diagram of Serial Interface UART0


Internal bus
Asynchronous serial interface mode register 0 (ASIM0)
TXE0
RXE0
PS01
PS00
CL0
SL0
ISRM0
Receive buffer register 0 (RXB0)
Asynchronous serial interface status register 0 (ASIS0)
RxD0/P24
Receive shift register 0 (RX0)
PE0
FE0
OVE0
Transmit shift register 0 (TXS0)
TxD0/P25
Receive controller (parity check)
INTSER0
INTSR0
Transmit controller (parity addition)
INTST0
Baud rate generator^{Note}
P26/ASK0
fx/2 to fx/2^7

Note For the configuration of the baud rate generator, refer to **Figure 13-2**.

★ **Figure 13-2. Block Diagram of Baud Rate Generator**



Remark TXE0: Bit 7 of asynchronous serial interface mode register 0 (ASIM0)
RXE0: Bit 6 of asynchronous serial interface mode register 0 (ASIM0)

13.2 Configuration of Serial Interface

Serial interface UART0 includes the following hardware.

Table 13-1. Configuration of Serial Interface UART0

Item	Configuration
Registers	Transmit shift register 0 (TXS0) Receive shift register 0 (RX0) Receive buffer register 0 (RXB0)
Control registers	Asynchronous serial interface mode register 0 (ASIM0) Asynchronous serial interface status register 0 (ASIS0) Baud rate generator control register 0 (BRGC0)

(1) Transmit shift register 0 (TXS0)

This is the register for setting transmit data. Data written to TXS0 is transmitted as serial data.
When the data length is set as 7 bits, bits 0 to 6 of the data written to TXS0 are transferred as transmit data.
Writing data to TXS0 starts the transmit operation.
TXS0 can be written using an 8-bit memory manipulation instruction. It cannot be read.
$\overline{\text{RESET}}$ input sets TXS0 to FFH.

Caution Do not write to TXS0 during a transmit operation.
The same address is assigned to TXS0 and receive buffer register 0 (RXB0). A read operation reads values from RXB0.

(2) Receive shift register 0 (RX0)

This register converts serial data input via the RxD0 pin to parallel data. When one byte of data is received at this register, the receive data is transferred to receive buffer register 0 (RXB0).
RX0 cannot be manipulated directly by a program.

(3) Receive buffer register 0 (RXB0)

This register is used to hold receive data. When one byte of data is received, one byte of new receive data is transferred from receive shift register 0 (RX0).
When the data length is set as 7 bits, receive data is sent to bits 0 to 6 of RXB0. In this case, the MSB of RXB0 is always 0.
RXB0 can be read by an 8-bit memory manipulation instruction. It cannot be written to.
$\overline{\text{RESET}}$ input sets RXB0 to FFH.

Caution The same address is assigned to RXB0 and transmit shift register 0 (TXS0). During a write operation, values are written to TXS0.

(4) Transmit controller

The transmit controller controls transmit operations, such as adding a start bit, parity bit, and stop bit to data that is written to transmit shift register 0 (TXS0), based on the values set to asynchronous serial interface mode register 0 (ASIM0).

(5) Receive controller

The receive controller controls receive operations based on the values set to asynchronous serial interface mode register 0 (ASIM0). During a receive operation, it performs error checking, such as for parity errors, and sets various values to asynchronous serial interface status register 0 (ASIS0) according to the type of error that is detected.

13.3 Registers Controlling Serial Interface

The following three registers are used to control serial interface UART0.

- Asynchronous serial interface mode register 0 (ASIM0)
- Asynchronous serial interface status register 0 (ASIS0)
- Baud rate generator control register 0 (BRGC0)

(1) Asynchronous serial interface mode register 0 (ASIM0)

This is an 8-bit register that controls serial transfer operations of serial interface UART0.
ASIM0 is set using a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets ASIM0 to 00H.
Figure 13-3 shows the format of ASIM0.

Caution In UART mode, set the port mode register (PMXX) as follows. Set the output latch to 0.

- **During receive operation**
  Set P24 (RxD0) to input mode (PM24 = 1)
- **During transmit operation**
  Set P25 (TxD0) to output mode (PM25 = 0)
- **During transmit/receive operation**
  Set P24 (RxD0) to input mode, and P25 (RxD0) to output mode

Figure 13-3. Format of Asynchronous Serial Interface Mode Register 0 (ASIM0)

Address: FFA0H After reset: 00H R/W

Symbol	$\boxed{7}$	$\boxed{6}$	5	4	3	2	1	0
ASIM0	TXE0	RXE0	PS01	PS00	CL0	SL0	ISRM0	0

TXE0	RXE0	Operation mode	RxD0/P24 pin function	TxD0/P25 pin function
0	0	Operation stop	Port function (P24)	Port function (P25)
0	1	UART mode (receive only)	Serial function (RxD0)	
1	0	UART mode (transmit only)	Port function (P24)	Serial function (TxD0)
1	1	UART mode (transmit and receive)	Serial function (RxD0)	

PS01	PS00	Parity bit specification
0	0	No parity
0	1	Zero parity always added during transmission No parity detection during reception (parity errors do not occur)
1	0	Odd parity
1	1	Even parity

CL0	Character length specification
0	7 bits
1	8 bits

SL0	Stop bit length specification for transmit data
0	1 bit
1	2 bits

ISRM0	Reception completion interrupt control when error occurs
0	Receive completion interrupt request issued when error occurs
1	Reception completion interrupt request not issued when error occurs

Cautions 1. Be sure to set bit 0 to 0.

2. Do not switch the operation mode until the current serial transmit/receive operation has stopped.

(2) Asynchronous serial interface status register 0 (ASIS0)

When a receive error occurs in UART mode, this register indicates the type of error.

ASIS0 can be read by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ASIS0 to 00H.

Figure 13-4. Format of Asynchronous Serial Interface Status Register 0 (ASIS0)

Address: FFA1H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS0	0	0	0	0	0	PE0	FE0	OVE0

PE0	Parity error flag
0	No parity error
0	Parity error (Incorrect parity bit detected)

FE0	Framing error flag
0	No framing error
1	Framing error$^{\text{Note 1}}$ (Stop bit not detected)

OVE0	Overrun error flag
0	No overrun error
1	Overrun error$^{\text{Note 2}}$ (Next receive operation was completed before data was read from receive buffer register)

Notes 1. Even if the stop bit length is set to two bits by setting bit 2 (SL0) in asynchronous serial interface mode register 0 (ASIM0), stop bit detection during a receive operation only applies to a stop bit length of 1 bit.

2. Be sure to read the contents of receive buffer register 0 (RXB0) when an overrun error has occurred. Until the contents of RXB0 are read, further overrun errors will occur when receiving data.

(3) Baud rate generator control register 0 (BRGC0)

This register sets the serial clock for serial interface.

BRGC0 is set using an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets BRGC0 to 00H.

Figure 13-5 shows the format of BRGC0.

Figure 13-5. Format of Baud Rate Generator Control Register 0 (BRGC0)

Address: FFA2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
BRGC0	0	TPS02	TPS01	TPS00	MDL03	MDL02	MDL01	MDL00

(fx = 4.19 MHz)

TPS02	TPS01	TPS00	Source clock selection for 5-bit counter	n
0	0	0	P26/ASCK0	0
0	0	1	fx/2	1
0	1	0	fx/2^2	2
0	1	1	fx/2^3	3
1	0	0	fx/2^4	4
1	0	1	fx/2^5	5
1	1	0	fx/2^6	6
1	1	1	fx/2^7	7

MDL03	MDL02	MDL01	MDL00	Input clock selection for baud rate generator	k
0	0	0	0	fSCK/16	0
0	0	0	1	fSCK/17	1
0	0	1	0	fSCK/18	2
0	0	1	1	fSCK/19	3
0	1	0	0	fSCK/20	4
0	1	0	1	fSCK/21	5
0	1	1	0	fSCK/22	6
0	1	1	1	fSCK/23	7
1	0	0	0	fSCK/24	8
1	0	0	1	fSCK/25	9
1	0	1	0	fSCK/26	10
1	0	1	1	fSCK/27	11
1	1	0	0	fSCK/28	12
1	1	0	1	fSCK/29	13
1	1	1	0	fSCK/30	14
1	1	1	1	Setting prohibited	—

Cautions 1. Be sure to set bit 7 to 0.

2. Writing to BRGC0 during a communication operation may cause abnormal output from the baud rate generator and disable further communication operations. Therefore, do not write to BRGC0 during a communication operation.

Remarks 1. fSCK: Source clock for 5-bit counter

2. n: Value set via TPS00 to TPS02 ($0 \leq n \leq 7$)

3. k: Value set via MDL00 to MDL03 ($0 \leq k \leq 14$)

13.4 Operation of Serial Interface

This section explains the three modes of serial interface UART0.

13.4.1 Operation stop mode

The power consumption can be reduced because serial transfer is not performed in this mode.
In addition, pins can be used as ordinary ports in this mode.

(1) Register settings

Operation stop mode is set by asynchronous serial interface mode register 0 (ASIM0).
ASIM0 is set using a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets ASIM0 to 00H.

Address: FFA0H After reset: 00H R/W

Symbol	$\boxed{7}$	$\boxed{6}$	5	4	3	2	1	0
ASIM0	TXE0	RXE0	PS01	PS00	CL0	SL0	ISRM0	0

TXE0	RXE0	Operation mode	RxD0/P24 pin function	TxD0/P25 pin function
0	0	Operation stop	Port function (P24)	Port function (P25)
0	1	UART mode (receive only)	Serial function (RxD0)	
1	0	UART mode (transmit only)	Port function (P24)	Serial function (TxD0)
1	1	UART mode (transmit and receive)	Serial function (RxD0)	

Cautions 1. Be sure to set bit 0 to 0.

2. Do not switch the operation mode until the current serial transmit/receive operation has stopped.

13.4.2 Asynchronous serial interface (UART) mode

This mode enables full-duplex operation wherein one byte of data after the start bit is transmitted or received.

The on-chip baud rate generator dedicated to UART enables communication using a wide range of selectable baud rates.

The UART baud rate generator can also be used to generate a MIDI-standard baud rate (31.25 Kbps).

(1) Register settings

UART mode settings are performed by asynchronous serial interface mode register 0 (ASIM0), asynchronous serial interface status register 0 (ASIS0), and baud rate generator control register 0 (BRGC0).

(a) Asynchronous serial interface mode register 0 (ASIM0)

ASIM0 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ASIM0 to 00H.

Caution In UART mode, set the port mode register (PMXX) as follows. Set the output latch to 0.

- **During receive operation**
  Set P24 (RxD0) to input mode (PM24 = 1)
- **During transmit operation**
  Set P25 (TxD0) to output mode (PM25 = 0)
- **During transmit/receive operation**
  Set P24 (RxD0) to input mode, and P25 (TxD0) to output mode

Address: FFA0H After reset: 00H R/W

Symbol	$\boxed{7}$	$\boxed{6}$	5	4	3	2	1	0
ASIM0	TXE0	RXE0	PS01	PS00	CL0	SL0	ISRM0	0

TXE0	RXE0	Operation mode	RxD0/P24 pin function	TxD0/P25 pin function
0	0	Operation stop	Port function (P24)	Port function (P25)
0	1	UART mode (receive only)	Serial function (RxD0)	
1	0	UART mode (transmit only)	Port function (P24)	Serial function (TxD0)
1	1	UART mode (transmit and receive)	Serial function (RxD0)	

PS01	PS00	Parity bit specification
0	0	No parity
0	1	Zero parity always added during transmission No parity detection during reception (parity errors do not occur)
1	0	Odd parity
1	1	Even parity

CL0	Character length specification
0	7 bits
0	8 bits

SL0	Stop bit length specification for transmit data
0	1 bit
1	2 bits

ISRM0	Reception completion interrupt control when error occurs
0	Reception completion interrupt request issued when error occurs
1	Reception completion interrupt request not issued when error occurs

Cautions 1. Be sure to set bit 0 to 0.

2. Do not switch the operation mode until the current serial transmit/receive operation has stopped.

(b) Asynchronous serial interface status register 0 (ASIS0)

ASIS0 can be read by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ASIS0 to 00H.

Address: FFA1H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS0	0	0	0	0	0	PE0	FE0	OVE0

PE0	Parity error flag
0	No parity error
1	Parity error (Incorrect parity bit detected)

FE0	Framing error flag
0	No framing error
1	Framing error^{Note 1} (Stop bit not detected)

OVE0	Overrun error flag
0	No overrun error
1	Overrun error^{Note 2} (Next receive operation was completed before data was read from receive buffer register 0)

Notes 1. Even if the stop bit length is set to two bits by setting bit 2 (SL0) in asynchronous serial interface mode register 0 (ASIM0), stop bit detection during a receive operation only applies to a stop bit length of 1 bit.

2. Be sure to read the contents of receive buffer register 0 (RXB0) when an overrun error has occurred.
Until the contents of RXB0 are read, further overrun errors will occur when receiving data.

(c) Baud rate generator control register 0 (BRGC0)

BRGC0 is set using an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets BRGC0 to 00H.

Address: FFA2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
BRGC0	0	TPS02	TPS01	TPS00	MDL03	MDL02	MDL01	MDL00

(f_X = 4.19 MHz)

TPS02	TPS01	TPS00	Source clock selection for 5-bit counter	n
0	0	0	P26/ASCK0	0
0	0	1	$f_X/2$	1
0	1	0	$f_X/2^2$	2
0	1	1	$f_X/2^3$	3
1	0	0	$f_X/2^4$	4
1	0	1	$f_X/2^5$	5
1	1	0	$f_X/2^6$	6
1	1	1	$f_X/2^7$	7

MDL03	MDL02	MDL01	MDL00	Input clock selection for baud rate generator	k
0	0	0	0	$f_{SCK}/16$	0
0	0	0	1	$f_{SCK}/17$	1
0	0	1	0	$f_{SCK}/18$	2
0	0	1	1	$f_{SCK}/19$	3
0	1	0	0	$f_{SCK}/20$	4
0	1	0	1	$f_{SCK}/21$	5
0	1	1	0	$f_{SCK}/22$	6
0	1	1	1	$f_{SCK}/23$	7
1	0	0	0	$f_{SCK}/24$	8
1	0	0	1	$f_{SCK}/25$	9
1	0	1	0	$f_{SCK}/26$	10
1	0	1	1	$f_{SCK}/27$	11
1	1	0	0	$f_{SCK}/28$	12
1	1	0	1	$f_{SCK}/29$	13
1	1	1	0	$f_{SCK}/30$	14
1	1	1	1	Setting prohibited	—

Cautions 1. Be sure to set bit 7 to 0.

2. Writing to BRGC0 during a communication operation may cause abnormal output from the baud rate generator and disable further communication operations. Therefore, do not write to BRGC0 during a communication operation.

Remarks 1. f_{SCK}: Source clock for 5-bit counter

2. n: Value set via TPS00 to TPS02 ($0 \leq n \leq 7$)

3. k: Value set via MDL00 to MDL03 ($0 \leq k \leq 14$)

The transmit/receive clock that is used to generate the baud rate is obtained by dividing the system clock.

- Transmit/receive clock generation for baud rate by using system clock
  The system clock is divided to generate the transmit/receive clock. The baud rate generated from the system clock is determined according to the following formula.

$$[\text{Baud rate}] = \frac{f_X}{2^{n+1}(k + 16)} \text{ [Hz]}$$

f_X: System clock oscillation frequency

★ When ASCK0 is selected as the source clock of the 5-bit counter, substitute the frequency of the clock input to the ASCK0 pin for f_X in the above expression.

n: Value set via TPS00 to TPS02 ($0 \le n \le 7$)
For details, see **Table 13-2**.

k: Value set via MDL00 to MDL03 ($0 \le k \le 14$)

Table 13-2 shows the relationship between the 5-bit counter's source clock assigned to bits 4 to 6 (TPS00 to TPS02) of BRGC0 and the "n" value in the above formula. Table 13-3 describes the relationship between the system clock and the baud rate.

Table 13-2. Relationship Between 5-Bit Counter's Source Clock and "n" Value

TPS02	TPS01	TPS00	5-Bit Counter's Source Clock Selection	n
0	0	0	P26/ASCK0	0
0	0	1	$f_X/2$	1
0	1	0	$f_X/2^2$	2
0	1	1	$f_X/2^3$	3
1	0	0	$f_X/2^4$	4
1	0	1	$f_X/2^5$	5
1	1	0	$f_X/2^6$	6
1	1	1	$f_X/2^7$	7

Table 13-3. Relationship Between System Clock and Baud Rate

Baud Rate (bps)	f_X = 4.19 MHz	
	BRGC0	ERR (%)
600	7BH	1.14
1200	6BH	1.14
2400	5BH	1.14
4800	4BH	1.14
9600	3BH	1.14
19200	2BH	1.14
31250	21BH	–1.3
38400	1BH	1.14

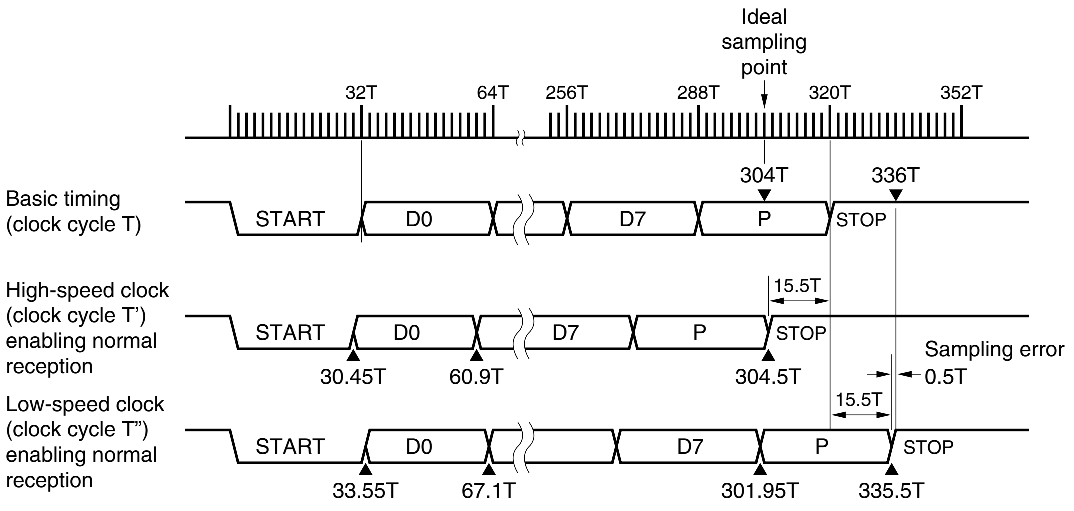
Remark f_X: System clock oscillation frequency
k: Value set via MDL00 to MDL03 ($0 \le k \le 14$)

- **Error tolerance range for baud rate**

  The tolerance range for the baud rate depends on the number of bits per frame and the counter's division rate [1/(16 + k)].

  Figure 13-6 shows an example of a baud rate error tolerance range.

Figure 13-6. Error Tolerance (When k = 0), Including Sampling Errors


Ideal sampling point
32T
64T
256T
288T
320T
352T
304T
336T
Basic timing (clock cycle T)
START
D0
D7
P
STOP
High-speed clock (clock cycle T') enabling normal reception
15.5T
30.45T
60.9T
304.5T
Sampling error 0.5T
Low-speed clock (clock cycle T") enabling normal reception
33.55T
67.1T
301.95T
335.5T

Remark T: 5-bit counter's source clock cycle

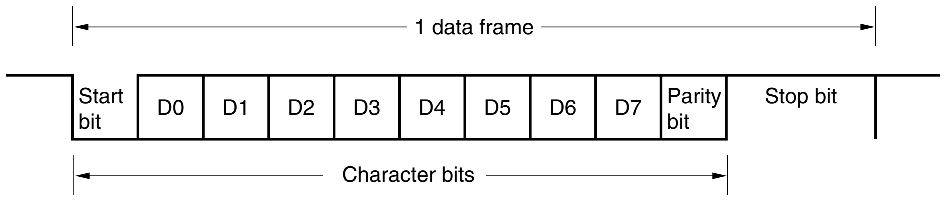
$$\text{Baud rate error tolerance (when k = 0)} = \frac{\pm 15.5}{320} \times 100 = 4.8438\ (\%)$$

(2) Communication operations

(a) Data format

Figure 13-7 shows the format of the transmit/receive data.

Figure 13-7. Format of Transmit/Receive Data in Asynchronous Serial Interface



1 data frame consists of the following bits.

- Start bit .... 1 bit
- Character bits ... 7 or 8 bits
- Parity bit .... Even parity, odd parity, zero parity, or no parity
- Stop bit(s) .... 1 or 2 bits

Asynchronous serial interface mode register 0 (ASIM0) is used to set the character bit length, parity selection, and stop bit length within each data frame.
When “7 bits” is selected as the number of character bits, only the lower 7 bits (bits 0 to 6) are valid, so that during a transmission the highest bit (bit 7) is ignored and during reception the highest bit (bit 7) must be set to 0.
ASIM0 and baud rate generator control register 0 (BRGC0) are used to set the serial transfer rate.
If a receive error occurs, information about the receive error can be ascertained by reading asynchronous serial interface status register 0 (ASIS0).

(b) Parity types and operations

The parity bit is used to detect bit errors in transfer data. Usually, the same type of parity bit is used by the transmitting and receiving sides. When odd parity or even parity is set, errors in the parity bit (the odd-number bit) can be detected. When zero parity or no parity is set, errors are not detected.

(i) Even parity

- During transmission

  The number of bits in transmit data that includes a parity bit is controlled so that there are an even number of bits whose value is 1. The value of the parity bit is as follows.

  If the transmit data contains an odd number of bits whose value is 1: The parity bit is “1”
  If the transmit data contains an even number of bits whose value is 1: The parity bit is “0”

- During reception

  The number of bits whose value is 1 is counted among the transfer data that include a parity bit, and a parity error occurs when the counted result is an odd number.

(ii) Odd parity

- During transmission

  The number of bits in transmit data that includes a parity bit is controlled so that there is an odd number of bits whose value is 1. The value of the parity bit is as follows.

  If the transmit data contains an odd number of bits whose value is 1: The parity bit is “0”
  If the transmit data contains an even number of bits whose value is 1: The parity bit is “1”

- During reception

  The number of bits whose value is 1 is counted among the transfer data that include a parity bit, and a parity error occurs when the counted result is an even number.

(iii) Zero parity

During transmission, the parity bit is set to 0 regardless of the transmit data.

During reception, the parity bit is not checked. Therefore, no parity errors will occur regardless of whether the parity bit is a “0” or a “1”.

(iv) No parity

No parity bit is added to the transmit data.

During reception, receive data is regarded as having no parity bit. Since there is no parity bit, no parity errors will occur.

(c) Transmission

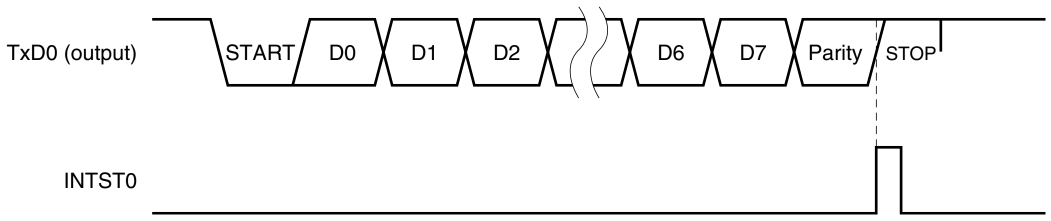
★ A transmit operation is enabled if bit 7 (TXE0) of asynchronous serial interface mode register 0 (ASIM0) is set to 1, and is started when transmit data is written to transmit shift register 0 (TXS0). A start bit, parity bit, and stop bit(s) are automatically added to the data.

Starting the transmit operation shifts out the data in TXS0, thereby emptying TXS0, after which a transmission completion interrupt request (INTST0) is issued.

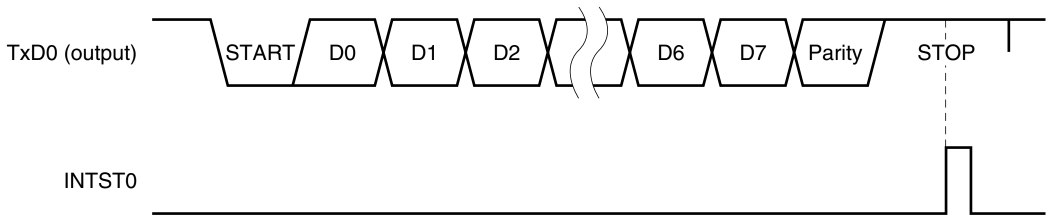
The timing of the transmission completion interrupt request is shown in Figure 13-8.

Figure 13-8. Timing of Asynchronous Serial Interface Transmit Completion Interrupt Request

(i) Stop bit length: 1 bit



(ii) Stop bit length: 2 bits



Caution Do not rewrite to asynchronous serial interface mode register 0 (ASIM0) during a transmit operation. Rewriting ASIM0 during a transmit operation may disable further transmit operations (in such cases, input $\overline{\text{RESET}}$ to restore normal operation).

Whether or not a transmit operation is in progress can be determined via software using the transmission completion interrupt request (INTST0) or the interrupt request flag (STIF0) set by INTST0.

(d) Reception

A receive operation is enabled when bit 6 (RXE0) of asynchronous serial interface mode register 0 (ASIM0) is set to 1, and input via the RxD0 pin is sampled.

The serial clock specified by ASIM0 is used to sample the RxD0 pin.

When the RxD0 pin goes low, the 5-bit counter of the baud rate generator begins counting and the start timing signal for data sampling is output when half of the specified baud rate time has elapsed. If sampling the RxD0 pin input with this start timing signal yields a low-level result, a start bit is recognized, after which the 5-bit counter is initialized and starts counting and data sampling begins. After the start bit is recognized, the character data, parity bit, and one-bit stop bit are detected, at which point reception of one data frame is completed.

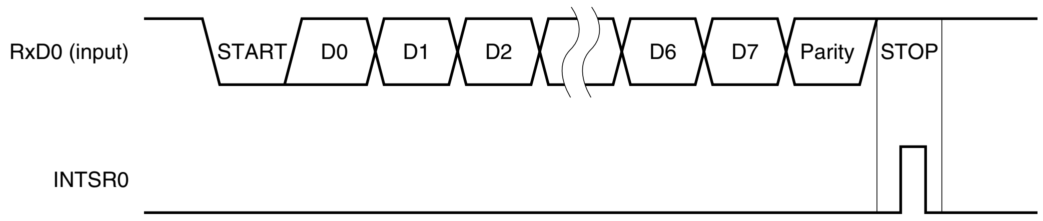
Once reception of one data frame is completed, the receive data in the shift register is transferred to receive buffer register 0 (RXB0) and a reception completion interrupt request (INTSR0) occurs.

Even if an error has occurred, the receive data in which the error occurred is still transferred to RXB0. When bit 1 (ISRM0) of ASIM0 is cleared (0) upon occurrence of an error, INTSR0 occurs (see **Figure 13-10**). When ISRM0 bit is set (1), INTSR0 does not occur.

If the RXE0 bit is reset (0) during a receive operation, the receive operation is stopped immediately. At this time, the contents of RXB0 and ASIS0 do not change, nor does INTSR0 or INTSER0 occur.

Figure 13-9 shows the timing of the asynchronous serial interface reception completion interrupt request.

Figure 13-9. Timing of Asynchronous Serial Interface Reception Completion Interrupt Request



Caution Be sure to read the contents of receive buffer register 0 (RXB0) even when a receive error has occurred. Overrun errors will occur during the next data receive operations and the receive error status will remain until the contents of RXB0 are read.

(e) Receive errors

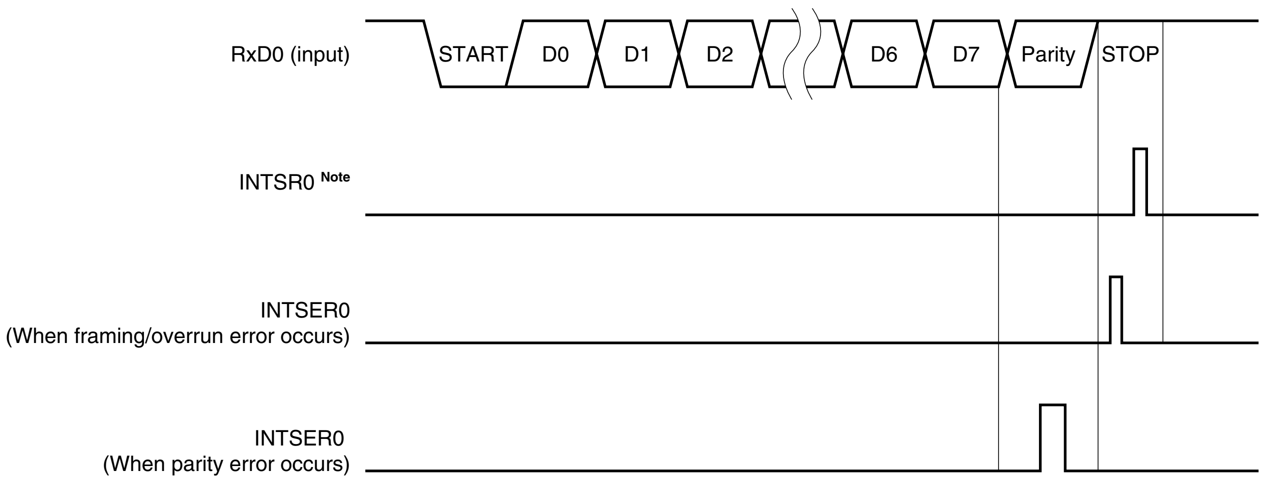
Three types of errors can occur during a receive operation: parity error, framing error, or overrun error. If, as the result of data reception, an error flag is set to asynchronous serial interface status register 0 (ASIS0), a receive error interrupt request (INTSER0) will occur. Receive error interrupts requests are generated before the reception completion interrupt request (INTSR0). Table 13-4 lists the causes behind receive errors.

As part of receive error interrupt request (INTSER0) servicing, the contents of ASIS0 can be read to determine which type of error occurred during the receive operation (see **Table 13-4** and **Figure 13-10**). The contents of ASIS0 are reset (0) when receive buffer register 0 (RXB0) is read or when the next data is received (if the next data contains an error, its error flag will be set).

Table 13-4. Causes of Receive Errors

Receive Error	Cause	ASIS0 Value
Parity error	Parity specified during transmission does not match parity of receive data	04H
Framing error	Stop bit was not detected	02H
Overrun error	Reception of next data was completed before data was read from receive buffer register	01H

Figure 13-10. Receive Error Timing



Note If a receive error occurs when the ISRM0 bit has been set (1), INTSR0 does not occur.

Cautions 1. The contents of asynchronous serial interface status register 0 (ASIS0) are reset (0) when receive buffer register 0 (RXB0) is read or when the next data is received. To obtain information about the error, be sure to read the contents of ASIS0 before reading RXB0.

2. Be sure to read the contents of receive buffer register 0 (RXB0) even when a receive error has occurred. Overrun errors will occur during the next data receive operations and the receive error status will remain until the contents of RXB0 are read.

CHAPTER 14 SERIAL INTERFACE SIO3

Serial interface SIO3 incorporates two 3-wire serial I/O mode channels (SIO30, SIO31).

These two channels have exactly the same functions.

Therefore, unless otherwise specified, SIO30 is used throughout this chapter to describe the functions of both SIO30 and SIO31. If using SIO31, refer to Table 14-1 for the register, bit, and pin names.

Table 14-1. SIO30 and SIO31 Name Differences

Item	SIO30	SIO31
Pins	P30/SI30 P31/SO30 P32/$\overline{\text{SCK30}}$	P20/SI31 P21/SO31 P22/$\overline{\text{SCK31}}$
Serial operation mode register 3	CSIM30	CSIM31
Address of serial operation mode register 3	FFB0H	FFB8H
Bit name of serial operation mode register 3	CSIE30 MODE0 SCL301 SCL300	CSIE31 MODE1 SCL311 SCL310
Serial I/O shift register 3	SIO30	SIO31
Address of serial I/O shift register 3	FF1AH	FF1BH
Interrupt request	INTCSI30	INTCSI31
Interrupt control register and bits mentioned in this chapter	CSIIF30 CSIMK30 CSIPR30	CSIIF31 CSIMK31 CSIPR31

14.1 Function of Serial Interface

Serial interface SIO3 has the following two modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed. For details, see **14.4.1 Operation stop mode**.

(2) 3-wire serial I/O mode (fixed as MSB first)

This is an 8-bit data transfer mode using three lines: a serial clock line ($\overline{\text{SCK30}}$), serial output line (SO30), and serial input line (SI30).

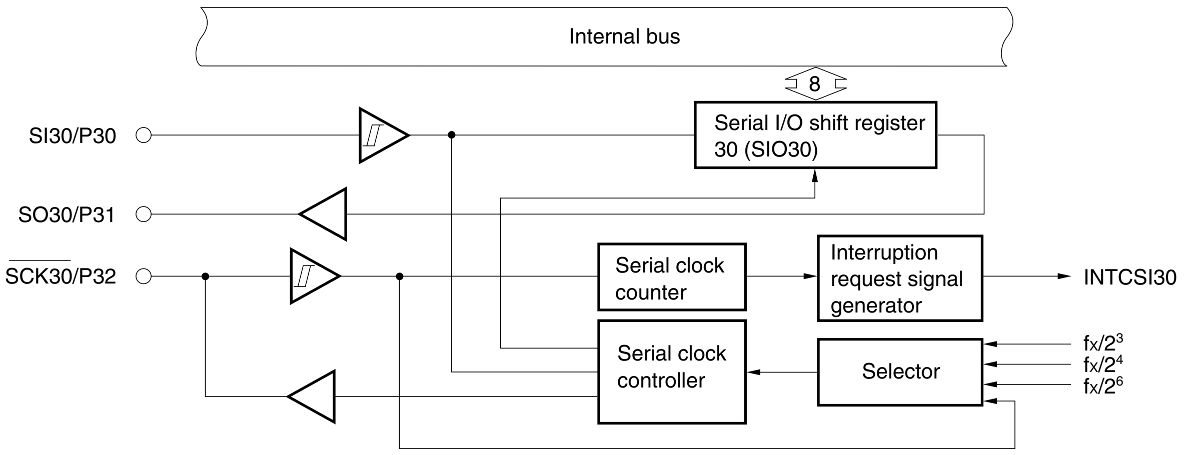
Since simultaneous transmit and receive operations are enabled in 3-wire serial I/O mode, the processing time for data transfers is reduced.

The first bit of the serially transferred 8-bit data is fixed as the MSB.

3-wire serial I/O mode is useful for connection to devices incorporating a clocked serial interface, such as a peripheral I/O or display controller. For details, see **14.4.2 3-wire serial I/O mode**.

Figure 14-1 shows a block diagram of serial interface SIO30.

★ **Figure 14-1. Block Diagram of Serial Interface SIO30**



14.2 Configuration of Serial Interface

Serial interface SIO30 includes the following hardware.

Table 14-2. Configuration of Serial Interface SIO30

Item	Configuration
Registers	Serial I/O shift register 30 (SIO30)
Control registers	Serial operation mode register 30 (CSIM30)

(1) Serial I/O shift register 30 (SIO30)

This is an 8-bit register that performs parallel-serial conversion and serial transmit/receive (shift operations) synchronized with the serial clock.

SIO30 is set using an 8-bit memory manipulation instruction.

When bit 7 (CSIE30) of serial operation mode register 30 (CSIM30) is set to 1, a serial operation can be started by writing data to or reading data from SIO30.

When transmitting, data written to SIO30 is output to the serial output (SO30).

When receiving, data is read from the serial input (SI30) and written to SIO30.

$\overline{\text{RESET}}$ input sets SIO30 to 00H.

Caution Do not access SIO30 during a transmit operation unless the access is triggered by a transfer start. (Read operations are disabled when MODE0 = 0 and write operations are disabled when MODE0 = 1.)

14.3 Register Controlling Serial Interface

The following register is used to control serial interface SIO30.

- Serial operation mode register 30 (CSIM30)

(1) Serial operation mode register 30 (CSIM30)

This register is used to select SIO30 serial clock and operation mode, and enable or disable specific operations.
CSIM30 is set using a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets CSIM30 to 00H.

Caution In 3-wire serial I/O mode, set the port mode register (PMXX) as follows. Set the output latch to 0.
- **During serial clock output (master transmission or master reception)**
  Set P32 ($\overline{\text{SCK30}}$) to output mode (PM32 = 0)
- **During serial clock input (slave transmission or slave reception)**
  Set P32 to input mode (PM32 = 1)
- **In transmit/transmit and receive mode**
  Set P31 (SO30) to output mode (PM31 = 0)
- **In receive mode**
  Set P30 (SI30) to input mode (PM30 = 1)

Figure 14-2. Format of Serial Operation Mode Register 30 (CSIM30)

Address: FFB0H After reset: 00H R/W

Symbol	$\boxed{7}$	6	5	4	3	2	1	0
CSIM30	CSIE30	0	0	0	0	MODE0	SCL301	SCL300

CSIE30	Enable/disable specification for SIO30		
	Shift register operation	Serial counter	Port
0	Operation stop	Clear	Port function[Note 1]
1	Operation enable	Count operation enable	Serial function + port function[Note 2]

MODE0	Transfer operation modes and flags		
	Operation mode	Transfer start trigger	SO30 output
0	Transmit/transmit and receive mode	Write to SIO30	Normal output
1	Receive-only mode	Read from SIO30	Fixed at low level

SCL301	SCL300	Clock selection
0	0	External clock input to $\overline{\text{SCK30}}$
0	1	$f_X/2^3$ (524 kHz)
1	0	$f_X/2^4$ (262 kHz)
1	1	$f_X/2^6$ (65.5 kHz)

Notes 1. When CSIE30 = 0 (SIO30 operation stop status), the SI30, SO30, and $\overline{\text{SCK30}}$ pins can be used for port functions.

2. When CSIE30 = 1 (SIO30 operation enabled state), the SI30 pin can be used as a port pin if only the send function is used, and the SO30 pin can be used as a port pin if only the receive-only mode is used.

Remarks 1. fx: System clock oscillation frequency

2. Figures in parentheses apply to operation with fx = 4.19 MHz.

14.4 Operation of Serial Interface

This section explains the two modes of serial interface SIO3.

14.4.1 Operation stop mode

The power consumption can be reduced because serial transfer is not performed in this mode.
In addition, pins can be used as normal I/O ports in this mode.

(1) Register settings

Operation stop mode is set by serial operation mode register 30 (CSIM30).
CSIM30 is set using a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets CSIM0 to 00H.

Address: FFB0H After reset: 00H R/W

Symbol	$\boxed{7}$	6	5	4	3	2	1	0
CSIM30	CSIE30	0	0	0	0	MODE0	SCL301	SCL300

CSIE30	SIO30 operation enable/disable specification		
	Shift register operation	Serial counter	Port
0	Operation stop	Clear	Port function[Note 1]
1	Operation enable	Count operation enable	Serial function + port function[Note 2]

Notes 1. When CSIE30 = 0 (SIO30 operation stop status), the SI30, SO30, and $\overline{\text{SCK30}}$ pins can be used for port functions.

2. When CSIE30 = 1 (SIO30 operation enabled state), the SI30 pin can be used as a port pin if only the send function is used, and the SO30 pin can be used as a port pin if only the receive-only mode is used.

14.4.2 3-wire serial I/O mode

3-wire serial I/O mode is useful for connection to a device incorporating a clocked serial interface, such as a peripheral I/O or display controller.

This mode executes data transfers via three lines: a serial clock line ($\overline{\text{SCK30}}$), serial output line (SO30), and serial input line (SI30).

(1) Register settings

3-wire serial I/O mode is set by serial operation mode register 30 (CSIM30).

CSIM30 is set using a 1-bit or 8-bit memory manipulation instructions.

$\overline{\text{RESET}}$ input sets CSIM30 to 00H .

Caution In 3-wire serial I/O mode, set the port mode register (PMXX) as follows. Set the output latch to 0.

- **During serial clock output (master transmission or master reception)**
  Set P32 ($\overline{\text{SCK30}}$) to output mode (PM32 = 0)
- **During serial clock input (slave transmission or slave reception)**
  Set P32 to input mode (PM32 = 1)
- **In transmit/transmit and receive mode**
  Set P31 (SO30) to output mode (PM31 = 0)
- **In receive mode**
  Set P30 (SI30) to input mode (PM30 = 1)

Address: FFB0H After reset: 00H R/W

Symbol	$\boxed{7}$	6	5	4	3	2	1	0
CSIM30	CSIE30	0	0	0	0	MODE0	SCL301	SCL300

CSIE30	Enable/disable specification for SIO30		
	Shift register operation	Serial counter	Port
0	Operation stop	Clear	Port function[Note 1]
1	Operation enable	Count operation enable	Serial function + port function[Note 2]

MODE0	Transfer operation modes and flags		
	Operation mode	Transfer start trigger	SO30 output
0	Transmit/transmit and receive mode	Write to SIO30	Normal output
1	Receive-only mode	Read from SIO30	Fixed at low level

SCL301	SCL300	Clock selection
0	0	External clock input to $\overline{\text{SCK30}}$
0	1	$f_X/2^3$ (524 kHz)
1	0	$f_X/2^4$ (262 kHz)
1	1	$f_X/2^6$ (65.5 kHz)

Notes 1. When CSIE30 = 0 (SIO30 operation stop status), the SI30, SO30, and $\overline{\text{SCK30}}$ pins can be used for port functions.

2. When CSIE30 = 1 (SIO30 operation enabled state), the SI30 pin can be used as a port pin if only the send function is used, and the SO30 pin can be used as a port pin if only the receive-only mode is used.

Remarks 1. fx: System clock oscillation frequency

2. Figures in parentheses apply to operation with fx = 4.19 MHz.

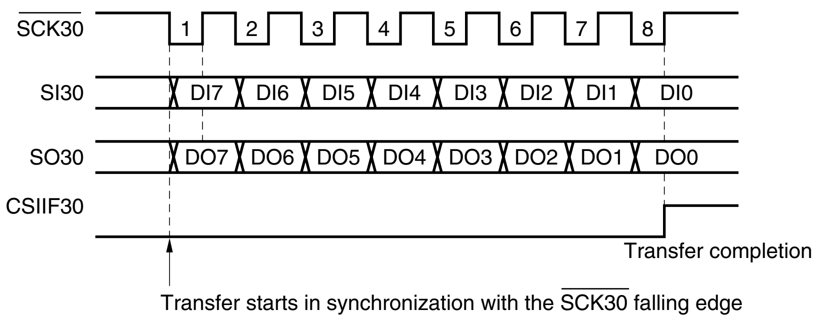
(2) Communication operations

In 3-wire serial I/O mode, data is transmitted and received in 8-bit units. Each bit of data is sent or received in synchronization with the serial clock.

Serial I/O shift register 30 (SIO30) is shifted in synchronization with the falling edge of the serial clock. Transmission data is held in the SO30 latch and is output from the SO30 pin. Data that is received via the SI30 pin in synchronization with the rising edge of the serial clock is latched to SIO30.

Completion of an 8-bit transfer automatically stops operation of SIO30 and sets the interrupt request flag (CSIIF30).

Figure 14-3. Timing of 3-Wire Serial I/O Mode


$\overline{\text{SCK30}}$
1
2
3
4
5
6
7
8
SI30
DI7
DI6
DI5
DI4
DI3
DI2
DI1
DI0
SO30
DO7
DO6
DO5
DO4
DO3
DO2
DO1
DO0
CSIIF30
Transfer completion
Transfer starts in synchronization with the $\overline{\text{SCK30}}$ falling edge

(3) Transfer start

A serial transfer starts when the following two conditions have been satisfied and transfer data has been set (or read) to serial I/O shift register 30 (SIO30).

- The SIO30 operation control bit ($\overline{\text{CSIE30}}$) = 1
- After an 8-bit serial transfer, either the internal serial clock is stopped or SCK30 is set to high level.
- Transmit/transmit and receive mode
  When CSIE30 = 1 and MODE0 = 0, transfer starts when writing to SIO30.
- Receive-only mode
  When CSIE30 = 1 and MODE0 = 1, transfer starts when reading from SIO30.

Caution After data has been written to SIO30, transfer will not start even if the CSIE30 bit value is set to 1.

Completion of an 8-bit transfer automatically stops the serial transfer operation and sets the interrupt request flag (CSIIF30).

CHAPTER 15 SERIAL INTERFACE IIC0

15.1 Function of Serial Interface

Serial interface IIC0 has the following two modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed. It can therefore be used to reduce power consumption.

(2) I^{2}C bus mode (multimaster supported)

This mode is used for 8-bit data transfers with several devices via two lines: a serial clock (SCL0) line and a serial data bus (SDA0) line.

This mode complies with the I^{2}C bus format and can output “start condition”, “data”, and “stop condition” data segments when transmitting via the serial data bus. These data segments are automatically detected by hardware during reception.

Since SCL0 and SDA0 are open-drain outputs, IIC0 requires pull-up resistors for the serial clock line (SCL0) and the serial data bus line (SDA0).

Figure 15-1 shows a block diagram of serial interface IIC0.

Figure 15-1. Block Diagram of Serial Interface IIC0

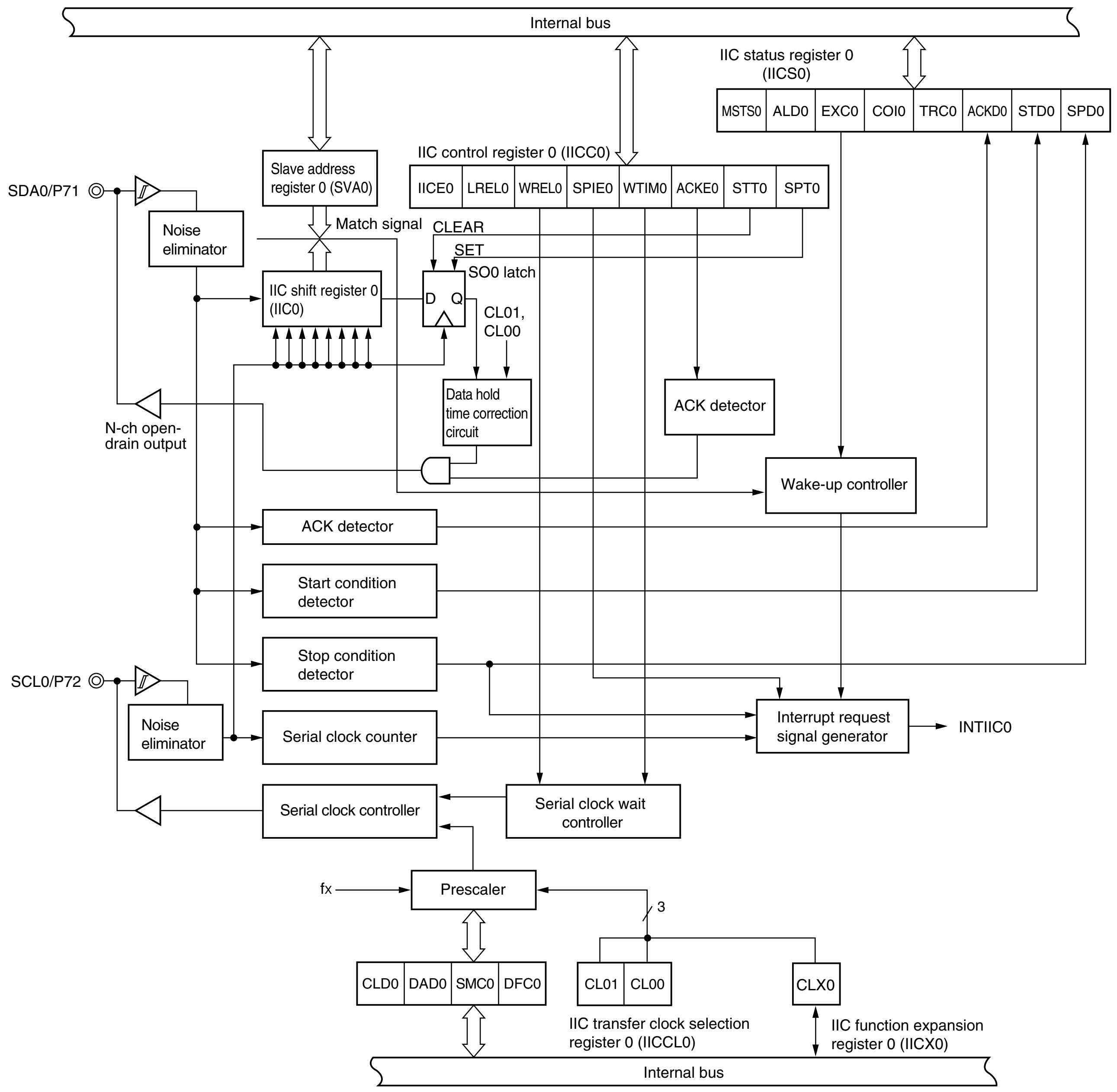
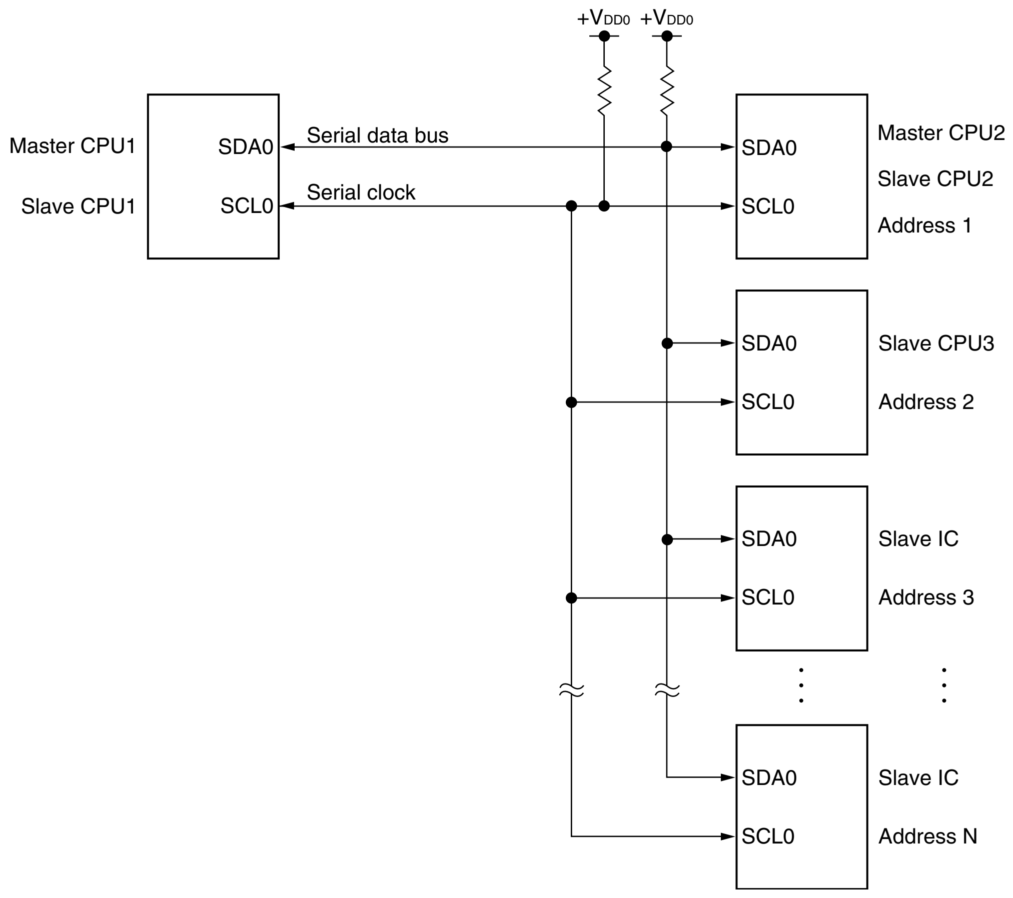


Figure 15-2 shows a serial bus configuration example.

Figure 15-2. Example of Serial Bus Configuration Using I²C Bus



15.2 Configuration of Serial Interface

Serial interface IIC0 includes the following hardware.

Table 15-1. Configuration of Serial Interface IIC0

Item	Configuration
Registers	IIC shift register 0 (IIC0) Slave address register 0 (SVA0)
Control registers	IIC control register 0 (IICC0) IIC status register 0 (IICS0) IIC transfer clock selection register 0 (IICCL0) IIC function expansion register 0 (IICX0)

(1) IIC shift register 0 (IIC0)

IIC0 is used to convert 8-bit serial data to 8-bit parallel data and to convert 8-bit parallel data to 8-bit serial data.
IIC0 can be used for both transmission and reception.
Write and read operations to IIC0 are used to control the actual transmit and receive operations.
IIC0 is set using an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets IIC0 to 00H.

(2) Slave address register 0 (SVA0)

This register sets local addresses when in slave mode.
SVA0 is set using an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets SVA0 to 00H.

(3) SO0 latch

The SO0 latch is used to retain the SDA0 pin's output level.

(4) Wake-up controller

This circuit generates an interrupt request when the address received by this register matches the address value set to slave address register 0 (SVA0) or when an extension code is received.

(5) Clock selector

This circuit selects the sampling clock to be used.

(6) Serial clock counter

This counter counts the serial clocks that are output or input during transmit/receive operations and is used to verify that 8-bit data was sent or received.

(7) Interrupt request signal generator

This circuit controls the generation of interrupt request signals (INTIIC0).
An I^2C interrupt request is generated following either of two triggers.

- Eighth or ninth clock of the serial clock (set by WTIM0 bit) **Note**
- Interrupt request generated when a stop condition is detected (set by SPIE0 bit) **Note**

Note WTIM0: Bit 3 of IIC control register 0 (IICC0)
SPIE0: Bit 4 of IIC control register 0 (IICC0)

(8) Serial clock controller

This circuit generates the clock output via the SCL0 pin from a sampling clock in master mode.

(9) Serial clock wait controller

This circuit controls the wait timing.

(10) ACK output circuit, stop condition detector, start condition detector, and ACK detector

These circuits are used to output and detect various control signals.

(11) Data hold time correction circuit

This circuit generates the hold time for data corresponding to the falling edge of the serial clock.

15.3 Registers Controlling Serial Interface

The following four registers are used to control serial interface IIC0 .

- IIC control register 0 (IICC0)
- IIC status register 0 (IICS0)
- IIC transfer clock selection register 0 (IICCL0)
- IIC function expansion register 0 (IICX0)

The following registers are also used.

- IIC shift register 0 (IIC0)
- Slave address register 0 (SVA0)

(1) IIC control register 0 (IICC0)

This register is used to enable/disable I^{2}C operations, and set the wait timing and other I^{2}C operations.
IICC0 is set using a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets IICC0 to 00H.

Caution In I^{2}C bus mode, set the port mode register (PMXX) as follows. Set the output latch to 0.
- **Set P71 (SDA0) to output mode (PM71 = 0)**
- **Set P72 (SCL0) to output mode (PM72 = 0)**

Figure 15-3. Format of IIC Control Register 0 (IICC0) (1/3)

Address: FFA8H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IICC0	IICE0	LREL0	WREL0	SPIE0	WTIM0	ACKE0	STT0	SPT0

IICE0	I^2C operation enable	
0	Operation stopped. IIC status register 0 (IICS0) preset. Internal operation stopped.	
1	Operation enabled.	
Condition for clearing (IICE0 = 0)		Condition for setting (IICE0 = 1)
• Cleared by instruction • When $\overline{\text{RESET}}$ is input		• Set by instruction

LREL0	Exit from communications	
0	Normal operation	
1	The current communications operation is exited and standby mode is set. This setting is automatically cleared after being executed. Its uses include cases in which a locally irrelevant extension code has been received. The SCL0 and SDA0 lines enter into the high impedance state. The following flags are cleared. • STD0 • ACKD0 • TRC0 • COI0 • EXC0 • MSTS0 • STT0 • SPT0	
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) Note		Condition for setting (LREL0 = 1)
• Automatically cleared after execution • When $\overline{\text{RESET}}$ is input		• Set by instruction

WREL0	Cancel wait	
0	Wait not canceled	
1	Wait canceled. This setting is automatically cleared after wait is canceled.	
If WREL0 is set (wait release) during the wait period of the ninth clock in the transmission status, the SDA0 line becomes high impedance (TRC0 = 0).		
Condition for clearing (WREL0 = 0) Note		Condition for setting (WREL0 = 1)
• Automatically cleared after execution • When $\overline{\text{RESET}}$ is input		• Set by instruction

SPIE0	Enable/disable generation of interrupt request when stop condition is detected	
0	Disabled	
1	Enabled	
Condition for clearing (SPIE0 = 0) Note		Condition for setting (SPIE0 = 1)
• Cleared by instruction • When $\overline{\text{RESET}}$ is input		• Set by instruction

Note This flag's signal is invalid when IICE0 = 0.

★ **Figure 15-3. Format of IIC Control Register 0 (IICC0) (2/3)**

WTIM0	Control of wait and interrupt request generation
0	Interrupt request generated at 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 the master device.
1	Interrupt request generated at 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 the master device.

This bit's setting is invalid during an address transfer and is valid after the 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 ACK signal is issued. 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) Note	Condition for setting (WTIM0 = 1)
• Cleared by instruction • When $\overline{\text{RESET}}$ is input	• Set by instruction

ACKE0	Acknowledge control
0	Acknowledge disabled.
1	Acknowledge enabled. During the ninth clock period, the SDA0 line is set to low level. However, ACK is invalid during address transfers and is valid when EXC0 = 1.

Condition for clearing (ACKE0 = 0) Note	Condition for setting (ACKE0 = 1)
• Cleared by instruction • When $\overline{\text{RESET}}$ is input	• Set by instruction

STT0	Start condition trigger
0	Start condition not generated.
1	When bus is released (during STOP mode): Start condition is generated (for starting as master). The SDA0 line is changed from high level to low level and then the start condition is generated. Next, after the rated amount of time has elapsed, SCL0 is changed to low level. When bus is not used: This trigger functions as a start condition reserve flag. When set, the bus is released and then a start condition is automatically generated. Wait status (in master mode): A restart condition is generated after wait is released.

Cautions concerning set timing
- For master reception: Cannot be set during transfer. Can be set only in the wait period when ACKE0 has been set to 0 and the slave has been notified of final reception.
- For master transmission: A start condition may not be generated normally during the ACK period. Therefore, set it during the wait period.
- Cannot be set at the same time as SPT0

Condition for clearing (STT0 = 0)	Condition for setting (STT0 = 1)
• Cleared by instruction • Cleared by loss in arbitration • Cleared after start condition is generated by master device • Cleared by LREL0 = 1 • When IICE0 = 0 • When $\overline{\text{RESET}}$ is input	• Set by instruction

★ **Figure 15-3. Format of IIC Control Register 0 (IICC0) (3/3)**

SPT0	Stop condition trigger
0	Stop condition not generated.
1	Stop condition generated (termination of master device's transfer). After the SDA0 line goes to low level, either set the SCL0 line to high level or wait until it goes to high level. Next, after the rated amount of time has elapsed, the SDA0 line changes from low level to high level and a stop condition is generated.

Cautions concerning set timing

- For master reception: Cannot be set during transfer. Can be set only in the wait period when ACKE0 has been set to 0 and the slave has been notified of final reception.
- For master transmission: A stop condition cannot be generated normally during the ACK0 period. Therefore, set it during the wait period.
- Cannot be set at the same time as STT0.
- SPT0 can be set only when in master mode.[Note]
- When WTIM0 has been set to 0, if SPT0 is set 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.
  When a ninth clock must be output, WTIM0 should be changed from 0 to 1 during the wait period following output of eight clocks, and SPT0 should be set during the wait period that follows output of the ninth clock.

Condition for clearing (SPT0 = 0)	Condition for setting (SPT0 = 1)
• Cleared by instruction • Cleared by loss in arbitration • Automatically cleared after stop condition is detected • Cleared by LREL0 = 1 • When IICE0 = 0 • When $\overline{\text{RESET}}$ is input	• Set by instruction

Note Set SPT0 only in master mode. However, SPT0 must be set and a stop condition generated before the first stop condition is detected following the switch to the operation enabled status. For details, see **15.5.15 Other cautions**.

Caution When bit 3 (TRC0) of IIC status register 0 (IICS0) is set to 1, WREL0 is set during the ninth clock and wait is canceled, after which TRC0 is cleared and the SDA0 line is set to high impedance.

Remarks 1.

STD0:	Bit 1 of IIC status register 0 (IICS0)
ACKD0:	Bit 2 of IIC status register 0 (IICS0)
TRC0:	Bit 3 of IIC status register 0 (IICS0)
COI0:	Bit 4 of IIC status register 0 (IICS0)
EXC0:	Bit 5 of IIC status register 0 (IICS0)
MSTS0:	Bit 7 of IIC status register 0 (IICS0)

2. Bits 0 and 1 (SPT0, STT0) become 0 when they are read after data is set.

(2) IIC status register 0 (IICS0)

This register indicates the status of the I^2C bus.

IICS0 is set using a 1-bit or 8-bit memory manipulation instruction. IICS0n is a read-only register.

$\overline{\text{RESET}}$ input sets IICS0 to 00H.

Figure 15-4. Format of IIC Status Register 0 (IICS0) (1/3)

Address: FFA9H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
IICS0	MSTS0	ALD0	EXC0	COI0	TRC0	ACKD0	STD0	SPD0

MSTS0	Master device status
0	Slave device status or communication standby status
1	Master device communication status

Condition for clearing (MSTS0 = 0)	Condition for setting (MSTS0 = 1)
• When a stop condition is detected • When ALD0 = 1 • Cleared by LREL0 = 1 • When IICE0 changes from 1 to 0 • When $\overline{\text{RESET}}$ is input	• When a start condition is generated

ALD0	Detection of arbitration loss
0	This status means either that there was no arbitration or that the arbitration result was a “win”.
1	This status indicates the arbitration result was a “loss”. MSTS0 is cleared.

Condition for clearing (ALD0 = 0)	Condition for setting (ALD0 = 1)
• Automatically cleared after IICS0 is read Note • When IICE0 changes from 1 to 0 • When $\overline{\text{RESET}}$ is input	• When the arbitration result is a “loss”.

EXC0	Detection of extension code reception
0	Extension code was not received.
1	Extension code was received.

Condition for clearing (EXC0 = 0)	Condition for setting (EXC0 = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL0 = 1 • When IICE0 changes from 1 to 0 • When $\overline{\text{RESET}}$ is input	• When the higher 4 bits of the received address data is either “0000” or “1111” (set at the rising edge of the eighth clock).

Note This register is also cleared when a bit manipulation instruction is executed for bits other than IICS0.

Figure 15-4. Format of IIC Status Register 0 (IICS0) (2/3)

COI0	Detection of matching addresses
0	Addresses do not match.
1	Addresses match.
Condition for clearing (COI0 = 0)	Condition for setting (COI0 = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL0 = 1 • When IICE0 changes from 1 to 0 • When $\overline{\text{RESET}}$ is input	• When the received address matches the local address (SVA0) (set at the rising edge of the eighth clock).

TRC0	Detection of transmit/receive status
0	Receive status (other than transmit status). The SDA0 line is set to high impedance.
1	Transmit status. The value in the SO0 latch is enabled for output to the SDA0 line (valid starting at the falling edge of the first byte's ninth clock).
Condition for clearing (TRC0 = 0)	Condition for setting (TRC0 = 1)
• When a stop condition is detected • Cleared by LREL0 = 1 • When IICE0 changes from 1 to 0 • Cleared by WREL0 = 1[Note] • When ALD0 changes from 0 to 1 • When $\overline{\text{RESET}}$ is input 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 at the first byte's LSB (transfer direction specification bit) • When not used for communication	Master • When a start condition is generated Slave • When 1 is input at the first byte's LSB (transfer direction specification bit)

ACKD0	Detection of ACK
0	ACK was not detected.
1	ACK 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 • When IICE0 changes from 1 to 0 • When $\overline{\text{RESET}}$ is input	• After the SDA0 line is set to low level at the rising edge of the SCL0's ninth clock

★ **Note** If the wait status is cleared by setting bit 5 (WREL0) of IIC control register 0 (IICC0) to 1 at the ninth clock when bit 3 (TRC0) of IIC status register 0 (IICS0) is 1, TRC0 is cleared, and the SDA0 line enters into a high-impedance state.

Figure 15-4. Format of IIC Status Register 0 (IICS0) (3/3)

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 • When IICE0 changes from 1 to 0 • When $\overline{\text{RESET}}$ is input	• 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 • When $\overline{\text{RESET}}$ is input	• When a stop condition is detected

Remark LREL0: Bit 6 of IIC control register 0 (IICC0)
IICE0: Bit 7 of IIC control register 0 (IICC0)

(3) IIC transfer clock selection register 0 (IICCL0)

This register is used to set the transfer clock for the I^{2}C bus.

IICCL0 is set using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets IICCL0 to 00H.

Figure 15-5. Format of IIC Transfer Clock Selection Register 0 (IICCL0) (1/2)

Address: FFAAH After reset: 00H R/W **Note**

Symbol	7	6	$\boxed{5}$	$\boxed{4}$	3	2	1	0
IICCL0	0	0	CLD0	DAD0	SMC0	DFC0	CL01	CL00

CLD0	Detection of SCL0 line level (valid only when IICE0 = 1)
0	SCL0 line was detected at low level.
1	SCL0 line was detected at high level.

Condition for clearing (CLD0 = 0)	Condition for setting (CLD0 = 1)
• When the SCL0 line is at low level • When IICE0 = 0 • When $\overline{\text{RESET}}$ is input	• When the SCL0 line is at high level

DAD0	Detection of SDA0 line level (valid only when IICE0 = 1)
0	SDA0 line was detected at low level.
1	SDA0 line was detected at high level.

Condition for clearing (DAD0 = 0)	Condition for setting (DAD0 = 1)
• When the SDA0 line is at low level • When IICE0 = 0 • When $\overline{\text{RESET}}$ is input	• When the SDA0 line is at high level

SMC0	Operation mode switching
0	Operation in standard mode
1	Operation in high-speed mode

Condition for clearing (SMC0 = 0)	Condition for setting (SMC0 = 1)
• Cleared by instruction • When $\overline{\text{RESET}}$ is input	• Set by instruction

Note Bits 4 and 5 are read-only bits.

Figure 15-5. Format of IIC Transfer Clock Selection Register 0 (IICCL0) (2/2)

DFC0	Control of digital filter operation[Note]
0	Digital filter off
1	Digital filter on

CL01	CL00	Selection of transfer rate	
		Standard mode	High-speed mode
0	0	f_X/44 (95.2 kHz)	f_X/24 (176 kHz)
0	1	f_X/86 (48.7 kHz)	
1	0	f_X/172 (24.4 kHz)	f_X/48 (setting prohibited)
1	1	f_X/264 (setting prohibited)	f_X/72 (setting prohibited)

Note The digital filter can be used when in high-speed mode. The response time is slower when the digital filter is used.

Caution Stop serial transfer once before rewriting CL01, CL00 to other than the same value.

Remarks 1. IICE0: Bit 7 of IIC control register 0 (IICC0)

2. f_X: System clock oscillation frequency

3. Figures in parentheses apply to operation with f_X = 4.19 MHz.

★ **4.** The transfer clock does not change in the high-speed mode by turning DFC0 on and off.

(4) IIC function expansion register 0 (IICX0)

This register expands the functions of the I^{2}C bus. The transfer frequency speed in high-speed mode is increased by this register.

IICX0 is set using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets IICX0 to 00H.

Figure 15-6. Format of IIC Function Expansion Register 0 (IICX0)

Address: FFA7H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	$\boxed{0}$
IICX0	0	0	0	0	0	0	0	CLX0

CLX0	Transfer clock selection bit
0	Operation using transfer clock set by IICCL0
1	Transfer frequency speed increased in high-speed mode

Caution CLX0 can be set only in high-speed mode (SMC0 = 1)

Remark

IICCL0:	IIC transfer clock selection register 0
SMC0:	Bit 3 of IICC0
CL00, CL01:	Bits 0 and 1 of IICCL0

The selectable transfer clocks are listed in Figure 15-2.

Table 15-2. Selectable Transfer Clocks

CLX0 Note	SMC0	CL01	CL00	Specified Clock	When fx = 4.19 MHz Operation
0	0	0	0	fx/44	95.2 kHz
0	0	0	1	fx/86	48.7 kHz
0	0	1	0	fx/172	24.4 kHz
0	1	0	×	fx/24	176 kHz
1	1	0	×	fx/12	349 kHz
Other than above				Setting prohibited	

Note CLX0 can be set only in high-speed mode (SMC0 = 1).

Remark

SMC0:	Bit 3 of IIC transfer clock selection register 0 (IICCL0)
CL00, CL01:	Bits 0 and 1 of IIC transfer clock selection register 0 (IICCL0)
fx:	System clock oscillation frequency
×:	Don't care

(5) I^{2}C shift register 0 (IIC0)

This register is used for serial transmission/reception (shift operations) synchronized with the serial clock. It can be read from or written to in 8-bit units, but data should not be written to IIC0 during a data transfer.

Address: FF1FH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IIC0								

(6) Slave address register 0 (SVA0)

This register holds the I^{2}C's slave addresses.
It can be read from or written to in 8-bit units, but bit 0 is fixed to 0.

Address: FFABH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SVA0								0

15.4 Function of I²C Bus Mode

15.4.1 Pin configuration

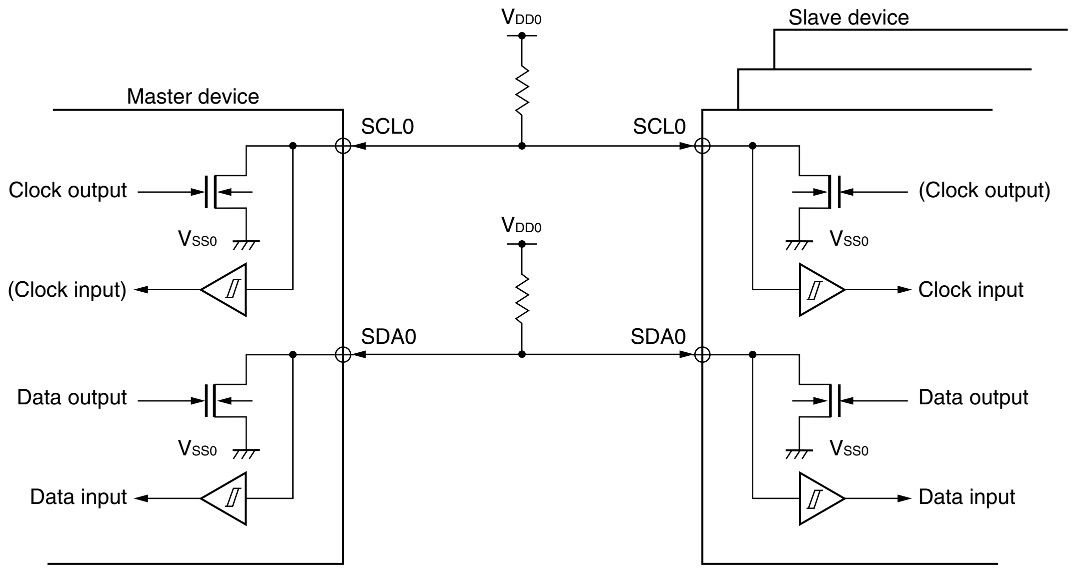
The serial clock pin (SCL0) and serial data bus pin (SDA0) are configured as follows.

(1) SCL0 ········· This pin is used for serial clock input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

(2) SDA0 ········· This pin is used for serial data input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

Since outputs from the serial clock line and the serial data bus line are N-ch open drain outputs, an external pull-up resistor is required.

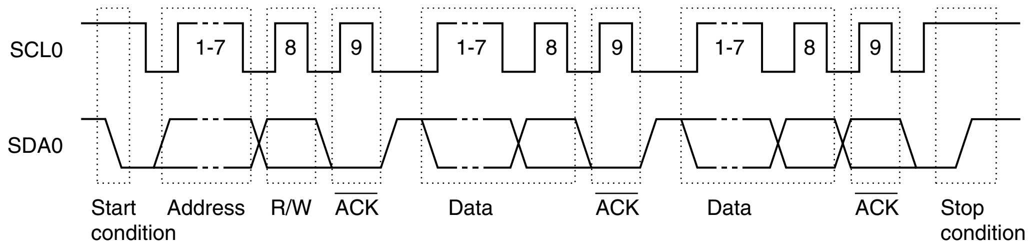
Figure 15-7. Pin Configuration Diagram



15.5 I^{2}C Bus Definitions and Control Methods

The following section describes the I^{2}C bus's serial data communication format and the signals used by the I^{2}C bus. Figure 15-8 shows the transfer timing for the "start condition", "data", and "stop condition" output via the I^{2}C bus's serial data bus.

Figure 15-8. Serial Data Transfer Timing of I^{2}C Bus



The master device outputs the start condition, slave address, and stop condition.

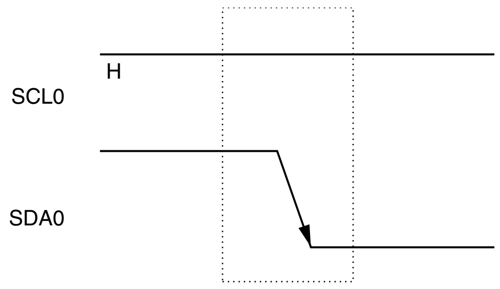
The acknowledge signal ($\overline{\text{ACK}}$) can be output by either the master or slave device (normally, it is output by the device that receives the 8-bit data).

The serial clock (SCL0) is continuously output by the master device. However, in the slave device, the SCL0's low level period can be extended and a wait can be inserted.

15.5.1 Start conditions

A start condition is met when the SCL0 pin is at high level and the SDA0 pin changes from high level to low level. The start conditions for the SCL0 and SDA0 pins are signals that the master device outputs to the slave device when starting a serial transfer. The slave device includes hardware for detecting start conditions.

Figure 15-9. Start Conditions



A start condition is output when bit 1 (STT0) of IIC control register 0 (IICC0) is set (1) after a stop condition has been detected (SPD0: Bit 0 = 1 in IIC status register 0 (IICS0)). When a start condition is detected, bit 1 (STD0) of IICS0 is set (1).

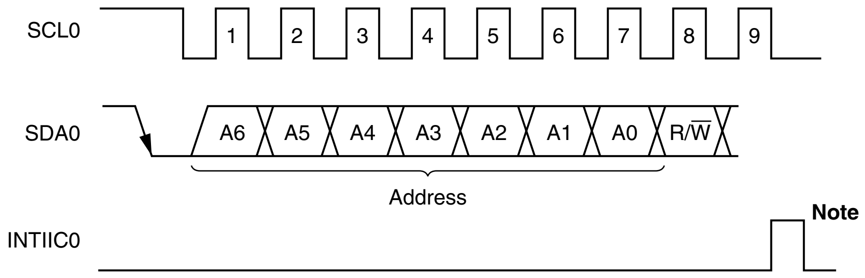
15.5.2 Addresses

The 7 bits of data that follow the start condition are defined as the address.

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 bus lines. Therefore, each slave device connected via the bus lines must have a unique address.

The slave devices include hardware that detects the start condition and checks whether or not the 7-bit address data matches the data values stored in slave address register 0 (SVA0). If the address data matches the SVA0 values, the slave device is selected and communicates with the master device until the master device transmits a start condition or stop condition.

Figure 15-10. Address



Note INTIIC0 is not issued if data other than a local address or extension code is received during slave device operation.

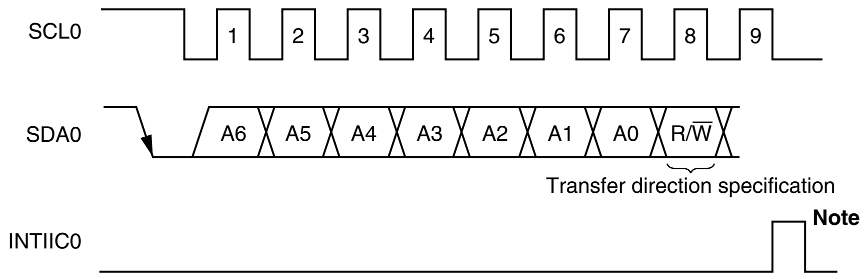
The slave address and the eighth bit, which specifies the transfer direction as described in **15.5.3 Transfer direction specification** below, are written together to IIC shift register 0 (IIC0) and are then output. Received addresses are written to IIC0.

The slave address is assigned to the higher 7 bits of IIC0.

15.5.3 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 15-11. Transfer Direction Specification



Note INTIIC0 is not issued if data other than a local address or extension code is received during slave device operation.

15.5.4 Acknowledge ($\overline{\text{ACK}}$) signal

The acknowledge ($\overline{\text{ACK}}$) signal is used by the transmitting and receiving devices to confirm serial data reception.

The receiving device returns one $\overline{\text{ACK}}$ signal for each 8 bits of data it receives. The transmitting device normally receives an $\overline{\text{ACK}}$ signal after transmitting 8 bits of data. However, when the master device is the receiving device, it does not output an $\overline{\text{ACK}}$ signal after receiving the final data to be transmitted. The transmitting device detects whether or not an $\overline{\text{ACK}}$ signal is returned after it transmits 8 bits of data. When an $\overline{\text{ACK}}$ signal is returned, the reception is judged as normal and processing continues. If the slave device does not return an $\overline{\text{ACK}}$ signal, the master device outputs either a stop condition or a restart condition and then stops the current transmission. Failure to return an $\overline{\text{ACK}}$ signal may be caused by the following two factors.

(a) Reception was not performed normally.
(b) The final data was received.

When the receiving device sets the SDA0 line to low level during the ninth clock, the $\overline{\text{ACK}}$ signal becomes active (normal receive response).

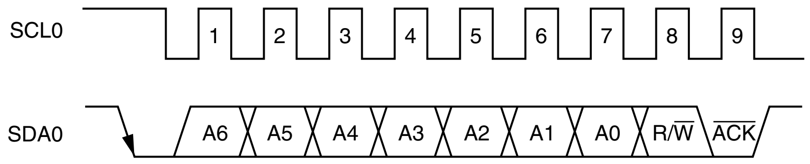
When bit 2 (ACKE0) of IIC control register 0 (IICC0) is set to 1, automatic $\overline{\text{ACK}}$ signal generation is enabled.

Transmission of the eighth bit following the 7 address data bits causes bit 3 (TRC0) of IIC status register 0 (IICS0) to be set. When this TRC0 bit's value is 0, it indicates receive mode. Therefore, ACKE0 should be set to 1.

When the slave device is receiving (when TRC0 = 0), if the slave devices does not need to receive any more data after receiving several bytes, setting ACKE0 to 0 will prevent the master device from starting transmission of the subsequent data.

Similarly, when the master device is receiving (when TRC0 = 0) and the subsequent data is not needed and when either a restart condition or a stop condition should therefore be output, setting ACKE0 to 0 will prevent the $\overline{\text{ACK}}$ signal from being returned. This prevents the MSB data from being output via the SDA line (i.e., stops transmission) during transmission from the slave device.

Figure 15-12. $\overline{\text{ACK}}$ Signal



When the local address is received, an $\overline{\text{ACK}}$ signal is automatically output in synchronization with the falling edge of the SCL's eighth clock regardless of the ACKE0 value. No $\overline{\text{ACK}}$ signal is output if the received address is not a local address.

The $\overline{\text{ACK}}$ signal output method during data reception is based on the wait timing setting, as described below.

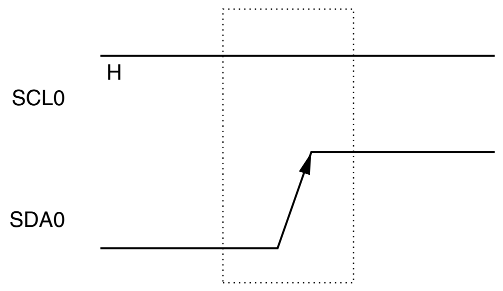
• When 8-clock wait is selected:	$\overline{\text{ACK}}$ signal is output when ACKE0 is set to 1 before wait cancellation.
• When 9-clock wait is selected:	$\overline{\text{ACK}}$ signal is automatically output at the falling edge of the SCL0's 8 clock if ACKE0 has already been set to 1.

15.5.5 Stop condition

When the SCL0 pin is high level, changing the SDA0 pin from low level to high level generates a stop condition.

A stop condition is a signal that the master device outputs to the slave device when serial transfer has been completed. The slave device includes hardware that detects stop conditions.

Figure 15-13. Stop Condition



A stop condition is generated when bit 0 (SPT0) of IIC control register 0 (IICC0) is set (1). When the stop condition is detected, bit 0 (SPD0) of IIC status register 0 (IICS0) is set (1) and INTIIC0 is generated when bit 4 (SPIE0) of IICC0 is set (1).

15.5.6 Wait signal ($\overline{\text{WAIT}}$)

The wait signal ($\overline{\text{WAIT}}$) is used to notify the communication partner that a device (master or slave) is preparing to transmit or receive data (i.e., is in a wait state).

Setting the SCL0 pin to low level notifies the communication partner of the wait status. When wait status has been canceled for both the master and slave devices, the next data transfer can begin.

Figure 15-14. Wait Signal (1/2)

(1) When master device has a 9-clock wait and slave device has an 8-clock wait (master transmits, slave receives, and ACKE0 = 1)

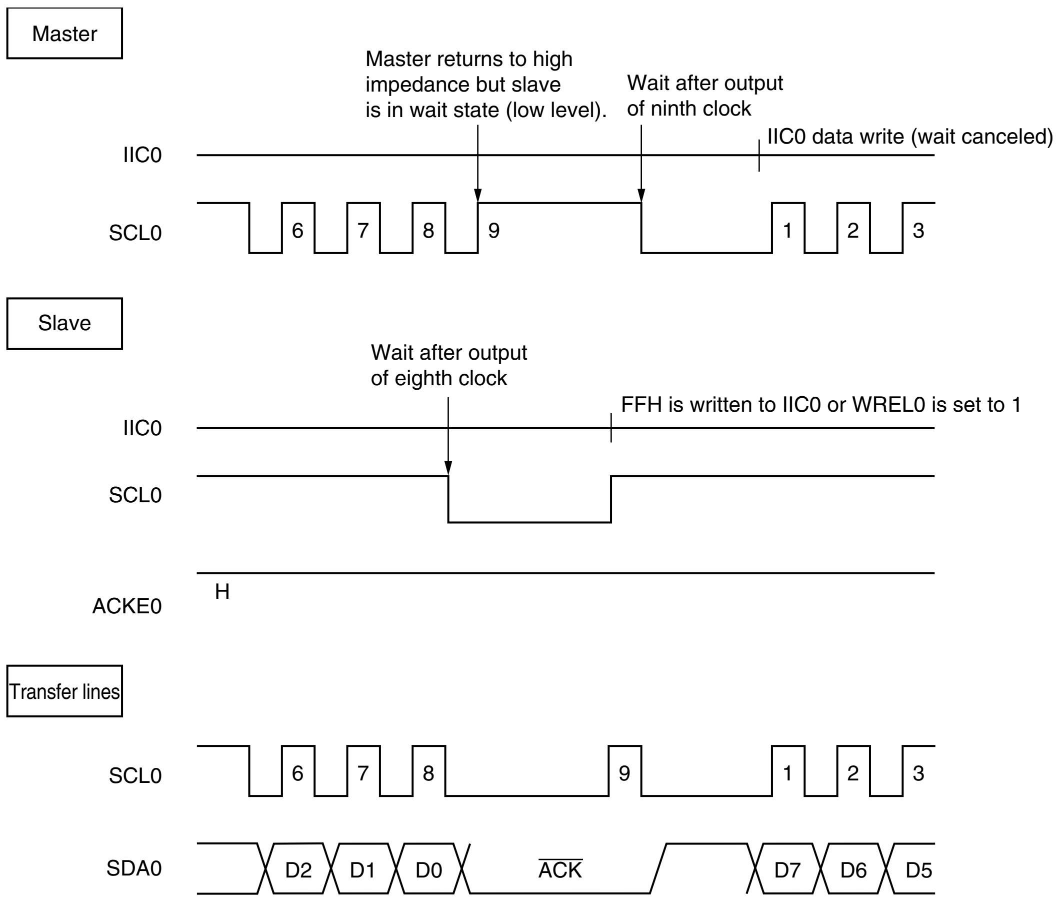
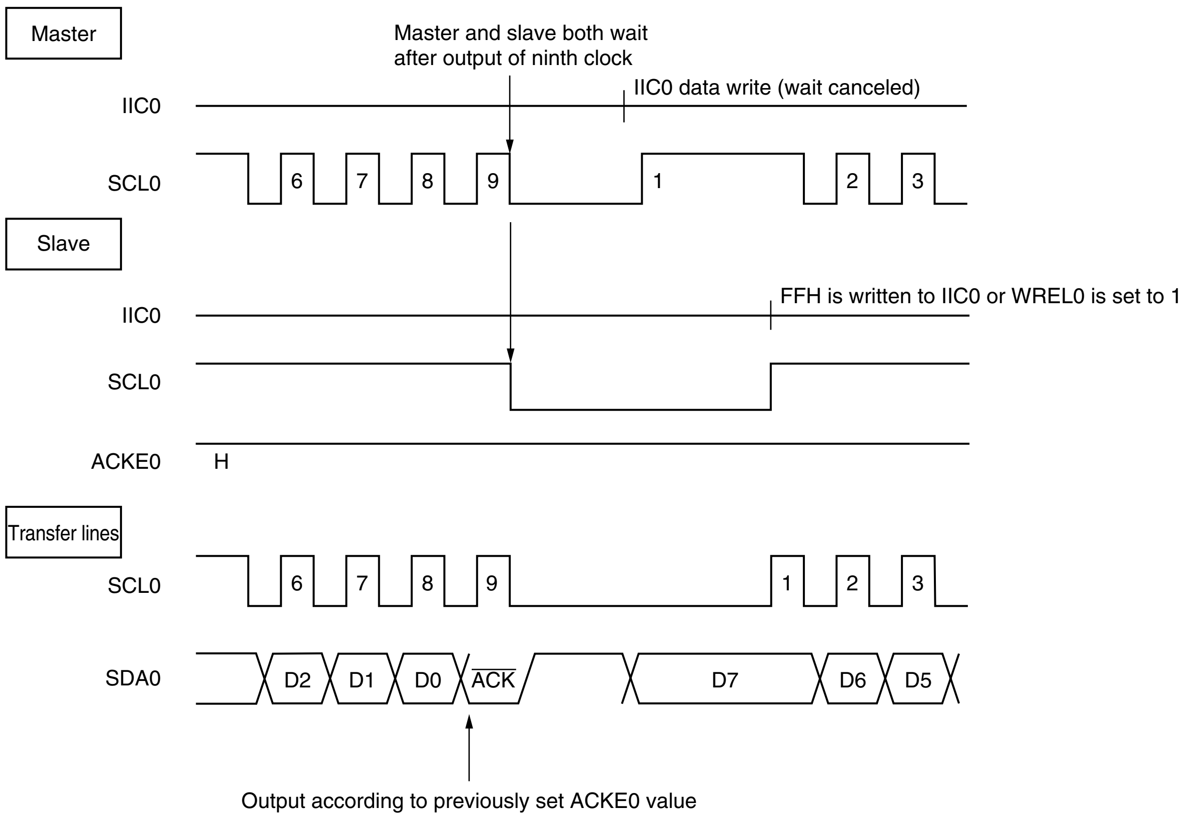

Master
Master returns to high impedance but slave is in wait state (low level).
Wait after output of ninth clock
IIC0 data write (wait canceled)
IIC0
SCL0
6
7
8
9
1
2
3
Slave
Wait after output of eighth clock
FFH is written to IIC0 or WREL0 is set to 1
IIC0
SCL0
ACKE0
H
Transfer lines
SCL0
SDA0
D2
D1
D0
ACK
D7
D6
D5

Figure 15-14. Wait Signal (2/2)

(2) When master and slave devices both have a 9-clock wait (master device transmits, slave receives, and ACKE0 = 1)


Master
Master and slave both wait after output of ninth clock
IIC0
IIC0 data write (wait canceled)
SCL0
6
7
8
9
1
2
3
Slave
IIC0
FFH is written to IIC0 or WREL0 is set to 1
SCL0
ACKE0
H
Transfer lines
SCL0
6
7
8
9
1
2
3
SDA0
D2
D1
D0
ACK
D7
D6
D5
Output according to previously set ACKE0 value

Remark ACKE0: Bit 2 of IIC control register 0 (IICC0)
WREL0: Bit 5 of IIC control register 0 (IICC0)

A wait may be automatically generated in accordance with the setting of bit 3 (WTIM0) of IIC control register 0 (IICC0).

Normally, when bit 5 (WREL0) of IICC0 is set to 1 or when FFH is written to IIC shift register 0 (IIC0), the wait status is canceled and the transmitting side writes data to IIC0 to cancel the wait status.

The master device can also cancel the wait status via either of the following methods.

- By setting bit 1 (STT0) of IICC0 to 1
- By setting bit 0 (SPT0) of IICC0 to 1

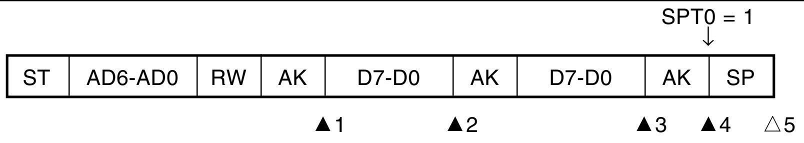
15.5.7 I²C interrupt requests (INTIIC0)

The INTIIC0 interrupt request timing and IIC status register 0 (IICS0) settings corresponding to that timing are described below.

★ **(1) Master device operation**

(a) Start ~ Address ~ Data ~ Data ~ Stop (normal transmission/reception)

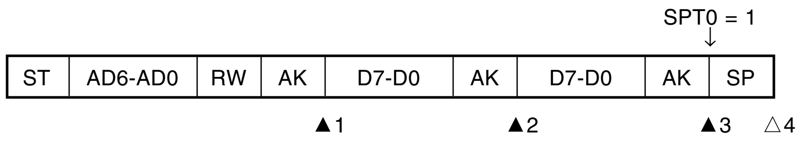
(i) When WTIM0 = 0



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B
▲3: IICS0 = 1000×000B (Sets WTIM0)
▲4: IICS0 = 1000××00B (Sets SPT0)
△5: IICS0 = 00000001B

Remark		
	▲:	Always generated
	△:	Generated only when SPIE0 = 1
	×:	Don't care

(ii) When WTIM0 = 1

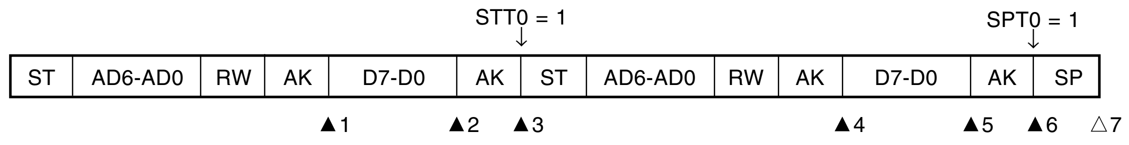


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×100B
▲3: IICS0 = 1000××00B (Sets SPT0)
△4: IICS0 = 00000001B

Remark		
	▲:	Always generated
	△:	Generated only when SPIE0 = 1
	×:	Don't care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

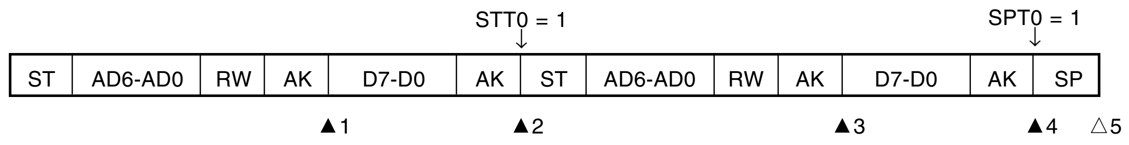
(i) When WTIM0 = 0



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B (Sets WTIM0)
▲3: IICS0 = 1000××00B (Clears WTIM0, sets STT0)
▲4: IICS0 = 1000×110B
▲5: IICS0 = 1000×000B (Sets WTIM0)
▲6: IICS0 = 1000××00B (Sets SPT0)
△7: IICS0 = 00000001B

Remark	▲:	Always generated
	△:	Generated only when SPIE0 = 1
	×:	Don’t care

(ii) When WTIM0 = 1

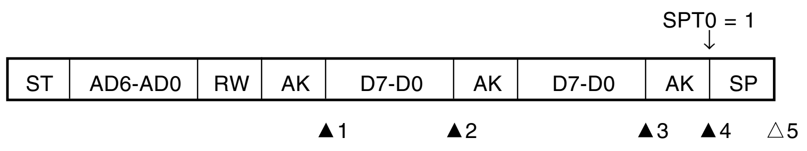


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000××00B (Sets STT0)
▲3: IICS0 = 1000×110B
▲4: IICS0 = 1000××00B (Sets SPT0)
△5: IICS0 = 00000001B

Remark	▲:	Always generated
	△:	Generated only when SPIE0 = 1
	×:	Don’t care

(c) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

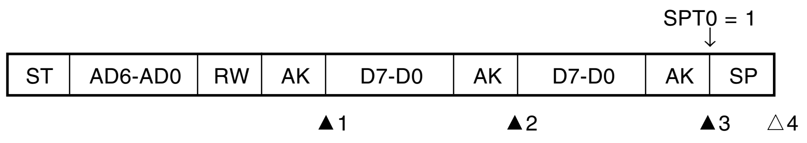
(i) When WTIM0 = 0



▲1: IICS0 = 1010×110B
▲2: IICS0 = 1010×000B
▲3: IICS0 = 1010×000B (Sets WTIM0)
▲4: IICS0 = 1010××00B (Sets SPT0)
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



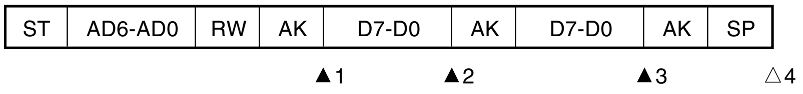
▲1: IICS0 = 1010×110B
▲2: IICS0 = 1010×100B
▲3: IICS0 = 1010××00B (Sets SPT0)
△4: IICS0 = 00001001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(2) Slave device operation (slave address data reception (matches with SVA0))

(a) Start ~ Address ~ Data ~ Data ~ Stop

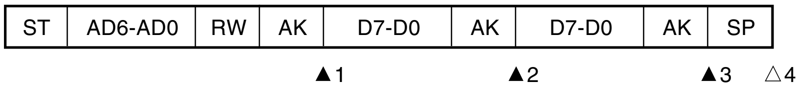
(i) When WTIM0 = 0



▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0001×000B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

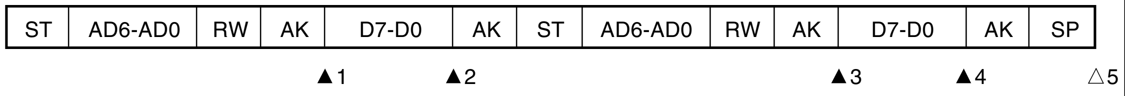


▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×100B
▲3: IICS0 = 0001××00B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

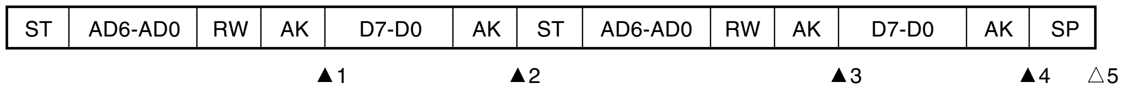
(i) When WTIM0 = 0 (after restart, matches with SVA0)



▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0001×110B
▲4: IICS0 = 0001×000B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, matches with SVA0)

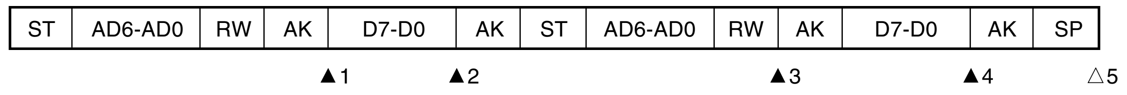


▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001××00B
▲3: IICS0 = 0001×110B
▲4: IICS0 = 0001××00B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

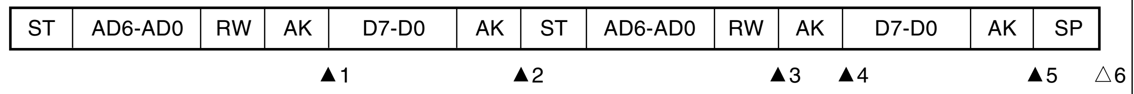
(i) When WTIM0 = 0 (after restart, extension code reception)



▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0010×010B
▲4: IICS0 = 0010×000B
△5: IICS0 = 00000001B

Remark		
	▲:	Always generated
	△:	Generated only when SPIE0 = 1
	×:	Don't care

(ii) When WTIM0 = 1 (after restart, extension code reception)

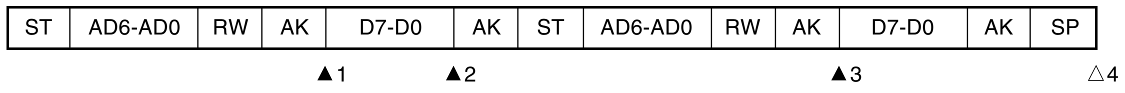


▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001××00B
▲3: IICS0 = 0010×010B
▲4: IICS0 = 0010×110B
▲5: IICS0 = 0010××00B
△6: IICS0 = 00000001B

Remark		
	▲:	Always generated
	△:	Generated only when SPIE0 = 1
	×:	Don't care

(d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIM0 = 0 (after restart, does not match with address (= not extension code))



▲1: IICS0 = 0001×110B

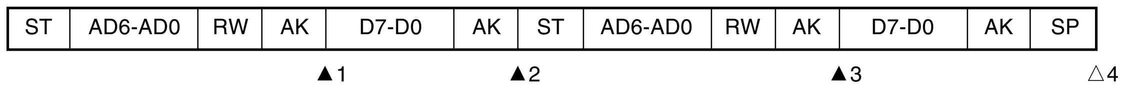
▲2: IICS0 = 0001×000B

▲3: IICS0 = 00000×10B

△4: IICS0 = 00000001B

Remark		
	▲:	Always generated
	△:	Generated only when SPIE0 = 1
	×:	Don't care

(ii) When WTIM0 = 1 (after restart, does not match with address (= not extension code))



▲1: IICS0 = 0001×110B

▲2: IICS0 = 0001××00B

▲3: IICS0 = 00000×10B

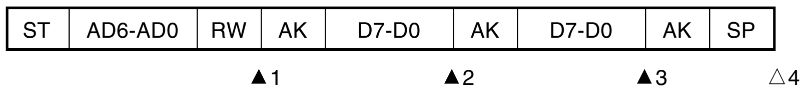
△4: IICS0 = 00000001B

Remark		
	▲:	Always generated
	△:	Generated only when SPIE0 = 1
	×:	Don't care

(3) Slave device operation (when receiving extension code)

(a) Start ~ Code ~ Data ~ Data ~ Stop

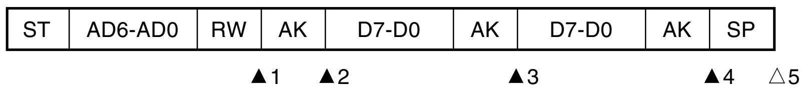
(i) When WTIM0 = 0



▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 0010×000B
△4: IICS0 = 00000001B

Remark		
	▲:	Always generated
	△:	Generated only when SPIE0 = 1
	×:	Don't care

(ii) When WTIM0 = 1

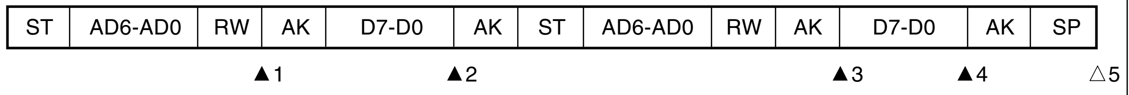


▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010×100B
▲4: IICS0 = 0010××00B
△5: IICS0 = 00000001B

Remark		
	▲:	Always generated
	△:	Generated only when SPIE0 = 1
	×:	Don't care

(b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

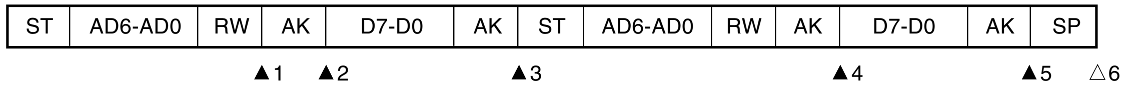
(i) When WTIM0 = 0 (after restart, matches with SVA0n)



▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 0001×110B
▲4: IICS0 = 0001×000B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(ii) When WTIM0 = 1 (after restart, matches with SVA0)

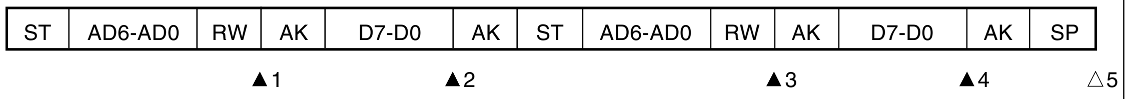


▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010××00B
▲4: IICS0 = 0001×110B
▲5: IICS0 = 0001××00B
△6: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

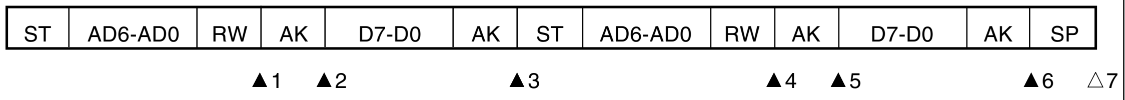
(i) When WTIM0 = 0 (after restart, extension code reception)



▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 0010×010B
▲4: IICS0 = 0010×000B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, extension code reception)

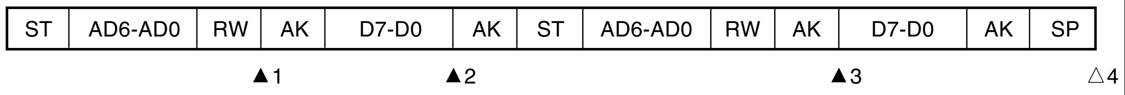


▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010××00B
▲4: IICS0 = 0010×010B
▲5: IICS0 = 0010×110B
▲6: IICS0 = 0010××00B
△7: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

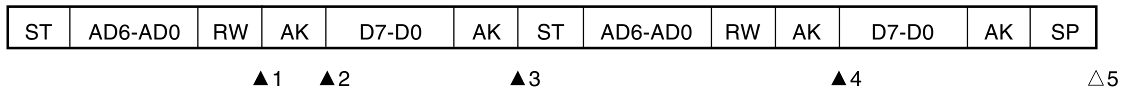
(i) When WTIM0 = 0 (after restart, does not match with address (= not extension code))



▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 00000×10B
△4: IICS0 = 00000001B

Remark	▲:	Always generated
	△:	Generated only when SPIE0 = 1
	×:	Don’t care

(ii) When WTIM0 = 1 (after restart, does not match with address (= not extension code))

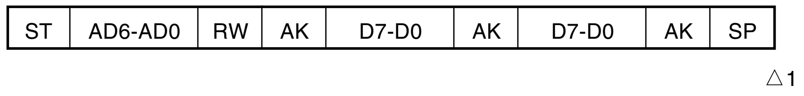


▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010××00B
▲4: IICS0 = 00000×10B
△5: IICS0 = 00000001B

Remark	▲:	Always generated
	△:	Generated only when SPIE0 = 1
	×:	Don’t care

(4) Operation without communication

(a) Start ~ Code ~ Data ~ Data ~ Stop



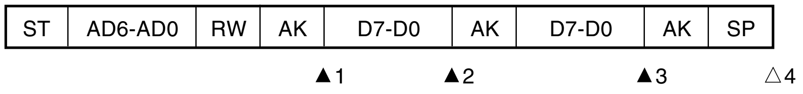
△1: IICS0 = 00000001B

Remark △: Generated only when SPIE0 = 1

(5) Arbitration loss operation (operation as slave after arbitration loss)

(a) When arbitration loss occurs during transmission of slave address data

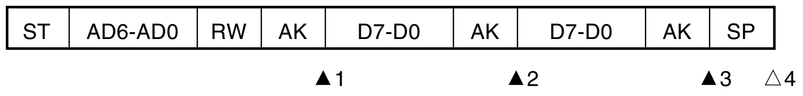
(i) When WTIM0 = 0



▲1: IICS0 = 0101×110B (**Example** When ALD0 is read during interrupt servicing)
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0001×000B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(ii) When WTIM0 = 1

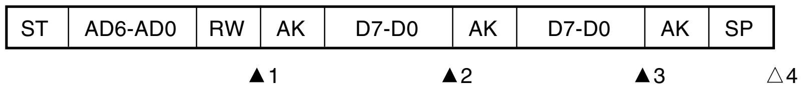


▲1: IICS0 = 0101×110B (**Example** When ALD0 is read during interrupt servicing)
▲2: IICS0 = 0001×100B
▲3: IICS0 = 0001××00B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(b) When arbitration loss occurs during transmission of extension code

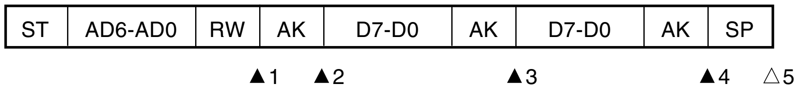
(i) When WTIM0 = 0



▲1: IICS0 = 0110×010B (**Example** When ALD0 is read during interrupt servicing)
▲2: IICS0 = 0010×000B
▲3: IICS0 = 0010×000B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

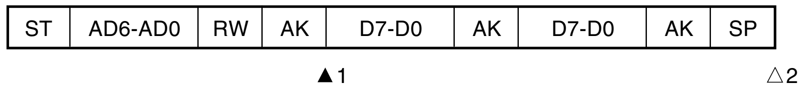


▲1: IICS0 = 0110×010B (**Example** When ALD0 is read during interrupt servicing)
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010×100B
▲4: IICS0 = 0010××00B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(6) Operation when arbitration loss occurs (no communication after arbitration loss)

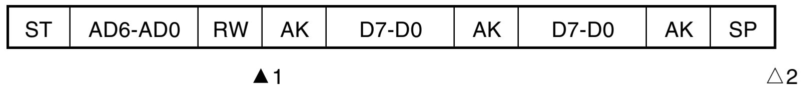
(a) When arbitration loss occurs during transmission of slave address data (when WTIM0 = 1)



▲1: IICS0 = 01000110B (**Example** When ALD0 is read during interrupt servicing)
△2: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1

(b) When arbitration loss occurs during transmission of extension code



▲1: IICS0 = 0110×010B (**Example** When ALD0 is read during interrupt servicing)
LREL0 is set to 1 by software
△2: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(c) When arbitration loss occurs during data transfer

(i) When WTIM0 = 0

ST	AD6-AD0	RW	AK	D7-D0	AK	D7-D0	AK	SP
			▲1	▲2				△3

▲1: IICS0 = 10001110B
▲2: IICS0 = 01000000B (**Example** When ALD0 is read during interrupt servicing)
△3: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1

(ii) When WTIM0 = 1

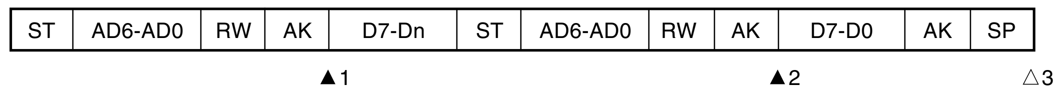
ST	AD6-AD0	RW	AK	D7-D0	AK	D7-D0	AK	SP
			▲1		▲2			△3

▲1: IICS0 = 10001110B
▲2: IICS0 = 01000100B (**Example** When ALD0 is read during interrupt servicing)
△3: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1

(d) When loss occurs due to restart condition during data transfer

(i) Not extension code (Example: Does not match with SVA0 or WTIM0 = 1)



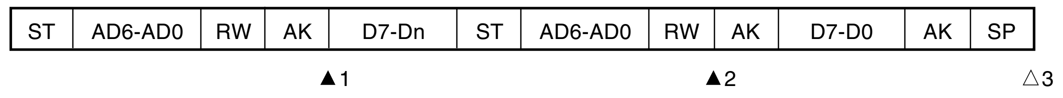
▲1: IICS0 = 1000×110B

▲2: IICS0 = 01000110B (**Example** When ALD0 is read during interrupt servicing)

△3: IICS0 = 00000001B

Remark	▲:	Always generated
	△:	Generated only when SPIE0 = 1
	×:	Don’t care
	n = 6 to 0	

(ii) Extension code



▲1: IICS0 = 1000×110B

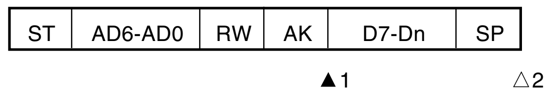
▲2: IICS0 = 0110×010B (**Example** When ALD0 is read during interrupt servicing)

Set LREL0 = 1

△3: IICS0 = 00000001B

Remark	▲:	Always generated
	△:	Generated only when SPIE0 = 1
	×:	Don’t care
	n = 6 to 0	

(e) When loss occurs due to stop condition during data transfer

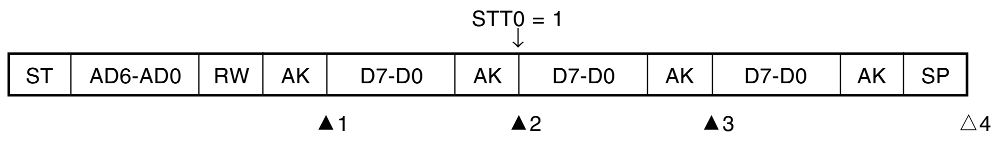


▲1: IICS0 = 1000×110B

△2: IICS0 = 01000001B

Remark	▲:	Always generated
	△:	Generated only when SPIE0 = 1
	×:	Don’t care
	n = 6 to 0	

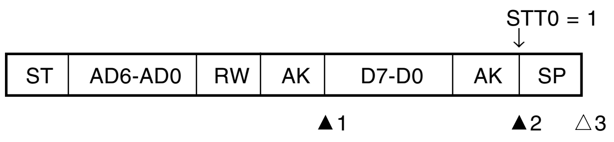
★ **(f) When arbitration loss occurs due to low-level data when attempting to generate a restart condition (When WTIM0 = 1)**



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×100B (Sets STT0)
▲3: IICS0 = 01000100B (**Example** When ALD0 is read during interrupt servicing)
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

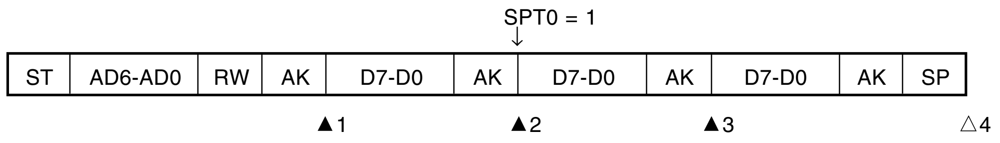
★ **(g) When arbitration loss occurs due to a stop condition when attempting to generate a restart condition (When WTIM0 = 1)**



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000××00B (Sets STT0)
△3: IICS0 = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

★ **(h) When arbitration loss occurs due to low-level data when attempting to generate a stop condition (When WTIM0 = 0)**



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000××00B (Sets SPT0)
▲3: IICS0 = 01000000B (**Example** When ALD0 is read during interrupt servicing)
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

15.5.8 Interrupt request (INTIIC0) generation timing and wait control

The setting of bit 3 (WTIM0) in IIC control register 0 (IICC0) determines the timing by which INTIIC0 is generated and the corresponding wait control, as shown in Table 15-3.

Table 15-3. INTIIC0 Timing and Wait Control

WTIM	During Slave Device Operation			During Master Device Operation		
	Address	Data Reception	Data Transmission	Address	Data Reception	Data Transmission
0	9^Notes 1, 2^	8^Note 2^	8^Note 2^	9	8	8
1	9^Notes 1, 2^	9^Note 2^	9^Note 2^	9	9	9

Notes 1. The slave device's INTIIC0 signal and wait period occurs at the falling edge of the ninth clock only when there is a match with the address set to slave address register 0 (SVA0).
At this point, $\overline{\text{ACK}}$ is output regardless of the value set to bit 2 (ACKE0) of IICC0. For a slave device that has received an extension code, INTIIC0 occurs at the falling edge of the eighth clock.

2. If the received address does not match the contents of slave address register 0 (SVA0), neither INTIIC0 nor a wait occurs.

Remark The numbers in the table indicate the number of serial clock 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: The interrupt and wait timing are determined regardless of the WTIM0 bit.
- Master device operation: The 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: The interrupt and wait timing are determined according to the WTIM0 bit.

(3) During data transmission

- Master/slave device operation: The interrupt and wait timing are determined according to the WTIM0 bit.

(4) Wait cancellation method

The four wait cancellation methods are as follows.

- By setting bit 5 (WREL0) of IIC control register 0 (IICC0) to 1
- By writing to IIC shift register 0 (IIC0)
- By setting a start condition (setting bit 1 (STT0) of IICC0 to 1)
- By setting a stop condition (setting bit 0 (SPT0) of IICC0 to 1)

When an 8-clock wait has been selected (WTIM0 = 0), the output level of $\overline{\text{ACK}}$ must be determined prior to wait cancellation.

(5) Stop condition detection

INTIIC0 is generated when a stop condition is detected.

15.5.9 Address match detection method

When in I^{2}C bus mode, the master device can select a particular slave device by transmitting the corresponding slave address.

An address match can be detected automatically by hardware. An interrupt request (INTIIC0) occurs when a local address has been set to slave address register 0 (SVA0) and when the address set to SVA0 matches the slave address sent by the master device, or when an extension code has been received.

15.5.10 Error detection

In I^{2}C bus mode, the status of the serial data bus (SDA0) during data transmission is captured by IIC shift register 0 (IIC0) of the transmitting device, so the IIC0 data prior to transmission can be compared with the transmitted IIC0 data to enable detection of transmission errors. A transmission error is judged as having occurred when the compared data values do not match.

15.5.11 Extension code

(1) When the higher 4 bits of the receive address are either “0000” or “1111”, the extension code flag (EXC0) is set for extension code reception and an interrupt request (INTIIC0) is issued at the falling edge of the eighth clock. The local address stored in slave address register 0 (SVA0) is not affected.

(2) If SVA0 is set to “111110××” by a 10-bit address transfer and “111110××0” is transferred from the master device, the results are as follows. Note that INTIIC0 occurs at the falling edge of the eighth clock.

- Higher 4 bits of data match: EXC0 = 1**Note**
- Seven bits of data match: COI0 = 1**Note**

Note EXC0: Bit 5 of IIC status register 0 (IICS0)
COI0: Bit 4 of IIC status register 0 (IICS0)

(3) Since the processing after the interrupt request occurs differs according to the data that follows the extension code, such processing is performed by software.
For example, after the extension code is received, if you do not wish to operate the target device as a slave device, you can set bit 6 (LREL0) of IIC control register 0 (IICC0) to 1 to set the standby mode for the next communication operation.

Table 15-4. Extension Code Bit Definitions

Slave Address	R/W Bit	Description
0000 000	0	General call address
0000 000	1	Start byte
0000 001	×	CBUS address
0000 010	×	Address reserved for different bus format
1111 0××	×	10-bit slave address specification

15.5.12 Arbitration

When several master devices simultaneously output a start condition (when STT0 is set to 1 before STD0 is set to 1[Note]), 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, the arbitration loss flag (ALD0) in IIC status register 0 (IICS0) is set via the timing at which the arbitration loss occurred, and the SCL0 and SDA0 lines are both set to high impedance, which releases the bus.

The arbitration loss is detected based on the timing of the next interrupt request (the eighth or ninth clock, when a stop condition is detected, etc.) and the ALD0 = 1 setting that has been made by software.

For details of interrupt request timing, see **15.5.7 I^{2}C interrupt requests (INTIIC0)**.

Note STD0: Bit 1 of IIC status register 0 (IICS0)
STT0: Bit 1 of IIC control register 0 (IICC0)

Figure 15-15. Arbitration Timing Example

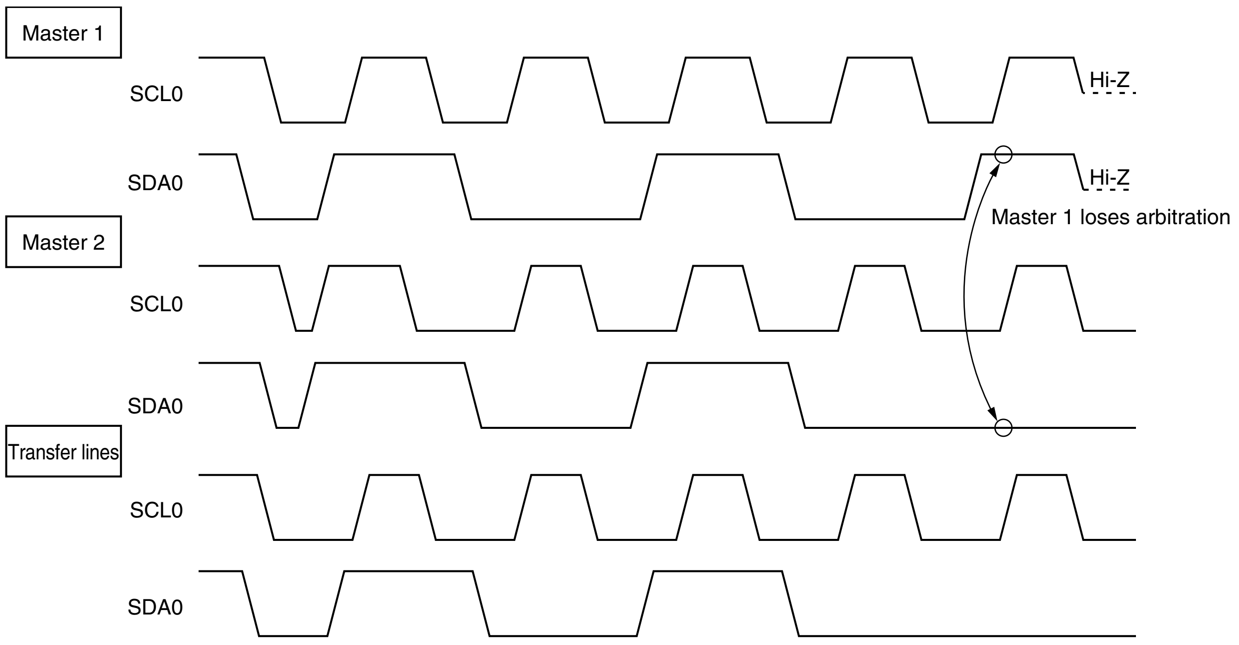


Table 15-5. Status During Arbitration and Interrupt Request Generation Timing

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 $\overline{\text{ACK}}$ signal 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 output (when SPIE0 = 1)[Note 2]
When data is low level while attempting to output a restart condition	At falling edge of eighth or ninth clock following byte transfer[Note 1]
When stop condition is detected while attempting to output a restart condition	When stop condition is output (when SPIE0 = 1)[Note 2]
When data is low level while attempting to output a stop condition	At falling edge of eighth or ninth clock following byte transfer[Note 1]
When SCL0 is low level while attempting to output a restart condition	

Notes 1. When WTIM0 (bit 3 of IIC control register 0 (IICC0)) = 1, an interrupt request occurs at the falling edge of the ninth clock. When WTIM0 = 0 and the extension code's slave address is received, an interrupt request occurs at the falling edge of the eighth clock.

2. When there is a chance that arbitration will occur, set SPIE0 = 1 for master device operation.

Remark SPIE0: Bit 5 of IIC control register 0 (IICC0)

15.5.13 Wake-up function

The I^2C bus slave function is a function that generates an interrupt request (INTIIC0) when a local address and extension code have been received.

This function makes processing more efficient by preventing unnecessary interrupt requests from occurring when addresses do not match.

When a start condition is detected, wake-up standby mode is set. This wake-up standby mode is in effect while addresses are transmitted due to the possibility that an arbitration loss may change the master device (which has output a start condition) to a slave device.

However, when a stop condition is detected, bit 5 (SPIE0) of IIC control register 0 (IICC0) is set regardless of the wake-up function, and this determines whether interrupt requests are enabled or disabled.

15.5.14 Communication reservation

To start master device communications when not currently using the bus, a communication reservation can be made to enable transmission of a start condition when the bus is released. There are two modes in 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 ($\overline{\text{ACK}}$ is not returned and the bus was released when bit 6 (LREL0) of IIC control register 0 (IICC0) was set to 1).

If bit 1 (STT0) of IICC0 is set while the bus is released (after a stop condition has been detected), a start condition is automatically generated and the wait status is set.

When the bus release is detected (when the stop condition is detected), writing to IIC shift register 0 (IIC0) causes the master to start address transfer. At this point, bit 4 (SPIE0) of IICC0 should be set.

When STT0 has been set, 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 using MSTS0 (bit 7 of IIC status register 0 (IICS0)) after SST0 is set and the wait time elapses.

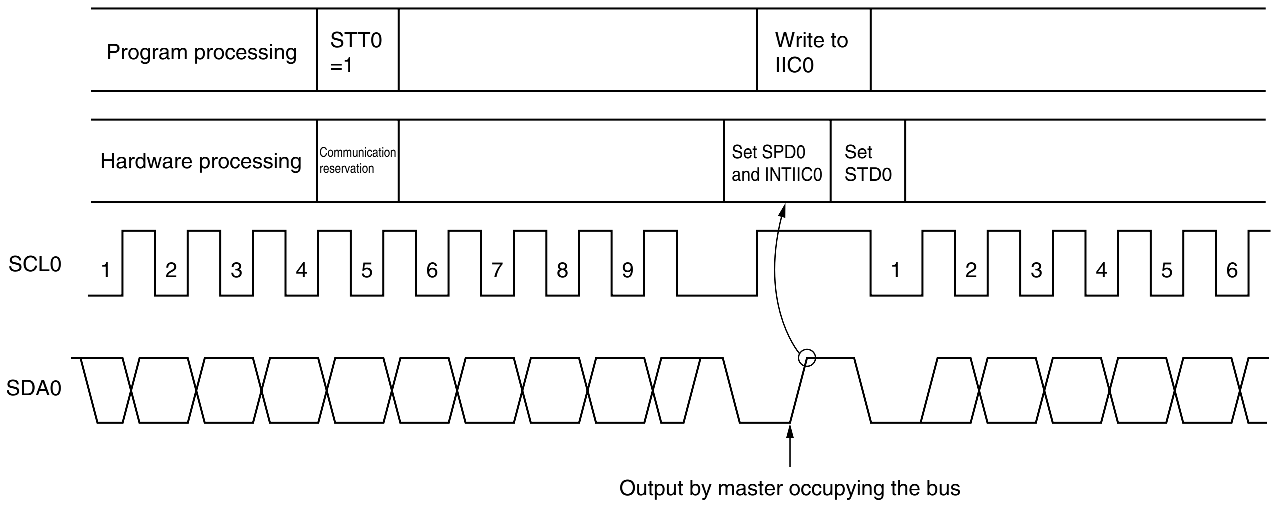
Wait periods, which should be set via software, are listed in Table 15-5. These wait periods can be specified via the settings for bits 3, 1, and 0 (SMC0, CL01, and CL00) in IIC clock selection register 0 (IICCL0).

Table 15-6. Wait Periods

SMC0	CL01	CL00	Wait Period
0	0	0	26 clocks × $1/f_X$
0	0	1	46 clocks × $1/f_X$
0	1	0	
0	1	1	37 clocks × $1/f_X$
1	0	0	16 clocks × $1/f_X$
1	0	1	
1	1	0	32 clocks × $1/f_X$
1	1	1	13 clocks × $1/f_X$

Figure 15-16 shows the communication reservation timing.

Figure 15-16. Communication Reservation Timing



Remark IIC0: IIC shift register 0
STT0: Bit 1 of IIC control register 0 (IICC0)
STD0: Bit 1 of IIC status register 0 (IICS0)
SPD0: Bit 0 of IIC status register 0 (IICS0)

Communication reservations are acknowledged via the following timing. After bit 1 (STD0) of IIC status register 0 (IICS0) is set to 1, a communication reservation can be made by setting bit 1 (STT0) of IIC control register 0 (IICC0) to 1 before a stop condition is detected.

Figure 15-17. Timing for Acknowledging Communication Reservations

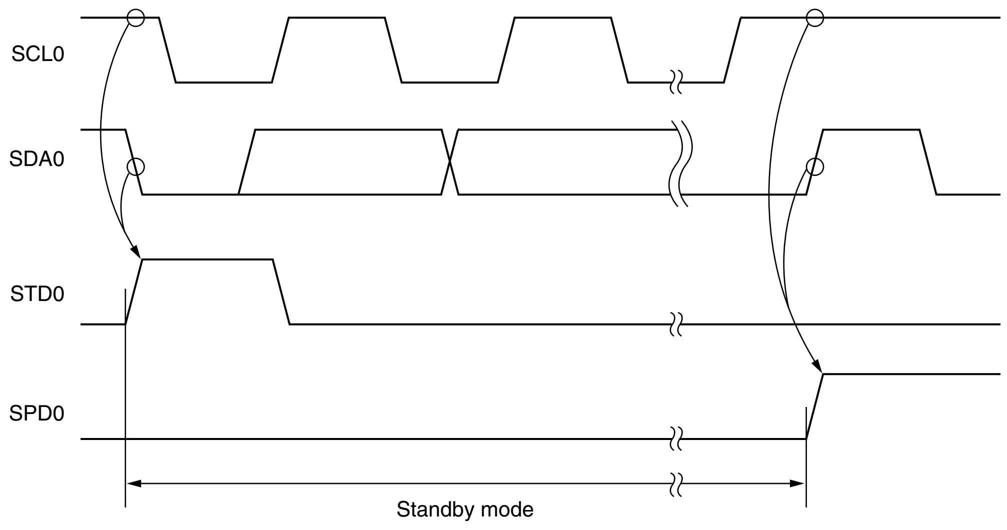
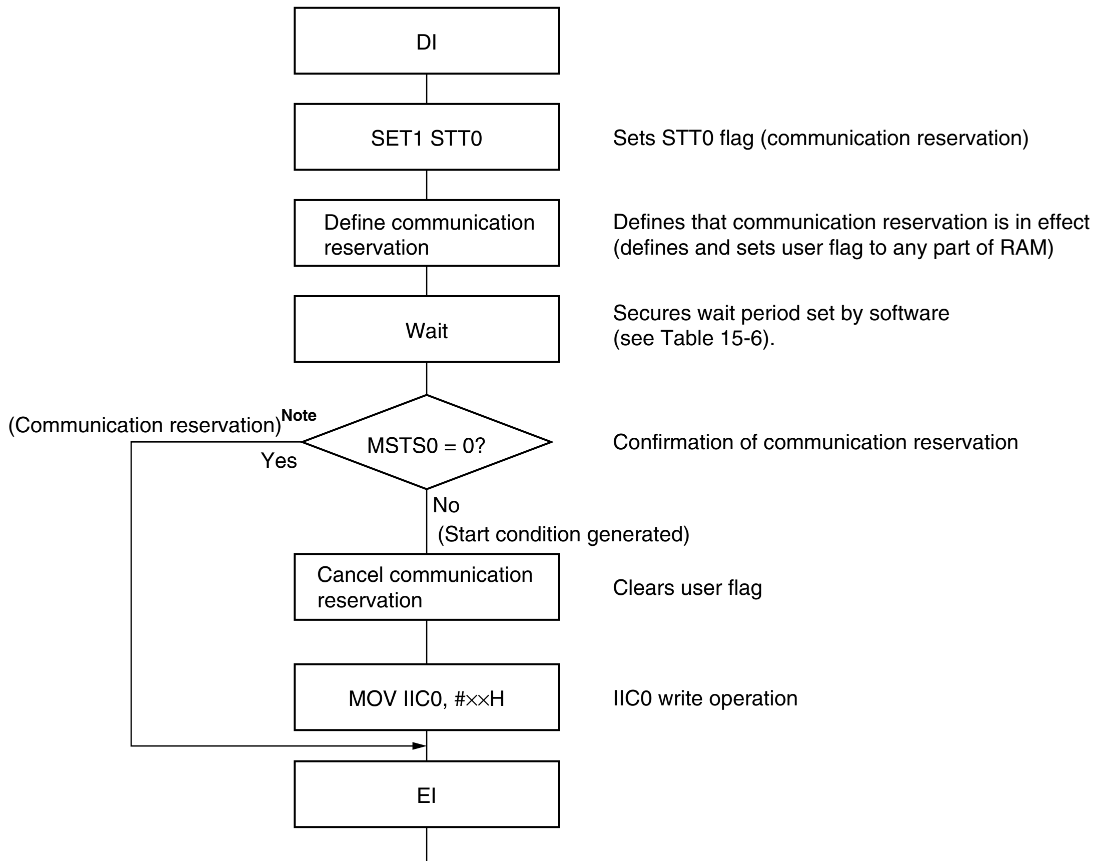


Figure 15-18 shows the communication reservation protocol.

Figure 15-18. Communication Reservation Protocol


DI
SET1 STT0
Sets STT0 flag (communication reservation)
Define communication reservation
Defines that communication reservation is in effect (defines and sets user flag to any part of RAM)
Wait
Secures wait period set by software (see Table 15-6).
(Communication reservation)Note
MSTS0 = 0?
Yes
Confirmation of communication reservation
No
(Start condition generated)
Cancel communication reservation
Clears user flag
MOV IIC0, #××H
IIC0 write operation
EI

Note The communication reservation operation executes a write to IIC shift register 0 (IIC0) when a stop condition interrupt request occurs.

Remark STT0: Bit 1 of IIC control register 0 (IICC0)
MSTS0: Bit 7 of IIC status register 0 (IICS0)
IIC0: IIC shift register 0

15.5.15 Other cautions

After a reset, when changing from a mode in which no stop condition has been detected (the bus has not been released) 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.

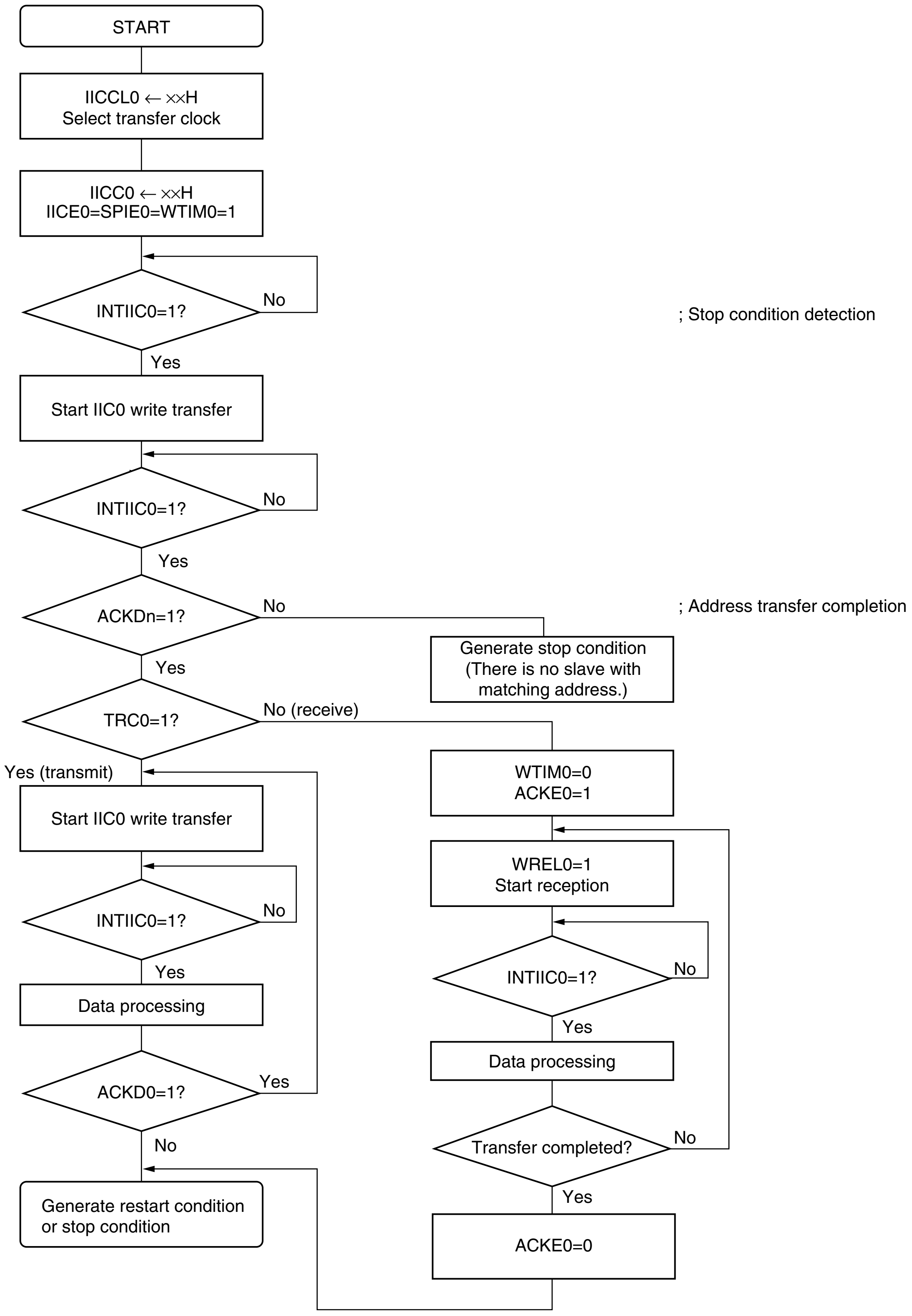
(a) Set IIC clock selection register 0 (IICCL0).
(b) Set bit 7 (IICE0) of IIC control register 0 (IICC0).
(c) Set bit 0 of IICC0.

15.5.16 Communication operations

(1) Master operations

The following is a flow chart of master operations.

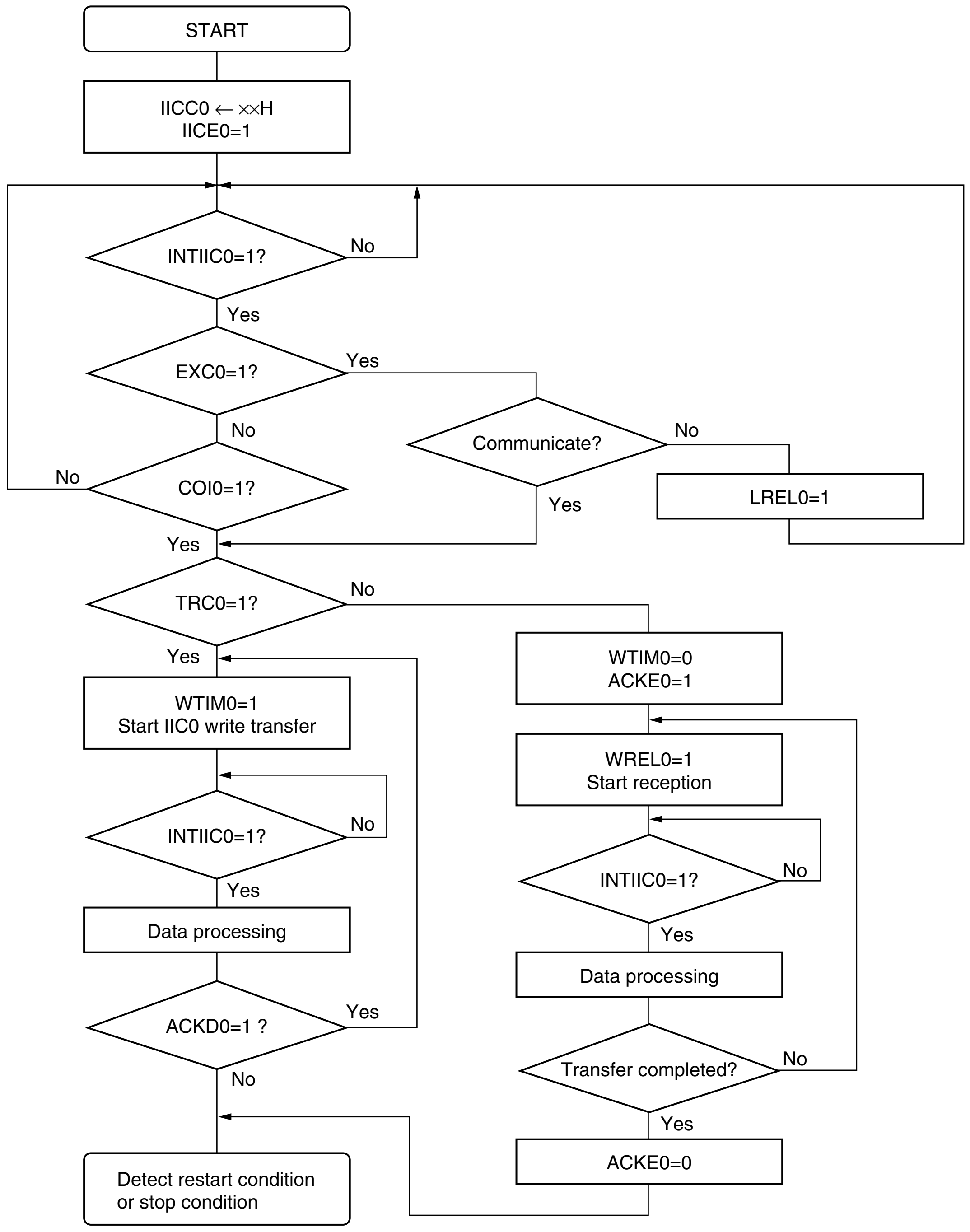
★ **Figure 15-19. Master Operation Flow Chart**



(2) Slave operation

An example of slave operation is shown below.

★ **Figure 15-20. Slave Operation Flow Chart**



15.6 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 TRC0 (bit 3 of IIC status register 0 (IICS0)), which specifies the data transfer direction, and then starts serial communication with the slave device.

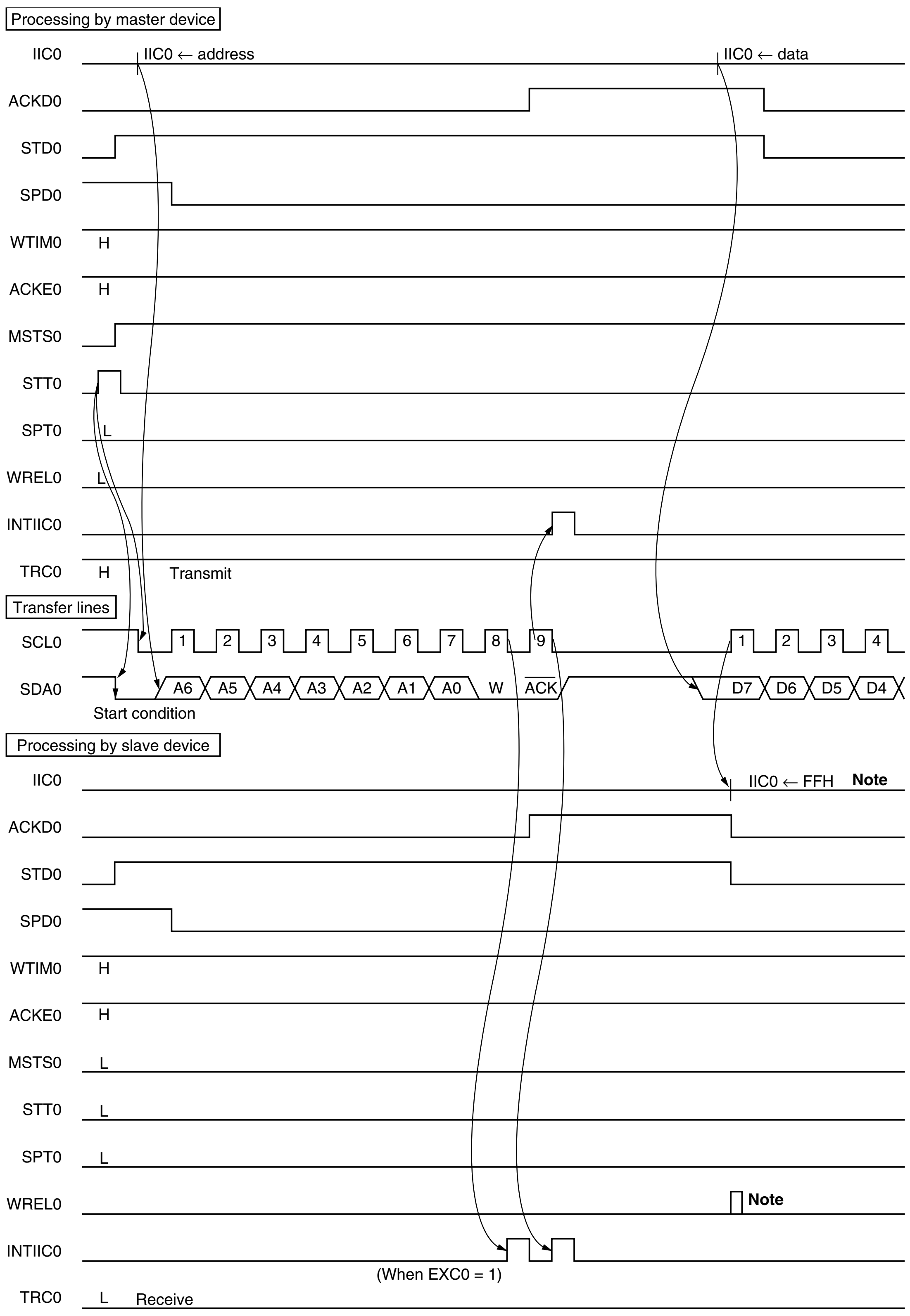
Figures 15-21 and 15-22 show timing charts of the data communication.

IIC shift register 0 (IIC0)'s shift operation is synchronized with the falling edge of the serial clock (SCL0). The transmit data is transferred to the SO0 latch and is output (MSB first) via the SDA0 pin.

Data input via the SDA0 pin is captured into IIC0 at the rising edge of SCL0.

Figure 15-21. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (1/3)

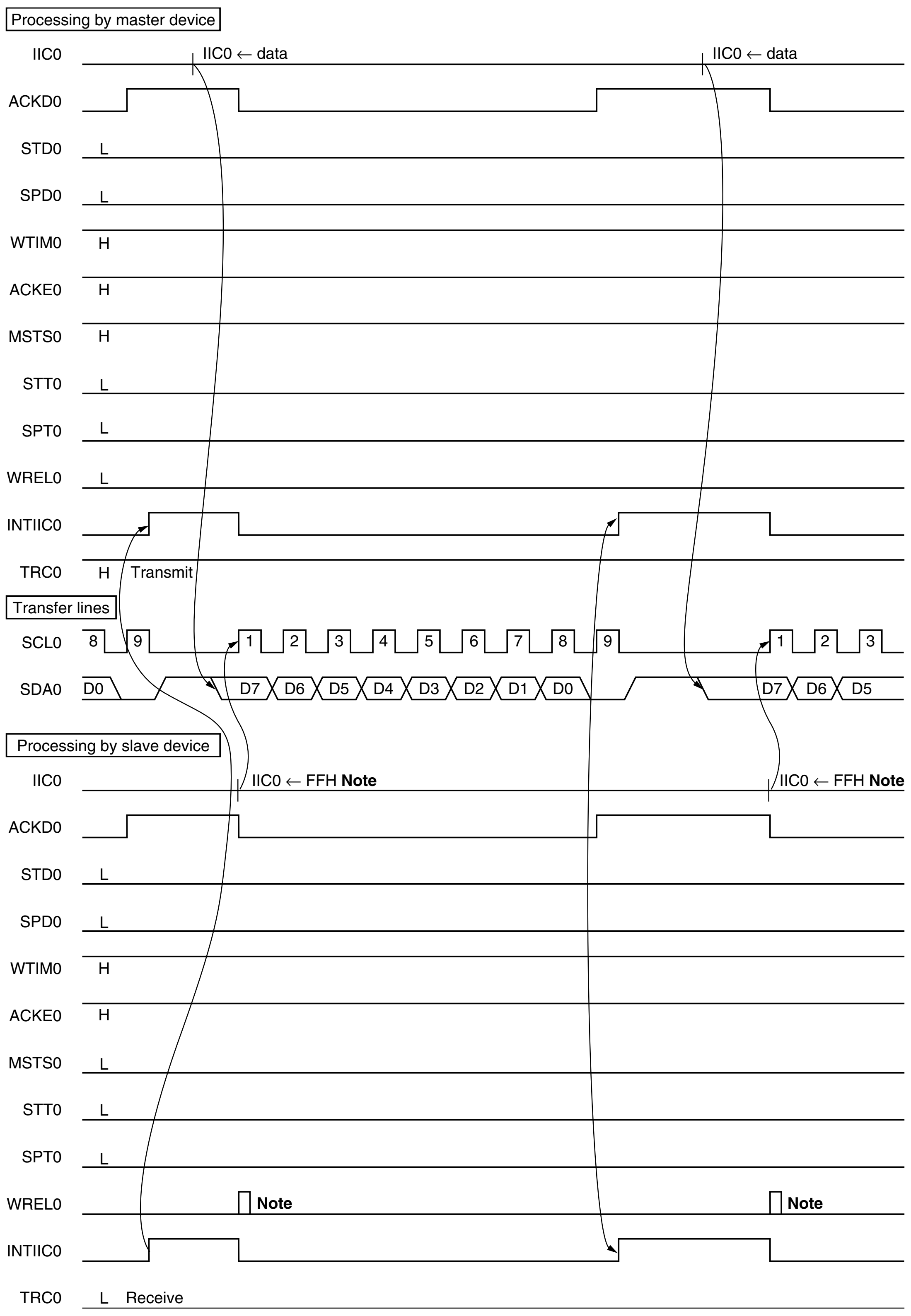
(1) Start condition ~ address



Note To cancel slave wait, write "FFH" to IIC0 or set WREL0.

Figure 15-21. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (2/3)

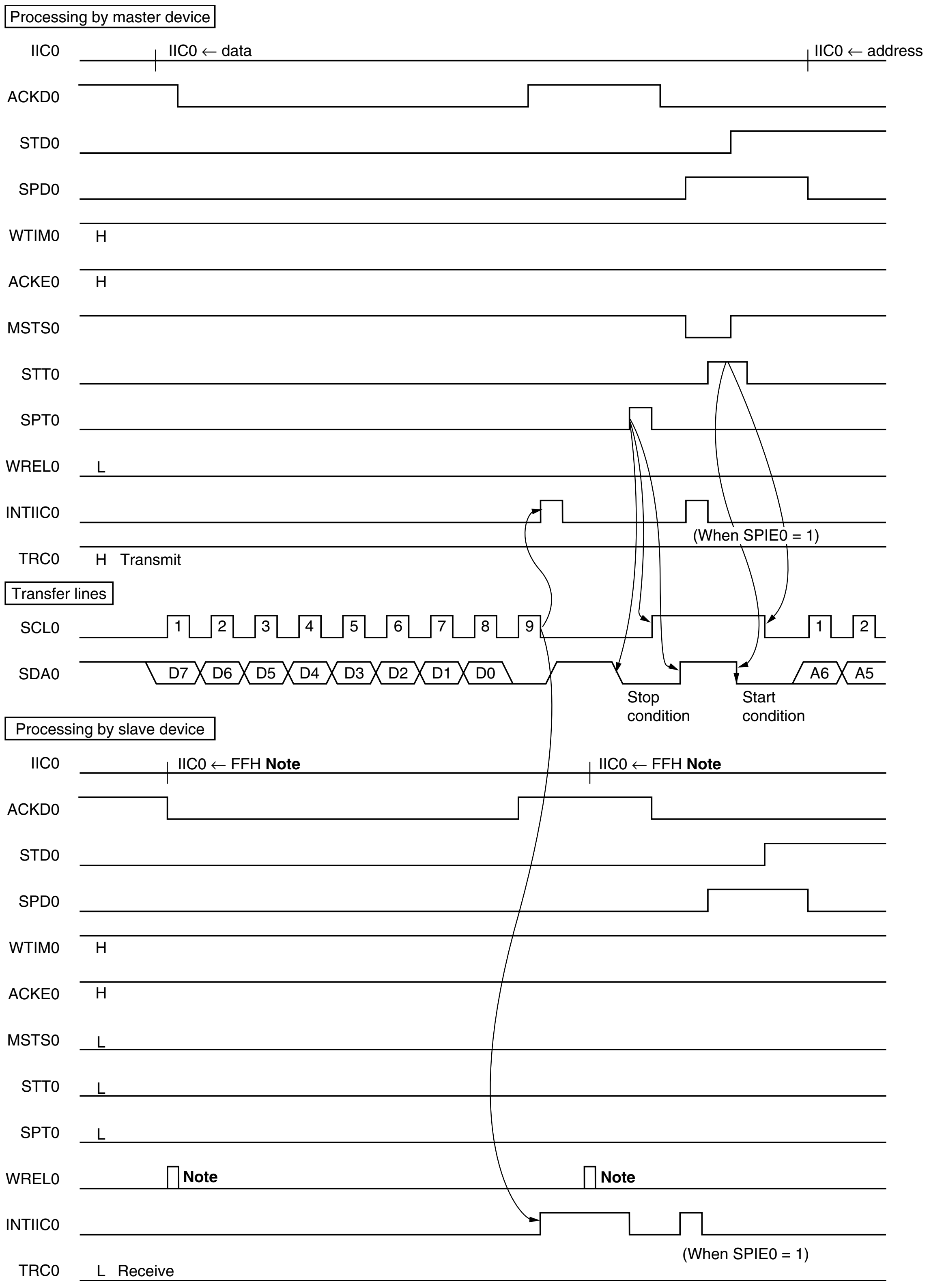
(2) Data



Note To cancel slave wait, write "FFH" to IIC0 or set WREL0.

Figure 15-21. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (3/3)

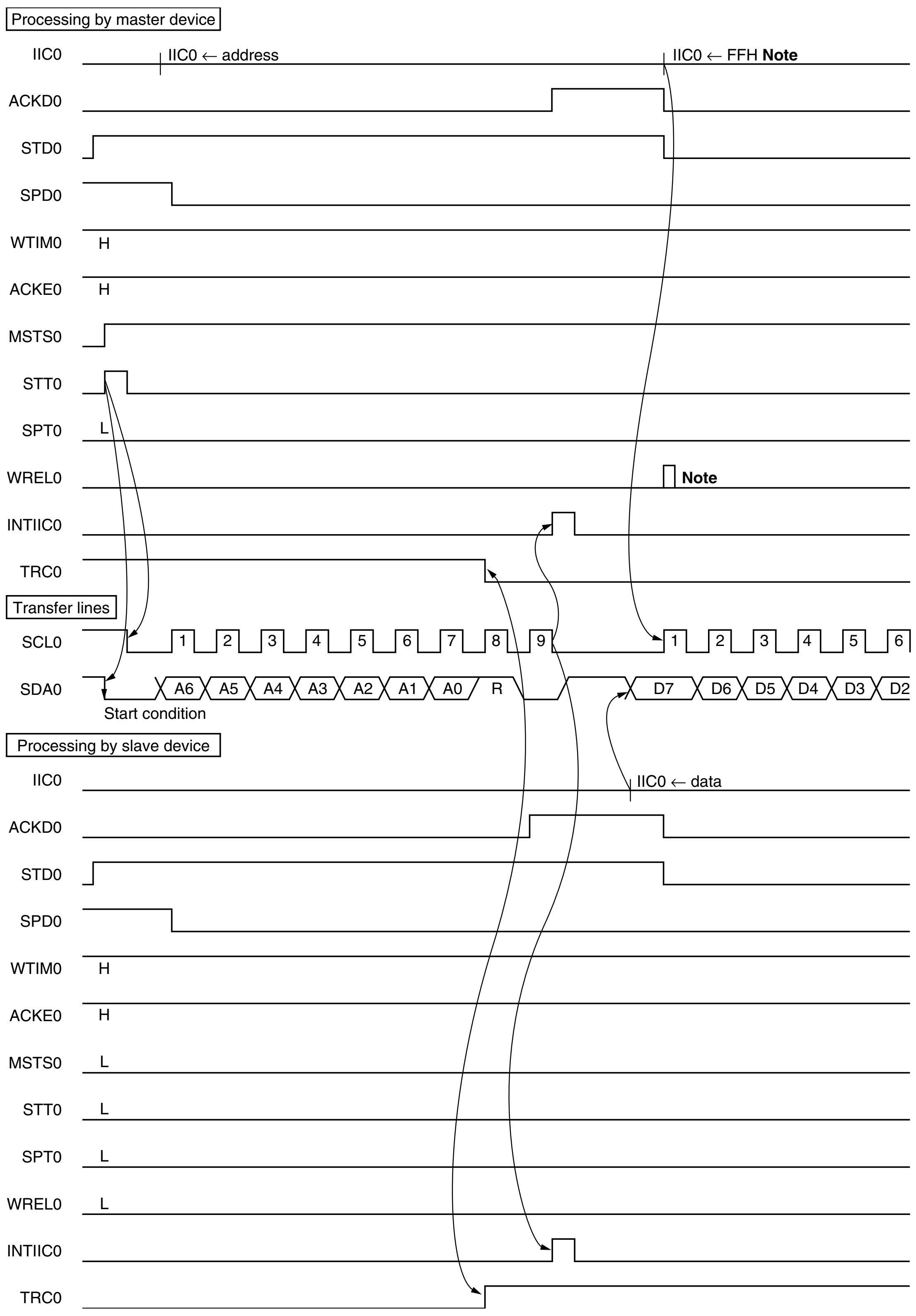
(3) Stop condition



Note To cancel slave wait, write “FFH” to IIC0 or set WREL0.

Figure 15-22. Example of Slave to Master Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (1/3)

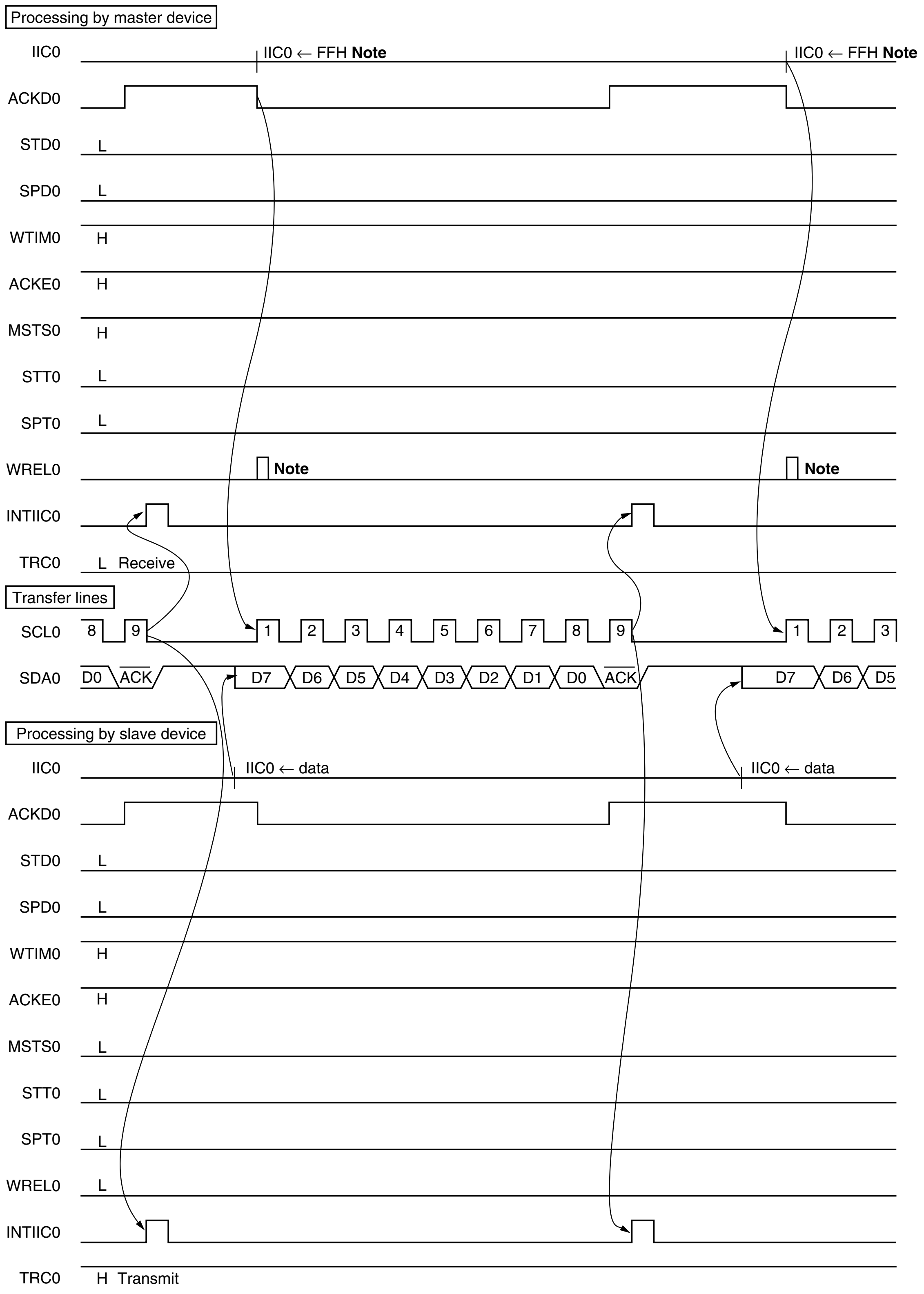
(1) Start condition ~ address



Note To cancel slave wait, write “FFH” to IIC0 or set WREL0.

Figure 15-22. Example of Slave to Master Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (2/3)

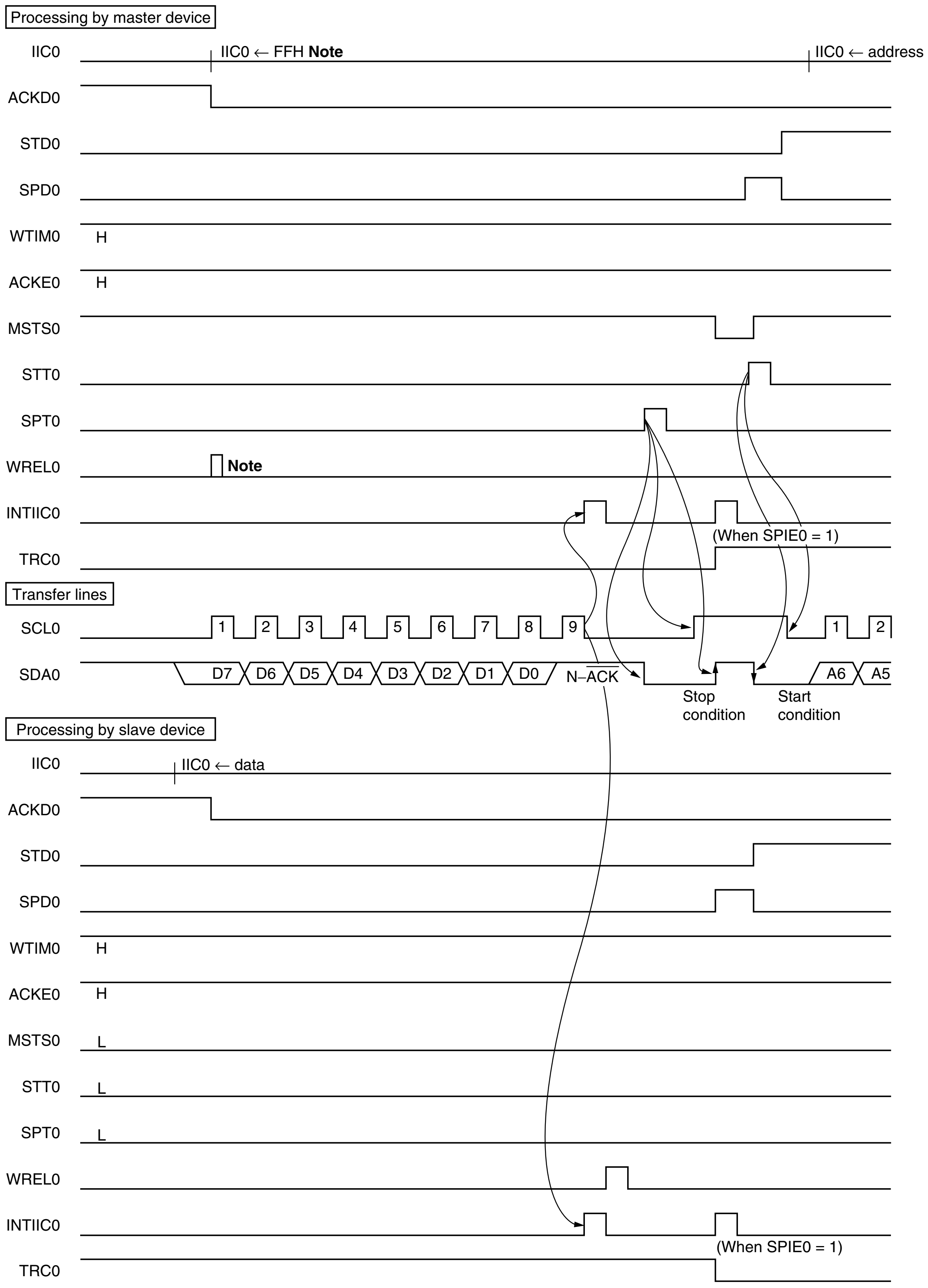
(2) Data



Note To cancel slave wait, write "FFH" to IIC0 or set WREL0.

Figure 15-22. Example of Slave to Master Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (3/3)

(3) Stop condition



Note To cancel slave wait, write "FFH" to IIC0 or set WREL0.

CHAPTER 16 INTERRUPT FUNCTION

16.1 Interrupt Types

The following three types of interrupts are used.

(1) Non-maskable interrupt

This interrupt is acknowledged unconditionally. It does not undergo priority control and is given top priority over all other interrupt requests.
It generates a standby release signal.
An interrupt request from the watchdog timer is provided as the single non-maskable interrupt source.

(2) 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 high priority interrupts can be applied to low priority interrupts. If two or more interrupts with the same priority are simultaneously generated, each interrupt has a predetermined priority (see **Table 16-1**).
A standby release signal is generated.
Nine external interrupt requests and 19 internal interrupt requests are provided as maskable interrupts.

(3) 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.

16.2 Interrupt Sources and Configuration

A total of 30 interrupt sources exist among non-maskable, maskable, and software interrupts (see **Table 16-1**).

★ **Remark** The watchdog timer interrupt source (INTWDT) can be selected as either a non-maskable interrupt or a maskable interrupt (internal).

Table 16-1. Interrupt Source List

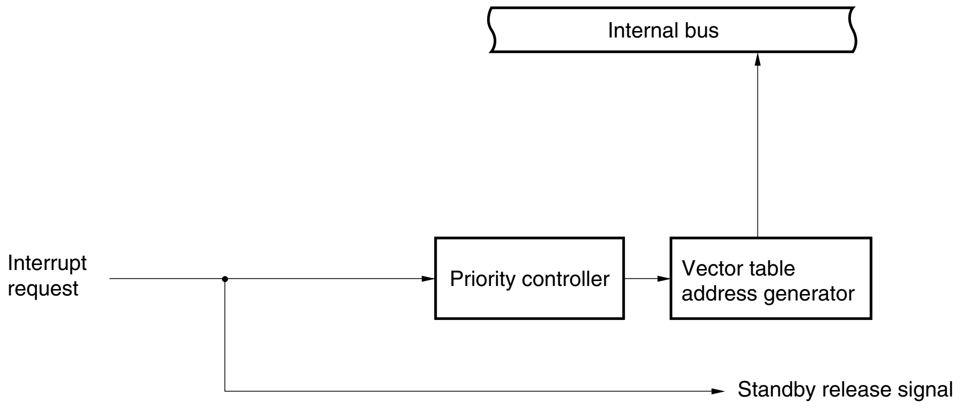
Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Non-maskable	—	INTWDT	Watchdog timer overflow (with watchdog timer mode 1 selected)	Internal	0004H	(A)
Maskable	0	INTWDT	Watchdog timer overflow (with interval timer mode selected)			(B)
	1	INTP0	Pin input edge detection	External	0006H	(C)
	2	INTP1			0008H	
	3	INTP2			000AH	
	4	INTP3			000CH	
	5	INTP4			000EH	
	6	INTP5			0010H	
	7	INTP6			0012H	
	8	INTP7			0014H	
	9	INTSER0	Serial interface UART0 reception error generation	Internal	0016H	(B)
	10	INTSR0	End of serial interface UART0 reception		0018H	
	11	INTST0	End of serial interface UART0 transmission		001AH	
	12	INTCSI30	End of serial interface SIO3 (SIO30) transfer		001CH	
	13	INTCSI31	End of serial interface SIO3 (SIO31) transfer		001EH	
	14	INTIIC0	End of serial interface IIC0 transfer		0020H	
	15	INTC2	Class 2 wake-up request, reception completion, sleep enable, reception error		0022H	
	16	INTWTNI0	Reference time interval signal from watch timer		0024H	
	★ 17	INTTM000	Match between TM00 and CR000 (when compare register is specified) or valid edge detection of TI000 pin (when capture register is specified).		0026H	
	★ 18	INTTM010	Match between TM00 and CR010 (when compare register is specified) or valid edge detection of TI010 pin (when capture register is specified).		0028H	
	★ 19	INTTM001	Match between TM01 and CR001 (when compare register is specified) or valid edge detection of TI001 pin (when capture register is specified).		002AH	
	★ 20	INTTM011	Match between TM01 and CR011 (when compare register is specified) or valid edge detection of TI011 pin (when capture register is specified).		002CH	
	21	INTAD00	End of A/D converter AD00 conversion		002EH	
	22	INTAD01	End of A/D converter AD01 conversion		0030H	
	23	INTWTN0	Watch timer overflow		0034H	
	24	INTKR	Port 4 falling edge detection	External	0036H	(D)
	25	INTTM50	Match between TM50 and CR50	Internal	0038H	(B)
	26	INTTM51	Match between TM51 and CR51		003AH	
	27	INTTM52	Match between TM52 and CR52		003CH	
Software	—	BRK	BRK instruction execution	—	003EH	(E)

★ **Notes 1.** The default priority is the priority applicable when two or more maskable interrupts are generated simultaneously. 0 is the highest priority, and 27 is the lowest.

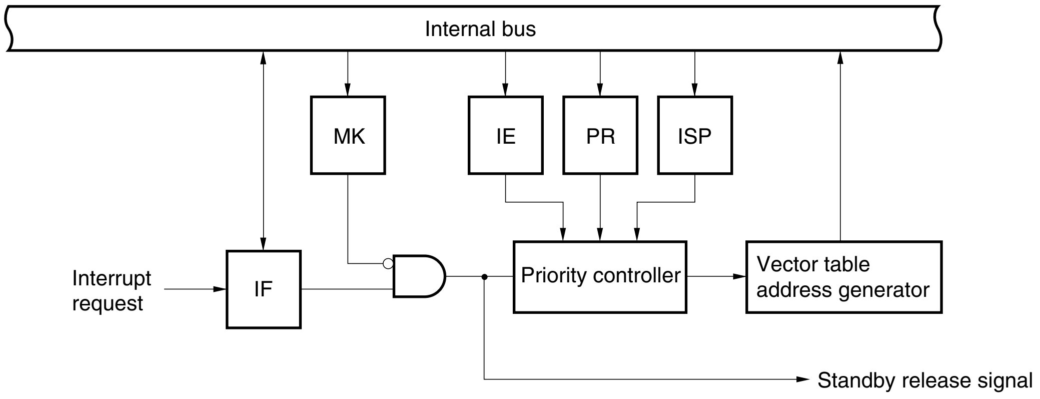
2. Basic configuration types (A) to (E) correspond to (A) to (E) in Figure 16-1.

Figure 16-1. Basic Configuration of Interrupt Function (1/2)

(A) Internal non-maskable interrupt



(B) Internal maskable interrupt



(C) External maskable interrupt (INTP0 to INTP7)

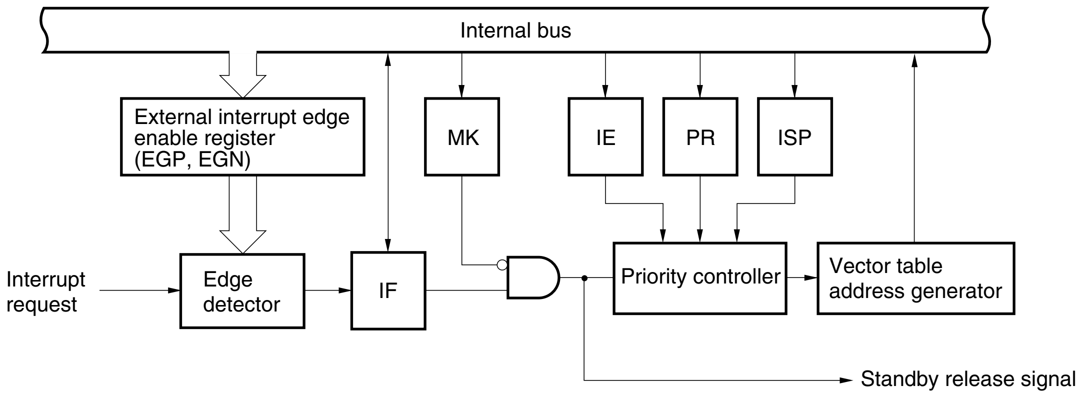
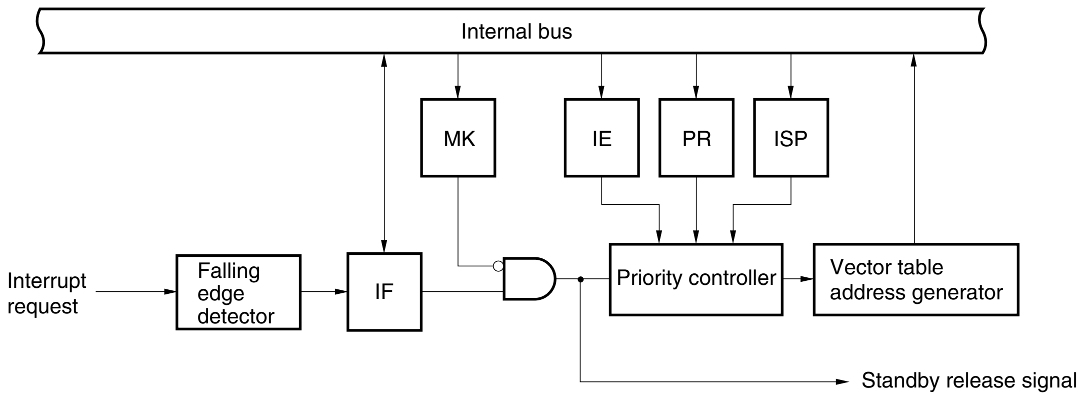
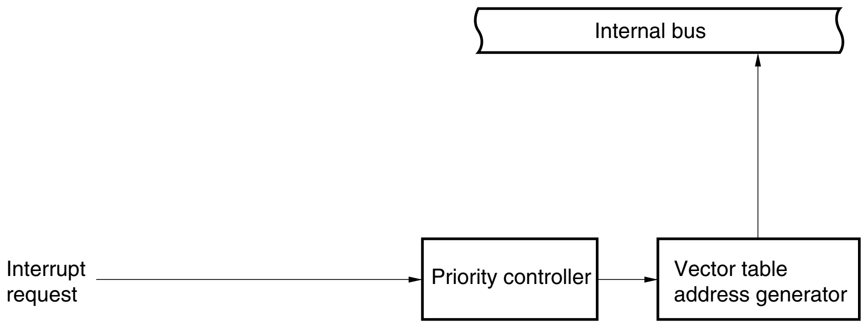


Figure 16-1. Basic Configuration of Interrupt Function (2/2)

(D) External maskable interrupt (INTKR)



(E) Software interrupt



IF:	Interrupt request flag
IE:	Interrupt enable flag
ISP:	In-service priority flag
MK:	Interrupt mask flag
PR:	Priority specification flag

16.3 Interrupt Function Control Registers

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 flag (EGP)
- External interrupt falling edge enable flag (EGN)
- Program status word (PSW)

Table 16-2 gives a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 16-2. Flags Corresponding to Interrupt Request Sources

Interrupt Request	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
		Register		Register		Register
INTWDT	WDTIF[Note]	IF0L	WDTMK	MK0L	WDTPR	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	
INTP6	PIF6		PMK6		PPR6	
INTP7	PIF7	IF0H	PMK7	MK0H	PPR7	PR0H
INTSER0	SERIF0		SERMK0		SERPR0	
INTSR0	SRIF0		SRMK0		SRPR0	
INTST0	STIF0		STMK0		STPR0	
INTCSI30	CSIIF30		CSIMK30		CSIPR30	
INTCSI31	CSIIF31		CSIMK31		CSIPR31	
INTIIC0	IICIF0		IICMK0		IICPR0	
INTC2	C2IF		C2MK		C2PR	
INTWTNI0	WTNIIF0	IF1L	WTNIMK0	MK1L	WTNIPR0	PR1L
INTTM000	TMIF000		TMMK000		TMPR000	
INTTM010	TMIF010		TMMK010		TMPR010	
INTTM001	TMIF001		TMMK001		TMPR001	
INTTM011	TMIF011		TMMK011		TMPR011	
INTAD00	ADIF00		ADMK00		ADPR00	
INTAD01	ADIF01		ADMK01		ADPR01	
INTWTN0	WTNIF0	IF1H	WTNMK0	MK1H	WTNPR0	PR1H
INTKR	KRIF		KRMK		KRPR	
INTTM50	TMIF50		TMMK50		TMPR50	
INTTM51	TMIF51		TMMK51		TMPR51	
INTTM52	TMIF52		TMMK52		TMPR52	

★ **Note** Interrupt control flag when watchdog timer is used as interval timer.

(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 application of $\overline{\text{RESET}}$ input.

IF0L, IF0H, IF1L, and IF1H are set using a 1-bit or 8-bit memory manipulation instruction. When IF0L and IF0H and IF1L and IF1H are respectively combined to form 16-bit registers IF0 and IF1, they are read by a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets these registers to 00H.

Figure 16-2. Format of Interrupt Request Flag Register (IF0L, IF0H, IF1L, IF1H)

Address: FFE0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IF0L	PIF6	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	WDTIF

Address: FFE1H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IF0H	C2IF	IICIF0	CSIIF31	CSIIF30	STIF0	SRIF0	SERIF0	PIF7

Address: FFE2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IF1L	0	ADIF01	ADIF00	TMIF011	TMIF001	TMIF010	TMIF000	WTNIIF0

Address: FFE3H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IF1H	0	0	0	TMIF52	TMIF51	TMIF50	KRIF	WTNIF0

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Cautions 1. The WDTIF flag is R/W enabled only when the watchdog timer is used as the interval timer. If watchdog timer modes 1 and 2 are used, set the WDTIF flag to 0.

★ **2. Be sure to clear the interrupt request flag before operating a timer, serial interface, or A/D converter after standby release; otherwise the interrupt request flag may be set by noise.**

3. Be sure to set bit 7 of IF1L and bits 5 to 7 of IF1LH 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 and set standby clear enable/disable.

MK0L, MK0H, MK1L, and MK1H are set using a 1-bit or 8-bit memory manipulation instruction. When MK0L and MK0H, and MK1L and MK1H are respectively combined to form 16-bit registers MK0 and MK1, they are set using a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets these registers to DFH.

Figure 16-3. Format of Interrupt Mask Flag Register (MK0L, MK0H, MK1L, MK1H)

Address: FFE4H After reset: FFH R/W

Symbol	$\boxed{7}$	$\boxed{6}$	$\boxed{5}$	$\boxed{4}$	$\boxed{3}$	$\boxed{2}$	$\boxed{1}$	$\boxed{0}$
MK0L	PMK6	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	WDTMK

Address: FFE5H After reset: FFH R/W

Symbol	$\boxed{7}$	$\boxed{6}$	$\boxed{5}$	$\boxed{4}$	$\boxed{3}$	$\boxed{2}$	$\boxed{1}$	$\boxed{0}$
MK0H	C2MK	IICMK0	CSIMK31	CSIMK30	STMK0	SRMK0	SERMK0	PMK7

Address: FFE6H After reset: FFH R/W

Symbol	7	$\boxed{6}$	$\boxed{5}$	$\boxed{4}$	$\boxed{3}$	$\boxed{2}$	$\boxed{1}$	$\boxed{0}$
MK1L	1	ADMK01	ADMK00	TMMK011	TMMK001	TMMK010	TMMK000	WTNIMK0

Address: FFE7H After reset: FFH R/W

Symbol	7	6	5	$\boxed{4}$	$\boxed{3}$	$\boxed{2}$	$\boxed{1}$	$\boxed{0}$
MK1H	1	1	0	TMMK52	TMMK51	TMMK50	KRMK	WTNMK0

XXMKX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Cautions 1. If the watchdog timer is used in watchdog timer mode 1, the contents of the WDTMK flag become undefined when read.

2. Because port 0 pins function alternately as external interrupt request inputs, when the output level is changed by specifying the output mode of the port function, the interrupt request flag is set. Therefore, the interrupt mask flag should be set to 1 before using the output mode.

3. Be sure to set bit 7 of MK1L and bits 6 and 7 of MK1H to 1, and bit 5 of MK1H to 0.

(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 using a 1-bit or 8-bit memory manipulation instruction. If PR0L and PR0H and PR1L and PR1H are respectively combined to form 16-bit registers PR0 and PR1, they are set using a 16-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets these registers to FFH.

Figure 16-4. Format of Priority Specification Flag Register (PR0L, PR0H, PR1L, PR1H)

Address: FFE8H After reset: FFH R/W

Symbol	$\boxed{7}$	$\boxed{6}$	$\boxed{5}$	$\boxed{4}$	$\boxed{3}$	$\boxed{2}$	$\boxed{1}$	$\boxed{0}$
PR0L	PPR6	PPR5	PPR4	PPR3	PPR2	PPR1	PPR0	WDTPR

Address: FFE9H After reset: FFH R/W

Symbol	$\boxed{7}$	$\boxed{6}$	$\boxed{5}$	$\boxed{4}$	$\boxed{3}$	$\boxed{2}$	$\boxed{1}$	$\boxed{0}$
PR0H	C2PR	IICPR0	CSIPR31	CSIPR30	STPR0	SRPR0	SERPR0	PPR7

Address: FFEAH After reset: FFH R/W

Symbol	7	$\boxed{6}$	$\boxed{5}$	$\boxed{4}$	$\boxed{3}$	$\boxed{2}$	$\boxed{1}$	$\boxed{0}$
PR1L	1	ADPR01	ADPR00	TMPR011	TMPR001	TMPR010	TMPR000	WTNIPR0

Address: FFEBH After reset: FFH R/W

Symbol	7	6	5	$\boxed{4}$	$\boxed{3}$	$\boxed{2}$	$\boxed{1}$	$\boxed{0}$
PR1H	1	1	1	TMPR52	TMPR51	TMPR50	KRPR	WTNPR0

XXPRX	Priority level selection
0	High priority level
1	Low priority level

Cautions 1. When the watchdog timer is used in watchdog timer mode 1, set the WDTPR flag to 1.
2. Be sure to set bit 7 of PR1L and bits 5 to 7 of PR1H to 1.

(4) External interrupt rising edge enable register (EGP), external interrupt falling edge enable register (EGN)

These registers specify the valid edge for INTP0 to INTP7.

EGP and EGN are set using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets these registers to 00H.

Figure 16-5. Format of External Interrupt Rising Edge Enable Register (EGP) and External Interrupt Falling Edge Enable Register (EGN)

Address: FF48H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP	EGP7	EGP6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

Address: FF49H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN	EGN7	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 7)
0	0	Interrupt disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

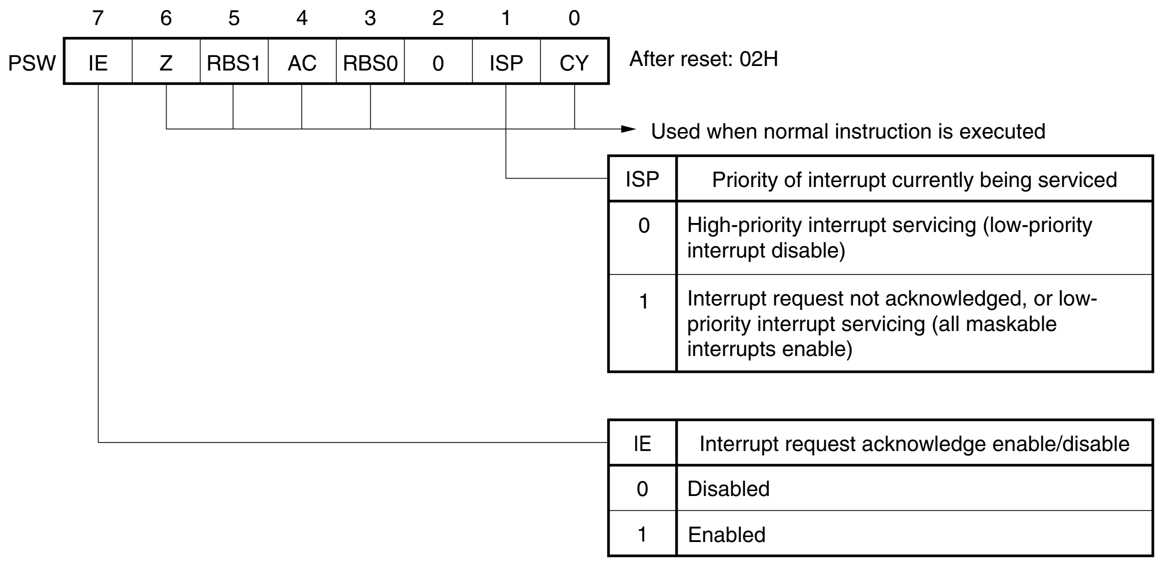
(5) Program status word (PSW)

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 to set maskable interrupt enable/disable and the ISP flag to control multiple interrupt servicing are mapped in this register.

Besides 8-bit read/write, this register can carry out operations via bit manipulation and dedicated (EI and DI) instructions. 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 via the PUSH PSW instruction. They are restored from the stack via the RETI, RETB, and POP PSW instructions.

$\overline{\text{RESET}}$ input sets the PSW to 02H.

Figure 16-6. Format of Program Status Word (PSW)



ISP	Priority of interrupt currently being serviced
0	High-priority interrupt servicing (low-priority interrupt disable)
1	Interrupt request not acknowledged, or low-priority interrupt servicing (all maskable interrupts enable)

IE	Interrupt request acknowledge enable/disable
0	Disabled
1	Enabled

16.4 Interrupt Servicing Operations

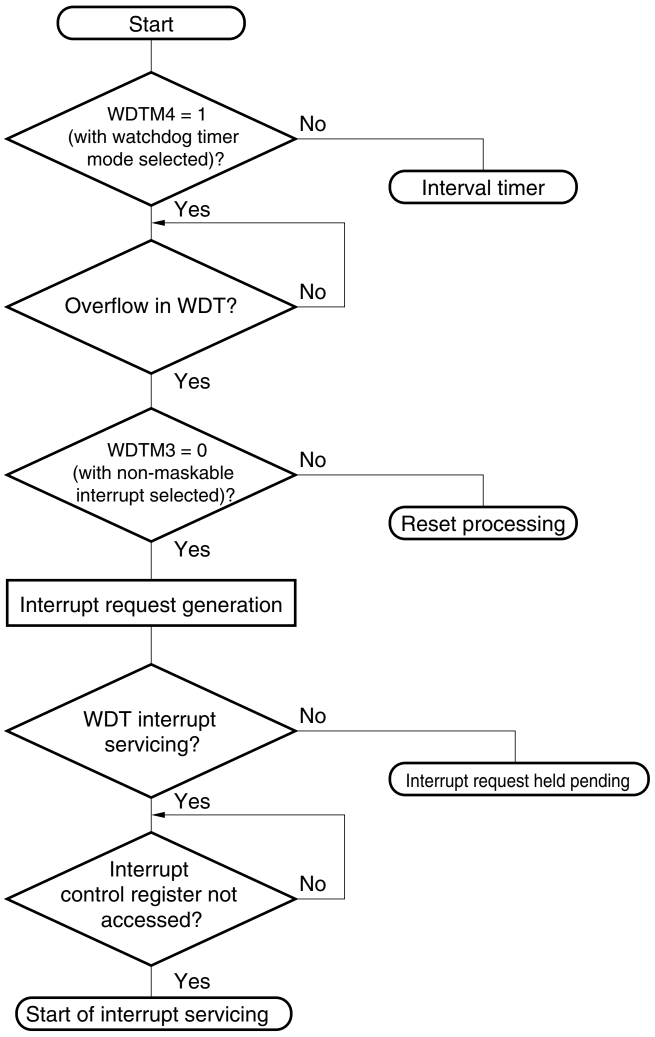
16.4.1 Non-maskable interrupt request acknowledgement operation

A non-maskable interrupt request is unconditionally acknowledged even if interrupt acknowledgement is disabled. It does not undergo interrupt priority control and has the highest priority of all interrupts.

If a non-maskable interrupt request is acknowledged, the contents are saved into the stack in the order of PSW, then PC, the IE flag and ISP flag are reset (0), and the contents of the vector table are loaded into the PC and branched.

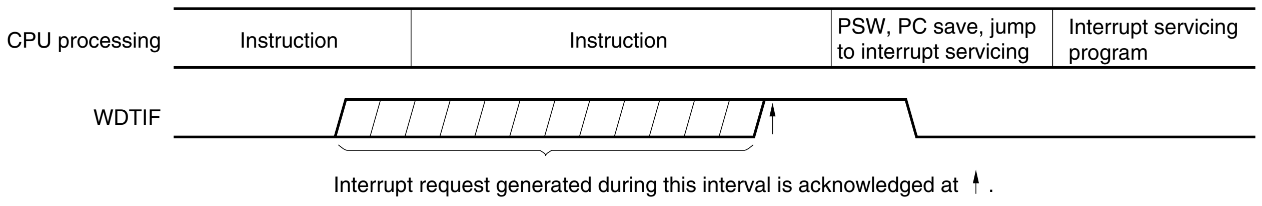
A new non-maskable interrupt request generated during execution of a non-maskable interrupt servicing program is acknowledged after the current execution of the non-maskable interrupt servicing program is terminated (following RETI instruction execution) and one main routine instruction is executed. However, if a new non-maskable interrupt request is generated twice or more during non-maskable interrupt servicing program execution, only one non-maskable interrupt request is acknowledged after termination of the non-maskable interrupt servicing program execution. Figures 16-7, 16-8, and 16-9 show the flowchart of the non-maskable interrupt request generation through to acknowledgement, the acknowledgement timing of a non-maskable interrupt request, and the acknowledgement operation when multiple non-maskable interrupt requests are generated, respectively.

Figure 16-7. Flowchart of Non-Maskable Interrupt Request Generation Through to Acknowledgement


Start
WDTM4 = 1 (with watchdog timer mode selected)?
No
Interval timer
Yes
Overflow in WDT?
No
Yes
WDTM3 = 0 (with non-maskable interrupt selected)?
No
Reset processing
Yes
Interrupt request generation
WDT interrupt servicing?
No
Interrupt request held pending
Yes
Interrupt control register not accessed?
No
Yes
Start of interrupt servicing

WDTM: Watchdog timer mode register
WDT: Watchdog timer

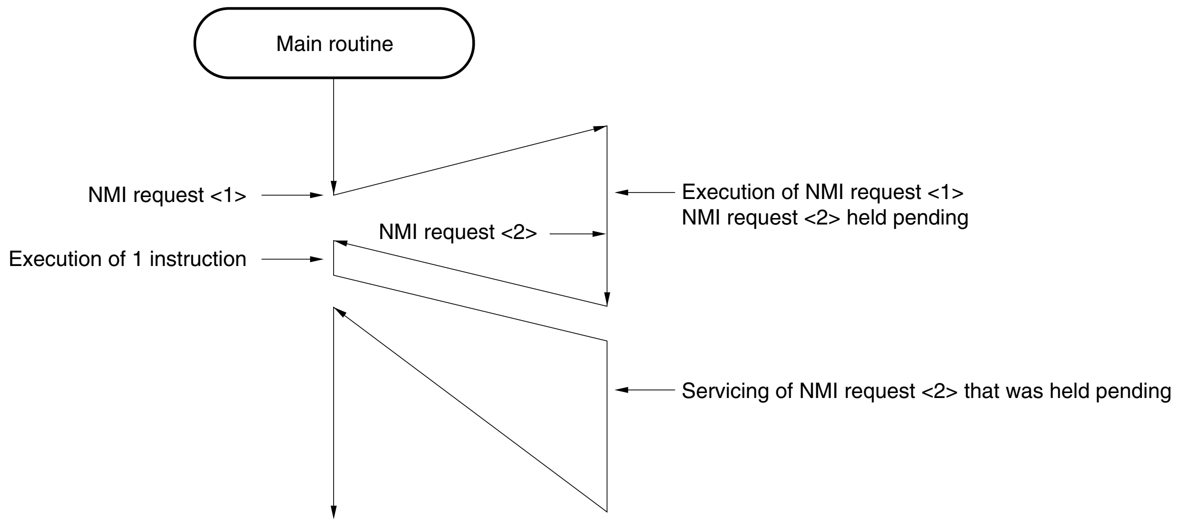
Figure 16-8. Non-Maskable Interrupt Request Acknowledgement Timing


CPU processing
Instruction
Instruction
PSW, PC save, jump to interrupt servicing
Interrupt servicing program
WDTIF
Interrupt request generated during this interval is acknowledged at ↑.

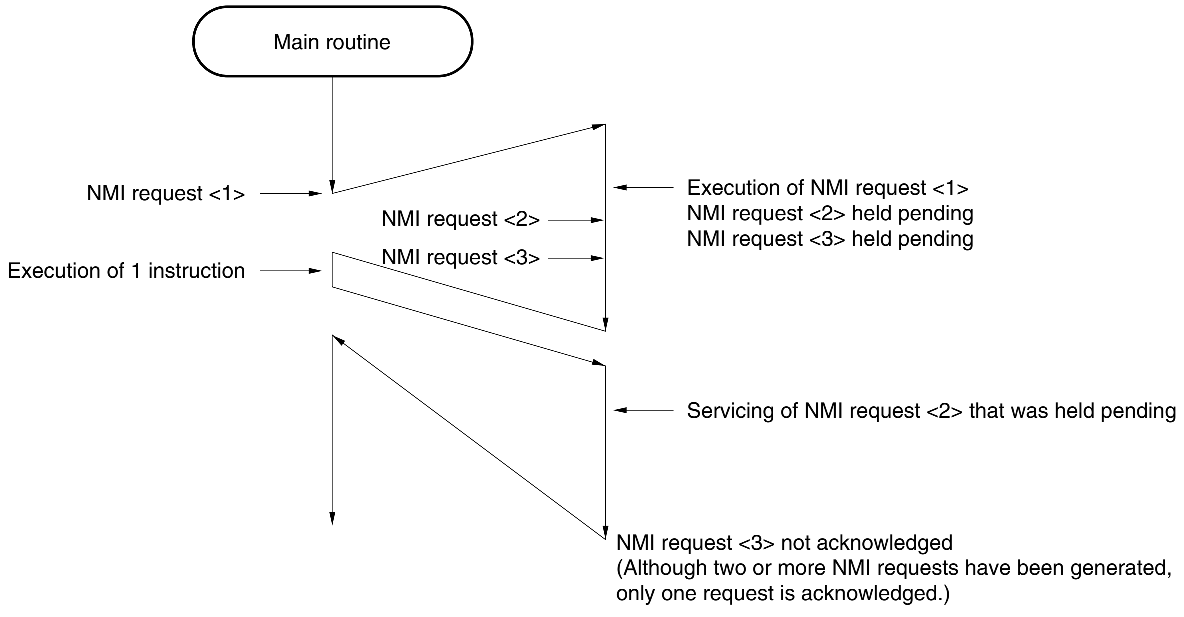
WDTIF: Watchdog timer interrupt request flag

Figure 16-9. Non-Maskable Interrupt Request Acknowledgement Operation

(a) If a non-maskable interrupt request is generated during non-maskable interrupt servicing program execution


Main routine
NMI request <1>
Execution of 1 instruction
NMI request <2>
Execution of NMI request <1>
NMI request <2> held pending
Servicing of NMI request <2> that was held pending

(b) If two non-maskable interrupt requests are generated during non-maskable interrupt servicing program execution


Main routine
NMI request <1>
Execution of 1 instruction
NMI request <2>
NMI request <3>
Execution of NMI request <1>
NMI request <2> held pending
NMI request <3> held pending
Servicing of NMI request <2> that was held pending
NMI request <3> not acknowledged
(Although two or more NMI requests have been generated, only one request is acknowledged.)

16.4.2 Maskable interrupt acknowledge operation

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 when interrupts are enabled (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 cleared to 0). The time from the generation of a maskable interrupt request until interrupt servicing is performed is shown in Table 16-3 below.

For the interrupt request acknowledgement timing, see Figures 16-11 and 16-12.

Table 16-3. 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 interrupt requests are generated simultaneously, the request with a higher priority level specified by the priority specification flag is acknowledged first. If two or more interrupt requests have the same priority level, the request with the highest default priority is acknowledged first.

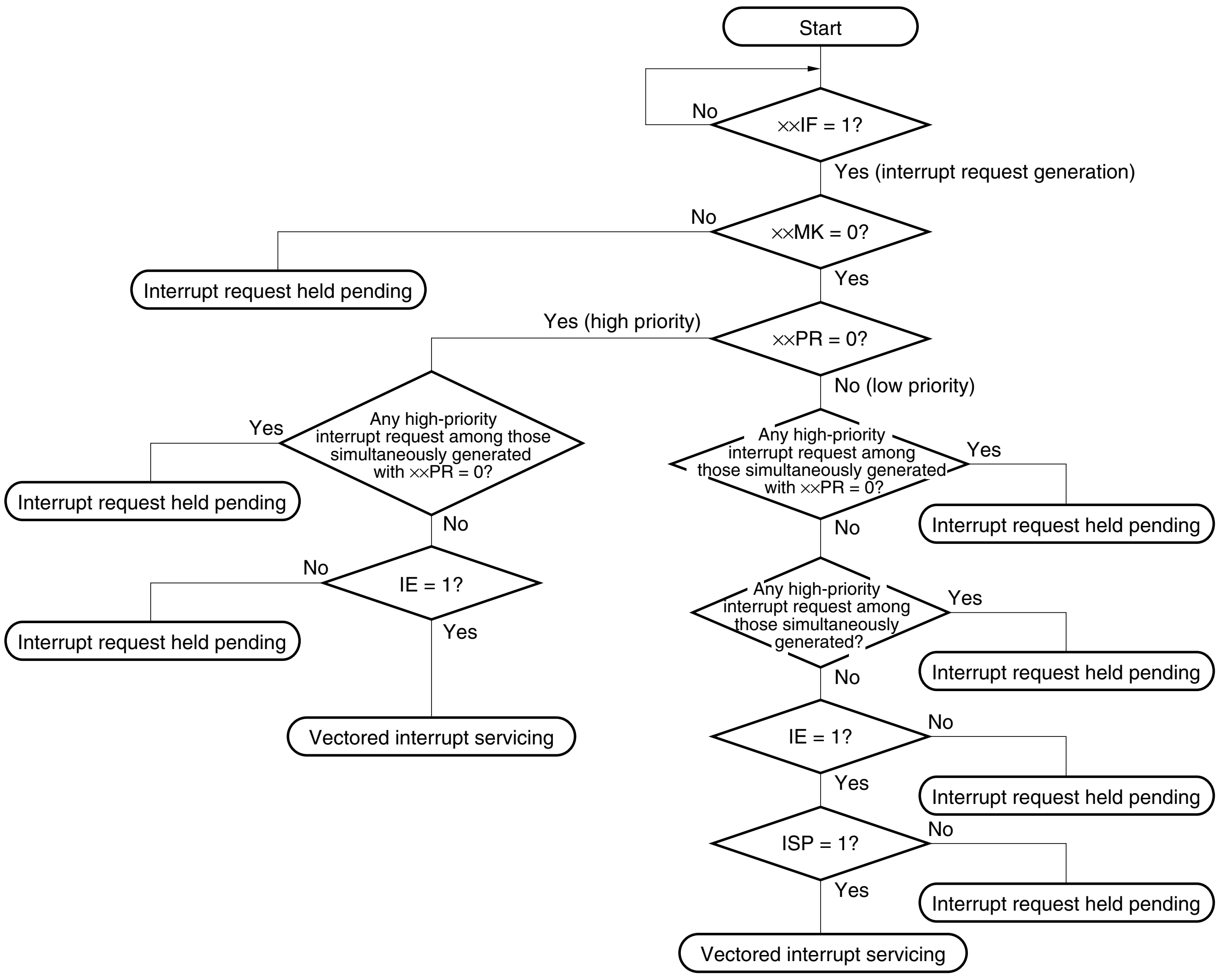
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 16-10 shows the interrupt request acknowledgement algorithm.

If a maskable interrupt request is acknowledged, the contents are saved into the stack 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. Further, the vector table data determined for each interrupt request is loaded into the PC and branched.

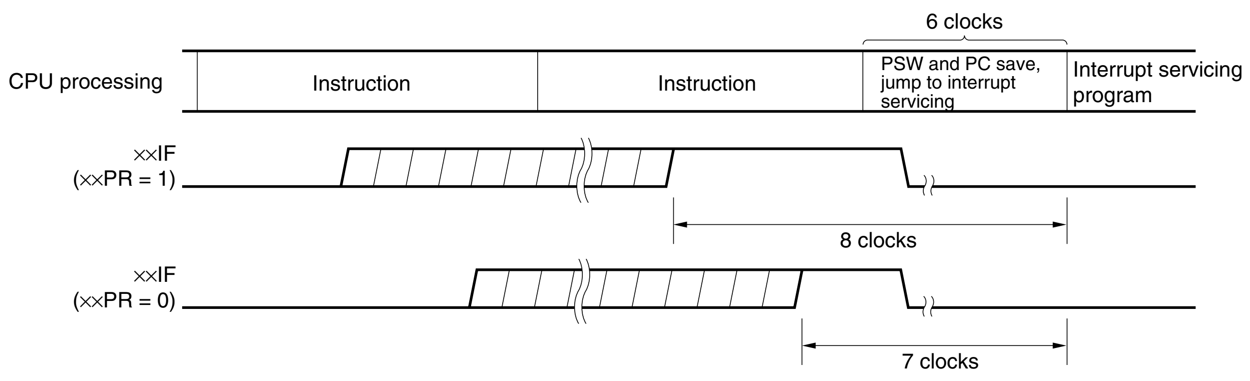
Restoration from an interrupt is possible via the RETI instruction.

Figure 16-10. Interrupt Request Acknowledgement Processing Algorithm


Start
××IF = 1?
No
Yes (interrupt request generation)
××MK = 0?
No
Interrupt request held pending
Yes
××PR = 0?
Yes (high priority)
No (low priority)
Any high-priority interrupt request among those simultaneously generated with ××PR = 0?
Yes
Interrupt request held pending
No
IE = 1?
No
Interrupt request held pending
Yes
Vectored interrupt servicing
Any high-priority interrupt request among those simultaneously generated with ××PR = 0?
Yes
Interrupt request held pending
No
Any high-priority interrupt request among those simultaneously generated?
Yes
Interrupt request held pending
No
IE = 1?
No
Interrupt request held pending
Yes
ISP = 1?
No
Interrupt request held pending
Yes
Vectored interrupt servicing

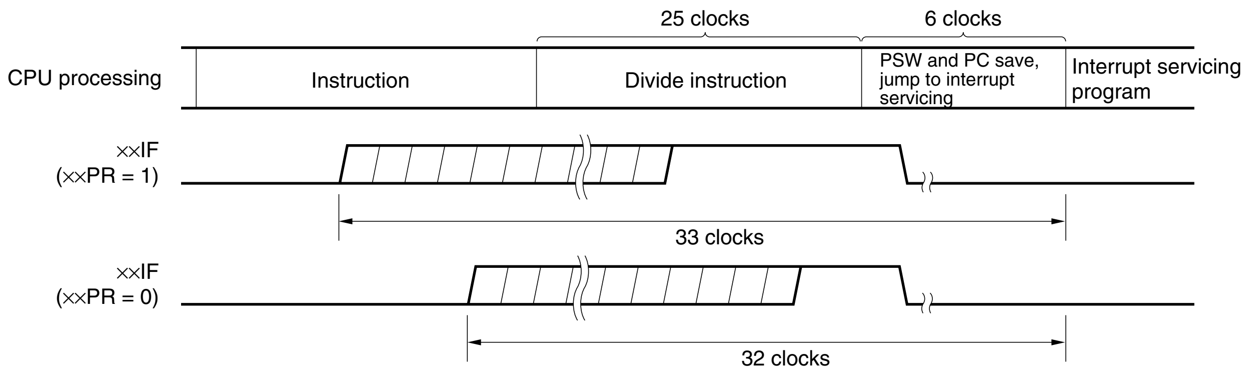
××IF:	Interrupt request flag
××MK:	Interrupt mask flag
××PR:	Priority specification flag
IE:	Flag that controls acknowledgement of maskable interrupt requests (1 = Enable, 0 = Disable)
ISP:	Flag that indicates the priority level of the interrupt currently being serviced (0 = High-priority interrupt servicing, 1 = No interrupt request received, or low-priority interrupt servicing)

Figure 16-11. Interrupt Request Acknowledgement Timing (Minimum Time)



Remark 1 clock: 1/fCPU (fCPU: CPU clock)

Figure 16-12. Interrupt Request Acknowledgement Timing (Maximum Time)



Remark 1 clock: 1/fCPU (fCPU: CPU clock)

16.4.3 Software interrupt request acknowledgement operation

A software interrupt request is acknowledged by BRK instruction execution. Software interrupts cannot be disabled.

If a software interrupt request is acknowledged, the contents are saved into the stack 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.

Restoration from a software interrupt is possible via the RETB instruction.

Caution Do not use the RETI instruction for restoring from a software interrupt.

16.4.4 Multiple interrupt servicing

Multiple interrupts occur when another interrupt request is acknowledged during execution of an interrupt.

Multiple interrupts do not occur unless the interrupt request acknowledge enable state is selected (IE = 1) (except non-maskable interrupts). Also, when an interrupt request is received, interrupt request acknowledgement becomes disabled (IE = 0). Therefore, to enable multiple interrupts, it is necessary to set (1) the IE flag with the EI instruction during interrupt servicing to enable interrupt acknowledgement.

Moreover, even if interrupts are enabled, multiple interrupts 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 interrupts.

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 of the interrupt disable state or because they have a lower priority are held pending. When servicing of the current interrupt ends, the perding interrupt request is acknowledged following execution of one main processing instruction.

Multiple interrupt servicing is not possible during non-maskable interrupt servicing.

Table 16-4 shows interrupt requests enabled for multiple interrupt servicing, and Figure 16-13 shows multiple interrupt examples.

Table 16-4. Interrupt Request Enabled for Multiple Interrupt During Interrupt Servicing

Multiple Interrupt Request / Interrupt Being Serviced		Non-Maskable Interrupt Request	Maskable Interrupt Request			
			PR = 0		PR = 1	
			IE = 1	IE = 0	IE = 1	IE = 0
Non-maskable interrupt		×	×	×	×	×
Maskable interrupt	ISP = 0	○	○	×	×	×
	ISP = 1	○	○	×	○	×
Software interrupt		○	○	×	○	×

Remarks 1. ○: Multiple interrupts enabled

2. ×: Multiple interrupts disabled

3. The 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 acknowledgement is disabled.

IE = 1: Interrupt request acknowledgement is enabled.

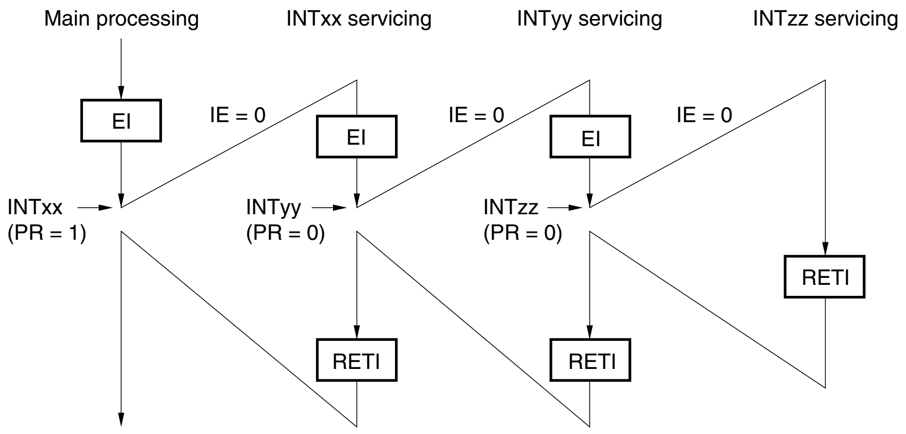
4. PR is a flag contained in PR0L, PR0H, and PR1L.

PR = 0: Higher priority level

PR = 1: Lower priority level

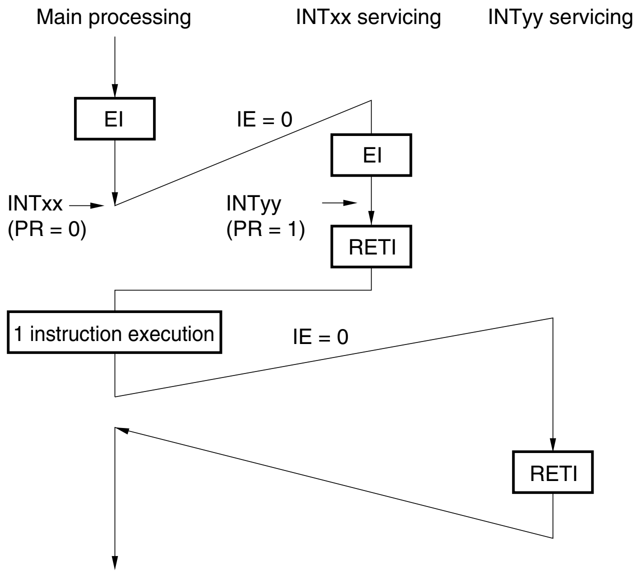
Figure 16-13. Multiple Interrupt Examples (1/2)

Example 1. Multiple interrupts occur 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 acknowledgement.

Example 2. Multiple interrupt servicing does not occur due to priority control

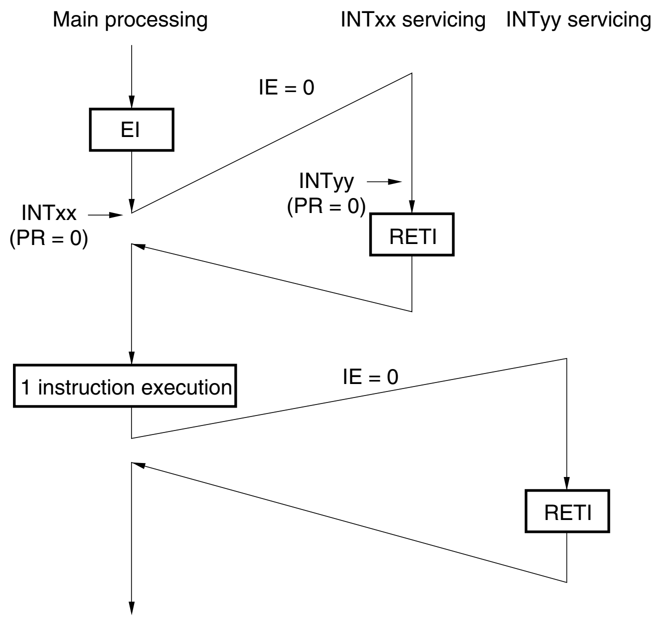


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 0:	Higher priority level
PR = 1:	Lower priority level
IE = 0:	Interrupt request acknowledge disabled

Figure 16-13. Multiple Interrupt Examples (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 acknowledge disabled

16.4.5 Interrupt request hold

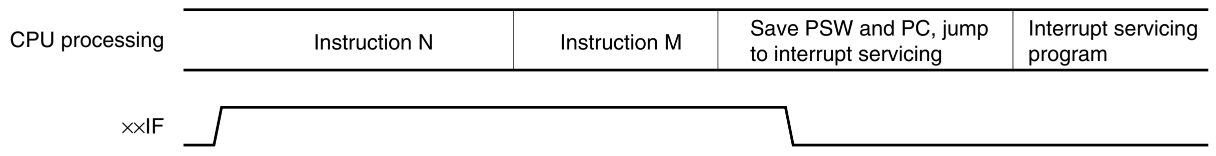
There are instructions where, even if an interrupt request is issued for them while another instruction is being executed, request acknowledgement is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- 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. However, a non-maskable interrupt request is acknowledged.

Figure 16-14 shows the timing at which interrupt requests are held pending.

Figure 16-14. Interrupt Request Hold


CPU processing
Instruction N
Instruction M
Save PSW and PC, jump to interrupt servicing
Interrupt servicing program
××IF

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 (instruction request)

CHAPTER 17 STANDBY FUNCTION

17.1 Standby Function and Configuration

17.1.1 Standby function

The standby function is designed to reduce the power consumption 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, but the system clock oscillator continues oscillating. In this mode, current consumption is not reduced as much as in the STOP mode. However, the HALT mode is effective for restarting operation immediately upon generation of an interrupt request and carrying out intermittent operations such as watch applications.

(2) STOP mode

STOP instruction execution sets the STOP mode. In the STOP mode, the system clock oscillator stops, stopping the whole system, thereby considerably reducing the CPU power consumption.

Data memory low-voltage hold (down to V_{DD} = 1.6 V) is possible. Thus, the STOP mode is effective for holding data memory contents with ultra-low current consumption. Because this mode can be cleared upon interrupt request, it enables intermittent operations to be carried out.

However, because wait time is required to secure the oscillation stabilization time after the STOP mode is released, 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 latch and output buffer statuses are also held.

Cautions 1. When operation is transferred to the STOP mode, be sure to stop the peripheral hardware operation and execute the STOP instruction.

2. The following sequence is recommended for reducing the power consumption of the A/D converter when the standby function is used: First clear bit 7 (ADCS00, ADCS01) of the A/D converter mode register (ADM00, ADM01) to 0 to stop the A/D conversion operation, and then execute the HALT or STOP instruction.

17.1.2 Standby function control register

The wait time after the STOP mode is released upon interrupt request generation is controlled with the oscillation stabilization time selection register (OSTS).

OSTS is set using an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets OSTS to 04H.

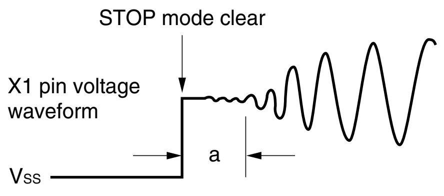
Figure 17-1. Format of Oscillation Stabilization Time Selection Register (OSTS)

Address: FFFAH After reset: 04H R/W

Symbol	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

OSTS2	OSTS1	OSTS0	Selection of oscillation stabilization time
0	0	0	$2^{12}/f_X$ (977 μs)
0	0	1	$2^{14}/f_X$ (3.91 ms)
0	1	0	$2^{15}/f_X$ (7.81 ms)
0	1	1	$2^{16}/f_X$ (15.6 ms)
1	0	0	$2^{17}/f_X$ (31.3 ms)
Other than above			Setting prohibited

Caution The wait time after the STOP mode is released does not include the time (see "a" in the illustration below) from STOP mode release to clock oscillation start. This applies whether STOP mode is released by $\overline{\text{RESET}}$ input or by interrupt request generation.



Remarks 1. f_X: System clock oscillation frequency

2. Values in parentheses apply to operation with f_X = 4.19 MHz.

17.2 Standby Function Operations

17.2.1 HALT mode

(1) HALT mode setting and operating statuses

The HALT mode is set by executing the HALT instruction.

The operating statuses in the HALT mode are described below.

Table 17-1. HALT Mode Operating Statuses

Item	Operating Status
Clock generator	Can be oscillated. Clock supply to CPU stops.
CPU	Operation stops.
Port (output latch)	Holds the state it was in just before the HALT instruction was executed.
16-bit timer/event counter	Operable
8-bit timer/event counter	
Watch timer	
Watchdog timer	
A/D converter	Operation stops
Serial interface	Operable
J1850 (Class 2)	

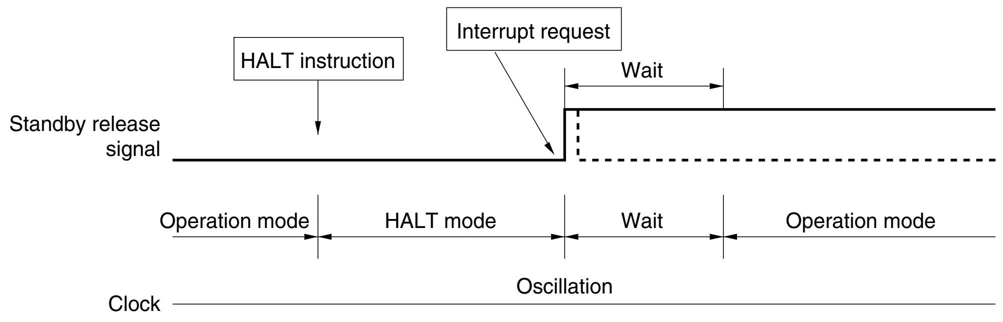
(2) HALT mode release

The HALT mode can be released by the following three sources.

(a) Release upon unmasked interrupt request

When an unmasked interrupt request is generated, the HALT mode is released. If interrupt acknowledgement is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgement is disabled, the next address instruction is executed.

Figure 17-2. HALT Mode Release by Interrupt Request Generation



Remarks 1. The broken lines indicate the case when the interrupt request that released the standby mode is acknowledged.

2. The wait times are as follows:

• When vectored interrupt servicing is carried out:	8 or 9 clocks
• When vectored interrupt servicing is not carried out:	2 or 3 clocks

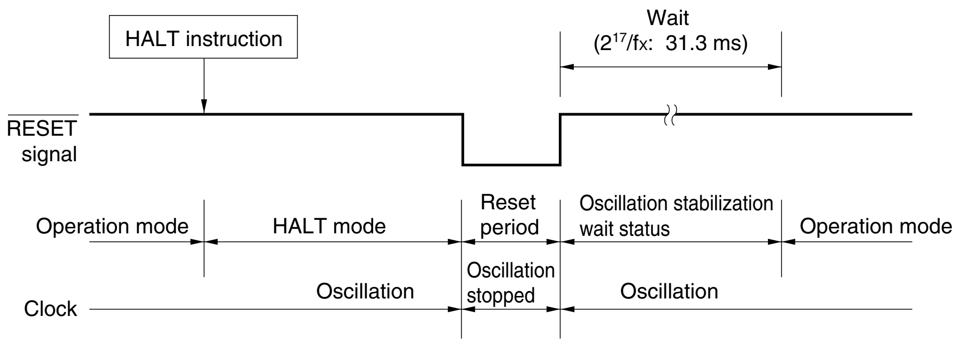
(b) Release upon non-maskable interrupt request

When a non-maskable interrupt request is generated, the HALT mode is released and vectored interrupt servicing is carried out, whether interrupt acknowledgement is enabled or disabled.

(c) Release by $\overline{\text{RESET}}$ input

When the $\overline{\text{RESET}}$ signal is input, HALT mode is released. The program is then executed after a branch to the reset vector address, as in the case of a normal reset operation.

Figure 17-3. HALT Mode Release by $\overline{\text{RESET}}$ Input



Remarks 1. fx: System clock oscillation frequency

2. Values in parentheses apply to operation with fx = 4.19 MHz.

Table 17-2. Operation After HALT Mode Release

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 hold
Non-maskable interrupt request	—	—	×	×	Interrupt servicing execution
$\overline{\text{RESET}}$ input	—	—	×	×	Reset processing

×: Don't care

17.2.2 STOP mode

(1) STOP mode setting and operating status

The STOP mode is set by executing the STOP instruction.

Cautions 1. When the STOP mode is set, the X2 pin is internally connected to V_{DD1} via a pull-up resistor to minimize the leakage current at the crystal oscillator. Thus, do not use the STOP mode in a system where an external clock is used for the system clock.

2. Because the interrupt request signal is used to release the standby mode, if there is an interrupt source with the interrupt request flag set and the interrupt mask flag reset, the standby mode is immediately released if set. Thus, the STOP mode is reset to the HALT mode immediately after execution of the STOP instruction. After the wait set using the oscillation stabilization time selection register (OSTS), the operation mode is set.

The operating status in the STOP mode is described in Table 17-3 below.

Table 17-3. STOP Mode Operating Status

Item		Operating Status
Clock generator		Oscillation stops
CPU		Operation stops
Port (output latch)		Holds the state it was in just before the STOP instruction was executed.
16-bit timer/event counter		Operable only when TI000 and TI001 are specified for the count clock.
8-bit timer/event counter		Operable only when TI50, TI51, TI52 are specified for the count clock.
Watch timer		Operation stops
Watchdog timer		
A/D converter		
Serial interface	Other than UART	Operable only when externally supplied clock is specified as the serial clock.
	UART	Operation stops. (Transmit shift register 0 (TXS0), receive shift register 0 (RX0), and receive buffer register 0 (RXB0) hold the value just before the clock stop.)
J1850 (Class 2)		Operation stops

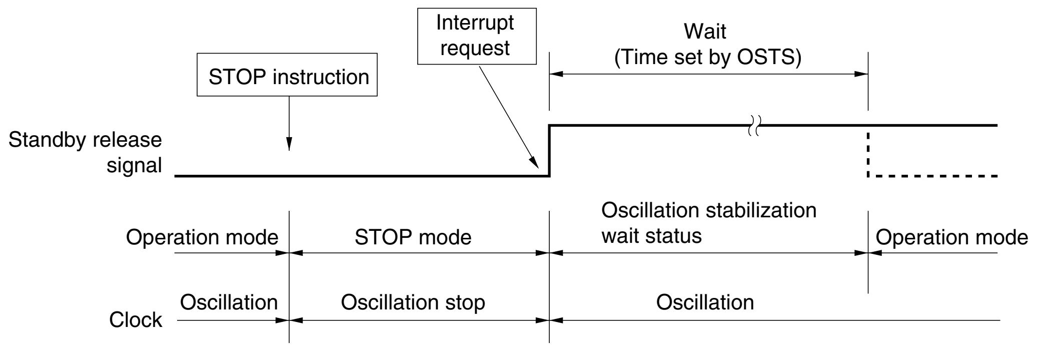
(2) STOP mode release

The STOP mode can be released by the following two sources.

(a) Release by unmasked interrupt request

When an unmasked interrupt request is generated, the STOP mode is released. If interrupt acknowledgement is enabled after the lapse of oscillation stabilization time, vectored interrupt servicing is carried out. If interrupt acknowledgement is disabled, the next address instruction is executed.

Figure 17-4. STOP Mode Release by Interrupt Request Generation

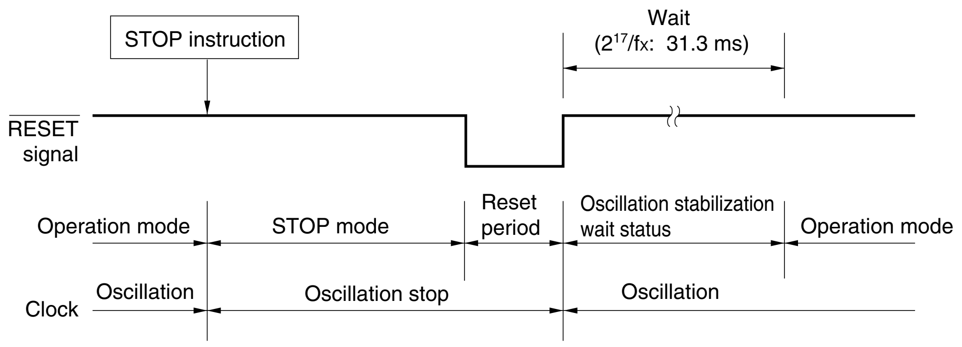


Remark The broken lines indicate the case when the interrupt request that released the standby status is acknowledged.

(b) Release by $\overline{\text{RESET}}$ input

The STOP mode is released when $\overline{\text{RESET}}$ signal is input, and after the lapse of oscillation stabilization time, the reset operation is carried out.

Figure 17-5. STOP Mode Release by $\overline{\text{RESET}}$ Input



Remarks 1. fx: System clock oscillation frequency

2. Values in parentheses apply to operation with fx = 4.19 MHz.

Table 17-4. Operation After STOP Mode Release

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 hold
$\overline{\text{RESET}}$ input	—	—	×	×	Reset processing

×: Don’t care

CHAPTER 18 RESET FUNCTION

18.1 Reset Function

The following two operations are available to generate a reset.

(1) External reset input via $\overline{\text{RESET}}$ pin
(2) Internal reset by watchdog timer program loop time detection

The external reset and internal reset have no functional differences. In both cases, program execution starts at the address at 0000H and 0001H by $\overline{\text{RESET}}$ input. When a low level is input to the $\overline{\text{RESET}}$ pin or the watchdog timer overflows, a reset is applied and each item of hardware is set to the status shown in Table 18-1. Each pin is high impedance during reset input or during the oscillation stabilization time just after reset release.

When a high level is input to the $\overline{\text{RESET}}$ pin, the reset is released and program execution starts after the lapse of the oscillation stabilization time, 2^{17}/fx. The reset applied by watchdog timer overflow is automatically released after the reset and program execution starts after the lapse of the oscillation stabilization time, 2^{17}/fx (see **Figures 18-2** to **18-4**).

Cautions 1. For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin.
2. When the STOP mode is released by reset, the STOP mode contents are held during reset input. However, the port pins become high-impedance.

Figure 18-1. Block Diagram of Reset Function

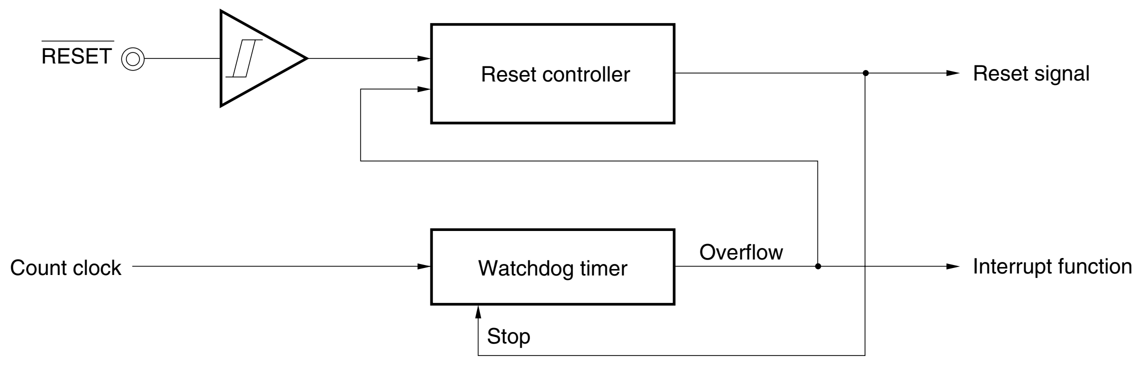


Figure 18-2. Timing of Reset by $\overline{\text{RESET}}$ Input

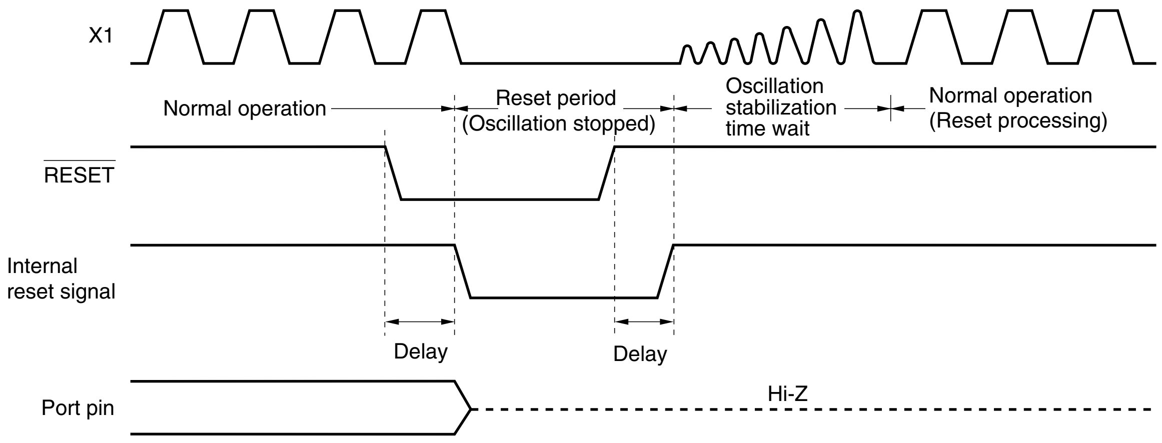


Figure 18-3. Timing of Reset Due to Watchdog Timer Overflow

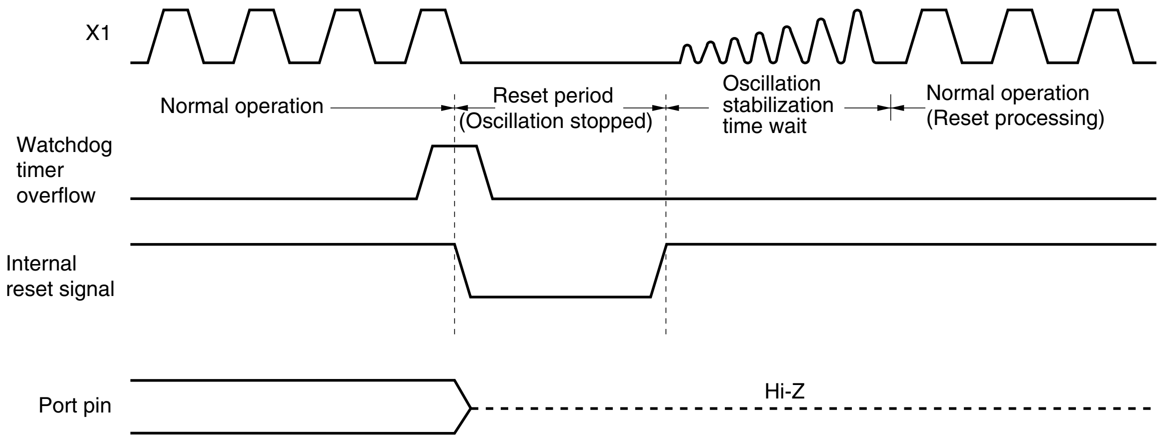
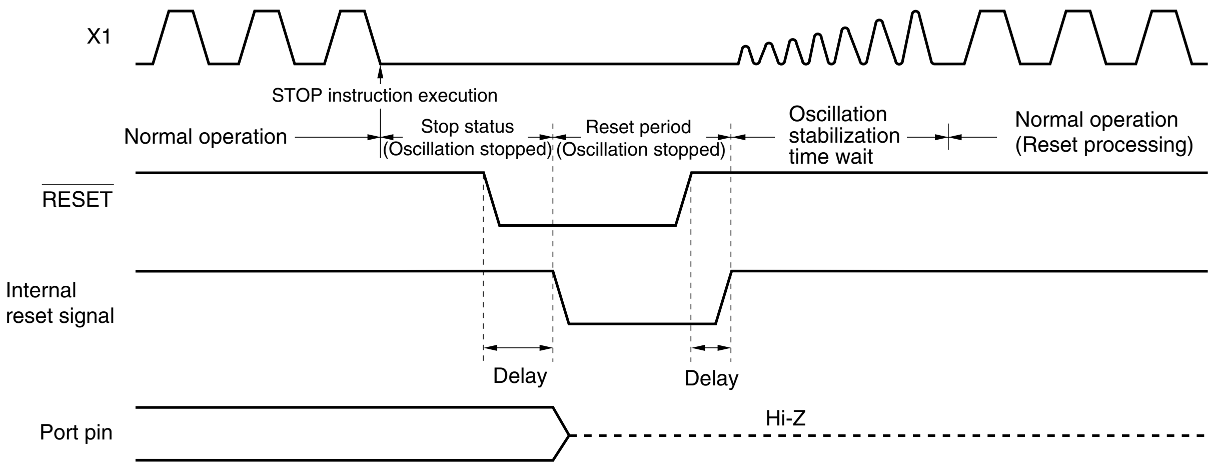


Figure 18-4. Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input



★ **Table 18-1. Hardware Statuses After Reset (1/2)**

Hardware		Status After Reset
Program counter (PC)Note 1		Contents of reset vector table (0000H, 0001H) are set.
Stack pointer (SP)		Undefined
Program status word (PSW)		02H
RAM	Data memory	UndefinedNote 2
	General-purpose registers	UndefinedNote 2
Ports (output latches)	Ports 0, 2, 3, 7 to 9 (P0, P2, P3, P7 to P9)	00H
	Ports 4 to 6 (P4 to P6)	Undefined
Port mode registers (PM0, PM2 to PM9)		FFH
Pull-up resistor option registers (PU0, PU2 to PU7)		00H
Processor clock control register (PCC)		04H
Memory size switching register (IMS) CFH		
Internal expansion RAM size switching register (IXS)		0CHNote 3
Oscillation stabilization time selection register (OSTS)		04H
16-bit timer/event counter	Timer registers 00, 01 (TM00, TM01)	0000H
	Capture/compare registers 000, 010, 001, 011 (CR000, CR010, CR001, CR011)	000H
	Capture/compare control registers 00, 01 (CRC00, CRC01)	00H
	Prescaler mode registers 00, 01 (PRM00, PRM01)	00H
	Mode control registers 00, 01 (TMC00, TMC01)	00H
	Output control registers 00, 01 (TOC00, TOC01)	00H
8-bit timer/event counter	Timer counters 50, 51, 52 (TM50, TM51, TM52)	00H
	Compare registers 50, 51, 52 (CR50, CR51, CR52)	00H
	Clock selection registers 50, 51, 52 (TCL50, TCL51, TCL52)	00H
	Mode control registers 50, 51, 52 (TMC50, TMC51, TMC52)	04H
Watch timer	Mode control register 0 (WTNM0)	00H
Watchdog timer	Clock selection register (WDCS)	00H
	Mode register (WDTM)	00H

Notes 1. During reset input 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 value is 0CH. However, be sure to use this register after setting it to 08H.

★ **Table 18-1. Hardware Statuses After Reset (2/2)**

Hardware		Status After Reset
Clock output controller	Clock output selection register (CKS)	00H
A/D converter	Conversion result registers (ADCR00, ADCR01)	Undefined
	Mode registers (ADM00, ADM01)	00H
	Analog input channel specification registers (ADS00, ADS01)	00H
Serial interface UART0	Asynchronous serial interface mode register 0 (ASIM0)	00H
	Asynchronous serial interface status register 0 (ASIS0)	00H
	Baud rate generator control register 0 (BRGC0)	00H
	Transmit shift register 0 (TXS0)	FFH
	Receive buffer register 0 (RXB0)	
Serial interface SIO3	Shift registers 30, 31 (SIO30, SIO31)	Undefined
	Operation mode registers 30, 31 (CSIM30, CSIM31)	00H
Serial interface IIC0	Transfer clock selection register 0 (IICCL0)	00H
	Shift register 0 (IIC0)	00H
	Control register 0 (IICC0)	00H
	Status register 0 (IICS0)	00H
	Function expansion register 0 (IICX0)	00H
	Slave address register 0 (SVA0)	00H
J1850 (Class 2)	Control registers 1, 2 (C2CT1, C2CT2)	00H
	Status register 1 (C2ST1)	C0H
	Status register 2 (C2ST2)	00H
	Transmit FIFO register (C2TXFIFO)	00H
	Receive FIFO register (C2RXFIFO)	
	Rise time propagation delay correction register (C2PDR)	00H
	Fall time propagation delay correction register (C2PDF)	00H
	Clock selection register (C2CLK)	00H
Interrupts	Request flag registers (IF0L, IF0H, IF1L, IF1H)	00H
	Mask flag registers (MK0L, MK0H, MK1L)	FFH
	Mask flag register (MK1H)	DFH
	Priority specification flag registers (PR0L, PR0H, PR1L, PR1H)	FFH
	External interrupt rising edge enable register (EGP)	00H
	External interrupt falling edge enable register (EGN)	00H
	Memory expansion mode register (MEM)	00H

CHAPTER 19 μPD78F0833Y

The μPD78F0833Y is provided as a flash memory product in the μPD780833Y Subseries.

In the μPD78F0833Y, the internal mask ROM of the μPD780833Y is replaced with a flash memory to which a program can be written, erased, and overwritten while mounted on the substrate. Table 19-1 lists the differences between the μPD78F0833Y and the mask ROM versions.

★ **Table 19-1. Differences Between μPD78F0833Y and Mask ROM Versions**

Item \ Version		μPD78F0833Y	μPD780833Y
Internal ROM configuration		Flash memory	Mask ROM
Internal ROM capacity		60 KB	60 KB
8-bit timer/event counters		3	
	Timer outputs	5 (8-bit PWM output: 3)	
Internal maskable interrupts		19	
IC		None	Available
V_{PP}		Available	None
Electrical specifications and soldering conditions		Refer to the data sheets of individual products.	

Caution **There are differences in noise immunity and noise radiation between the flash memory and mask ROM versions. When preproducing an application set with the flash memory version and then mass producing it with the mask ROM version, be sure to conduct sufficient evaluations on the commercial samples (CS) (not engineering samples (ES)) of the mask ROM version.**

19.1 Memory Size Switching Register

In the μPD78F0833Y, the internal memory capacity is set using the memory size switching register (IMS).
IMS is set using an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets IMS to CFH.

★ **Figure 19-1. Format of 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 setting
1	1	0	1024 bytes
Other than above			Setting prohibited

ROM3	ROM2	ROM1	ROM0	Internal ROM capacity setting
1	1	1	1	60 KB
Other than above				Setting prohibited

19.2 Internal Expansion RAM Size Switching Register

IXS sets the internal expansion RAM capacity.
IXS is set using an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets IXS to 0CH.

Caution The initial value of IXS is 0CH. Be sure to set it to 08H before use.

Figure 19-2. Format of Internal Expansion RAM Size Switching Register (IXS)

Address: FFF4H After reset: 0CH R/W

Symbol	7	6	5	4	3	2	1	0
IXS	0	0	0	IXRAM4	IXRAM3	IXRAM2	IXRAM1	IXRAM0

IXRAM4	IXRAM3	IXRAM2	IXRAM1	IXRAM0	Internal expansion RAM capacity setting
0	1	0	0	0	2048 bytes
Other than above					Setting prohibited

19.3 Flash Memory Programming

On-board writing of the flash memory (with the device mounted on target system) is supported.

On-board writing is performed after connecting a dedicated flash programmer (Flashpro III (part number FL-PR3 or PG-FP3)) to the host machine and target system.

Moreover, writing to flash memory can also be performed using a flash memory writing adapter connected to Flashpro III.

Remark FL-PR3 is a product of Naito Densei Machida Mfg. Co., Ltd.

19.3.1 Selection of communication mode

Writing to flash memory is performed using Flashpro III and serial communication. Select the communication mode for writing from Table 19-2. For the selection of the communication mode, a format like the one shown in Figure 19-3 is used. The communication modes are selected using the number of V_{PP} pulses shown in Table 19-2.

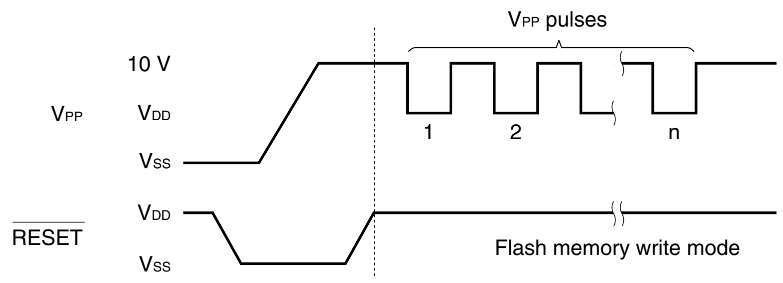
★ **Table 19-2. Communication Mode List**

Communication Method	Number of Channels	Pin Used[Note]	PG-FP3 Setting	
			Rank E	Rank I
3-wire serial I/O	2	SI30/P30 SO30/P31 $\overline{\text{SCK30}}$/P32	COMM PORT: ch-2 Number of V_{PP} pulses: 2	COMM PORT: ch-0 Number of V_{PP} pulses: 0
		SI31/P20 SO31/P21 $\overline{\text{SCK31}}$/P22	COMM PORT: ch-1 Number of V_{PP} pulses: 1	COMM PORT: ch-1 Number of V_{PP} pulses: 1

★ **Note** Shifting to the flash memory programming mode sets all pins not used for flash memory programming to the same state as immediately after reset. Therefore, all ports enter an output high-impedance state. If the external device does not acknowledge an output high-impedance state, pin handling is required.

Caution Be sure to select the number of V_{PP} pulses shown in Table 19-3 for the communication mode.

Figure 19-3. Format of Communication Mode Selection



19.3.2 Flash memory programming function

Flash memory writing is performed via command and data transmit/receive operations using the selected communication mode. The main functions are listed in Table 19-3.

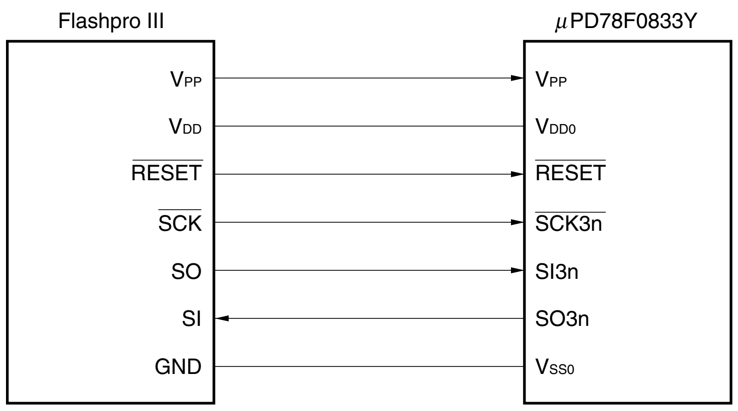
Table 19-3. Main Functions of Flash Memory Programming

Function	Description
Reset	Used to detect write stop and communication synchronization.
Batch verify	Compares the entire memory contents with the input data.
Batch contents verify	Compares the entire memory contents in the different modes.
Batch erase	Erases the entire memory contents.
Batch blank check	Checks the erase status of the entire memory.
High-speed write	Performs writing to flash memory according to the write start address and number of write data (bytes).
Continuous write	Performs successive write operations using the data input by the high-speed write operation.
Batch prewrite	Writes 00H to the entire memory.
Status	Checks the current operation mode and operation end.
Oscillation frequency setting	Inputs the resonator oscillation frequency information.
Delete time setting	Inputs the memory erase time.
Silicon signature read	Outputs the device name, memory capacity, and device block information.

★ ### 19.3.3 Flashpro III connection

The connection of the Flashpro III and the μPD78F0833Y is shown in Figure 19-4.

Figure 19-4. Connection of Flashpro III Using 3-Wire Serial I/O Mode



n = 0, 1

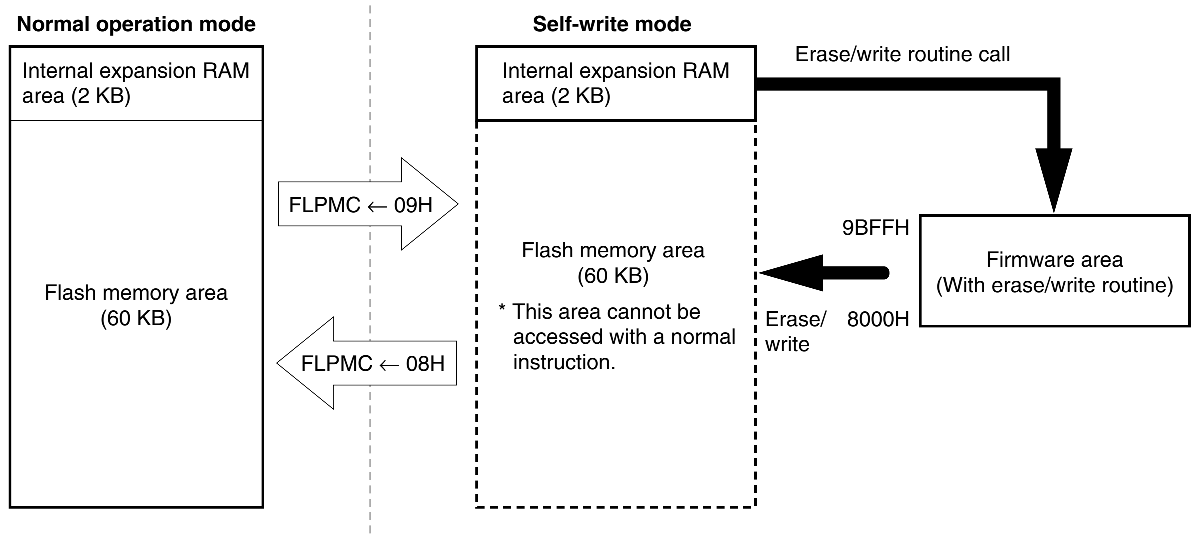
19.4 Flash Memory Programming by Self-Write

With the μPD78F0833Y, it is possible to rewrite the flash memory using a program.

19.4.1 Flash memory configuration

The configuration of the flash memory is shown in Figure 19-5.

Figure 19-5. Flash Memory Configuration


Normal operation mode
Internal expansion RAM area (2 KB)
Flash memory area (60 KB)
FLPMC ← 09H
FLPMC ← 08H
Self-write mode
Internal expansion RAM area (2 KB)
Flash memory area (60 KB)
* This area cannot be accessed with a normal instruction.
Erase/write routine call
9BFFH
8000H
Firmware area (With erase/write routine)
Erase/ write

19.4.2 Flash programming mode control register (FLPMC)

The flash programming mode control register (FLPMC) is a register for checking the operation mode selection and V_{PP} pin status.

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

$\overline{\text{RESET}}$ input sets FLPMC to 08H.

Figure 19-6. Format of Flash Programming Mode Control Register (FLMPC)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
FLPMC	0	0	0	0	1	VPP	0	FLSPM0	FF89H	08H[Note 1]	R/W[Note 2]

VPP	V_{PP} pin voltage status
0	The voltage required for erase/write is not applied to the V_{PP} pin.
1	A voltage greater than that of the V_{DD} pin is applied to the V_{PP} pin.

FLSPM0	Operation mode selection
0	Normal operation mode
1	Self-write mode

Notes 1. Bit 2 changes depending on the level of V_{PP}.

2. Bit 2 is read only.

Cautions 1. Be sure to set bits 1 and bits 4 to 7 to 0, and set bit 3 to 1.

2. The VPP bit gives the status of the voltage applied to the V_{PP} pin. If the VPP bit is 0, the voltage required for erase/write is not being applied. However, even if the VPP bit is 1, it does necessarily mean that the voltage required for erase/write is being applied. Configure the hardware so the voltage required for erase/write is applied to the V_{PP} pin.

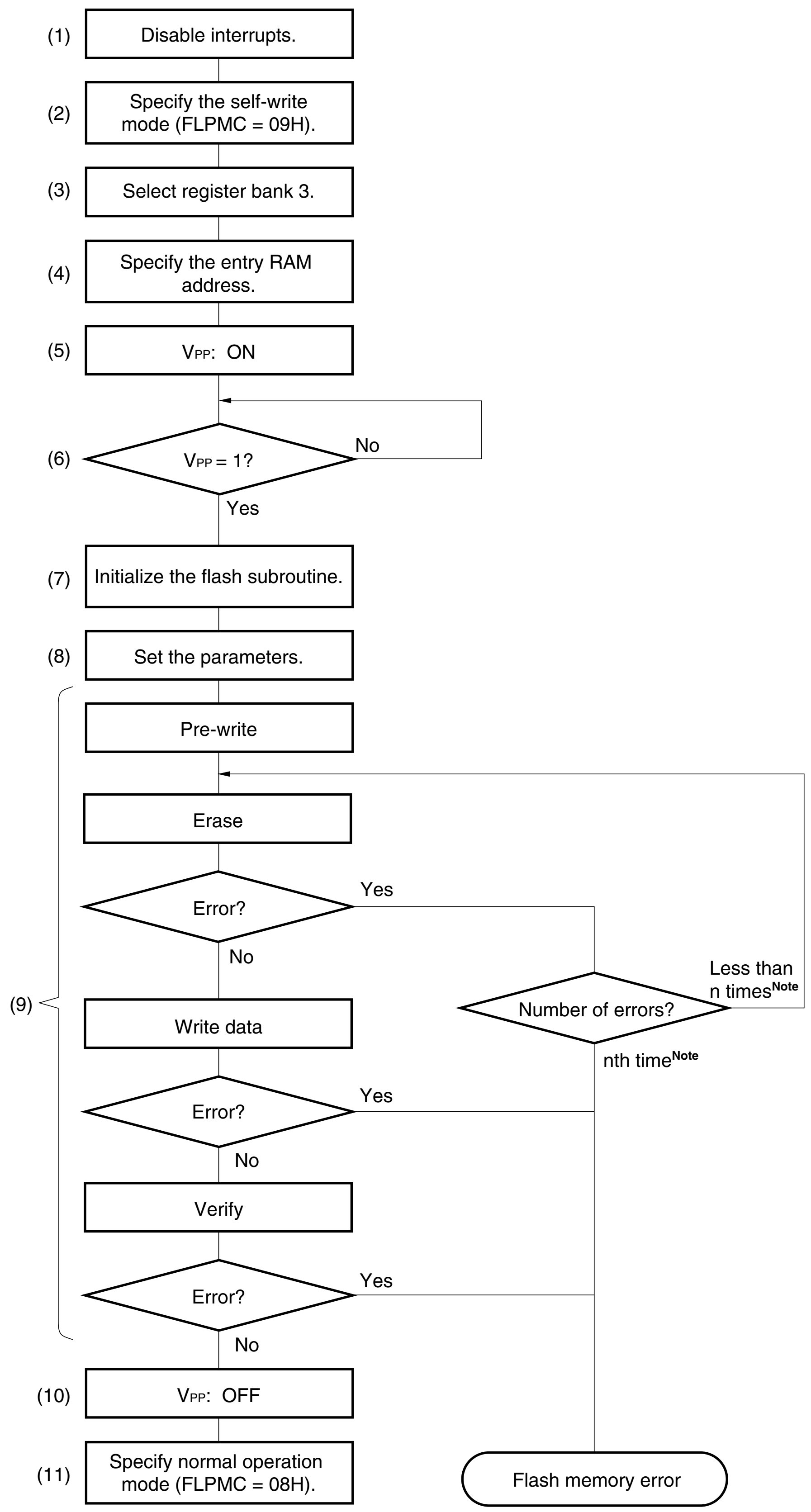
Also, if software will be used in addition to hardware to check that the voltage required for erase/write is being applied, provide an external hardware detector and use its output.

19.4.3 Self-write procedure

The procedure for performing self-write is shown below (see **Figure 19-7**).

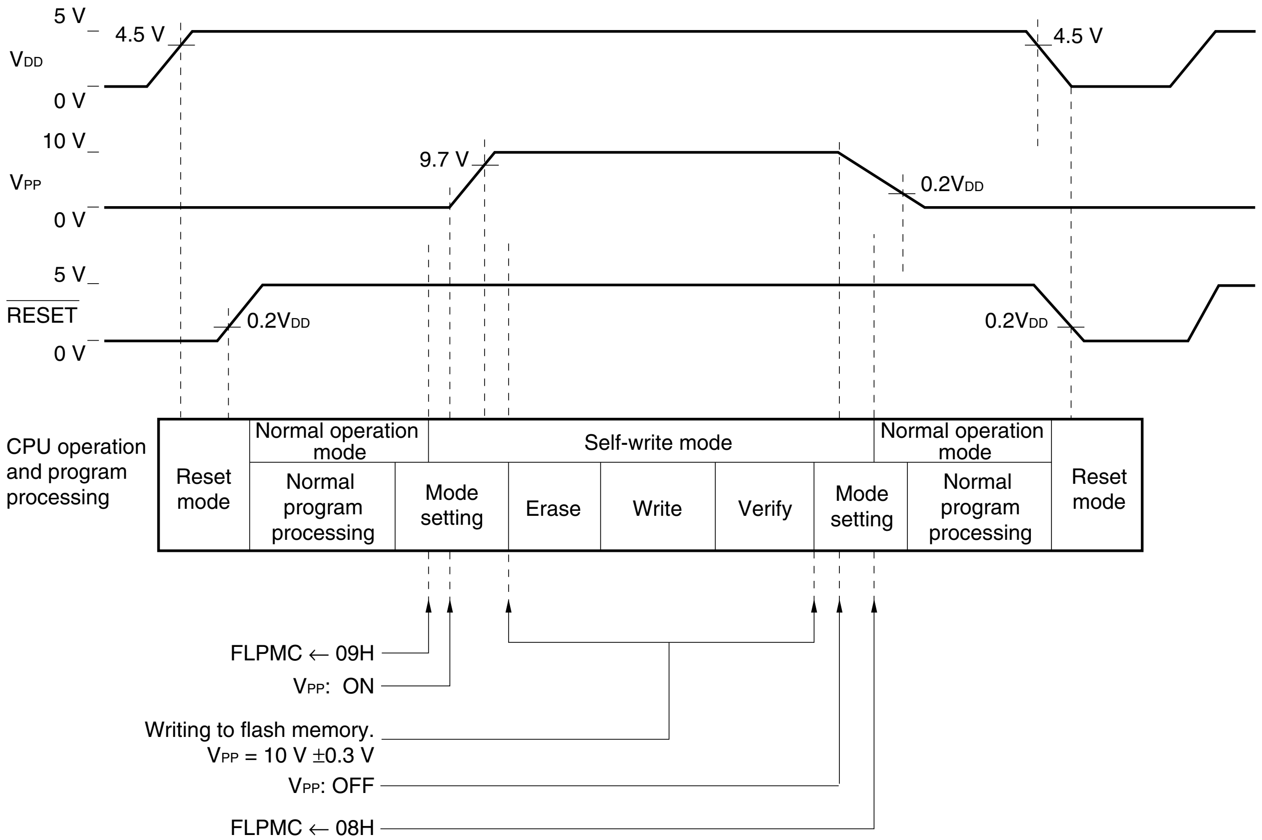
(1) Disable interrupts.
(2) Designate the self-write mode (FLPMC = 09H).
(3) Select register bank 3.
(4) Specify the start address of the entry RAM for the HL register.
(5) V_{PP}: ON (ON signal for voltage IC)
(6) Check the V_{PP} level.
(7) Initialize the flash subroutine.
(8) Set the parameters.
(9) Control the flash memory (erase, write, etc.).
(10) V_{PP}: OFF (OFF signal for voltage IC)
(11) Designate the normal operation mode (FLPMC = 08H).

Figure 19-7. Self-Write Flowchart



Note Differs depending on the user program.

Figure 19-8. Self-Write Timing



19.4.4 CPU resources

The CPU resources used during self-write are as follows:

- Register bank: BANK3 (8 bytes)
  B register: Status flag
  C register: Function number
  HL register: Entry RAM area starting address

- Stack area: Maximum 16 bytes

- Write data storage area: 1 to 256 bytes

- Entry RAM area: 32 bytes
  RAM area used by the self-write subroutines.
  Can be specified by the user using the HL register.

- Status flag

7	6	5	4	3	2	1	0
Parameter setting error	—	—	Verify error	Write error	—	Blank check error	—

19.4.5 Entry RAM area

A description of the entry RAM area is shown in Table 19-4.

Table 19-4. Entry RAM Area

Offset Value	Description
+0	Reserved area (1 byte)
+1	Reserved area (1 byte)
+2	Flash memory start address (2 bytes)
+4	Flash memory end address (2 bytes)
+6	No. of bytes written to flash memory (1 byte)
+7	Write time data (1 byte)
+8	Erase time data (3 bytes)
+11	Reserved area (3 bytes)
+14	Write data storage buffer for address (2 bytes)
+16	Total block number (1 byte)
+17	Total area number (1 byte)
+18 . .	Reserved area (14 bytes)

Example When the value of the HL register of register bank 3 is 0FD00H
0FD00H: Status
0FD02H: Flash memory start address
0FD06H: Number of bytes written to flash memory
.
.

Next, the entry RAM area will be explained in detail.

(a) Flash memory start address

This is the flash memory address value used by the _FlashByteWrite subroutine.

(b) Flash memory end address

This is the flash memory address value used by the _FlashGetInfo subroutine.

(c) Number of bytes written in flash memory

Area number and number of bytes written in the flash memory.

(d) Write time data

Set the following values according to the operating frequency.

f_X (MHz)	Setting Value
1.00 to 1.28	20H
1.29 to 2.56	40H
2.57 to 5.12	60H
5.13 to 8.38	80H

(e) Erase time data

Setting value = Erase time (s) × Operating frequency/2^9 + 1

(Erase time range: 0.5 to 20 seconds)

Example Erase time: 2 seconds, operating frequency: 4.19 MHz

Setting value = 2 × 4194304/512 + 1

= 16385 (decimal)

= 4001H (hexadecimal)

(f) Write data storage buffer top address

This area contains the top address of the write data storage buffer area. The RAM data (write data), specified by the address data in this area, is written to the flash memory (_FlashByteWrite subroutine). The data in this area is specified as the top address and it is possible to specify up to a maximum of 256 bytes of write data.

(g) Total block number

This is the total flash memory block number used by the _FlashGetInfo subroutine.

(h) Total area number

This is the total flash memory area used by the _FlashGetInfo subroutine.

19.4.6 Self-write subroutines

The self-write subroutines and their functions are shown in Table 19-5 below.

Table 19-5. List of Self-Write Subroutines

Function Number		Subroutine Name	Function
Decimal	Hexadecimal		
0	00H	_FlashEnv	Initializes the flash subroutine.
1	01H	_FlashSetEnv	Sets the parameters.
2	02H	_FlashGetInfo	Reads flash memory data
16	10H	_FlashAreaBlankCheck	Performs a blank check of a specified area.
32	20H	_FlashAreaPreWrite	Performs prewrite for a specified area.
48	30H	_FlashAreaErase	Erases a specified area.
80	50H	_FlashByteWrite	Writes continuously in byte units.
96	60H	_FlashAreaIVerify	Performs internal verification of a specified area.

(1) _FlashEnv subroutine

[Function]

Initializes the flash subroutine.

[Argument]

Entry RAM address ...... 2 bytes (HL register)

[Return value]

None

[Register/memory status after called]

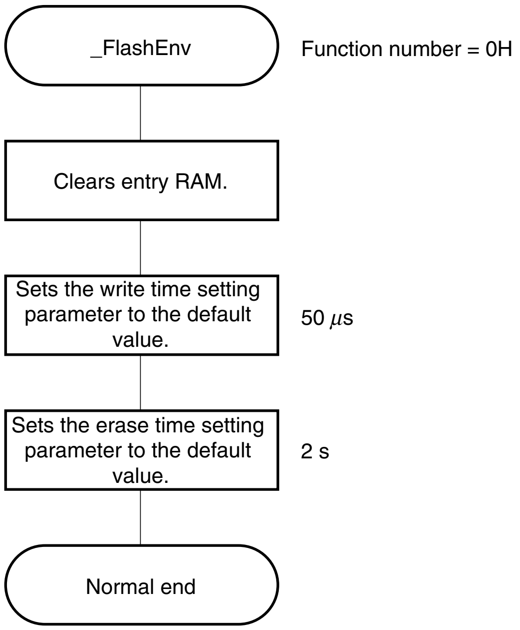
Entry RAM address

[Call example]

When the entry RAM address = 0FC30H

```
                DI
                SET1    FLSPM0
  LOOP:         BF      VPP, $LOOP
                SEL     RB3
                MOVW    HL, #0FC30H         ; Entry RAM address
  ; * * * * * * * * * * Initialization * * * * * * * * * *
                MOV     C, #0H              ; FlashEnv (function number setting)
                CALL    !8100H
                        .
                        .
                        .
```

[Flowchart]


_FlashEnv
Function number = 0H
Clears entry RAM.
Sets the write time setting parameter to the default value.
50 μs
Sets the erase time setting parameter to the default value.
2 s
Normal end

(2) _FlashSetEnv subroutine

[Function]

Sets the parameters.

[Argument]

Write time data: 2 bytes (offset value: +7)

Erase time data: 3 bytes (offset value: +8 to 10)

[Return value]

Status (B register or offset value: +0)

00H: Normal end

80H: Parameter setting error

[Register/memory status after called]

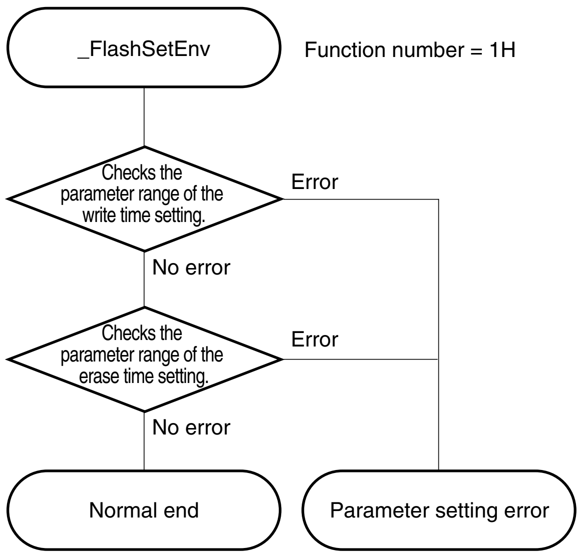
Entry RAM address, write time data, erase time data

[Call example]

When the entry RAM address = 0FC30H

```
        MOV     A, #20H         ; Write time data
        MOV     !0FC37H, A
        MOV     A, #30H         ; Erase time data
        MOV     !0FC38H, A      ; 8 s : 13130H
        MOV     A, #31H
        MOV     !0FC39H, A
        MOV     A, #01H
        MOV     !0FC3AH, A
;
        MOV     C, #1H          ;  FlashSetEnv (function number setting)
        CALL    !8100H
                .
                .
                .
```

[Flowchart]


_FlashSetEnv
Function number = 1H
Checks the parameter range of the write time setting.
Error
No error
Checks the parameter range of the erase time setting.
Error
No error
Normal end
Parameter setting error

(3) _FlashGetInfo subroutine

[Function]

Reads flash memory data

[Argument]

Option number (=0, 1, 2, 3): 1 byte (offset value: +6)

0: Flash firmware version
1: Area number and block number
2: Area 0 end address
3: Area 1 end address

[Return Value]

Status (B register or offset value: +6)

00H: Normal end
80H: Optional number specification error

One of the items below can be specified by an optional number.

- Flash firmware version (A register or offset value: +6)
- Area number and block number (AX register or offset value: +16 to 17)
- Specification area end address (AX register or offset value: +4 to 5)

[Register/memory status after called]

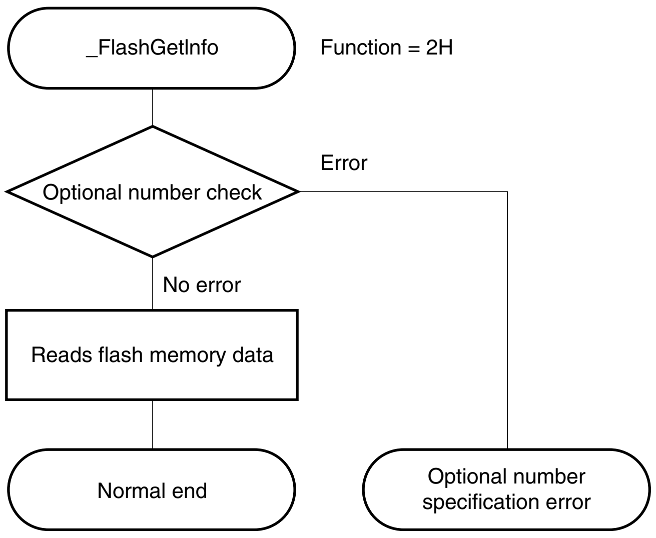
Entry RAM address optional number (flash firmware version)

[Call example]

When the entry RAM address = 0FC30H

```
MOV     A, #02H      ; Area 1 end address information
MOV     !0FC36H, A
MOV     C, #02H      ; FlashGetInfo (Function number setting)
CALL    !8100H
        .
        .
        .
```

[Flowchart]



(4) _FlashAreaBlankCheck subroutine

[Function]

Performs a blank check of a specified area.

[Argument]

Area number (= 0, 1): 1 byte (offset value: +6)
  0: Blank check of area 0000H to 7FFFH
  1: Blank check of area 8000H to EFFFH

[Return value]

Status (B register or offset value: +0)
  00H: Normal end
  02H: Blank check error
  80H: Area number specification error

[Register/memory status after called]

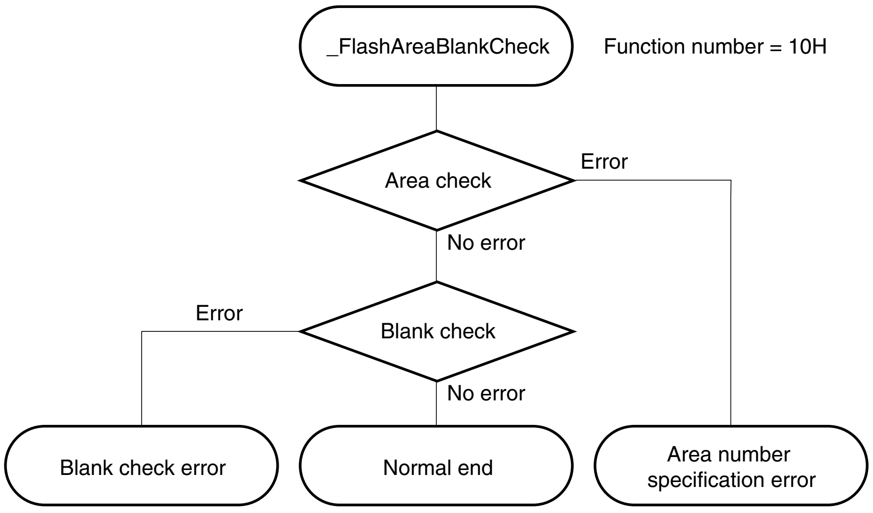
Entry RAM address, area number

[Call example]

When the entry RAM address = 0FC30H

```
        MOV     A, #01H         ; Specifies area 1
        MOV     !0FC36H, A
        MOV     C, #10H         ; FlashAreaBlankCheck (function number setting)
        CALL    !8100H
                .
                .
                .
```

[Flowchart]



(5) _FlashAreaPreWrite subroutine

[Function]

Performs prewrite for a specified area (writes 00H to a specified area).

[Argument]

Area number (= 0, 1): 1 byte (offset value: +6)
  0: Prewrites area 0000H to 7FFFH.
  1: Prewrites area 8000H to EFFFH.

[Return value]

Status (B register or offset value: +0)
  00H: Normal end
  08H: Write error
  80H: Area number specification error

[Register/memory status after called]

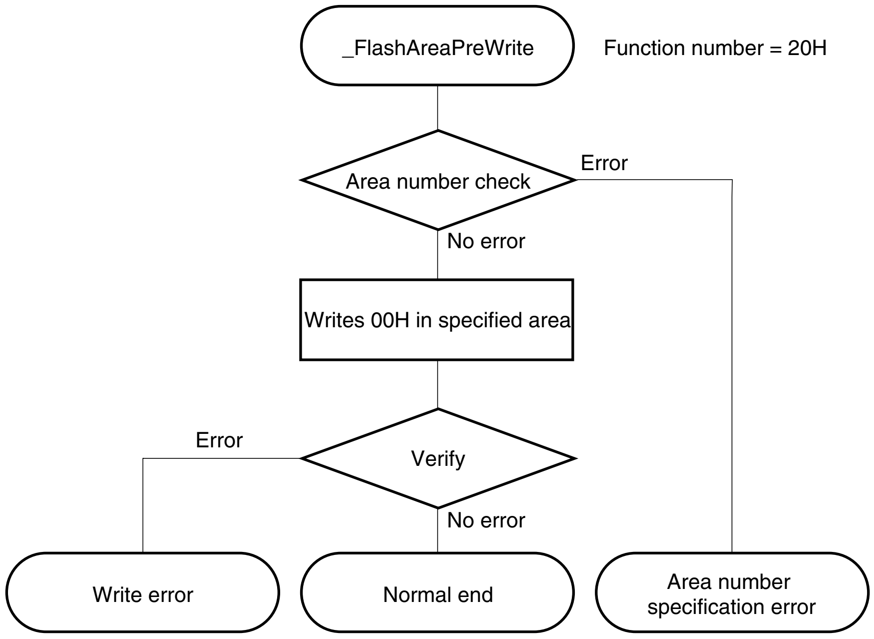
Entry RAM address, area number

[Call example]

When the entry RAM address = 0FC30H

```
        MOV     A, #1H          ; Specifies 8000H to EFFFH.
        MOV     !0FC36H, A
  ;
        MOV     C, #20H         ; FlashAreaPreWrite (function number setting)
        CALL    !8100H
                .
                .
                .
```

[Flowchart]



(6) _FlashAreaErase subroutine

[Function]

Erases a specified area.

[Argument]

Area number (= 0, 1): 1 byte (offset value: +6)
0: Erases area 0000H to 7FFFH.
1: Erases area 8000H to EFFFH.

[Return value]

Status (B register or offset value: +0)
00H: Normal end
02H: Blank check error
80H: Area number specification error

[Register/memory status after called]

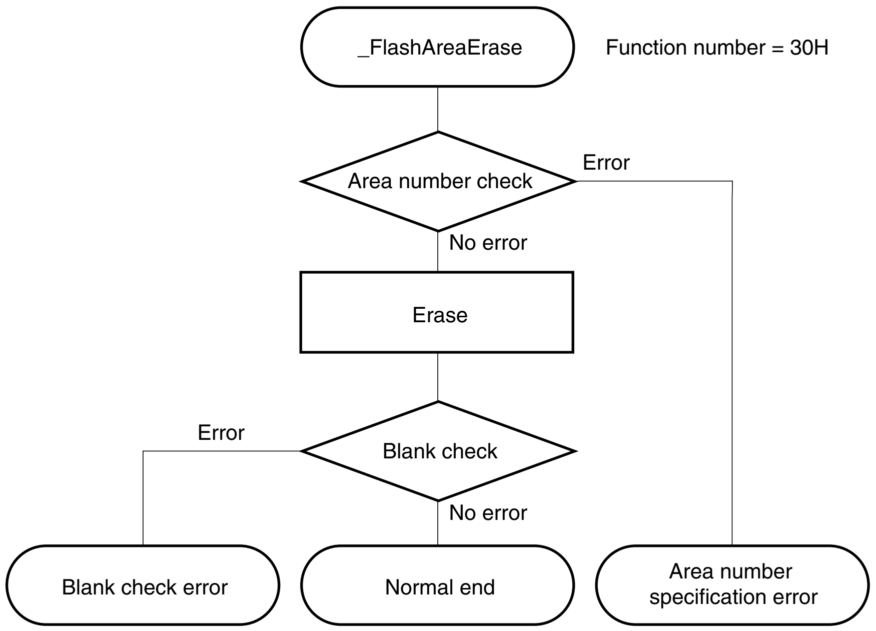
Entry RAM address, area number

[Call example]

When the entry RAM address = 0FC30H

```
      MOV     A, #1H         ; Specifies 8000H to EFFFH.
      MOV     !0FC36H, A
;
      MOV     C, #30H        ; FlashAreaErase (function number setting)
      CALL    !8100H
              .
              .
              .
```

[Flowchart]



(7) _FlashByteWrite subroutine

[Function]

Writes continuously in byte units.

[Argument]

Flash memory write start address: 2 bytes (offset value: +2)
Number of bytes[Note] written to flash memory: 1 byte (offset value: +6)
Write data storage buffer top address: 2 bytes (offset value: +14)

Note If 0 is set, it is possible to set a maximum of 256 bytes.

[Return value]

Status (B register or offset value: +0)
00H: Normal end
08H: Write error
80H: Write address error

[Register/memory status after called]

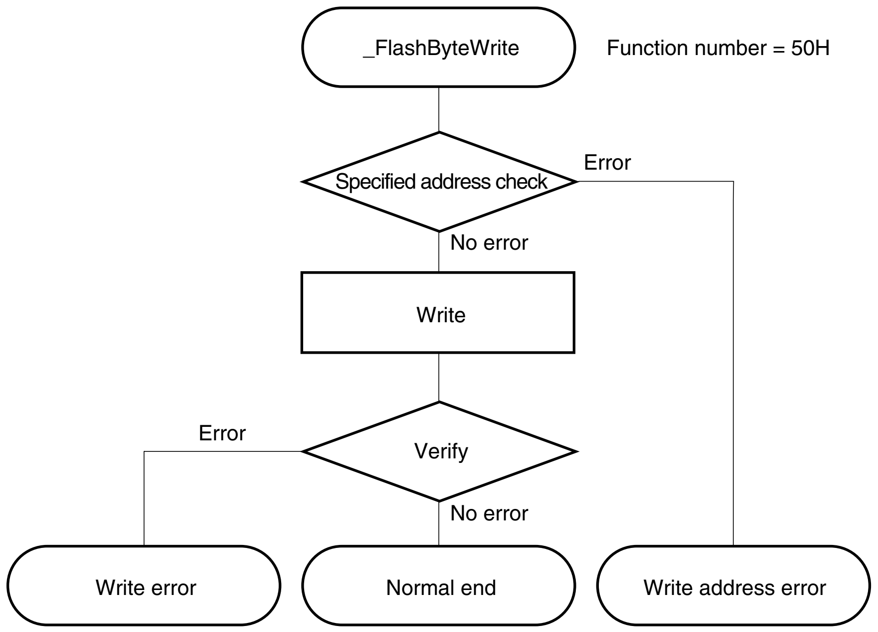
Entry RAM address, number of bytes written in flash memory
The flash memory write start address is updated to the address at the end of writing.

[Call example]

When the entry RAM address = 0FC30H

```
     MOVW    AX, #0FD00H    ; Write data storage buffer top address
     MOVW    !0FC3EH, AX
     MOVW    AX, #200H      ; Flash memory write start address
     MOVW    !0FC32H, AX
     MOV     A, #0H         ; Number of bytes written to flash memory (256 bytes)
     MOV     !0FC36H, A
;
     MOV     C, #50H        ; FlashByteWrite (function number setting)
     CALL    !8100H
             .
             .
             .
```

[Flowchart]


_FlashByteWrite
Function number = 50H
Specified address check
Error
No error
Write
Verify
Error
No error
Write error
Normal end
Write address error

(8) _FlashArealVerify subroutine

[Function]

Performs internal verification of a specified area (reads the flash memory of a specified area in a different mode, and makes a comparison).

[Argument]

Area number (= 0, 1): 1 byte (offset value: +6)

0: Performs internal verification of area 0000H to 7FFFH.

1: Performs internal verification of area 8000H to EFFFH.

[Return value]

Status (B register or offset value: +0)

00H: Normal end

10H: Verify error

80H: Area number specification error

[Register/memory status after called]

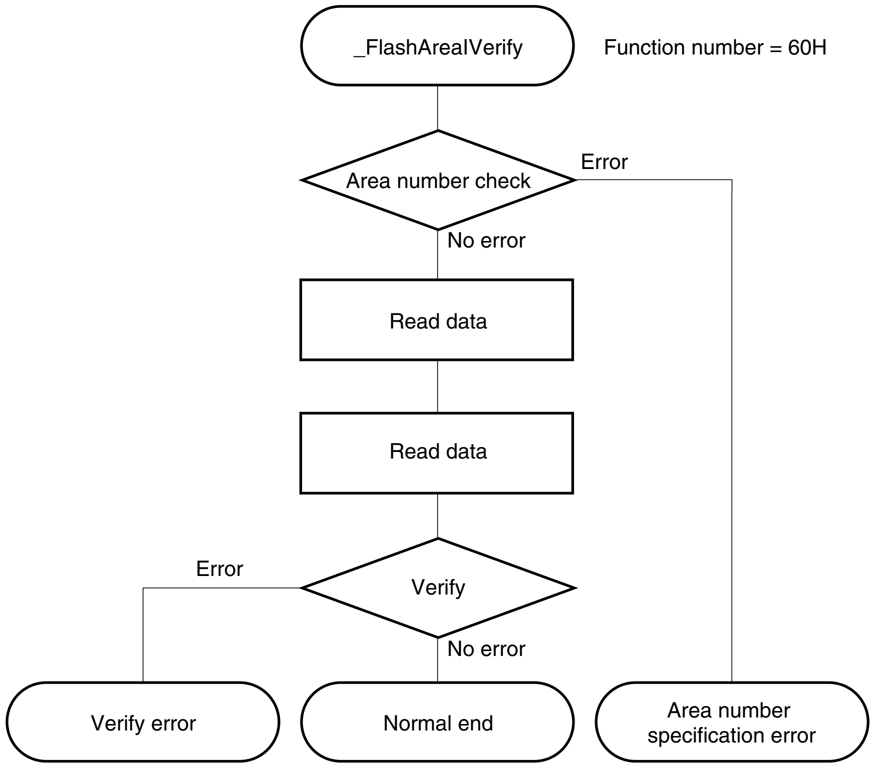
Entry RAM address, area number

[Call example]

When the entry RAM address = 0FC30H

```
        MOV     A, #01H       ; Area 1 specification
        MOV     !0FC36H, A
;
        MOV     C, #60H       ; FlashArealVerify (function number setting)
        CALL    !8100H
                .
                .
                .
```

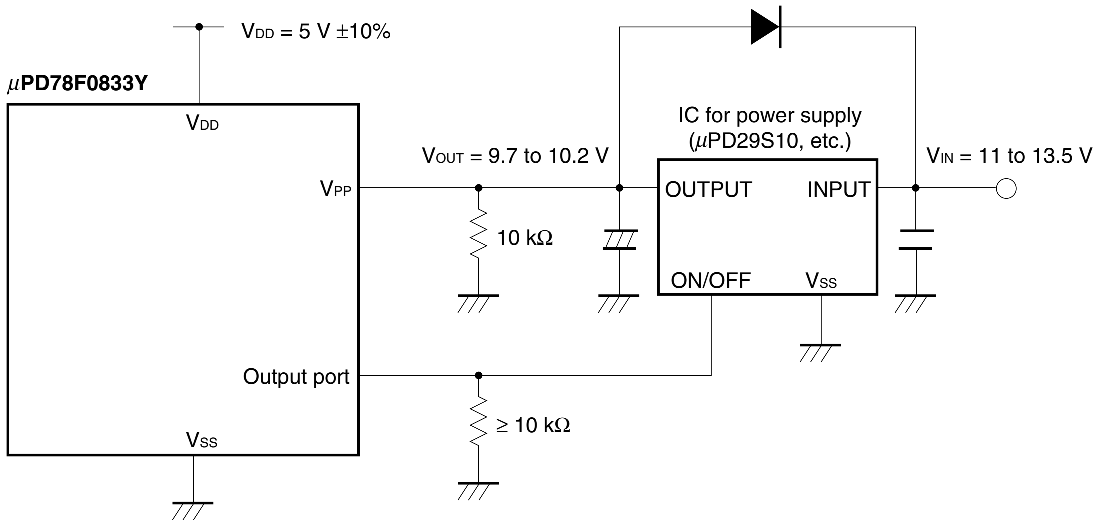
[Flowchart]



19.4.7 Self-write circuit configuration

The configuration of the self-write circuit is shown in Figure 19-9.

Figure 19-9. Configuration of Self-Write Circuit



CHAPTER 20 INSTRUCTION SET

This chapter lists the instruction set of the μPD780833Y Subseries. For details of the operation and operation code of each instruction, refer to the separate document **78K/0 Series User's Manual Instruction (U12326E)**.

20.1 Conventions

20.1.1 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 key words and must be described 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, write an appropriate numeric value or a label. When using a label, be sure to write the #, !, $, and [ ] symbols.

For the 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 20-1. 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 addresses 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 addresses only)
word	16-bit immediate data or label
byte	8-bit immediate data or label
bit	3-bit immediate data or label
RBn	RB0 to RB3

Note Addresses from FFD0H to FFDFH cannot be accessed with these operands.

Remark For special function register symbols, refer to **Table 3-2 Special Function Register List**.

20.1.2 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 selection flag
IE:	Interrupt request enable flag
NMIS:	Non-maskable interrupt servicing 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)
——:	Inverted data
addr16:	16-bit immediate data or label
jdisp8:	Signed 8-bit data (displacement value)

20.1.3 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

20.2 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 instruction with no data access is executed.
2. When an area except the internal high-speed RAM area is accessed.
3. Except when r = A

Remark One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

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 instruction with no data access is executed.
2. When an area except the internal high-speed RAM area is accessed.
3. Only when rp = BC, DE or HL
4. Except when r = A

Remark One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

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 instruction with no data access is executed.
2. When an area except the internal high-speed RAM area is accessed.
3. Except when r = A

Remark One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

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 instruction with no data access is executed.
2. When an area except the internal high-speed RAM area is accessed.
3. Except when r = A

Remark One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

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 $\leftarrow$ AX + word	×	×	×
	SUBW	AX, #word	3	6	–	AX, CY $\leftarrow$ AX – word	×	×	×
	CMPW	AX, #word	3	6	–	AX – word	×	×	×
Multiply/ divide	**MULU**	X	2	16	–	AX $\leftarrow$ A × X			
	DIVUW	C	2	25	–	AX (Quotient), C (Remainder) $\leftarrow$ AX ÷ C			
Increment/ decrement	**INC**	r	1	2	–	r $\leftarrow$ r + 1	×	×	
		saddr	2	4	6	(saddr) $\leftarrow$ (saddr) + 1	×	×	
	DEC	r	1	2	–	r $\leftarrow$ r – 1	×	×	
		saddr	2	4	6	(saddr) $\leftarrow$ (saddr) – 1	×	×	
	INCW	rp	1	4	–	rp $\leftarrow$ rp + 1			
	DECW	rp	1	4	–	rp $\leftarrow$ rp – 1			
Rotate	**ROR**	A, 1	1	2	–	(CY, $A_7 \leftarrow A_0$, $A_{m-1} \leftarrow A_m$) × 1 time			×
	ROL	A, 1	1	2	–	(CY, $A_0 \leftarrow A_7$, $A_{m+1} \leftarrow A_m$) × 1 time			×
	RORC	A, 1	1	2	–	(CY $\leftarrow A_0$, $A_7 \leftarrow$ CY, $A_{m-1} \leftarrow A_m$) × 1 time			×
	ROLC	A, 1	1	2	–	(CY $\leftarrow A_7$, $A_0 \leftarrow$ CY, $A_{m+1} \leftarrow A_m$) × 1 time			×
	ROR4	[HL]	2	10	12	$A_{3-0} \leftarrow (HL)_{3-0}$, $(HL)_{7-4} \leftarrow A_{3-0}$, $(HL)_{3-0} \leftarrow (HL)_{7-4}$			
	ROL4	[HL]	2	10	12	$A_{3-0} \leftarrow (HL)_{7-4}$, $(HL)_{3-0} \leftarrow A_{3-0}$, $(HL)_{7-4} \leftarrow (HL)_{3-0}$			
BCD adjust	**ADJBA**		2	4	–	Decimal Adjust Accumulator after Addition	×	×	×
	ADJBS		2	4	–	Decimal Adjust Accumulator after Subtract	×	×	×
Bit manipu- late	**MOV1**	CY, saddr.bit	3	6	7	CY $\leftarrow$ (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY $\leftarrow$ sfr.bit			×
		CY, A.bit	2	4	–	CY $\leftarrow$ A.bit			×
		CY, PSW.bit	3	–	7	CY $\leftarrow$ PSW.bit			×
		CY, [HL].bit	2	6	7	CY $\leftarrow$ (HL).bit			×
		saddr.bit, CY	3	6	8	(saddr.bit) $\leftarrow$ CY			
		sfr.bit, CY	3	–	8	sfr.bit $\leftarrow$ CY			
		A.bit, CY	2	4	–	A.bit $\leftarrow$ CY			
		PSW.bit, CY	3	–	8	PSW.bit $\leftarrow$ CY	×	×	
		[HL].bit, CY	2	6	8	(HL).bit $\leftarrow$ CY			

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access is executed.

2. When an area except the internal high-speed RAM area is accessed.

Remark One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Bit manipu-late	**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 ← $\overline{\text{CY}}$			×

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access is executed.

2. When an area except the internal high-speed RAM area is accessed.

Remark One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

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) ← $(PC + 2)_L$, PC_{15-11} ← 00001, PC_{10-0} ← addr11, SP ← SP – 2			
	CALLT	[addr5]	1	6	–	(SP – 1) ← $(PC + 1)_H$, (SP – 2) ← $(PC + 1)_L$, PC_H ← (00000000, addr5 + 1), PC_L ← (00000000, addr5), SP ← SP – 2			
	BRK		1	6	–	(SP – 1) ← PSW, (SP – 2) ← $(PC + 1)_H$, (SP – 3) ← $(PC + 1)_L$, PC_H ← (003FH), PC_L ← (003EH), SP ← SP – 3, IE ← 0			
	RET		1	6	–	PC_H ← (SP + 1), PC_L ← (SP), SP ← SP + 2			
	RETI		1	6	–	PC_H ← (SP + 1), PC_L ← (SP), PSW ← (SP + 2), SP ← SP + 3, NMIS ← 0	R	R	R
	RETB		1	6	–	PC_H ← (SP + 1), PC_L ← (SP), PSW ← (SP + 2), SP ← SP + 3	R	R	R
Stack manipu-late	**PUSH**	PSW	1	2	–	(SP – 1) ← PSW, SP ← SP – 1			
		rp	1	4	–	(SP – 1) ← rp_H, (SP – 2) ← rp_L, SP ← SP – 2			
	POP	PSW	1	2	–	PSW ← (SP), SP ← SP + 1	R	R	R
		rp	1	4	–	rp_H ← (SP + 1), rp_L ← (SP), SP ← SP + 2			
	MOVW	SP, #word	4	–	10	SP ← word			
		SP, AX	2	–	8	SP ← AX			
		AX, SP	2	–	8	AX ← SP			
Uncondi-tional branch	**BR**	!addr16	3	6	–	PC ← addr16			
		$addr16	2	6	–	PC ← PC + 2 + jdisp8			
		AX	2	8	–	PC_H ← A, PC_L ← 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 instruction with no data access is executed.
2. When an area except the internal high-speed RAM area is accessed.

Remark One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

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 instruction with no data access is executed.

2. When an area except the internal high-speed RAM area is accessed.

Remark One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

20.3 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

Second Operand First 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 when r = A

(2) 16-bit instructions

MOVW, XCHW, ADDW, SUBW, CMPW, PUSH, POP, INCW, DECW

Second Operand / First 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

Second Operand / First 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

Second Operand / First 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

APPENDIX A   DEVELOPMENT TOOLS

The following development tools are available for the development of systems that employ the μPD780833Y Subseries.

Figure A-1 shows the development tool configuration.

Support of PC98-NX Series

Unless otherwise specified, products that operate in IBM PC/AT™ or compatibles can operate in the PC98-NX series. When using the PC98-NX series, refer to the descriptions for IBM PC/AT™ or compatibles.

Windows

Unless otherwise specified, “Windows” refers the following OSs.

- Windows 3.1
- Windows 95, 98, 2000
- Windows NT™ Ver.4.0

Figure A-1. Development Tool Configuration (1/2)

(1) When using the in-circuit emulator IE-78K0-NS

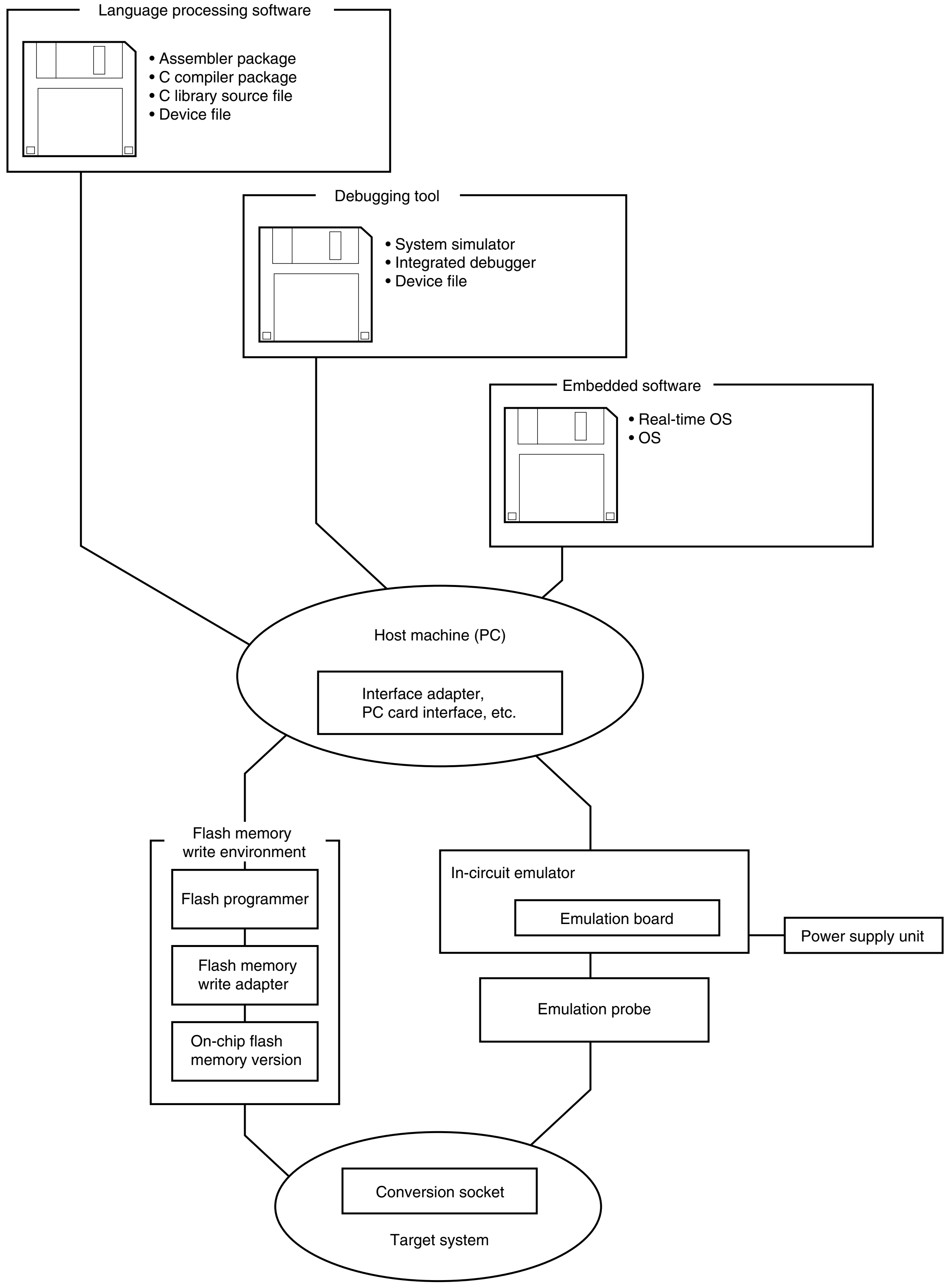
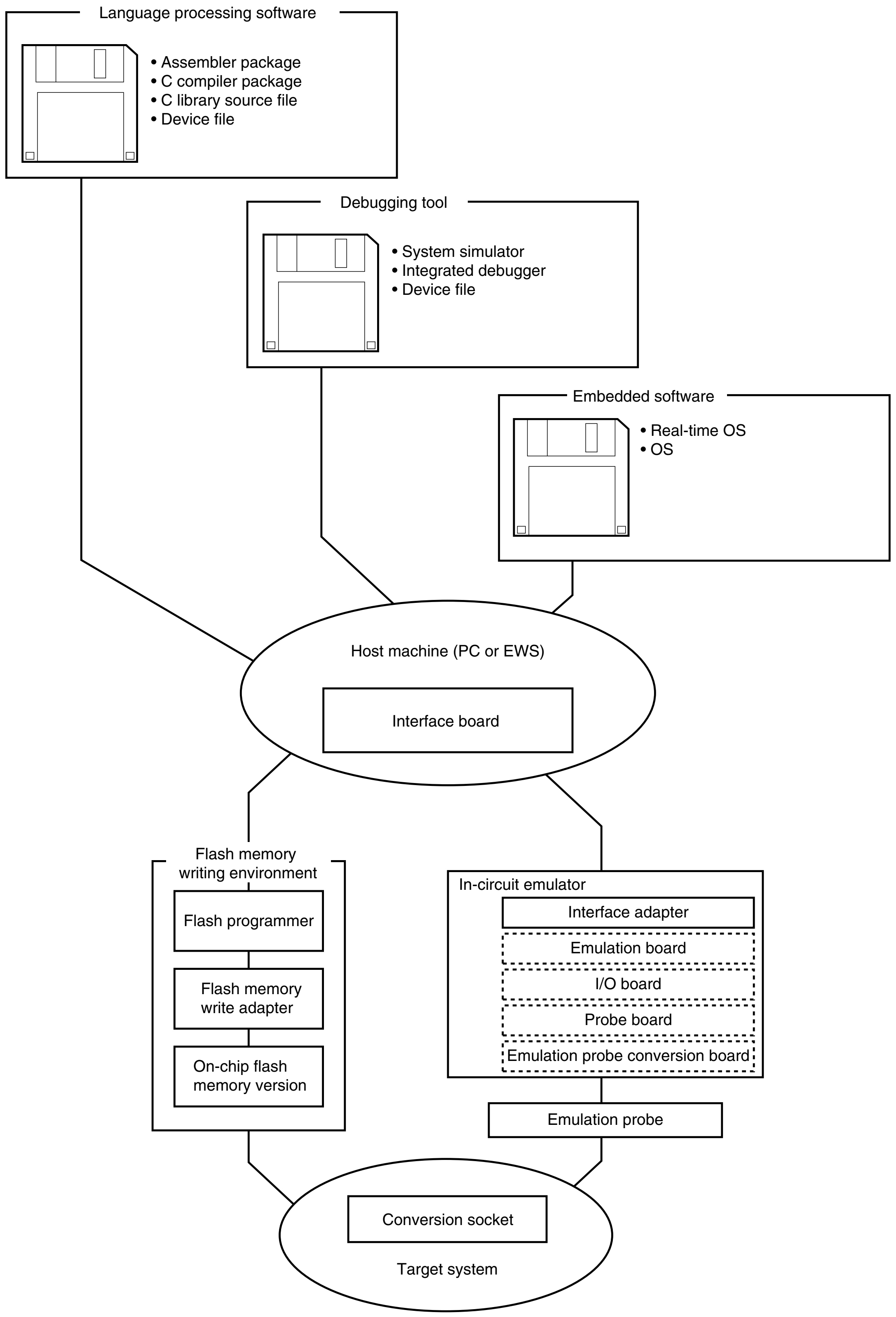


Figure A-1. Development Tool Configuration (2/2)

(2) When using the in-circuit emulator IE-78001-R-A



Remark Items in broken line boxes differ according to the development environment. Refer to **A.3.1. Hardware**.

A.1 Language Processing Software

★ SP78K0 78K/0 Series software package	This is a software package that includes the development tools common to the 78K/0 Series.
	Part number: μS××××SP78K0
RA78K0 Assembler package	This assembler converts programs written in mnemonics into object codes executable by a microcontroller. Further, this assembler is provided with functions capable of automatically creating symbol tables and branch instruction optimization. This assembler should be used in combination with a device file (DF780833) (sold separately). **<Precaution when using RA78K0 in PC environment>** This assembler package is a DOS-based application. It can also be used in Windows, however, by using the Project Manager (included in the assembler package) on Windows.
	Part number: μS××××RA78K0
CC78K0 C compiler package	This compiler converts programs written in C language into object codes executable by a microcontroller. This compiler should be used in combination with an optical assembler package and device file (DF780833). **<Precaution when using CC78K0 in PC environment>** This C compiler package is a DOS-based application. It can also be used in Windows, however, by using the Project Manager (included in the assembler package) on Windows.
	Part number: μS××××CC78K0
DF780833**Note** Device file	This file contains information peculiar to the device. This device file should be used in combination with tool (RA78K0, CC78K0, SM78K0, ID78K0-NS, and ID78K0) (sold separately). The corresponding OS and host machine differ depending on the tool used.
	Part number: μS××××DF780833
CC78K0-L C library source file	This is a source file of functions configuring the object library included in the C compiler package (CC78K0). This file is required to match the object library included in the C compiler package to the user's specifications.
	Part number: μS××××CC78K0-L

Note The DF780833 can be used commonly for the RA78K0, CC78K0, SM78K0, ID78K0-NS, and ID78K0.

Remark ×××× in the part number differs depending on the host machine and OS used.

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××SP78K0

××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese)	CD-ROM
BB17		Windows (English)	

μS××××RA78K0
μS××××CC78K0

××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese)	3.5-inch 2HD FD
BB13		Windows (English)	
AB17		Windows (Japanese)	CD-ROM
BB17		Windows (English)	
3P17	HP9000 Series 700™	HP-UX™ (Rel. 10.10)	
3K17	SPARCstation™	SunOS™ (Rel. 4.1.4), Solaris (Rel. 2.5.1)	

μS××××DF780833
μS××××CC78K0-L

××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese)	3.5-inch 2HD FD
BB13		Windows (English)	
3P16	HP9000 Series 700™	HP-UX (Rel. 10.10)	DAT
3K13	SPARCstation	SunOS™ (Rel. 4.1.4)	3.5-inch 2HD FD
3K15		Solaris (Rel. 2.5.1)	1/4-inch CGMT

A.2 Flash Memory Writing Tools

Flashpro III (part number FL-PR3, PG-FP3) Flash programmer	Flash programmer dedicated to microcontrollers with on-chip flash memory.
FA-80GC Flash memory writing adapter	Flash memory writing adapter used connected to the Flashpro III. (80-pin plastic QFP (GC-8BT type))
Flashpro III controller	This is a program controlled by a personal computer. It is included in the Flashpro III and runs on Windows 95 or other Windows OSs.

Remark The FL-PR3 and FA-80GC are manufactured by Naito Densei Machida Mfg. Co., Ltd.
For more information, consult Naito Densei Machida Mfg. Co., Ltd. (TEL: +8-45-475-4191)

A.3 Debugging Tools

A.3.1 Hardware (1/2)

(1) When using the in-circuit emulator IE-78K0-NS(-A)

IE-78K0-NS(-A) In-circuit emulator		This in-circuit emulator is used to debug hardware and software when developing application systems using a 78K/0 Series product. It supports an integrated debugger (ID78K0-NS). This emulator should be used in combination with a power supply unit, emulation probe, and interface adapter, which is required to connect this emulator to the host machine.
IE-70000-MC-PS-B Power supply unit		This adapter is used for supplying power from a socket of 100 V to 240 V AC.
IE-70000-98-IF-C Interface adapter		This adapter is required when using a PC-9800 series computer (except notebook PC) as the IE-78K0-NS host machine (C-bus supported).
IE-70000-CD-IF-A PC card interface		This is the PC card and interface cable required when using a notebook PC as the IE-78K0-NS host machine (PCMCIA-socket supported).
IE-70000-PC-IF-C Interface adapter		This adapter is required when using an IBM PC compatible computer as the IE-78K0-NS host machine (ISA-bus supported).
IE-70000-PCI-IF Interface adapter		This adapter is required when connecting a personal computer that includes a PCI bus as the host machine of the IE-78K0-NS.
IE-780833-NS-EM4 Probe board		This board emulates the operations of the peripheral hardware peculiar to a device. It should be used in combination with an in-circuit emulator and I/O board.
★ IE-78K0-NS-P04 I/O board		This I/O board is necessary when using the IE-780833-NS-EM4.
NP-80GC Emulation probe		This probe is used to connect the in-circuit emulator to the target system and is designed for an 80-pin plastic QFP (GC-8BT type).
	EV-9200GC-80 Conversion socket (Refer to **Figures A-2** and **A-3**)	This conversion socket connects the NP-80GC to a target system board designed to mount an 80-pin plastic QFP (GC-8B type).

Remarks 1. The NP-80GC is manufactured by Naito Densei Machida Mfg. Co., Ltd.
For more information, consult Naito Densei Machida Mfg. Co., Ltd. (TEL: +8-45-475-4191)

2. The EV-9200GC-80 is sold in sets of five.

A.3.1 Hardware (2/2)

(2) When using the in-circuit emulator IE-78001-R-A

	IE-78001-R-A In-circuit emulator		This in-circuit emulator is used to debug hardware and software when developing application systems using a 78K/0 Series product. It supports an integrated debugger (ID78K0). This emulator should be used in combination with an emulation probe and interface adapter, which is required to connect this emulator to the host machine.
	IE-7000-98-IF-C Interface adapter		This adapter is required when using a PC-9800 series (except notebook PC) as the IE-78001-R-A host machine (C-bus supported).
	IE-7000-PC-IF-C Interface adapter		This adapter is required when using an IBM PC/AT compatible as the IE-78001-R-A host machine (ISA-bus supported).
	IE-78000-R-SV3 Interface adapter		This is the adapter and cable required when using an EWS as the IE-78001-R-A host machine, and is used connected to the board in the IE-78000-R-A. 10Base-5 is supported for Ethernet^{TM}. For other methods, a commercially available conversion adapter is required.
	IE-70000-PCI-IF Interface adapter		This adapter is required when connecting a personal computer that includes a PCI bus as the host machine of the IE-78001-R-A.
★	IE-780831-NS-EM4 Probe board		This board emulates the operations of the peripheral hardware peculiar to a device. It should be used in combination with an in-circuit emulator, I/O board, and emulation conversion board.
★		IE-78K0-NS-P04 I/O board	This I/O board is necessary when using the IE-780833-NS-EM4.
★		IE-78K0-R-EX1 Emulation probe conversion board	This board is required when using the IE-780833-NS-EM4 + IE-78K0-NS-P04 on the IE-78001-R-A.
	EP-78230GC-R Emulation probe		This probe is used to connect the in-circuit emulator to the target system and is designed for an 80-pin plastic QFP (GC-8BT type).
		EV-9200GC-80 Conversion socket (Refer to **Figures A-2** and **A-3**)	This conversion socket connects the EP-78230GC-R to a target system board designed to mount an 80-pin plastic QFP (GC-8BT type).

Remark The EV-9200GC-80 is sold in sets of five.

A.3.2 Software

SM78K0 System simulator	This system simulator is used to perform debugging at the C source level or assembler level while simulating the operation of the target system on a host machine. This simulator runs on Windows. Use of the SM78K0 allows the execution of application logical testing and performance testing on an independent basis from hardware development without having to use an in-circuit emulator, thereby providing higher development efficiency and software quality. The SM78K0 should be used in combination with a device file (DF780833) (sold separately).
	Part number: μS××××SM78K0
ID78K0-NS Integrated debugger (supporting in-circuit emulator IE-78K0-NS)	This debugger is a control program to debug 78K/0 Series microcontrollers. It employs a graphical user interface, which is equivalent visually and operationally to Windows or OSF/Motif™. It also has an enhanced debugging function for C language programs, and thus trace results can be displayed on screen at the C-language level by using the windows integration function, which links a trace result with its source program, disassembled display, and memory display. In addition, by incorporating function modules such as a task debugger and system performance analyzer, the efficiency of debugging programs that run on real-time OSs can be improved. It should be used in combination with a device file (sold separately).
ID78K0 Integrated debugger (supporting in-circuit emulator IE-78001-R-A)	
	Part number: μS××××ID78K0-NS, μS××××ID78K0

μS$\underline{××××}$SM78K0

××××	Host Machine	OS	Supply Medium
AB13	IBM PC/AT compatibles	Windows (Japanese)	3.5-inch 2HC FD
BB13		Windows (English)	
AB17		Windows (Japanese)	CD-ROM
BB17		Windows (English)	

μS××××ID78K0-NS
μS$\underline{××××}$ID78K0

××××	Host Machine	OS	Supply Medium
AB13	IBM PC/AT compatibles	Windows (Japanese)	3.5-inch 2HC FD
BB13		Windows (English)	
AB17		Windows (Japanese)	CD-ROM
BB17		Windows (English)	

Remark ×××× in the part number differs depending on the host machine and OS used.

A.4 System Upgrade from Former In-Circuit Emulator for 78K/0 Series to IE-78001-R-A

Users with a previous version of the in-circuit emulator for 78K/0 Series microcontrollers (IE-78000-R or IE-78000-R-A) can achieve operation equivalent to the IE-78001-R-A by replacing the internal break board of the older version with the IE-78001-R-BK.

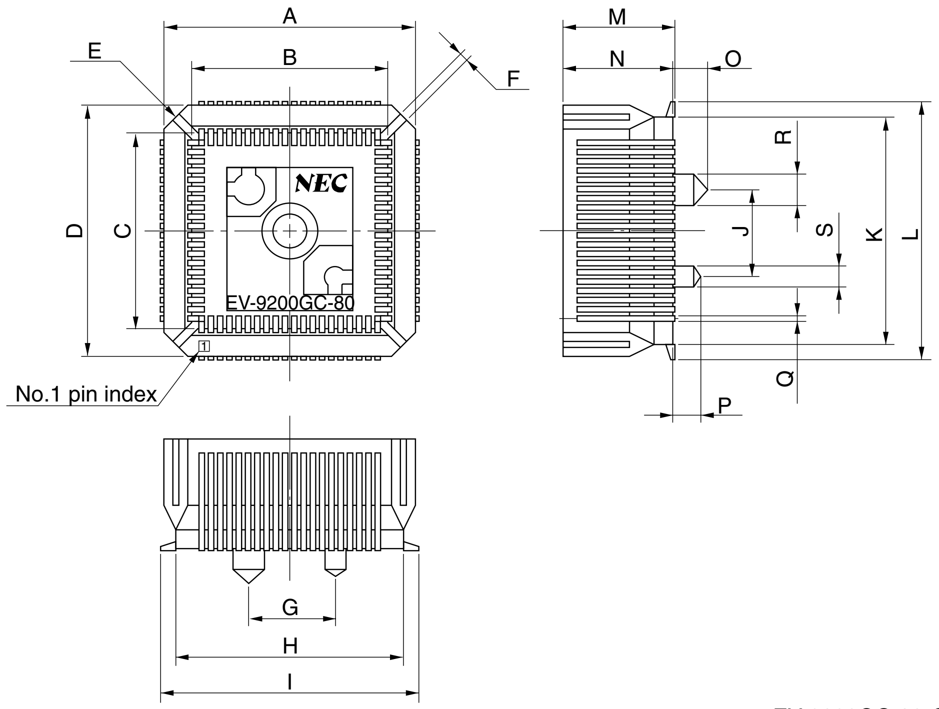
Table A-1. Upgrade Method from Former In-Circuit Emulator for 78K/0 Series to IE-78001-R-A

In-Circuit Emulator Owned	In-Circuit Emulator Unit Upgrade[Note]	Board to Be Purchased
IE-78000-R	Required	IE-78001-R-BK
IE-78000-R-A	Not required	

Note To upgrade the unit, send your in-circuit emulator to NEC.

Conversion Socket Drawing (EV-9200GC-80) and Footprints

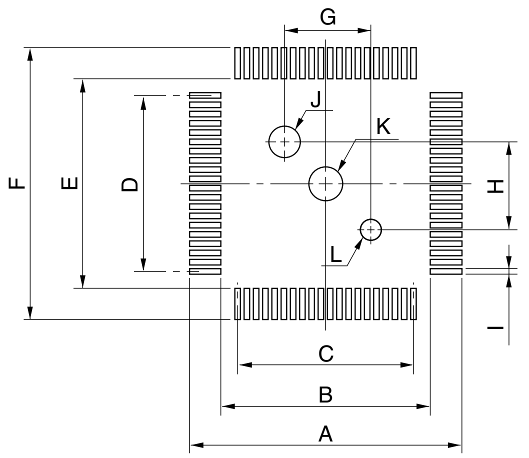
Figure A-2. EV-9200GC-80 Drawing (for reference only)



EV-9200GC-80-G1E

ITEM	MILLIMETERS	INCHES
A	18.0	0.709
B	14.4	0.567
C	14.4	0.567
D	18.0	0.709
E	4-C 2.0	4-C 0.079
F	0.8	0.031
G	6.0	0.236
H	16.0	0.63
I	18.7	0.736
J	6.0	0.236
K	16.0	0.63
L	18.7	0.736
M	8.2	0.323
N	8.0	0.315
O	2.5	0.098
P	2.0	0.079
Q	0.35	0.014
R	ϕ2.3	ϕ0.091
S	ϕ1.5	ϕ0.059

Figure A-3. EV-9200GC-80 Footprints (for reference only)



EV-9200GC-80-P1E

ITEM	MILLIMETERS	INCHES
A	19.7	0.776
B	15.0	0.591
C	0.65±0.02 × 19=12.35±0.05	$0.026^{+0.001}_{-0.002}$ × 0.748=0.486 $^{+0.003}_{-0.002}$
D	0.65±0.02 × 19=12.35±0.05	$0.026^{+0.001}_{-0.002}$ × 0.748=0.486 $^{+0.003}_{-0.002}$
E	15.0	0.591
F	19.7	0.776
G	6.0±0.05	$0.236^{+0.003}_{-0.002}$
H	6.0±0.05	$0.236^{+0.003}_{-0.002}$
I	0.35±0.02	$0.014^{+0.001}_{-0.001}$
J	ϕ2.36±0.03	$\phi 0.093^{+0.001}_{-0.002}$
K	ϕ2.3	ϕ0.091
L	ϕ1.57±0.03	$\phi 0.062^{+0.001}_{-0.002}$

Caution **Dimensions of mount pad for EV-9200 and that for target device (QFP) may be different in some parts. For the recommended mount pad dimensions for QFP, refer to "SEMICONDUCTOR DEVICE MOUNTING TECHNOLOGY MANUAL" (C10535E).**

APPENDIX B EMBEDDED SOFTWARE

For efficient development and maintenance of the μPD780833Y Subseries, the following embedded products are available.

Real-time OS (1/2)

RX78K0 Real-time OS	The RX78K0 is a real-time OS conforming to the μITRON specifications. A tool (configurator) for generating the nucleus of the RX78K0 and plural information tables is supplied. This product is used in combination with an assembler package (RA78K0) and device file (DF780833) (both sold separately). **<Precaution when using RX78K0 in PC environment>** The real-time OS is a DOS-based application. It should be used in the DOS Prompt when using in Windows.
	Part number: μS××××RX78013-ΔΔΔΔ

Caution When purchasing the RX78K0, fill in the purchase application form in advance and sign the user agreement.

Remark ×××× and ΔΔΔΔ in the part number differ depending on the host machine and OS used.

μS××××RX78013-ΔΔΔΔ

ΔΔΔΔ	Product Outline	Maximum Number for Use in Mass Production
001	Evaluation object	Do not use for mass-produced product.
100K	Mass-production object	0.1 million units
001M		1 million units
010M		10 million units
S01	Source program	Source program for mass-produced object

××××	Host Machine	OS	Supply Medium
AA13	PC-9800 series	Windows (Japanese)Note	3.5-inch 2HD FD
AB13	IBM PC/AT compatibles	Windows (Japanese)Note	3.5-inch 2HC FD
BB13		Windows (English)Note	
3P16	HP9000 series 700	HP-UX (Rel. 10.10)	DAT (DDS)
3K13	SPARCstation	SunOS (Rel. 4.1.4),	3.5-inch 2HC FD
3K15		Solaris (Rel. 2.5.1)	1/4-inch CGMT
3R13	NEWS (RISC)	NEWS-OS (Rel. 6.1)	3.5-inch 2HC FD

Note Can also be operated in a DOS environment.

Real-time OS (2/2)

MX78K0	The MX78K0 is an OS for μITRON specification subsets. A nucleus for the MX78K0 is OS also included as a companion product. This manages tasks, events, and time. In task management, determining the task execution order and switching from one task to the next task are performed. **<Precaution when using MX78K0 in PC environment>** The MX78K0 is a DOS-based application. It should be used in the DOS Prompt when using in Windows.
	Part number: μS××××MX78K0-ΔΔΔ

Remark ×××× and ΔΔΔ in the part number differ depending on the host machine and OS used.

μS××××MX78K0-ΔΔΔ

ΔΔΔ	Product Outline	Maximum Number for Use in Mass Production
001	Evaluation object	Use in preproduction stages.
××	Mass-production object	Use in mass production stages.
S01	Source program	Only the users who purchased mass-production objects are allowed to purchase this program.

××××	Host Machine	OS	Supply Medium
AA13	PC-9800 series	Windows (Japanese)Note	3.5-inch 2HD FD
AB13	IBM PC/AT compatibles	Windows (Japanese)Note	3.5-inch 2HC FD
BB13		Windows (English)Note	
3P16	HP9000 series 700	HP-UX (Rel. 10.10)	DAT (DDS)
3K13	SPARCstation	SunOS (Rel. 4.1.4),	3.5-inch 2HC FD
3K15		Solaris (Rel. 2.5.1)	1/4-inch CGMT
3R13	NEWS (RISC)	NEWS-OS (Rel. 6.1)	3.5-inch 2HC FD

Note Can also be operated in a DOS environment.

C.1 Register Index (In Alphabetical Order with Respect to Register Names)

[A]
A/D conversion result register 00 (ADCR00) … 172
A/D conversion result register 01 (ADCR01) … 172
A/D converter mode register 00 (ADM00) … 172
A/D converter mode register 01 (ADM01) … 172
Analog input channel specification register 00 (ADS00) … 172
Analog input channel specification register 01 (ADS01) … 172
Asynchronous serial interface mode register 0 (ASIM0) … 242
Asynchronous serial interface status register 0 (ASIS0) … 243

[B]
Baud rate generator control register 0 (BRGC0) … 244

[C]
Capture/compare control register 00 (CRC00) … 106
Capture/compare control register 01 (CRC01) … 106
Class 2 clock selection register (C2CLK) … 203
Class 2 control register 1 (C2CT1) … 191
Class 2 control register 2 (C2CT2) … 194
Class 2 fall time propagation daily correction register (C2PDF) … 202
Class 2 receive FIFO register (C2RXFIFO) … 200
Class 2 rise time propagation daily correction register (C2PDR) … 201
Class 2 status register 1 (C2ST1) … 196
Class 2 status register 2 (C2ST2) … 198
Class 2 transmit FIFO register (C2TXFIFO) … 200
Clock output selection register (CKS) … 169

[E]
8-bit timer counter 50 (TM50) ... 138
8-bit timer counter 51 (TM51) ... 138
8-bit timer counter 52 (TM52) ... 138
8-bit timer compare register 50 (CR50) ... 138
8-bit timer compare register 51 (CR51) ... 138
8-bit timer compare register 52 (CR52) ... 138
8-bit timer mode control register 50 (TMC50) ... 139
8-bit timer mode control register 51 (TMC51) ... 139
8-bit timer mode control register 52 (TMC52) ... 139
External interrupt falling edge enable register (EGN) … 328
External interrupt rising edge enable register (EGP) … 328

[F]
Flash programming mode control register (FLPMC) ... 357

[I]
IIC control register 0 (IICC0) … 270
IIC function expansion register 0 (IICX0) ... 278
IIC shift register 0 (IIC0) … 268, 279
IIC status register 0 (IICS0) … 273
IIC transfer clock selection register 0 (IICCL0) … 276
Internal expansion RAM size switching register (IXS) … 353
Interrupt mask flag register 0H (MK0H) … 326
Interrupt mask flag register 0L (MK0L) … 326
Interrupt mask flag register 1H (MK1H) … 326
Interrupt mask flag register 1L (MK1L) … 326
Interrupt request flag register 0H (IF0H) … 325
Interrupt request flag register 0L (IF0L) … 325
Interrupt request flag register 1H (IF1H) … 325
Interrupt request flag register 1L (IF1L) … 325

[M]
Memory size switching register (IMS) … 353
Memory expansion mode register (MEM) … 90

[O]
Oscillation stabilization time selection register (OSTS) … 165, 341

[P]
Port 0 (P0) … 75
Port 2 (P2) … 77
Port 3 (P3) … 78
Port 4 (P4) … 80
Port 5 (P5) … 81
Port 6 (P6) … 82
Port 7 (P7) … 83
Port 8 (P8) … 84
Port 9 (P9) … 85
Port mode register 0 (PM0) … 86
Port mode register 2 (PM2) … 86, 145
Port mode register 3 (PM3) … 86, 112
Port mode register 4 (PM4) … 86
Port mode register 5 (PM5) … 86
Port mode register 6 (PM6) … 86
Port mode register 7 (PM7) … 86, 112, 170
Port mode register 8 (PM8) … 86
Port mode register 9 (PM9) … 86
Prescaler mode register 00 (PRM00) … 110
Prescaler mode register 01 (PRM01) … 110
Priority specification flag register 0H (PR0H) … 327
Priority specification flag register 0L (PR0L) … 327
Priority specification flag register 1H (PR1H) … 327
Priority specification flag register 1L (PR1L) … 327

Processor clock control register (PCC) … 93
Program status word (PSW) … 53, 329
Pull-up resistor option register 0 (PU0) … 88
Pull-up resistor option register 2 (PU2) … 88
Pull-up resistor option register 3 (PU3) … 88
Pull-up resistor option register 4 (PU4) … 88
Pull-up resistor option register 5 (PU5) … 88
Pull-up resistor option register 6 (PU6) … 88
Pull-up resistor option register 7 (PU7) … 88

[R]
Receive buffer register 0 (RXB0) … 240

[S]
Serial I/O shift register 30 (SIO30) … 258
Serial I/O shift register 31 (SIO31) … 258
Serial operation mode register 30 (CSIM30) … 258
Serial operation mode register 31 (CSIM31) … 258
16-bit timer capture/compare register 000 (CR000) … 101
16-bit timer capture/compare register 010 (CR010) … 103
16-bit timer capture/compare register 001 (CR001) … 101
16-bit timer capture/compare register 011 (CR011) … 103
16-bit timer mode control register 00 (TMC00) … 103
16-bit timer mode control register 01 (TMC01) … 103
16-bit timer output control register 00 (TOC00) … 108
16-bit timer output control register 01 (TOC01) … 108
16-bit timer counter 00 (TM00) … 101
16-bit timer counter 01 (TM01) … 101
Slave address register 0 (SVA0) … 268, 279

[T]
Timer clock selection register 50 (TCL50) ... 143
Timer clock selection register 51 (TCL51) ... 143
Timer clock selection register 52 (TCL52) ... 143
Transmit shift register 0 (TXS0) … 240

[W]
Watch timer operation mode register 0 (WTNM0) … 158
Watchdog timer clock selection register (WDCS) … 163
Watchdog timer mode register (WDTM) … 164

C.2 Register Index (In Alphabetical Order with Respect to Register Symbol)

[A]

ADCR00: A/D conversion result register 00 … 172
ADCR01: A/D conversion result register 01 … 172
ADM00: A/D converter mode register 00 … 172
ADM01: A/D converter mode register 01 … 172
ADS00: Analog input channel specification register 00 … 172
ADS01: Analog input channel specification register 01 … 172
ASIM0: Asynchronous serial interface mode register 0 … 242
ASIS0: Asynchronous serial interface status register 0 … 243

[B]

BRGC0: Baud rate generator control register 0 … 244

[C]

C2CLK: Class 2 clock selection register … 203
C2CT1: Class 2 control register 1 … 191
C2CT2: Class 2 control register 2 … 194
C2PDF: Class 2 fall time propagation daily correction register … 202
C2PDR: Class 2 rise time propagation daily correction register … 201
C2RXFIFO: Class 2 receive FIFO register … 200
C2ST1: Class 2 status register 1 … 196
C2ST2: Class 2 status register 2 … 198
C2TXFIFO: Class 2 transmit FIFO register … 200
CKS: Clock output selection register … 169
CRC00: Capture/compare control register 00 … 106
CRC01: Capture/compare control register 01 … 106
CR000: 16-bit timer capture/compare register 00 … 101
CR010: 16-bit timer capture/compare register 01 … 103
CR001: 16-bit timer capture/compare register 001 … 101
CR011: 16-bit timer capture/compare register 011 … 103
CR50: 8-bit timer compare register 50 ... 138
CR51: 8-bit timer compare register 51 ... 138
CR52: 8-bit timer compare register 52 ... 138
CSIM30: Serial operation mode register 30 … 258
CSIM31: Serial operation mode register 31 … 258

[E]

EGN: External interrupt falling edge enable register … 328
EGP: External interrupt rising edge enable register … 328

[F]

FLPMC: Flash programming mode control register … 357

[I]

IF0H: Interrupt request flag register 0H … 325
IF0L: Interrupt request flag register 0L … 325

IF1L: Interrupt request flag register 1L … 325
IF1H: Interrupt request flag register 1H … 325
IIC0: IIC shift register 0 … 268, 279
IICC0: IIC control register 0 … 270
IICCL0: IIC transfer clock selection register 0 … 276
IICS0: IIC status register 0 … 273
IICX0: IIC function expansion register 0 ... 278
IMS: Memory size switching register … 353
IXS: Internal expansion RAM size switching register … 353

[M]
MEM: Memory extpasion mode register … 90
MK0H: Interrupt mask flag register 0H … 326
MK0L: Interrupt mask flag register 0L … 326
MK1L: Interrupt mask flag register 1L … 326
MK1H: Interrupt mask flag register 1H … 326

[O]
OSTS: Oscillation stabilization time selection register … 165, 341

[P]
P0: Port 0 … 75
P2: Port 2 … 77
P3: Port 3 … 78
P4: Port 4 … 80
P5: Port 5 … 81
P6: Port 6 … 82
P7: Port 7 … 83
P8: Port 8 … 84
P9: Port 9 … 85
PCC: Processor clock control register … 93
PM0: Port mode register 0 … 86
PM2: Port mode register 2 … 86, 145
PM3: Port mode register 3 … 86, 112
PM4: Port mode register 4 … 86
PM5: Port mode register 5 … 86
PM6: Port mode register 6 … 86
PM7: Port mode register 7 … 86, 112, 170
PM8: Port mode register 8 … 86
PM9: Port mode register 9 … 86
PR0H: Priority specification flag register 0H … 327
PR0L: Priority specification flag register 0L … 327
PR1L: Priority specification flag register 1L … 327
PR1H: Priority specification flag register 1H … 327
PRM00: Prescaler mode register 00 … 110
PRM01: Prescaler mode register 01 … 110
PSW: Program status word … 53, 329
PU0: Pull-up resistor option register 0 … 88

PU2:	Pull-up resistor option register 2 … 88
PU3:	Pull-up resistor option register 3 … 88
PU4:	Pull-up resistor option register 4 … 88
PU5:	Pull-up resistor option register 5 … 88
PU6:	Pull-up resistor option register 6 … 88
PU7:	Pull-up resistor option register 7 … 88

[R]

RXB0:	Receive buffer register 0 … 240

[S]

SIO30:	Serial I/O shift register 30 … 258
SIO31:	Serial I/O shift register 31 … 258
SVA0:	Slave address register 0 … 268, 279

[T]

TCL50:	Timer clock selection register 50 ... 143
TCL51:	Timer clock selection register 51 ... 143
TCL52:	Timer clock selection register 52 ... 143
TM00:	16-bit timer counter 00 … 101
TM01:	16-bit timer counter 01 ... 101
TM50:	8-bit timer counter 50 ... 138
TM51:	8-bit timer counter 51 ... 138
TM52:	8-bit timer counter 52 ... 138
TMC00:	16-bit timer mode control register 00 … 103
TMC01:	16-bit timer mode control register 01 … 103
TMC50:	8-bit timer mode control register 50 ... 139
TMC51:	8-bit timer mode control register 51 ... 139
TMC52:	8-bit timer mode control register 52 ... 139
TOC00:	16-bit timer output control register 00 … 108
TOC01:	16-bit timer output control register 01 … 108
TXS0:	Transmit shift register 0 … 240

[W]

WDCS:	Watchdog timer clock selection register … 163
WDTM:	Watchdog timer mode register … 164
WTNM0:	Watch timer operation mode register 0 … 158

APPENDIX D   REVISION HISTORY

(1/3)

Edition	Description	Applied to
2nd edition	Deletion of the following products. μPD780831Y, μPD780832Y Flashpro II	Throughout
	Change of **1.5 78K/0 Series Lineup** Modification of timer overview.	CHAPTER 1 OUTLINE
	Modification of **Table 2-1 Types of Pin I/O Circuits and Recommended Connection of Unused Pins**	CHAPTER 2 PIN FUNCTION
	Addition of **Caution 2** to **4.3 Registers Controlling Port Function**	CHAPTER 4 PORT FUNCTIONS
	Modification of **Caution 2** in **(2) Pull-up resistor option registers (PU0, PU2 to PU7)**	
	Addition of oscillation frequency in **5.1 Function of Clock Generator**	CHAPTER 5 CLOCK GENERATOR
	Modification of **Caution 1** in **(2) 16-bit capture/compare registers 000, 001 EVENT COUNTERS 00, 01**	CHAPTER 6 16-BIT TIMER/EVENT COUNTERS 00, 01
	Modification of **Caution** in **(3) 16-bit capture/compare registers 010, 011 (CR010, CR011)**	
	Modification of **Figure 6-19 Timing of Pulse Width Measurement Operation by Free-Running Counter and One Capture Register (with Both Edges Specified)**	
	Modification of **Figure 6-21 CR01n Capture Operation with Rising Edge Specified**	
	Modification of **Figure 6-22 Timing of Pulse Width Measurement Operation by Free-Running Counter (with Both Edges Specified)**	
	Modification of **Figure 6-24 Timing of Pulse Width Measurement Operation by Free-Running Counter and Two Capture Registers (with Rising Edge Specified)**	
	Modification of **Figure 6-26 Timing of Pulse Width Measurement Operation by Means of Restart (with Rising Edge Specified)**	
	Addition of **Note** to **6.5.6 One-shot pulse output operation**	
	Modification of **Caution** in **Figure 6-32 Control Register Settings for One-Shot Pulse Output Operation Using Software Trigger**	
	Addition of **Note** to **(2) One-shot pulse output by external trigger**	
	Modification of **Caution** in **Figure 6-34 Control Register Settings for One-Shot Pulse Output Operation Using External Trigger**	
	Modification of **Figure 6-38 Capture Register Data Retention Timing**	
	Addition of **(c) One-shot pulse output function**	
	Modification of **Figure 6-39 Operation Timing of OVF0n Flag**	

(2/3)

Edition	Description	Applied to
2nd edition	Addition of **7.1 Outline of 8-Bit Timer/Event Counters 50, 51, 52**	CHAPTER 7 8-BIT TIMER/ EVENT COUNTERS 50, 51, 52
	Modification of **Caution** in **(2) 8-bit timer compare registers 50, 51, and 52 (CR50, CR51, and CR52)**	
	Addition of **Caution 1** to **Figure 7-4 Format of 8-Bit Timer Mode Control Register 50 (TMC50)**	
	Addition of **Caution** to **Figure 7-5 Format of 8-Bit Timer Mode Control Register 51 (TMC51)**	
	Addition of **Caution 1** to **Figure 7-6 Format of 8-Bit Timer Mode Control Register 52 (TMC52)**	
	Addition of **Caution** to **7.5.4 8-bit PWM output operation**2nd edition	
	Addition of **8.1 Outline of Watch Timer**	CHAPTER 8 WATCH TIMER
	Addition of **Caution** to **Figure 8-3 Operation Timing of Watch Timer/ Interval Timer**	
	Addition of **9.1 Outline of Watchdog Timer**	CHAPTER 9 WATCHDOG TIMER
	Modification of **Figure 9-1 Block Diagram of Watchdog Timer**	
	Modification of **Caution** in **Figure 9-3 Format of Watchdog Timer Mode Register (WDTM)**	
	Addition of **10.1 Outline of Clock Output Controller**	CHAPTER 10 CLOCK OUTPUT CONTROLLER
	Addition of **11.5 How to Read A/D Converter Characteristics Table**	CHAPTER 11 A/D CONVERTER
	Addition of items (11) to (14) to **11.6 Notes on A/D Converter**	
	Addition of **Figure 11-14 Conversion Result Read Timing (When Conversion Result Is Undefined)**	
	Addition of **Figure 11-15 Conversion Result Read Timing (When Conversion Result Is Normal)**	
	Addition of **Figure 11-16 Example of Capacitor Connection to AV$_{REF0}$ Pin**	
	Addition of **Figure 11-17 Internal Equivalence Circuit of ANI00 to ANI70 Pins**	
	Addition of **Table 11-3 Resistance and Capacitance Values for Equivalence Circuits (Reference Values)**	
	Addition of **Figure 11-18 Example of Circuit When Signal Source Impedance Is High**	
	Addition of **Caution 3** and **4** to **Figure 12-4 Format of Class 2 Status Register 1 (C2ST1)**	CHAPTER 12 J1850 (Class 2)
	Addition of **Remark 2 to Figure 12-8 Format of Class 2 Clock Selection Register (C2CLK)**	
	12.3.6 Communication control based on CPU	
	Addition of **Caution** and block diagram for **<2> When CRC is used** to **(2) Reception control (b) Example of reception control in block mode**	
	Addition of **Caution** to **(3) Transmission control (a) Example of transmission control in normal mode**	
	Modification of a block diagram in **(3) Transmission control (b) Example of transmission/reception control in block mode**	
	Addition of **Figure 13-2 Block Diagram of Baud Rate Generator**	CHAPTER 13 SERIAL INTERFACE UART0
	Addition of description on 5-bit counter to baud rate expression	
	Modification of **(c) Transmission**	
	Modification of **Figure 14-1 Block Diagram of Serial Interface SIO30**	CHAPTER 14 SERIAL INTERFACE SIO3

(3/3)

Edition	Description	Applied to
2nd edition	Modification of **Figure 15-3 Format of IIC Control Register 0 (IICC0)**	CHAPTER 15 SERIAL INTERFACE IIC0
	Addition of **Note** to **Figure 15-4 Format of IIC Status Register 0 (IICS0)**	
	Addition of **Remark 4** to **Figure 15-5 Format of IIC Transfer Clock Selection Register 0 (IICCL0)**	
	Modification of timing diagram in **15.5.7 I^{2}C interrupt requests (INTIIC0)**	
	Modification of **Figure 15-19 Master Operation Flow Chart**	
	Modification of **Figure 15-20 Slave Operation Flow Chart**	
	Addition of **Remark** to **16.2 Interrupt Sources and Configuration** Modification of **Table 16-1 Interrupt Source List**	CHAPTER 16 INTERRUPT FUNCTIONS
	Change of **Note** in **Table 16-2 Flags Corresponding to Interrupt Request Sources**	
	Addition of **Caution 2** to **Figure 16-2 Format of Interrupt Request Flag Register (IF0L, IF0H, IF1L, IF1H)**	
	Modification of **Table 18-1 Hardware Statuses After Reset**	CHAPTER 18 RESET FUNCTION
	Modification of **Table 19-1 Differences Between μPD78F0833Y and Mask ROM Versions**	CHAPTER 19 μPD78F0833Y
	Modification of **Figure 19-1 Format of Memory Size Switching Register (IMS)**	
	Modification of **Table 19-2 Communication Mode List**	
	Addition of SP78K0	APPENDIX A DEVELOPMENT TOOLS

NEC

Facsimile Message

Although NEC has taken all possible steps to ensure that the documentation supplied to our customers is complete, bug free and up-to-date, we readily accept that errors may occur. Despite all the care and precautions we've taken, you may encounter problems in the documentation. Please complete this form whenever you'd like to report errors or suggest improvements to us.

From:

Name

Company

Tel. FAX

Address

Thank you for your kind support.

North America NEC Electronics Inc. Corporate Communications Dept. Fax: +1-800-729-9288 +1-408-588-6130	**Hong Kong, Philippines, Oceania** NEC Electronics Hong Kong Ltd. Fax: +852-2886-9022/9044	**Asian Nations except Philippines** NEC Electronics Singapore Pte. Ltd. Fax: +65-250-3583
Europe NEC Electronics (Europe) GmbH Technical Documentation Dept. Fax: +49-211-6503-274	**Korea** NEC Electronics Hong Kong Ltd. Seoul Branch Fax: +82-2-528-4411	**Japan** NEC Semiconductor Technical Hotline Fax: +81- 44-435-9608
South America NEC do Brasil S.A. Fax: +55-11-6462-6829	**Taiwan** NEC Electronics Taiwan Ltd. Fax: +886-2-2719-5951	

I would like to report the following error/make the following suggestion:

Document title: ____________________

Document number: ____________________ Page number: __________

If possible, please fax the referenced page or drawing.

Document Rating	Excellent	Good	Acceptable	Poor
Clarity	❏	❏	❏	❏
Technical Accuracy	❏	❏	❏	❏
Organization	❏	❏	❏	❏

CS 01.2