User's Manual

μPD780078, 780078Y Subseries

8-Bit Single-Chip Microcontrollers

μPD780076
μPD780078
μPD78F0078
μPD780076Y
μPD780078Y
μPD78F0078Y

Document No. U14260EJ1V1UM00 (1st edition)
Date Published August 2000 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:

μPD78F0078GC-AB8, 78F0078GK-9ET, 78F0078YGC-AB8, 78F0078YGK-9ET

The customer must judge the need for a license for the following products:

μPD780076GC-×××-AB8, 780076GK-×××-9ET
μPD780078GC-×××-AB8, 780078GK-×××-9ET
μPD780076YGC-×××-AB8, 780076YGK-×××-9ET
μPD780078YGC-×××-AB8, 780078YGK-×××-9ET

Purchase of NEC I^{2}C components conveys a license under the Philips I^{2}C Patent Rights to use these components in an I^{2}C system, provided that the system conforms to the I^{2}C Standard Specification as defined by Philips.

- **The information in this document is current as of July, 2000. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC's data sheets or data books, etc., for the most up-to-date specifications of NEC semiconductor products. Not all products and/or types are available in every country. Please check with an NEC sales representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without prior written consent of NEC. NEC assumes no responsibility for any errors that may appear in this document.
- NEC does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC semiconductor products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of customer's equipment shall be done under the full responsibility of customer. NEC assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC endeavours to enhance the quality, reliability and safety of NEC semiconductor products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC semiconductor products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment, and anti-failure features.
- NEC semiconductor products are classified into the following three quality grades:
  "Standard", "Special" and "Specific". The "Specific" quality grade applies only to semiconductor products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of a semiconductor product depend on its quality grade, as indicated below. Customers must check the quality grade of each semiconductor product before using it in a particular application.

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

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

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

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

  (Note)

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

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

M8E 00. 4

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: 01-30-67 58 00
Fax: 01-30-67 58 99

NEC Electronics (France) S.A.
Madrid Office
Madrid, Spain
Tel: 91-504-2787
Fax: 91-504-2860

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.
United Square, Singapore
Tel: 65-253-8311
Fax: 65-250-3583

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: 55-11-6462-6810
Fax: 55-11-6462-6829

J00.7

Major Revisions in this Edition

Page	Description
p.335, 338	Modification of the description of count clock CSI1 selection of serial clock select register 1 (CSIC1)
p.437	Modification of the description of 16-bit timer/event counter in **Table 21-1. HALT Mode Operating Statuses**

The mark ★ shows major revised points.

[MEMO]

INTRODUCTION

Readers

This manual has been prepared for user engineers who understand the functions of the μPD780078, 780078Y Subseries and wish to design and develop application systems and programs for these devices.

μPD780078 Subseries : μPD780076, 780078
μPD780078Y Subseries : μPD780076Y, 780078Y

Purpose

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

Organization

The μPD780078, 780078Y Subseries manual is separated into two parts: this manual and the instructions edition (common to the 78K/0 Series).

μPD780078, 780078Y **Subseries User's Manual** **(This Manual)**	**78K/0 Series** **User's Manual** **Instructions**
• Pin functions	• CPU functions
• Internal block functions	• Instruction set
• Interrupt	• Explanation of each instruction
• Other on-chip peripheral functions	

How To Read This Manual

Before reading this manual, you should have general knowledge of electric and logic circuits and microcontrollers.

- To gain a general understanding of functions:
  Read this manual in the order of the contents.
- How to interpret the register format:
  → For the bit number enclosed in square, the bit name is defined as a reserved word in RA78K/0, and in CC78K/0, already defined in the header file named sfrbit.h.
- To check the details of a register when you know the register name.
  → Refer to **APPENDIX D REGISTER INDEX.**

Differences between μPD780078 and 780078Y Subseries

The configuration of the serial interface differs on μPD780078 and 780078Y subseries products.

Item / Subseries		μPD780078 subseries	μPD780078Y subseries
Configuration of serial interface	UART0	1ch	1ch
	UART2/SIO3	1ch	1ch
	CSI1	1ch	1ch
	IIC0	None	1ch

Chapter Organization This manual divides the descriptions for the subseries into different chapters as shown below. Read only the chapters related to the device you use.

Chapter	μPD780078 Subseries	μPD780078Y Subseries
Chapter 1 Outline (μPD780078 Subseries)	○	—
Chapter 2 Outline (μPD780078Y Subseries)	—	○
Chapter 3 Pin Function (μPD780078 Subseries)	○	—
Chapter 4 Pin Function (μPD780078Y Subseries)	—	○
Chapter 5 CPU Architecture	○	○
Chapter 6 Port Functions	○	○
Chapter 7 Clock Generator	○	○
Chapter 8 16-Bit Timer/Event Counters 00, 01	○	○
Chapter 9 8-Bit Timer/Event Counters 50, 51	○	○
Chapter 10 Watch Timer	○	○
Chapter 11 Watchdog Timer	○	○
Chapter 12 Clock Output/Buzzer Output Control Circuits	○	○
Chapter 13 A/D Converter	○	○
Chapter 14 Serial Interface (UART0)	○	○
Chapter 15 Serial Interface (UART2)	○	○
Chapter 16 Serial Interface (SIO3)	○	○
Chapter 17 Serial Interface (CSI1)	○	○
Chapter 18 Serial Interface (IIC0) (μPD780078Y Subseries only)	—	○
Chapter 19 Interrupt Functions	○	○
Chapter 20 External Device Expansion Function	○	○
Chapter 21 Standby Function	○	○
Chapter 22 Reset Function	○	○
Chapter 23 μPD78F0078, 78F0078Y	○	○
Chapter 24 Instruction Set	○	○

Conventions

Data representation weight : High digits on the left and low digits on the right
Active low representations : $\overline{\times\times\times}$ (line over the pin and signal names)
Note : Description of note in the text.
Caution : Information requiring particular attention
Remark : Additional explanatory material
Numerical representations : Binary ··· ×××× or ××××B
Decimal ··· ××××
Hexadecimal ··· ××××H

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

- **Device-related documents**

Document Name	Document No.	
	Japanese	English
μPD780076, 780078, 780076Y, 780078Y Preliminary Product Information	U14259J	U14259E
μPD78F0078, 78F0078Y Preliminary Product Information	U14258J	U14258E
μPD780078, 780078Y Subseries User's Manual	U14046J	This manual
78K/0 Series User's Manual -Instructions	U12326J	U12326E
78K/0 Series Instruction Application Table	U10903J	—
78K/0 Series Instruction Set	U10904J	—
78K/0 Series Application Note Basics (I)	U12704J	U12704E
78K/0, 78K/0S Series PAN Flash Memory Write	U14458J	U14458E

- **Related documents for development tool (User's Manuals)**

Document Name		Document No.	
		Japanese	English
RA78K0 Assembler Package	Operation	U11802J	U11802E
	Language	U11801J	U11801E
	Structured Assembly Language	U11789J	U11789E
RA78K Series Structured Assembler Preprocessor		U12323J	EEU-1402
CC78K0 C Compiler	Operation	U11517J	U11517E
	Language	U11518J	U11518E
IE-78K0-NS		U13731J	—
IE-78001-R-A		To be prepared	To be prepared
IE-78K0-R-EX1		To be prepared	To be prepared
IE-780078-NS-EM1		To be prepared	To be prepared
EP-78240		EEU-986	U10332E
EP-78012GK-R		EEU-5012	EEU-1538
SM78K0 System Simulator Windows^{TM} Based	Reference	U10181J	U10181E
SM78K Series System Simulator Interface Specifications	External Part User Open	U10092J	U10092E
ID78K0-NS Integrated Debugger Windows Based	Reference	U12900J	U12900E
ID78K0 Integrated Debugger EWS Based	Reference	U11151J	—
ID78K0 Integrated Debugger PC Based	Reference	U11539J	U11539E
ID78K0 Integrated Debugger Windows Based	Guide	U11649J	U11649E

Caution The above documents are subject to change without prior notice. Be sure to use the latest version document when starting design.

- **Related documents for embedded software (User's Manual)**

Document Name		Document No.	
		Japanese	English
78K/0 Series Real-Time OS	Fundamentals	U11537J	U11537E
	Installation	U11536J	U11536E
78K/0 Series OS MX78K0	Fundamental	U12257J	U12257E

- **Other Documents**

Document Name	Document No.	
	Japanese	English
SEMICONDUCTOR SELECTION GUIDE Product & Packages (CD-ROM)	X13769X	
Semiconductor Device Mounting Technology Manual	C10535J	C10535E
Quality Grades on NEC Semiconductor Devices	C11531J	C11531E
NEC Semiconductor Device Reliability/Quality Control System	C10983J	C10983E
Guide to Prevent Damage for Semiconductor Devices by Electrostatic Discharge (ESD)	C11892J	C11892E
Microcomputer Related Product Series Guide - Third Party Manufacturers	U11416J	—

Caution The above documents are subject to change without prior notice. Be sure to use the latest version document when starting design.

CONTENTS

CHAPTER 1 OUTLINE (μPD780078 Subseries) 31

1.1 Features 31
1.2 Applications 32
1.3 Ordering Information 32
1.4 Pin Configuration (Top View) 33
1.5 78K/0 Series Expansion 35
1.6 Block Diagram 37
1.7 Outline of Function 38
1.8 Mask Options 39

CHAPTER 2 OUTLINE (μPD780078Y Subseries) 41

2.1 Features 41
2.2 Applications 42
2.3 Ordering Information 42
2.4 Pin Configuration (Top View) 43
2.5 78K/0 Series Expansion 45
2.6 Block Diagram 47
2.7 Outline of Function 48
2.8 Mask Options 49

CHAPTER 3 PIN FUNCTION (μPD780078 Subseries) 51

3.1 Pin Function List 51
3.2 Description of Pin Functions 54
3.2.1 P00 to P03 (Port 0) 54
3.2.2 P10 to P17 (Port 1) 54
3.2.3 P20 to P25 (Port 2) 54
3.2.4 P30 to P36 (Port 3) 55
3.2.5 P40 to P47 (Port 4) 56
3.2.6 P50 to P57 (Port 5) 56
3.2.7 P64 to P67 (Port 6) 56
3.2.8 P70 to P75 (Port 7) 57
3.2.9 P80 (Port 8) 58
3.2.10 AV_{REF} 58
3.2.11 AV_{SS} 58
3.2.12 $\overline{\text{RESET}}$ 58
3.2.13 X1 and X2 58
3.2.14 XT1 and XT2 58
3.2.15 V_{DD0} and V_{DD1} 58
3.2.16 V_{SS0} and V_{SS1} 58
3.2.17 V_{PP} (flash memory versions only) 58
3.2.18 IC (mask ROM version only) 59
3.3 Pin Input/output Circuits and Recommended Connection of Unused Pins 60

CHAPTER 4 PIN FUNCTION (μPD780078Y Subseries) **63**

4.1 Pin Function List **63**

4.2 Description of Pin Functions **66**

4.2.1 P00 to P03 (Port 0) 66

4.2.2 P10 to P17 (Port 1) 66

4.2.3 P20 to P25 (Port 2) 66

4.2.4 P30 to P36 (Port 3) 67

4.2.5 P40 to P47 (Port 4) 68

4.2.6 P50 to P57 (Port 5) 68

4.2.7 P64 to P67 (Port 6) 68

4.2.8 P70 to P75 (Port 7) 69

4.2.9 P80 (Port 8) 70

4.2.10 AV_{REF} 70

4.2.11 AV_{SS} 70

4.2.12 $\overline{\text{RESET}}$ 70

4.2.13 X1 and X2 70

4.2.14 XT1 and XT2 70

4.2.15 V_{DD0} and V_{DD1} 70

4.2.16 V_{SS0} and V_{SS1} 70

4.2.17 V_{PP} (flash memory versions only) 70

4.2.18 IC (Mask ROM version only) 71

4.3 Pin Input/output Circuits and Recommended Connection of Unused Pins **72**

CHAPTER 5 CPU ARCHITECTURE **75**

5.1 Memory Spaces **75**

5.1.1 Internal program memory space 78

5.1.2 Internal data memory space 79

5.1.3 Special Function Register (SFR) area 79

5.1.4 External memory space 79

5.1.5 Data memory addressing 80

5.2 Processor Registers **83**

5.2.1 Control registers 83

5.2.2 General-purpose registers 86

5.2.3 Special Function Register (SFR) 87

5.3 Instruction Address Addressing **91**

5.3.1 Relative addressing 91

5.3.2 Immediate addressing 92

5.3.3 Table indirect addressing 93

5.3.4 Register addressing 94

5.4 Operand Address Addressing **95**

5.4.1 Implied addressing 95

5.4.2 Register addressing 96

5.4.3 Direct addressing 97

5.4.4 Short direct addressing 98

5.4.5 Special function register (SFR) addressing 100

5.4.6 Register indirect addressing 101

5.4.7 Based addressing 102
5.4.8 Based indexed addressing 103
5.4.9 Stack addressing 103

CHAPTER 6 PORT FUNCTIONS 105

6.1 Port Functions 105
6.2 Port Configuration 108
6.2.1 Port 0 108
6.2.2 Port 1 109
6.2.3 Port 2 110
6.2.4 Port 3 (μPD780078 Subseries) 112
6.2.5 Port 3 (μPD780078Y Subseries) 114
6.2.6 Port 4 117
6.2.7 Port 5 119
6.2.8 Port 6 120
6.2.9 Port 7 121
6.2.10 Port 8 122
6.3 Port Function Control Registers 123
6.4 Port Function Operations 127
6.4.1 Writing to input/output port 127
6.4.2 Reading from input/output port 127
6.4.3 Operations on input/output port 127
6.5 Selection of Mask Option 128

CHAPTER 7 CLOCK GENERATOR 129

7.1 Clock Generator Functions 129
7.2 Clock Generator Configuration 129
7.3 Clock Generator Control Register 131
7.4 System Clock Oscillator 133
7.4.1 Main system clock oscillator 133
7.4.2 Subsystem clock oscillator 134
7.4.3 Scaler 137
7.4.4 When no subsystem clocks are used 137
7.5 Clock Generator Operations 138
7.5.1 Main system clock operations 139
7.5.2 Subsystem clock operations 140
7.6 Changing System Clock and CPU Clock Settings 140
7.6.1 Time required for switchover between system clock and CPU clock 140
7.6.2 System clock and CPU clock switching procedure 142

CHAPTER 8 16-BIT TIMER/EVENT COUNTERS 00, 01 143

8.1 Outline of 16-bit Timer/Event Counters 00, 01 143
8.2 16-Bit Timer/Event Counters 00, 01 Functions 143
8.3 16-Bit Timer/Event Counters 00, 01 Configuration 144
8.4 Registers to Control 16-Bit Timer/Event Counters 00, 01 148

8.5 16-Bit Timer/Event Counters 00, 01 Operations 158
8.5.1 Interval timer operation 158
8.5.2 PPG output operation 160
8.5.3 Pulse width measurement operations 161
8.5.4 External event counter operation 168
8.5.5 Square-wave output operation 169
8.5.6 One-shot pulse output operation 171
8.6 16-Bit Timer/Event Counters 00, 01 Cautions 174

CHAPTER 9 8-BIT TIMER/EVENT COUNTERS 50, 51 179

9.1 Outline of 8-bit Timer/Event Counters 50, 51 179
9.2 8-Bit Timer/Event Counter Functions 179
9.3 8-Bit Timer/Event Counters 50, 51 Configurations 181
9.4 Registers to Control 8-Bit Timer/Event Counters 50, 51 182
9.5 8-Bit Timer/Event Counters 50, 51 Operations 187
9.5.1 8-bit interval timer operation 187
9.5.2 External event counter operation 191
9.5.3 Square-wave output (8-bit resolution) operation 192
9.5.4 8-bit PWM output operation 193
9.5.5 Interval timer (16-bit) operations 196
9.6 8-Bit Timer/Event Counters 50, 51 Cautions 197

CHAPTER 10 WATCH TIMER 199

10.1 Outline of Watch Time 199
10.2 Watch Timer Functions 199
10.3 Watch Timer Configuration 200
10.4 Register to Control Watch Timer 200
10.5 Watch Timer Operations 202
10.5.1 Watch timer operation 202
10.5.2 Interval timer operation 202

CHAPTER 11 WATCHDOG TIMER 205

11.1 Outline of Watchdog Timer 205
11.2 Watchdog Timer Functions 205
11.3 Watchdog Timer Configuration 207
11.4 Registers to Control the Watchdog Timer 207
11.5 Watchdog Timer Operations 211
11.5.1 Watchdog timer operation 211
11.5.2 Interval timer operation 212

CHAPTER 12 CLOCK OUTPUT/BUZZER OUTPUT CONTROL CIRCUITS 213

12.1 Outline of Clock Output/Buzzer 213
12.2 Clock Output/Buzzer Output Control Circuit Functions 213
12.3 Clock Output/Buzzer Output Control Circuit Configuration 214

12.4 Register to Control Clock Output/Buzzer Output Control Circuit 214
12.5 Clock Output/Buzzer Output Control Circuit Operations 217
12.5.1 Operation as clock output 217
12.5.2 Operation as buzzer output 217

CHAPTER 13 A/D CONVERTER 219

13.1 A/D Converter Functions 219
13.2 A/D Converter Configuration 220
13.3 Registers to Control A/D Converter 221
13.4 A/D Converter Operation 224
13.4.1 Basic operations of A/D converter 224
13.4.2 Input voltage and conversion results 226
13.4.3 A/D converter operation mode 227
13.5 How to Read the A/D Converter Characteristics Table 230
13.6 A/D Converter Cautions 233

CHAPTER 14 SERIAL INTERFACE (UART0) 239

14.1 Serial Interface (UART0) Functions 239
14.2 Serial Interface (UART0) Configuration 241
14.3 Registers to Control Serial Interface (UART0) 242
14.4 Serial Interface (UART0) Operations 246
14.4.1 Operation stop mode 246
14.4.2 Asynchronous serial interface (UART) mode 247
14.4.3 Infrared data transfer mode 259

CHAPTER 15 SERIAL INTERFACE (UART2) 263

15.1 Serial Interface (UART2) Functions 263
15.2 Serial Interface (UART2) Configuration 265
15.3 Registers to Control Serial Interface (UART2) 267
15.4 Serial Interface (UART2) Operations 277
15.4.1 Operation stop mode 277
15.4.2 Asynchronous serial interface (UART) mode 278
15.4.3 Multi-processor transfer mode 300
15.4.4 Infrared data transfer (IrDA) mode 313

CHAPTER 16 SERIAL INTERFACE (SIO3) 323

16.1 Serial Interface (SIO3) Functions 323
16.2 Serial Interface (SIO3) Configuration 324
16.3 Register to Control Serial Interface (SIO3) 325
16.4 Serial Interface (SIO3) Operations 327
16.4.1 Operation stop mode 327
16.4.2 3-wire serial I/O mode 328

CHAPTER 17 SERIAL INTERFACE (CSI1) 331

17.1 Serial Interface (CSI1) Functions 331
17.2 Serial Interface (CSI1) Configuration 331
17.3 Registers to Control Serial Interface (CSI1) 333
17.4 Serial Interface (CSI1) Operations 336
17.4.1 Operation stop mode 336
17.4.2 3-wire serial I/O mode 336

CHAPTER 18 SERIAL INTERFACE (IIC0) (μPD780078Y SUBSERIES ONLY) 349

18.1 Serial Interface (IIC0) Functions 349
18.2 Serial Interface (IIC0) Configuration 352
18.3 Registers to Control Serial Interface (IIC0) 353
18.4 I^{2}C Bus Mode Functions 362
18.4.1 Pin configuration 362
18.5 I^{2}C Bus Definitions and Control Methods 363
18.5.1 Start conditions 363
18.5.2 Addresses 364
18.5.3 Transfer direction specification 364
18.5.4 Acknowledge ($\overline{\text{ACK}}$) signal 365
18.5.5 Stop condition 366
18.5.6 Wait signal ($\overline{\text{WAIT}}$) 367
18.5.7 I^{2}C interrupt requests (INTIIC0) 369
18.5.8 Interrupt request (INTIIC0) generation timing and wait control 387
18.5.9 Address match detection method 388
18.5.10 Error detection 388
18.5.11 Extension code 388
18.5.12 Arbitration 389
18.5.13 Wake up function 390
18.5.14 Communication reservation 391
18.5.15 Other cautions 393
18.5.16 Communication operations 394
18.6 Timing Charts 396

CHAPTER 19 INTERRUPT FUNCTIONS 403

19.1 Interrupt Function Types 403
19.2 Interrupt Sources and Configuration 403
19.3 Interrupt Function Control Registers 408
19.4 Interrupt Servicing Operations 414
19.4.1 Non-maskable interrupt request acknowledge operation 414
19.4.2 Maskable interrupt acknowledge operation 417
19.4.3 Software interrupt request acknowledge operation 419
19.4.4 Nesting interrupt servicing 420
19.4.5 Interrupt request hold 423

CHAPTER 20 EXTERNAL DEVICE EXPANSION FUNCTION 425

20.1 External Device Expansion Function 425
20.2 External Device Expansion Function Control Register 427
20.3 External Device Expansion Function Timing 429
20.4 Example of Connection with Memory 434

CHAPTER 21 STANDBY FUNCTION 435

21.1 Standby Function and Configuration 435
21.1.1 Standby function 435
21.1.2 Standby function control register 436
21.2 Standby Function Operations 437
21.2.1 HALT mode 437
21.2.2 STOP mode 440

CHAPTER 22 RESET FUNCTION 443

22.1 Reset Function 443

CHAPTER 23 μPD78F0078, 78F0078Y 447

23.1 Memory Size Switching Register 448
23.2 Internal Expansion RAM Size Select Register (IXS) 449
23.3 Flash Memory Programming 450
23.3.1 Selection of transmission method 450
23.3.2 Flash memory programming function 451
23.3.3 Connection of Flashpro II and Flashpro III 452

CHAPTER 24 INSTRUCTION SET 455

24.1 Legends Used in Operation List 456
24.1.1 Operand identifiers and specification methods 456
24.1.2 Description of “operation” column 457
24.1.3 Description of “flag operation” column 457
24.2 Operation List 458
24.3 Instructions Listed by Addressing Type 466

APPENDIX A DIFFERENCES BETWEEN μPD780024A, 780034A, AND 780078 SUBSERIES 471

APPENDIX B DEVELOPMENT TOOLS 473

B.1 Language Processing Software 476
B.2 Flash Memory Writing Tools 477
B.3 Debugging Tools 478
B.3.1 Hardware 478
B.3.2 Software 480
B.4 System Upgrade from Former In-circuit Emulator for 78K/0 Series to IE-78001-R-A 482

APPENDIX C EMBEDDED SOFTWARE 487

APPENDIX D REGISTER INDEX 489

D.1 Register Index (In Alphabetical Order with Respect to Register Names) 489
D.2 Register Index (In Alphabetical Order with Respect to Register Symbol) 492

LIST OF FIGURES (1/6)

Figure No.	Title	Page
3-1	Pin Input/Output Circuit List	62
4-1	Pin Input/Output Circuit List	74
5-1	Memory Map (μPD780076, 780076Y)	75
5-2	Memory Map (μPD780078, 780078Y)	76
5-3	Memory Map (μPD78F0078, 78F0078Y)	77
5-4	Data Memory Addressing (μPD780076, 780076Y)	80
5-5	Data Memory Addressing (μPD780078, 780078Y)	81
5-6	Data Memory Addressing (μPD78F0078, 78F0078Y)	82
5-7	Program Counter Format	83
5-8	Program Status Word Format	83
5-9	Stack Pointer Format	85
5-10	Data To Be Saved to Stack Memory	85
5-11	Data To Be Restored from Stack Memory	85
5-12	General-purpose Register Configuration	86
6-1	Port Types	105
6-2	P00 to P03 Block Diagram	109
6-3	P10 to P17 Block Diagram	109
6-4	P20, P22 to P25 Block Diagram	110
6-5	P21 Block Diagram	111
6-6	P30 to P33 Block Diagram (μPD780078 Subseries)	112
6-7	P34 to P36 Block Diagram (μPD780078 Subseries)	113
6-8	P30 and P31 Block Diagram (μPD780078Y Subseries)	115
6-9	P32 and P33 Block Diagram (μPD780078Y Subseries)	115
6-10	P34 to P36 Block Diagram (μPD780078Y Subseries)	116
6-11	P40 to P47 Block Diagram	117
6-12	Falling Edge Detection Circuit Block Diagram	118
6-13	P50 to P57 Block Diagram	119
6-14	P64 to P67 Block Diagram	120
6-15	P70 to P75 Block Diagram	121
6-16	P80 Block Diagram	122
6-17	Port Mode Register (PM0, PM2 to PM8) Format	124
6-18	Pull-Up Resistor Option Register (PU0, PU2 to PU8) Format	126
7-1	Clock Generator Block Diagram	130
7-2	Subsystem Clock Feedback Resistor	131
7-3	Processor Clock Control Register (PCC) Format	132
7-4	External Circuit of Main System Clock Oscillator	133
7-5	External Circuit of Subsystem Clock Oscillator	134
7-6	Examples of Incorrect Oscillator Connection	135
7-7	Main System Clock Stop Function	139
7-8	System Clock and CPU Clock Switching	142

LIST OF FIGURES (2/6)

Figure No. Title Page

8-1 16-Bit Timer/Event Counter 00 Block Diagram 144
8-2 16-Bit Timer/Event Counter 01 Block Diagram 145
8-3 16-Bit Timer Mode Control Register 00 (TMC00) Format 149
8-4 16-Bit Timer Mode Control Register 01 (TMC01) Format 150
8-5 Capture/Compare Control Register 00 (CRC00) Format 151
8-6 Capture/Compare Control Register 01 (CRC01) Format 152
8-7 16-Bit Timer Output Control Register 00 (TOC00) Format 153
8-8 16-Bit Timer Output Control Register 01 (TOC01) Format 154
8-9 Prescaler Mode Register 00 (PRM00) Format 155
8-10 Prescaler Mode Register 01 (PRM01) Format 156
8-11 Port Mode Register 7 (PM7) Format 157
8-12 Control Register Settings for Interval Timer Operation 158
8-13 Interval Timer Configuration Diagram 159
8-14 Timing of Interval Timer Operation 159
8-15 Control Register Settings for PPG Output Operation 160
8-16 Control Register Settings for Pulse Width Measurement with Free-Running Counter and One Capture Register 161
8-17 Configuration Diagram for Pulse Width Measurement by Free-Running Counter 162
8-18 Timing of Pulse Width Measurement Operation by Free-Running Counter and One Capture Register (with Both Edges Specified) 162
8-19 Control Register Settings for Measurement of Two Pulse Widths with Free-Running Counter 163
8-20 CR01n Capture Operation with Rising Edge Specified 164
8-21 Timing of Pulse Width Measurement Operation with Free-Running Counter (with Both Edges Specified) 164
8-22 Control Register Settings for Pulse Width Measurement with Free-Running Counter and Two Capture Registers 165
8-23 Timing of Pulse Width Measurement Operation by Free-Running Counter and Two Capture Registers (with Rising Edge Specified) 166
8-24 Control Register Settings for Pulse Width Measurement by Means of Restart 167
8-25 Timing of Pulse Width Measurement Operation by Means of Restart (with Rising Edge Specified) 167
8-26 Control Register Settings in External Event Counter Mode 168
8-27 External Event Counter Configuration Diagram 169
8-28 External Event Counter Operation Timings (with Rising Edge Specified) 169
8-29 Control Register Settings in Square-Wave Output Mode 170
8-30 Square-Wave Output Operation Timing 171
8-31 Control Register Settings for One-Shot Pulse Output Operation Using Software Trigger 172
8-32 Timing of One-Shot Pulse Output Operation Using Software Trigger 173
8-33 16-Bit Timer Counter 0n (TM0n) Start Timing 174
8-34 Timings after Change of Compare Register during Timer Count Operation 174
8-35 Capture Register Data Retention Timing 175
8-36 Operation Timing of OVF0n Flag 176

LIST OF FIGURES (3/6)

Figure No.	Title	Page
9-1	8-Bit Timer/Event Counter 50 Block Diagram	180
9-2	8-Bit Timer/Event Counter 51 Block Diagram	180
9-3	Timer Clock Select Register 50 (TCL50) Format	182
9-4	Timer Clock Select Register 51 (TCL51) Format	183
9-5	8-Bit Timer Mode Control Register 50 (TMC50) Format	184
9-6	8-Bit Timer Mode Control Register 51 (TMC51) Format	185
9-7	Port Mode Register 7 (PM7) Format	186
9-8	Interval Timer Operation Timings	188
9-9	External Event Counter Operation Timing (with Rising Edge Specified)	191
9-10	Square-Wave Output Operation Timing	192
9-11	PWM Output Operation Timing	194
9-12	Timing of Operation by Change of CR5n	195
9-13	16-Bit Resolution Cascade Connection Mode	197
9-14	8-Bit Timer/Counter Start Timing	197
9-15	Timing after Compare Register Transition during Timer Count Operation	198
10-1	Watch Timer Block Diagram	199
10-2	Watch Timer Operation Mode Register (WTM) Format	201
10-3	Operation Timing of Watch Timer/Interval Timer	203
11-1	Watchdog Timer Block Diagram	205
11-2	Watchdog Timer Clock Select Register (WDCS) Format	208
11-3	Watchdog Timer Mode Register (WDTM) Format	209
11-4	Oscillation Stabilization Time Select Register (OSTS) Format	210
12-1	Clock Output/Buzzer Output Control Circuit Block Diagram	213
12-2	Clock Output Select Register (CKS) Format	215
12-3	Port Mode Register 7 (PM7) Format	216
12-4	Remote Control Output Application Example	217
13-1	10-Bit A/D Converter Block Diagram	219
13-2	A/D Converter Mode Register 0 (ADM0) Format	222
13-3	Analog Input Channel Specification Register 0 (ADS0) Format	223
13-4	External Interrupt Rising Edge Enable Register (EGP), External Interrupt Falling Edge Enable Register (EGN) Format	223
13-5	Basic Operation of A/D Converter	225
13-6	Relationship between Analog Input Voltage and A/D Conversion Result	226
13-7	A/D Conversion by Hardware Start (When Falling Edge Is Specified)	228
13-8	A/D Conversion by Software Start	229
13-9	Overall Error	230
13-10	Quantization Error	230
13-11	Zero Scale Offset	231
13-12	Full Scale Offset	231
13-13	Integral Linearity Error	231

LIST OF FIGURES (4/6)

Figure No. Title Page

13-14 Differential Linearity Error 231
13-15 Example of Method of Reducing Current Consumption in Standby Mode 233
13-16 Analog Input Pin Connection 234
13-17 A/D Conversion End Interrupt Request Generation Timing 235
13-18 Timing of Reading Conversion Result (when conversion result is undefined) 236
13-19 Timing of Reading Conversion Result (when conversion result is normal) 236
13-20 Example of Connecting Capacitor to AV_{REF} Pin 237
13-21 Internal Equivalent Circuit of Pins ANI0 through ANI7 238
13-22 Example of Connection If Signal Source Impedance Is High 238

14-1 Serial Interface (UART0) Block Diagram 240
14-2 Baud Rate Generator Block Diagram 240
14-3 Asynchronous Serial Interface Mode Register 0 (ASIM0) Format 243
14-4 Asynchronous Serial Interface Status Register 0 (ASIS0) Format 244
14-5 Baud Rate Generator Control Register 0 (BRGC0) Format 245
14-6 Error Tolerance (when k = 0), Including Sampling Errors 253
14-7 Format of Transmit/Receive Data in Asynchronous Serial Interface 254
14-8 Timing of Asynchronous Serial Interface Transmit Completion Interrupt Request 256
14-9 Timing of Asynchronous Serial Interface Receive Completion Interrupt Request 257
14-10 Receive Error Timing 258
14-11 Data Format Comparison between Infrared Data Transfer Mode and UART Mode 259

15-1 Serial Interface (UART2) Block Diagram 264
15-2 Baud Rate Generator Block Diagram 264
15-3 Asynchronous Serial Interface Mode Register 2 (ASIM2) Format 268
15-4 Asynchronous Serial Interface Status Register 2 (ASIS2) Format 270
15-5 Baud Rate Generator Control Register 2 (BRGC2) Format 271
15-6 Asynchronous Serial Interface Transmit Status Register 2 (ASIF2) Format 272
15-7 Clock Select Register 2 (CKSEL2) Format 273
15-8 Transfer Mode Specification Register 2 (TRMC2) Format 275
15-9 Minimum Permissible Transfer Rate and Maximum Permissible Transfer Rate 289
15-10 Format of Transmit/Receive Data in Asynchronous Serial Interface 291
15-11 Timing of Asynchronous Serial Interface Transmit Completion Interrupt Request 293
15-12 Timing of Starting Successive Transmission 294
15-13 Timing of Completing Successive Transmission 296
15-14 Timing of Asynchronous Serial Interface Receive Completion Interrupt Request 297
15-15 Receive Error Timing 298
15-16 Reception Error Interrupts (INTSR2/INTSER2) 299
15-17 Format of Transmit/Receive Data in Multi-Processor Transfer Mode 308
15-18 Timing of Transmission Completion Interrupt in Multi-Processor Transfer Mode 310
15-19 Timing of Reception Completion Interrupt Request in Multi-Processor Transfer Mode 311
15-20 Data Format of Transmission/Reception in Infrared Data Transfer (IrDA) Mode 318
15-21 Timing of Transmission Completion Interrupt in Infrared Data (IrDA) Transfer Mode 320
15-22 Timing of Reception Completion Interrupt Request in Infrared Data (IrDA) Transfer Mode 321

LIST OF FIGURES (5/6)

Figure No. Title Page

16-1 Serial Interface (SIO3) Block Diagram 323
16-2 Serial Operation Mode Register 3 (CSIM3) Format 326
16-3 Timing of 3-Wire Serial I/O Mode 330

17-1 Serial Interface (CSI1) Block Diagram 332
17-2 Serial Operation Mode Register 1 (CSIM1) Format 334
17-3 Serial Clock Select Register 1 (CSIC1) Format 335
17-4 Timing of 3-Wire Serial I/O Mode 341
17-5 Timing of Clock/Data Phase 343
17-6 Timing of Output to SO1 Pin (first bit) 344
17-7 Output Value of SO1 Pin (last bit) 345

18-1 Serial Interface (IIC0) Block Diagram 350
18-2 Serial Bus Configuration Example Using I^{2}C Bus 351
18-3 IIC Control Register 0 (IICC0) Format 354
18-4 IIC Status Register 0 (IICS0) Format 357
18-5 IIC Transfer Clock Select Register 0 (IICCL0) Format 360
18-6 Pin Configuration Diagram 362
18-7 I^{2}C Bus Serial Data Transfer Timing 363
18-8 Start Conditions 363
18-9 Address 364
18-10 Transfer Direction Specification 364
18-11 $\overline{\text{ACK}}$ Signal 365
18-12 Stop Condition 366
18-13 Wait Signal 367
18-14 Arbitration Timing Example 389
18-15 Communication Reservation Timing 392
18-16 Timing for Accepting Communication Reservations 392
18-17 Communication Reservation Protocol 393
18-18 Master Operation Flow Chart 394
18-19 Slave Operation Flow Chart 395
18-20 Example of Master to Slave Communication (When 9-clock Wait Is Selected for Both Master and Slave) 397
18-21 Example of Slave to Master Communication (When 9-clock Wait Is Selected for Both Master and Slave) 400

19-1 Basic Configuration of Interrupt Function 406
19-2 Interrupt Request Flag Register (IF0L, IF0H, IF1L) Format 409
19-3 Interrupt Mask Flag Register (MK0L, MK0H, MK1L) Format 410
19-4 Priority Specification Flag Register (PR0L, PR0H, PR1L) Format 411
19-5 External Interrupt Rising Edge Enable Register (EGP), External Interrupt Falling Edge Enable Register (EGN) Format 412
19-6 Program Status Word Format 413
19-7 Non-Maskable Interrupt Request Generation to Acknowledge Flowchart 415

LIST OF FIGURES (6/6)

Figure No. Title Page

19-8 Non-Maskable Interrupt Request Acknowledge Timing 415
19-9 Non-Maskable Interrupt Request Acknowledge Operation 416
19-10 Interrupt Request Acknowledge Processing Algorithm 418
19-11 Interrupt Request Acknowledge Timing (Minimum Time) 419
19-12 Interrupt Request Acknowledge Timing (Maximum Time) 419
19-13 Nesting Examples 421
19-14 Interrupt Request Hold 423

20-1 Memory Map When Using External Device Function 426
20-2 Memory Expansion Mode Register (MEM) Format 427
20-3 Memory Expansion Wait Setting Register (MM) Format 428
20-4 Instruction Fetch from External Memory 430
20-5 External Memory Read Timing 431
20-6 External Memory Write Timing 432
20-7 External Memory Read Modify Write Timing 433
20-8 Connection Example of μPD780078 and Memory 434

21-1 Oscillation Stabilization Time Select Register (OSTS) Format 436
21-2 HALT Mode Release by Interrupt Request Generation 438
21-3 HALT Mode Release by $\overline{\text{RESET}}$ Input 439
21-4 STOP Mode Release by Interrupt Request Generation 441
21-5 STOP Mode Release by $\overline{\text{RESET}}$ Input 442

22-1 Reset Function Block Diagram 443
22-2 Timing of Reset by $\overline{\text{RESET}}$ Input 444
22-3 Timing of Reset Due to Watchdog Timer Overflow 444
22-4 Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input 444

23-1 Memory Size Switching Register (IMS) Format 448
23-2 Internal Expansion RAM Size Select Register (IXS) Format 449
23-3 Transmission Method Selection Format 451
23-4 Connection of Flashpro II and Flashpro III Using 3-Wire Serial I/O (SIO3) Method 452
23-5 Connection of Flashpro III Using 3-wire Serial I/O (SIO3) (When Handshake Is Used) 452
23-6 Connection of Flashpro II and Flashpro III Using I^{2}C (IIC0) Bus Method (μPD78F0078Y Only) 453
23-7 Connection of Flashpro II and Flashpro III Using UART (UART0) Method 453

B-1 Development Tool Configuration 474
B-2 EV-9200GC-64 Package Drawing (for reference only) 483
B-3 EV-9200GC-64 Recommended Board Mounting Pattern (for reference only) 484
B-4 TGK-064SBP Package Drawing (for reference only) 485

LIST OF TABLES (1/3)

Table No.	Title	Page
1-1	Mask Options of Mask ROM Versions	39
2-1	Mask Options of Mask ROM Versions	49
3-1	Pin Input/Output Circuit Types	60
4-1	Pin Input/Output Circuit Types	72
5-1	Internal ROM Capacity	78
5-2	Vector Table	78
5-3	Special Function Register List	88
6-1	Port Functions (μPD780078 Subseries)	106
6-2	Port Functions (μPD780078Y Subseries)	107
6-3	Port Configuration	108
6-4	Pull-Up Resistor of Port 3 (μPD780078 Subseries)	112
6-5	Pull-Up Resistor of Port 3 (μPD780078Y Subseries)	114
6-6	Comparison between Mask ROM Version and Flash Memory Version	128
7-1	Clock Generator Configuration	129
7-2	Relationship of CPU Clock and Min. Instruction Execution Time	133
7-3	Maximum Time Required for CPU Clock Switchover	141
8-1	16-Bit Timer/Event Counters 00, 01 Configuration	144
8-2	TI00n Pin Valid Edge and CR00n, CR01n Capture Trigger	146
8-3	TI01n Pin Valid Edge and CR00n Capture Trigger	146
9-1	8-Bit Timer/Event Counters 50, 51 Configuration	181
10-1	Interval Timer Interval Time	200
10-2	Watch Timer Configuration	200
10-3	Interval Timer Interval Time	202
11-1	Watchdog Timer Runaway Detection Time	206
11-2	Interval Time	206
11-3	Watchdog Timer Configuration	207
11-4	Watchdog Timer Runaway Detection Time	211
11-5	Interval Timer Interval Time	212
12-1	Clock Output/Buzzer Output Control Circuits Configuration	214
13-1	A/D Converter Configuration	220
13-2	Resistances and Capacitances of Equivalent Circuit (reference values)	238

LIST OF TABLES (2/3)

Table No.	Title	Page
14-1	Serial Interface (UART0) Configuration	241
14-2	Relationship between 5-Bit Counter’s Source Clock and “n” Value	251
14-3	Relationship between Main System Clock and Baud Rate	252
14-4	Causes of Receive Errors	258
14-5	Bit Rate and Pulse Width Values	260
15-1	Serial Interface (UART2) Configuration	265
15-2	Relationship between 8-Bit Counter’s Source Clock and “n” Value	286
15-3	Relationship between Input Clock of Baud Rate Generator and “k” Value	287
15-4	Relationship between Main System Clock and Baud Rate	288
15-5	Maximum Permissible Baud Rate Error and Minimum Permissible Baud Rate Error	290
15-6	Transmission Status and Writing to TXB2	294
15-7	Timing of Starting Successive Transmission	295
15-8	Timing of Completing Successive Transmission	296
15-9	Causes of Receive Errors	298
15-10	Bit Rate and Pulse Width	322
16-1	Serial Interface (SIO3) Configuration	324
17-1	Serial Interface (CSI1) Configuration	331
17-2	Status of SCK1 Pin	347
17-3	Status of SO1 Pin	347
18-1	Serial Interface (IIC0) Configuration	352
18-2	INTIIC0 Timing and Wait Control	387
18-3	Extension Code Bit Definitions	388
18-4	Status during Arbitration and Interrupt Request Generation Timing	390
18-5	Wait Periods	391
19-1	Interrupt Source List	404
19-2	Flags Corresponding to Interrupt Request Sources	408
19-3	Times from Generation of Maskable Interrupt until Servicing	417
19-4	Interrupt Request Enabled for Nesting during Interrupt Servicing	420
20-1	Pin Functions in External Memory Expansion Mode	425
20-2	State of Port 4 to 6 Pins in External Memory Expansion Mode	425
21-1	HALT Mode Operating Statuses	437
21-2	Operation after HALT Mode Release	439
21-3	STOP Mode Operating Status	440
21-4	Operation after STOP Mode Release	442
22-1	Hardware Statuses after Reset	445

LIST OF TABLES (3/3)

Table No.	Title	Page
23-1	Differences among μPD78F0078 and Mask ROM Versions	447
23-2	Memory Size Switching Register Settings	448
23-3	Transmission Method List	450
23-4	Main Functions of Flash Memory Programming	451
24-1	Operand Identifiers and Specification Methods	456
A-1	Major Differences between μPD780024A, 780034A, and 780078 Subseries	471
B-1	System Upgrade Method from Former In-Circuit Emulator for 78K/0 Series To the IE-78001-R-A	482

[MEMO]

CHAPTER 1 OUTLINE (μPD780078 SUBSERIES)

1.1 Features

- Internal Memory

Type	Program Memory	Data Memory	
Part Number	(Mask ROM/Flash memory)	High-Speed RAM	Expansion RAM
μPD780076	48 Kbytes	1024 bytes	1024 bytes
μPD780078	60 Kbytes		
μPD78F0078	60 Kbytes[Note]		

Note The capacities of RAM can be changed by means of the memory size switching register (IMS).

- External Memory Expansion Space: 64 Kbytes
- Minimum Instruction execution time changeable from high speed (0.24 μs: @ 8.38-MHz operation with main system clock) to ultra-low speed (122 μs: @ 32.768-kHz operation with subsystem clock)
- Instruction set suited to system control
  - Bit manipulation possible in all address spaces
  - Multiply and divide instructions
- 52 I/O ports: (Four N-ch open-drain ports)
- 10-bit resolution A/D converter : 8 channels
- Serial interface : 3 channels
  - 3-wire serial I/O mode : 1 channel
  - UART mode : 1 channel
  - 3-wire serial I/O/UART mode selectable: 1 channel
- Timer: Six channels
  - 16-bit timer/event counter : 2 channels
  - 8-bit timer/event counter : 2 channels
  - Watch timer : 1 channel
  - Watchdog timer : 1 channel
- Vectored interrupts: 25
- Two types of on-chip clock oscillators (main system clock and subsystem clock)
- Power supply voltage: V_{DD} = 1.8 to 5.5 V (μPD780076, 780078)
  V_{DD} = 2.7 to 5.5 V (μPD78F0078)

1.2 Applications

Personal computers, air conditioners, dash boards, air bags, audio, etc.

1.3 Ordering Information

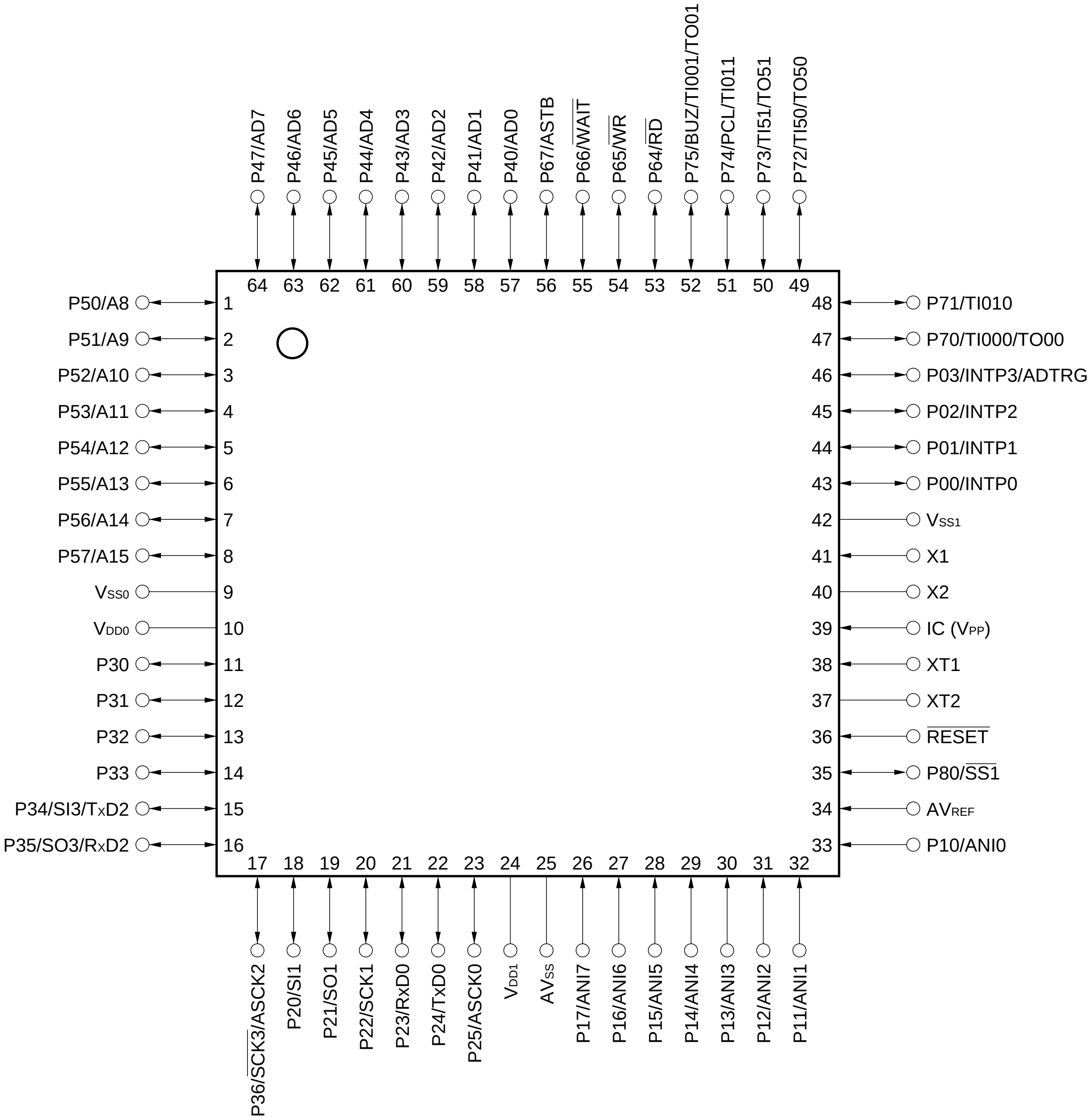
Part Number	Package	Internal ROM
μPD780076GC-×××-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780076GK-×××-9ET	64-pin plastic TQFP (12 × 12 mm)	Mask ROM
μPD780078GC-×××-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780078GK-×××-9ET	64-pin plastic TQFP (12 × 12 mm)	Mask ROM
μPD78F0078GC-AB8	64-pin plastic QFP (14 × 14 mm)	Flash memory
μPD78F0078GK-9ET	64-pin plastic TQFP (12 × 12 mm)	Flash memory

Remark ××× indicates ROM code suffix.

1.4 Pin Configuration (Top View)

- **64-pin plastic QFP (14 × 14 mm)**
  μPD780076GC-×××-AB8, 780078GC-×××-AB8, 78F0078GC-AB8

- **64-pin plastic TQFP (12 × 12 mm)**
  μPD780076GK-×××-9ET, 780078GK-×××-9ET, 78F0078GK-9ET



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

2. Connect AV_{SS} pin to V_{SS0}.

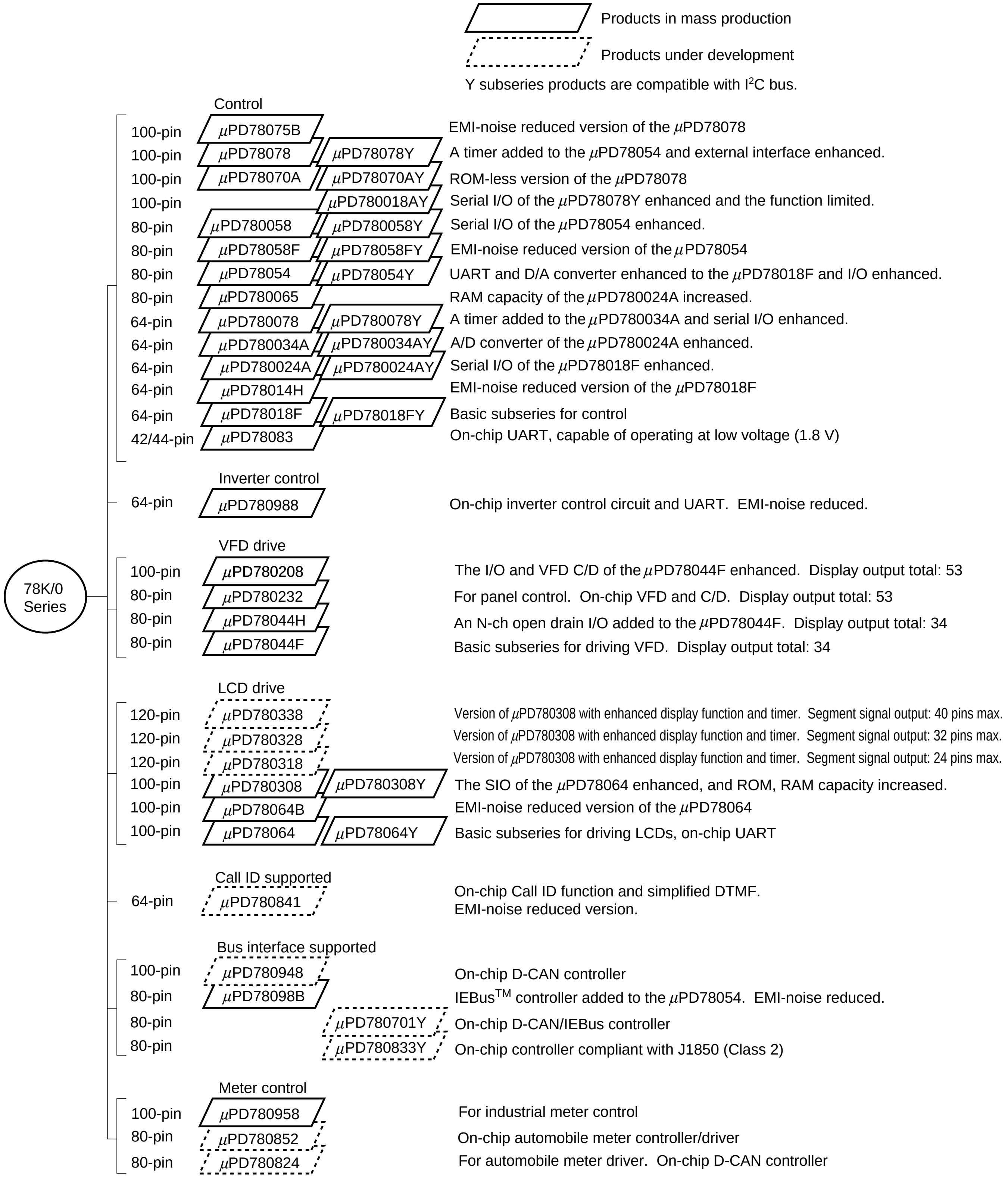
Remarks 1. When these devices are used in applications that require the reduction of noise generated from an on-chip microcontroller, the implementation of noise measures is recommended, such as supplying V_{DD0} and V_{DD1} independently, connecting V_{SS0} and V_{SS1} independently to ground lines, and so on.

2. Pin connection in parentheses is intended for the μPD78F0078.

A8 to A15	: Address Bus	PCL	: Programmable Clock
AD0 to AD7	: Address/Data Bus	$\overline{\text{RD}}$	: Read Strobe
ADTRG	: AD Trigger Input	$\overline{\text{RESET}}$	: Reset
ANI0 to ANI7	: Analog Input	R_XD0, R_XD2	: Receive Data
ASCK0, ASCK2	: Asynchronous Serial Clock	SCK1, $\overline{\text{SCK3}}$	: Serial Clock
ASTB	: Address Strobe	SI1, SI3	: Serial Input
AV_{REF}	: Analog Reference Voltage	SO1, SO3	: Serial Output
AV_{SS}	: Analog Ground	$\overline{\text{SS1}}$	: Serial Interface Chip Select Input
BUZ	: Buzzer Clock	TI000, TI010, TI001,	: Timer Input
IC	: Internally Connected	TI011, TI50, TI51	
INTP0 to INTP3	: External Interrupt Input	TO00, TO01, TO50,	: Timer Output
P00 to P03	: Port0	TO51	
P10 to P17	: Port1	T_XD0, T_XD2	: Transmit Data
P20 to P25	: Port2	V_{DD0}, V_{DD1}	: Power Supply
P30 to P36	: Port3	V_{PP}	: Programming Power Supply
P40 to P47	: Port4	V_{SS0}, V_{SS1}	: Ground
P50 to P57	: Port5	$\overline{\text{WAIT}}$	: Wait
P64 to P67	: Port6	$\overline{\text{WR}}$	: Write Strobe
P70 to P75	: Port7	X1, X2	: Crystal (Main System Clock)
P80	: Port8	XT1, XT2	: Crystal (Subsystem Clock)

1.5 78K/0 Series Expansion

The following shows the products organized according to usage. The names in the parallelograms are subseries.



Y subseries products are compatible with I^{2}C bus.

Control			
100-pin	μPD78075B		EMI-noise reduced version of the μPD78078
100-pin	μPD78078	μPD78078Y	A timer added to the μPD78054 and external interface enhanced.
100-pin	μPD78070A	μPD78070AY	ROM-less version of the μPD78078
100-pin		μPD780018AY	Serial I/O of the μPD78078Y enhanced and the function limited.
80-pin	μPD780058	μPD780058Y	Serial I/O of the μPD78054 enhanced.
80-pin	μPD78058F	μPD78058FY	EMI-noise reduced version of the μPD78054
80-pin	μPD78054	μPD78054Y	UART and D/A converter enhanced to the μPD78018F and I/O enhanced.
80-pin	μPD780065		RAM capacity of the μPD780024A increased.
64-pin	μPD780078	μPD780078Y	A timer added to the μPD780034A and serial I/O enhanced.
64-pin	μPD780034A	μPD780034AY	A/D converter of the μPD780024A enhanced.
64-pin	μPD780024A	μPD780024AY	Serial I/O of the μPD78018F enhanced.
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 control circuit and UART. EMI-noise reduced.
VFD drive			
100-pin	μPD780208		The I/O and VFD C/D of the μPD78044F enhanced. Display output total: 53
80-pin	μPD780232		For panel control. On-chip VFD and C/D. Display output total: 53
80-pin	μPD78044H		An N-ch open drain I/O added to the μPD78044F. Display output total: 34
80-pin	μPD78044F		Basic subseries for driving VFD. Display output total: 34
LCD drive			
120-pin	μPD780338		Version of μPD780308 with enhanced display function and timer. Segment signal output: 40 pins max.
120-pin	μPD780328		Version of μPD780308 with enhanced display function and timer. Segment signal output: 32 pins max.
120-pin	μPD780318		Version of μPD780308 with enhanced display function and timer. Segment signal output: 24 pins max.
100-pin	μPD780308	μPD780308Y	The SIO of the μPD78064 enhanced, and ROM, RAM capacity increased.
100-pin	μPD78064B		EMI-noise reduced version of the μPD78064
100-pin	μPD78064	μPD78064Y	Basic subseries for driving LCDs, on-chip UART
Call ID supported			
64-pin	μPD780841		On-chip Call ID function and simplified DTMF. EMI-noise reduced version.
Bus interface supported			
100-pin	μPD780948		On-chip D-CAN controller
80-pin	μPD78098B		IEBus™ controller added to the μPD78054. EMI-noise reduced.
80-pin		μPD780701Y	On-chip D-CAN/IEBus controller
80-pin		μPD780833Y	On-chip controller compliant with J1850 (Class 2)
Meter control			
100-pin	μPD780958		For industrial meter control
80-pin	μPD780852		On-chip automobile meter controller/driver
80-pin	μPD780824		For automobile meter driver. On-chip D-CAN 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.

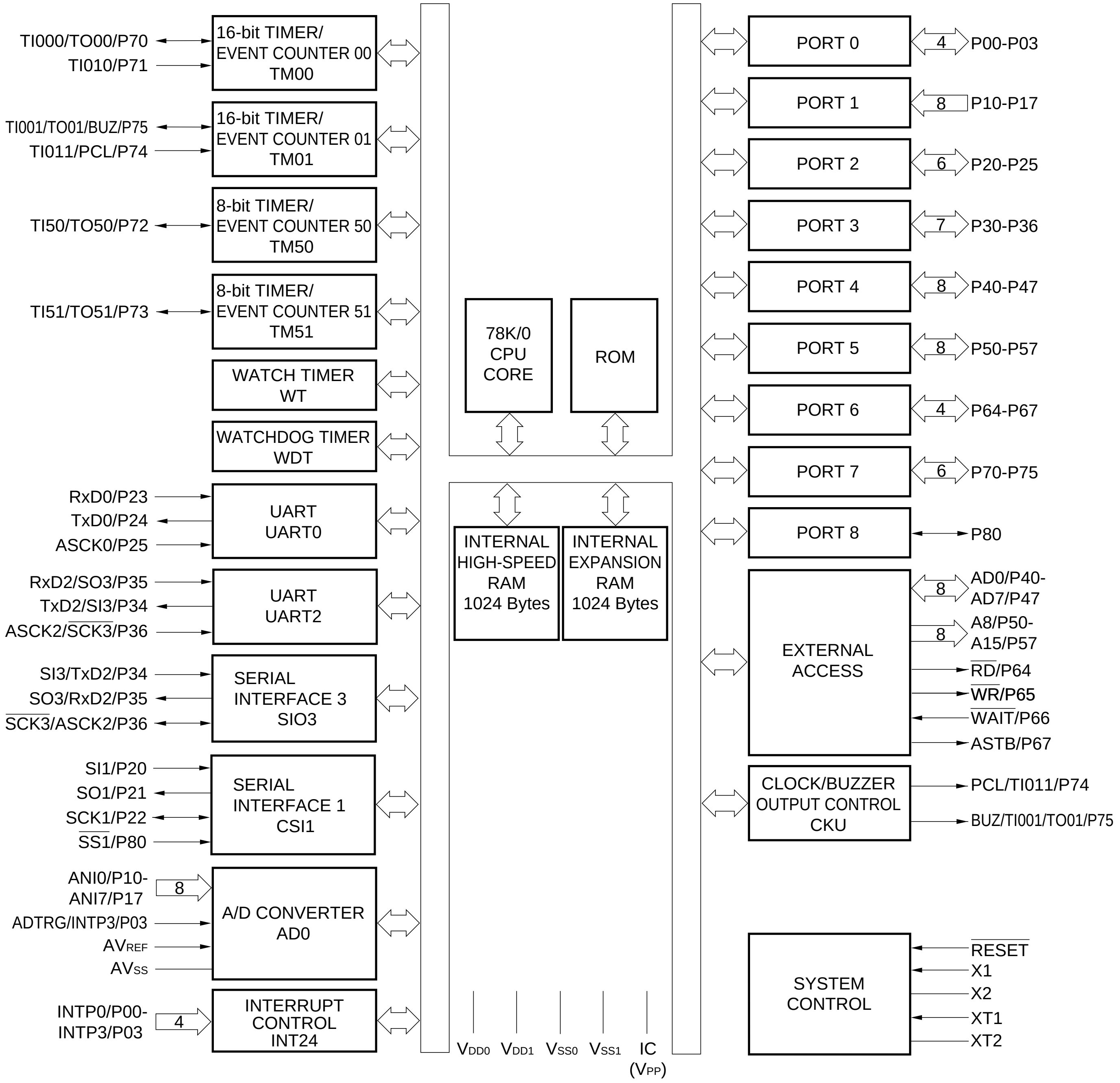
The major functional differences among the subseries are shown below.

- **Subseries without the suffix Y**

	Function	ROM	Timer				8-bit	10-bit	8-bit	Serial Interface	I/O	V_{DD} MIN. Value	External Expansion
Subseries Name		Capacity	8-bit	16-bit	Watch	WDT	A/D	A/D	D/A				
Control	μPD78075B	32 K-40 K	4ch	1ch	1ch	1ch	8ch	–	2ch	3ch (UART: 1 ch)	88	1.8 V	Yes
	μPD78078	48 K-60 K											
	μPD78070A	–									61	2.7 V	
	μPD780058	24 K-60 K	2ch							3ch (time-division UART: 1ch)	68	1.8 V	
	μPD78058F	48 K-60 K								3ch (UART: 1ch)	69	2.7 V	
	μPD78054	16 K-60 K										2.0 V	
	μPD780065	40 K-48 K							–	4ch (UART: 1ch)	60	2.7 V	
	μPD780078	48 K-60 K		2ch			–	8ch		3ch (UART: 2ch)	52	1.8 V	
	μPD780034A	8 K-32 K		1ch						3ch (UART: 1ch, time-division 3-wire: 1ch)	51		
	μPD780024A						8ch	–					
	μPD78014H									2ch	53		
	μPD78018F	8 K-60 K											
	μPD78083	8 K-16 K		–	–					1ch (UART: 1ch)	33		–
Inverter control	μPD780988	32 K-60 K	3ch	**Note**	–	1ch	–	8ch	–	3ch (UART: 2ch)	47	4.0 V	Yes
VFD drive	μPD780208	32 K-60 K	2ch	1ch	1ch	1ch	8ch	–	–	2ch	74	2.7 V	–
	μPD780232	16 K-24 K	3ch	–	–		4ch				40	4.5 V	
	μPD78044H	32 K-48 K	2ch	1ch	1ch		8ch			1ch	68	2.7 V	
	μPD78044F	16 K-40 K								2ch			
LCD drive	μPD780338	48 K-60 K	3ch	2ch	1ch	1ch	–	10ch	1ch	2ch (UART: 1ch)	54	1.8 V	–
	μPD780328										62		
	μPD780318										70		
	μPD780308	48 K-60 K	2ch	1ch			8ch	–	–	3ch (time-division UART: 1ch)	57	2.0 V	
	μPD78064B	32 K								2ch (UART: 1ch)			
	μPD78064	16 K-32 K											
Call ID supported	μPD780841	24 K-32 K	1ch	1ch	1ch	1ch	2ch	–	–	2ch (UART: 1ch)	57	2.7 V	–
Bus interface supported	μPD780948	60 K	2ch	2ch	1ch	1ch	8ch	–	–	3ch (UART: 1ch)	79	4.0 V	Yes
	μPD78098B	40 K-60 K		1ch					2ch		69	2.7 V	–
Meter control	μPD780958	48 K-60 K	4ch	2ch	–	1ch	–	–	–	2ch (UART: 1ch)	69	2.2 V	–
Dash board control	μPD780852	32 K-40 K	3ch	1ch	1ch	1ch	5ch	–	–	3ch (UART: 1ch)	56	4.0 V	–
	μPD780824	32 K-60 K								2ch (UART: 1ch)	59		

Note 16-bit timer: 2 channels
10-bit timer: 1 channel

1.6 Block Diagram



Remarks 1. The internal ROM and RAM capacities depend on the product.

2. Pin connection in parentheses is intended for the μPD78F0078.

1.7 Outline of Function

Part Number Item		μPD780076	μPD780078	μPD78F0078
Internal memory	ROM	48 Kbytes (Mask ROM)	60 Kbytes (Mask ROM)	60 Kbytes[Note 1] (Flash memory)
	High-speed RAM	1024 bytes		
	Expansion RAM	1024 bytes		
Memory space		64 Kbytes		
General-purpose register		8 bits × 32 registers (8 bits × 8 registers × 4 banks)		
Minimum instruction execution time		Minimum instruction execution time changeable function		
	When main system clock selected	0.24 μs/0.48 μs/0.95 μs/1.91 μs/3.81 μs (@ 8.38-MHz operation)		
	When subsystem clock selected	122 μs (@ 32.768-kHz operation)		
Instruction set		• 16-bit operation • Multiply/divide (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulate (set, reset, test, and Boolean operation) • BCD adjust, etc.		
I/O port		Total : 52 • CMOS input : 8 • CMOS I/O : 40 • N-ch open-drain I/O 5-V breakdown : 4		
A/D converter		• 10-bit resolution × 8 channels • Low-voltage operation: AV_{REF} = 2.7 to 5.5 V		
Serial interface		• 3-wire serial I/O mode : 1 channel • UART mode : 1 channel • 3-wire serial I/O/UART mode selectable[Note 2] : 1 channel		
Timer		• 16-bit timer/event counter : 2 channels • 8-bit timer/event counter : 2 channels • Watch timer : 1 channel • Watchdog timer : 1 channel		
Timer output		4 outputs: (8-bit PWM output enable: 2)		
Clock output		• 65.5 kHz, 131 kHz, 262 kHz, 524 kHz, 1.05 MHz, 2.10 MHz, 4.19 MHz, 8.38 MHz (8.38 MHz with main system clock) • 32.768 kHz (32.768 kHz with subsystem clock)		
Buzzer output		1.02 kHz, 2.05 kHz, 4.10 kHz, 8.19 kHz (8.38 MHz with main system clock)		
Vectored interrupt	Maskable	Internal: 18, External: 5		
	Non-maskable	Internal: 1		
	Software	1		
Power supply voltage		V_{DD} = 1.8 to 5.5 V		V_{DD} = 2.7 to 5.5 V
Operating ambient temperature		T_A = –40 to +85°C		
Package		• 64-pin plastic QFP (14 × 14 mm) • 64-pin plastic TQFP (12 × 12 mm)		

Notes 1. The capacities of internal flash memory can be changed by means of the memory size switching register (IMS).

2. Select either of the functions of these multiplexed pins.

The following table outlines the timers/event counters (for details, refer to **CHAPTER 8 16-BIT TIMER/EVENT COUNTERS 00, 01; CHAPTER 9 8-BIT TIMER/EVENT COUNTERS 50, 51; CHAPTER 10 WATCH TIMER; and CHAPTER 11 WATCHDOG TIMER)**:

		16-Bit Timer/Event Counters 00, 01	8-Bit Timer/Event Counters 50, 51	Watch Timer	Watchdog Timer
Operation mode	Interval timer	2 channels	2 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 be used both as a watch timer and an interval timer at the same time.

2. The watchdog timer can be used as either a watchdog timer or interval timer. Select one of the functions.

1.8 Mask Options

The mask ROM versions (μPD780076 and 780078) provide pull-up resistor mask options which allow users to specify whether to connect a pull-up resistor to a specific port pin when the user places an order for device production. Using the mask option when pull-up resistors are required reduces the number of components to add to the device, resulting in board space saving.

The mask options provided in the μPD780078 Subseries are shown in Table 1-1.

Table 1-1. Mask Options of Mask ROM Versions

Pin Names	Mask Option
P30 to P33	Pull-up resistor connection can be specified in 1-bit units.

[MEMO]

CHAPTER 2 OUTLINE (μPD780078Y SUBSERIES)

2.1 Features

- Internal Memory

Type / Part Number	Program Memory (Mask ROM/flash memory)	Data Memory	
		High-Speed RAM	Expansion RAM
μPD780076Y	48 Kbytes	1024 bytes	1024 bytes
μPD780078Y	60 Kbytes		
μPD78F0078Y	60 Kbytes[Note]		

Note The capacities of internal flash memory can be changed by means of the memory size switching register.

- External Memory Expansion Space: 64 Kbytes
- Minimum instruction execution time changeable from high speed (0.24 μs: @ 8.38-MHz operation with main system clock) to ultra-low speed (122 μs: @ 32.768-kHz operation with subsystem clock)
- Instruction set suited to system control
  - Bit manipulation possible in all address spaces
  - Multiply and divide instructions
- 52 I/O ports: (Four N-ch open-drain ports)
- 10-bit resolution A/D converter : 8 channels
- Serial interface : 4 channels
  - 3-wire serial mode : 1 channel
  - UART mode : 1 channel
  - 3-wire serial I/O/UART mode selectable : 1 channel
  - I^{2}C mode : 1 channel
- Timer: 6 channels
  - 16-bit timer/event counter : 2 channels
  - 8-bit timer/event counter : 2 channels
  - Watch timer : 1 channel
  - Watchdog timer : 1 channel
- Vectored interrupts: 26
- Two types of on-chip clock oscillators (main system clock and subsystem clock)
- Power supply voltage: V_{DD} = 1.8 to 5.5 V (μPD780076Y, 780078Y)
  V_{DD} = 2.7 to 5.5 V (μPD78F0078Y)

2.2 Applications

Personal computers, air conditioners, dash boards, air bags, car audio, etc.

2.3 Ordering Information

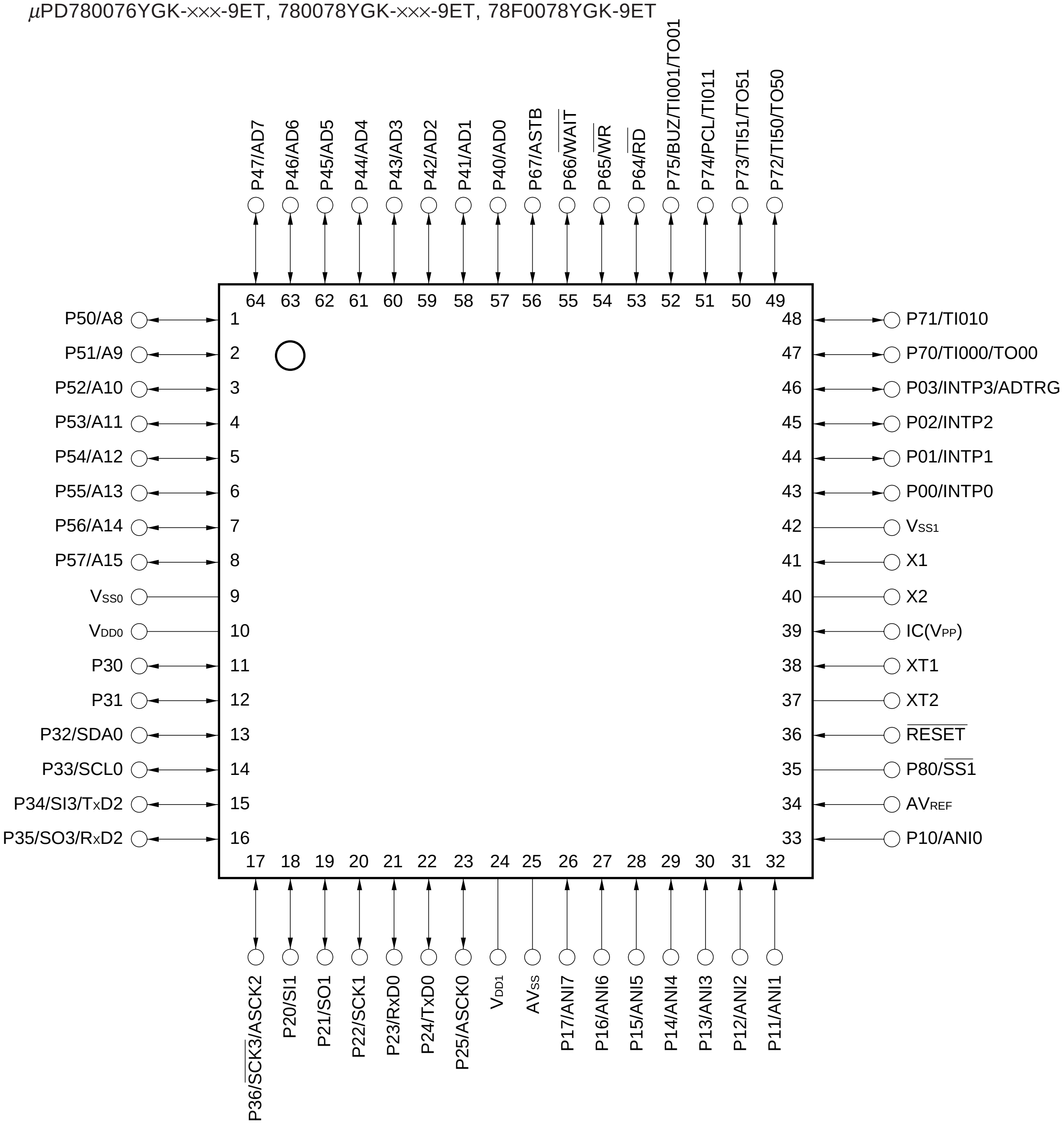
Part Number	Package	Internal ROM
μPD780076YGC-×××-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780076YGK-×××-9ET	64-pin plastic TQFP (12 × 12 mm)	Mask ROM
μPD780078YGC-×××-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780078YGK-×××-9ET	64-pin plastic TQFP (12 × 12 mm)	Mask ROM
μPD78F0078YGC-AB8	64-pin plastic QFP (14 × 14 mm)	Flash memory
μPD78F0078YGK-9ET	64-pin plastic TQFP (12 × 12 mm)	Flash memory

Remark ××× indicates ROM code suffix.

2.4 Pin Configuration (Top View)

- **64-pin plastic QFP (14 × 14 mm)**
  μPD780076YGC-×××-AB8, 780078YGC-×××-AB8, 78F0078YGC-AB8

- **64-pin plastic TQFP (12 × 12 mm)**
  μPD780076YGK-×××-9ET, 780078YGK-×××-9ET, 78F0078YGK-9ET



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

2. Connect AV_{SS} pin to V_{SS0}.

Remarks 1. When these devices are used in applications that require the reduction of noise generated from an on-chip microcontroller, the implementation of noise measures is recommended, such as supplying V_{DD0} and V_{DD1} independently, connecting V_{SS0} and V_{SS1} independently to ground lines, and so on.

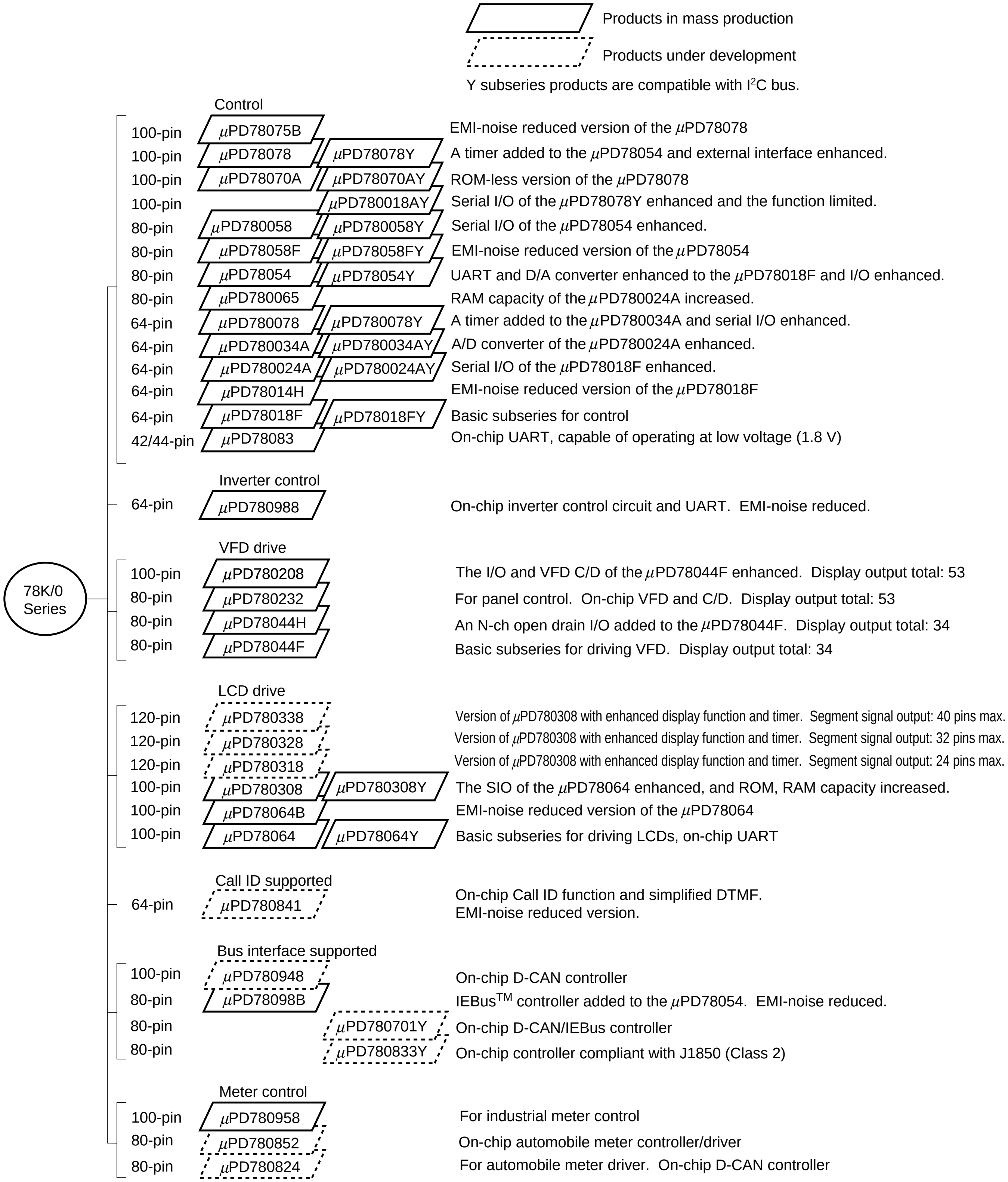
2. Pin connection in parentheses is intended for the μPD78F0078Y.

A8 to A15	: Address Bus
AD0 to AD7	: Address/Data Bus
ADTRG	: AD Trigger Input
ANI0 to ANI7	: Analog Input
ASCK0, ASCK2	: Asynchronous Serial Clock
ASTB	: Address Strobe
AV_{REF}	: Analog Reference Voltage
AV_{SS}	: Analog Ground
BUZ	: Buzzer Clock
IC	: Internally Connected
INTP0 to INTP3	: External Interrupt Input
P00 to P03	: Port 0
P10 to P17	: Port 1
P20 to P25	: 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	: Port 8

PCL	: Programmable Clock
$\overline{RD}$	: Read Strobe
$\overline{RESET}$	: Reset
$R_{X}D0$, $R_{X}D2$	: Receive Data
SCK1, $\overline{SCK3}$, SCL0	: Serial Clock
SDA0	: Serial Data
SI1, SI3	: Serial Input
SO1, SO3	: Serial Output
$\overline{SS1}$	: Serial Interface Chip Select Input
TI000, TI010, TI001, TI011, TI50, TI51	: Timer Input
TO00, TO01, TO50, TO51	: Timer Output
$T_{X}D0$, $T_{X}D2$	: Transmit Data
V_{DD0}, V_{DD1}	: Power Supply
V_{PP}	: Programming Power Supply
V_{SS0}, V_{SS1}	: Ground
$\overline{WAIT}$	: Wait
$\overline{WR}$	: Write Strobe
X1, X2	: Crystal (Main System Clock)
XT1, XT2	: Crystal (Subsystem Clock)

2.5 78K/0 Series Expansion

The following shows the products organized according to usage. The names in the parallelograms are subseries.


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 A timer added to the μPD78054 and external interface enhanced.
100-pin μPD78070A μPD78070AY ROM-less version of the μPD78078
100-pin μPD780018AY Serial I/O of the μPD78078Y enhanced and the function limited.
80-pin μPD780058 μPD780058Y Serial I/O of the μPD78054 enhanced.
80-pin μPD78058F μPD78058FY EMI-noise reduced version of the μPD78054
80-pin μPD78054 μPD78054Y UART and D/A converter enhanced to the μPD78018F and I/O enhanced.
80-pin μPD780065 RAM capacity of the μPD780024A increased.
64-pin μPD780078 μPD780078Y A timer added to the μPD780034A and serial I/O enhanced.
64-pin μPD780034A μPD780034AY A/D converter of the μPD780024A enhanced.
64-pin μPD780024A μPD780024AY Serial I/O of the μPD78018F enhanced.
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 control circuit and UART. EMI-noise reduced.
VFD drive
100-pin μPD780208 The I/O and VFD C/D of the μPD78044F enhanced. Display output total: 53
80-pin μPD780232 For panel control. On-chip VFD and C/D. Display output total: 53
80-pin μPD78044H An N-ch open drain I/O added to the μPD78044F. Display output total: 34
80-pin μPD78044F Basic subseries for driving VFD. Display output total: 34
LCD drive
120-pin μPD780338 Version of μPD780308 with enhanced display function and timer. Segment signal output: 40 pins max.
120-pin μPD780328 Version of μPD780308 with enhanced display function and timer. Segment signal output: 32 pins max.
120-pin μPD780318 Version of μPD780308 with enhanced display function and timer. Segment signal output: 24 pins max.
100-pin μPD780308 μPD780308Y The SIO of the μPD78064 enhanced, and ROM, RAM capacity increased.
100-pin μPD78064B EMI-noise reduced version of the μPD78064
100-pin μPD78064 μPD78064Y Basic subseries for driving LCDs, on-chip UART
Call ID supported
64-pin μPD780841 On-chip Call ID function and simplified DTMF.
EMI-noise reduced version.
Bus interface supported
100-pin μPD780948 On-chip D-CAN controller
80-pin μPD78098B IEBus™ controller added to the μPD78054. EMI-noise reduced.
80-pin μPD780701Y On-chip D-CAN/IEBus controller
80-pin μPD780833Y On-chip controller compliant with J1850 (Class 2)
Meter control
100-pin μPD780958 For industrial meter control
80-pin μPD780852 On-chip automobile meter controller/driver
80-pin μPD780824 For automobile meter driver. On-chip D-CAN 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.

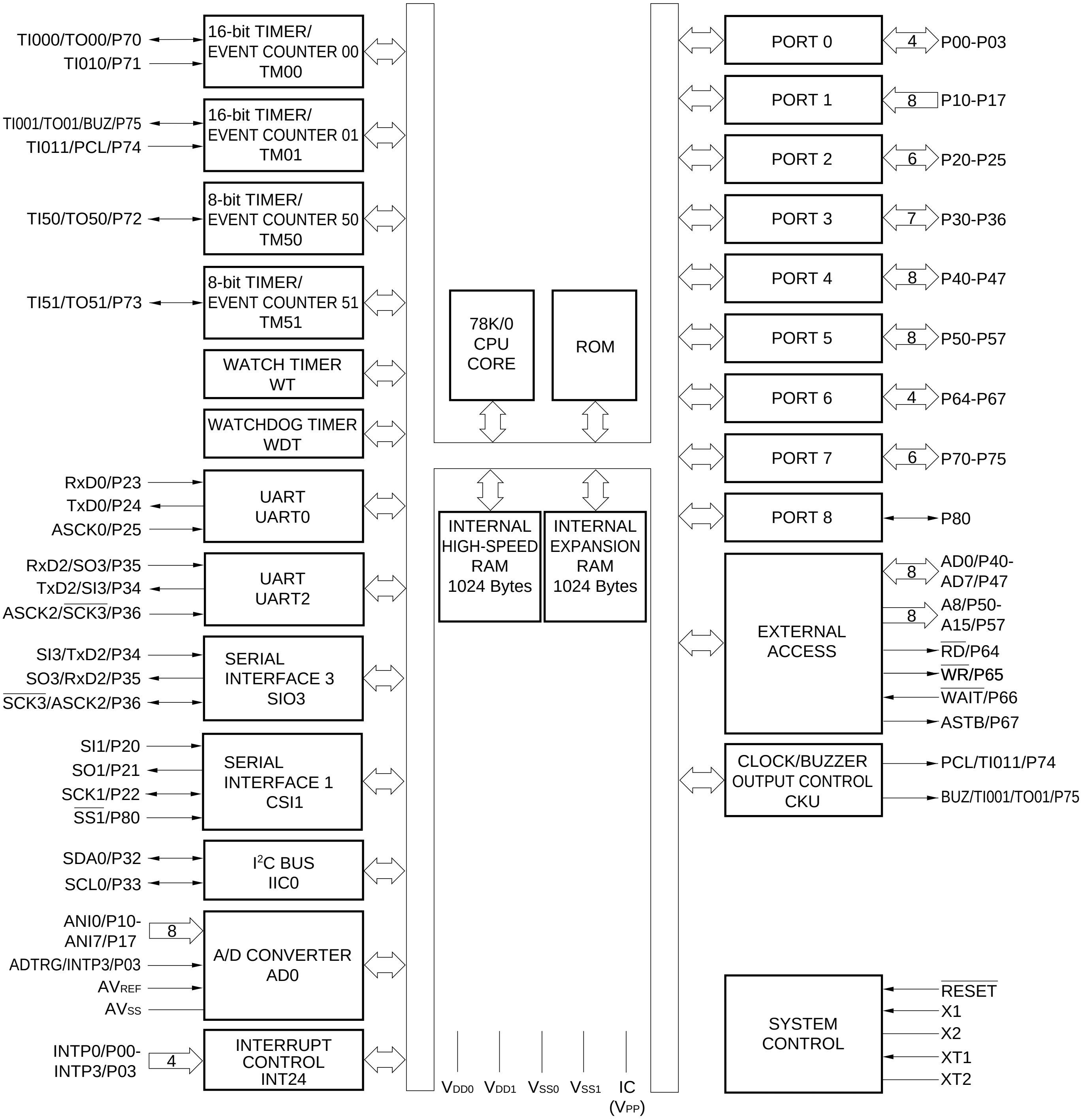
The major functional differences among the subseries are shown below.

- **Subseries with the suffix Y**

Function Subseries Name		ROM	Timer				8-bit	10-bit	8-bit	Serial Interface	I/O	V$_{DD}$	External
		Capacity	8-bit	16-bit	Watch	WDT	A/D	A/D	D/A			MIN. Value	Expansion
Control	μPD78078Y	48 K-60 K	4 ch	1 ch	1 ch	1 ch	8 ch	—	2 ch	3 ch (UART: 1 ch, I^{2}C: 1 ch)	88	1.8 V	○
	μPD78070AY	—									61	2.7 V	
	μPD780018AY	48 K-60 K							—	3 ch (I^{2}C: 1 ch)	88		
	μPD780058Y	24 K-60 K	2 ch						2 ch	3 ch (Time division UART: 1 ch, I^{2}C: 1 ch)	68	1.8 V	
	μPD78058FY	48 K-60 K								3 ch (UART: 1 ch, I^{2}C: 1 ch)	69	2.7 V	
	μPD78054Y	16 K-60 K										2.0 V	
	μPD780078Y	48 K-60 K		2 ch			—	8 ch	—	4 ch (UART: 2 ch, I^{2}C: 1 ch)	52	1.8 V	
	μPD780034AY	8 K-32 K		1 ch						3 ch (UART: 1 ch, I^{2}C: 1 ch)	51		
	μPD780024AY						8 ch	—					
	μPD78018FY	8 K-60 K								2 ch (I^{2}C: 1 ch)	53		
LCD drive	μPD780308Y	48 K-60 K	2 ch	1 ch	1 ch	1 ch	8 ch	—	—	3 ch (Time division UART: 1 ch, I^{2}C: 1 ch)	57	2.0 V	—
	μPD78064Y	16 K-32 K								2 ch (UART: 1 ch, I^{2}C: 1 ch)			
For bus interface	μPD780701Y	60 K	3 ch	2 ch	1 ch	1 ch	16 ch	—	—	4 ch (UART: 1 ch, I^{2}C: 1 ch)	67	3.5 V	—
	μPD780833Y										65	4.5 V	

Remark The functions of the subseries without the suffix Y and the subseries with the suffix Y are the same, except for the serial interface (if a subseries without the suffix Y is available).

2.6 Block Diagram



Remarks 1. The internal ROM and RAM capacities depend on the product.

2. Pin connection in parentheses is intended for the μPD78F0078Y.

2.7 Outline of Function

Item \ Part Number		μPD780076Y	μPD780078Y	μPD78F0078Y
Internal memory	ROM	48 Kbytes (Mask ROM)	60 Kbytes (Mask ROM)	60 Kbytes[Note 1] (Flash memory)
	High-speed RAM	1024 bytes		
	Expansion RAM	1024 bytes		
Memory space		64 Kbytes		
General-purpose register		8 bits × 32 registers (8 bits × 8 registers × 4 banks)		
Minimum instruction		Minimum instruction execution time changeable function		
execution time	When main system clock selected	0.24 μs/0.48 μs/0.95 μs/1.91 μs/3.81 μs (@ 8.38-MHz operation)		
	When subsystem clock selected	122 μs (@ 32.768-kHz operation)		
Instruction set		• 16-bit operation • Multiply/divide (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulate (set, reset, test, and Boolean operation) • BCD adjust, etc.		
I/O port		Total : 52 • CMOS input : 8 • CMOS I/O : 40 • N-ch open-drain I/O 5-V breakdown : 4		
A/D converter		• 10-bit resolution × 8 channels • Low-voltage operation: AV_{REF} = 2.7 to 5.5 V		
Serial interface		• 3-wire serial I/O mode : 1 channel • UART mode : 1 channel • 3-wire serial I/O/UART mode selectable[Note 2] : 1 channel • I^2C bus mode : 1 channel		
Timer		• 16-bit timer/event counter : 2 channels • 8-bit timer/event counter : 2 channels • Watch timer : 1 channel • Watchdog timer : 1 channel		
Timer output		4 outputs: (8-bit PWM output enable: 2)		
Clock output		• 65.5 kHz, 131 kHz, 262 kHz, 524 kHz, 1.05 MHz, 2.10 MHz, 4.19 MHz, 8.38 MHz (8.38 MHz with main system clock) • 32.768 kHz (32.768 kHz with subsystem clock)		
Buzzer output		1.02 kHz, 2.05 kHz, 4.10 kHz, 8.19 kHz (8.38 MHz with main system clock)		
Vectored interrupt	Maskable interrupt	Internal: 19, External: 5		
	Non-maskable interrupt	Internal: 1		
	Software interrupt	1		
Power supply voltage		V_{DD} = 1.8 to 5.5 V		V_{DD} = 2.7 to 5.5 V
Operating ambient temperature		T_A = –40 to +85°C		
Package		• 64-pin plastic QFP (14 × 14 mm) • 64-pin plastic TQFP (12 × 12 mm)		

Notes 1. The capacities of internal flash memory can be changed by means of the memory size switching register (IMS).

2. Select either of the functions of these multiplexed pins.

The following table outlines the timers/event counters (for details, refer to **CHAPTER 8 16-BIT TIMER/EVENT COUNTERS 00, 01; CHAPTER 9 8-BIT TIMER/EVENT COUNTERS 50, 51; CHAPTER 10 WATCH TIMER; and CHAPTER 11 WATCHDOG TIMER)**:

		16-Bit Timer/Event Counters 00, 01	8-Bit Timer/Event Counters 50, 51	Watch Timer	Watchdog Timer
Operation mode	Interval timer	2 channels	2 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 be used both as a watch timer and an interval timer at the same time.

2. The watchdog timer can be used as either a watchdog timer or interval timer. Select one of the functions.

2.8 Mask Options

The mask ROM versions (μPD780076Y and 780078Y) provide pull-up resistor mask options which allow users to specify whether to connect a pull-up resistor to a specific port pin when the user places an order for device production. Using this mask option when pull-up resistors are required reduces the number of components to add to the device, resulting in board space saving.

The mask options provided in the μPD780078Y Subseries are shown in Table 2-1.

Table 2-1. Mask Options of Mask ROM Versions

Pin Names	Mask Option
P30-P33	Pull-up resistor connection can be specified in 1-bit units.

[MEMO]

3.1 Pin Function List

(1) Port Pins (1/2)

Pin Name	Input/Output	Function		At Reset	Alternate Function
P00	Input/Output	Port 0 4-bit input/output port Input/output mode can be specified bit-wise. An on-chip pull-up resistor can be used by software.		Input	INTP0
P01					INTP1
P02					INTP2
P03					INTP3/ADTRG
P10 to P17	Input	Port 1 8-bit input only port.		Input	ANI0 to ANI7
P20	Input/Output	Port 2 6-bit input/output port Input/output mode can be specified bit-wise. An on-chip pull-up resistor can be used by software.		Input	SI1
P21					SO1
P22					SCK1
P23					RxD0
P24					TxD0
P25					ASCK0
P30	Input/Output	Port 3 7-bit input/output port Input/output mode can be specified bit-wise.	N-ch open-drain input/output port On-chip pull-up resistor can be specified by mask option. (Mask ROM version only) LEDs can be driven directly.	Input	—
P31					
P32					
P33					
P34			An on-chip pull-up resistor can be used by software.		SI3/TxD2
P35					SO3/RxD2
P36					$\overline{\text{SCK3}}$/ASCK2
P40 to P47	Input/Output	Port 4 8-bit input/output port Input/output mode can be specified bit-wise. An on-chip pull-up resistor can be used by software. Interrupt request flag (KRIF) is set to 1 by falling edge detection.		Input	AD0 to AD7
P50 to P57	Input/Output	Port 5 8-bit input/output port LEDs can be driven directly. Input/output mode can be specified bit-wise. An on-chip pull-up resistor can be used by software.		Input	A8 to A15
P64	Input/Output	Port 6 4-bit input/output port Input/output mode can be specified bit-wise. An on-chip pull-up resistor can be used by software.		Input	$\overline{\text{RD}}$
P65					$\overline{\text{WR}}$
P66					$\overline{\text{WAIT}}$
P67					ASTB

(1) Port Pins (2/2)

Pin Name	Input/Output	Function	At Reset	Alternate Function
P70	Input/Output	Port 7 6-bit input/output port Input/output mode can be specified bit-wise. An on-chip pull-up resistor can be used by software.	Input	TI000/TO00
P71				TI010
P72				TI50/TO50
P73				TI51/TO51
P74				TI011/PCL
P75				TI001/TO01/BUZ
P80		Port 8 1-bit input/output port Input/output mode can be specified bit-wise. An on-chip pull-up resistor can be used by software.		$\overline{\text{SS1}}$

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

Pin Name	Input/Output	Function	At Reset	Alternate Function
INTP0	Input	External interrupt request input with specifiable valid edges (rising edge, falling edge, both rising and falling edges)	Input	P00
INTP1				P01
INTP2				P02
INTP3				P03/ADTRG
SI1	Input	Serial interface serial data input	Input	P20
SI3				P34/TxD2
SO1	Output	Serial interface serial data output	Input	P21
SO3				P35/RxD2
SCK1	Input/Output	Serial interface serial clock input/output	Input	P22
$\overline{\text{SCK3}}$				P36/ASCK2
$\overline{\text{SS1}}$	Input	Serial interface chip select input	Input	P80
RxD0	Input	Asynchronous serial interface serial data input	Input	P23
RxD2				P35/SO3
TxD0	Output	Asynchronous serial interface serial data output	Input	P24
TxD2				P34/SI3
ASCK0	Input	Asynchronous serial interface serial clock input	Input	P25
SCK2				P36/$\overline{\text{SCK3}}$
TI000	Input	External count clock input to 16-bit timer/event counter 00. Capture trigger input to capture register (CR000, CR010) of 16-bit timer/event counter 00	Input	P70/TO00
TI010		Capture trigger input to capture register (CR000) of 16-bit timer/event counter 00		P71
TI001		External count clock input to 16-bit timer/event counter 01. Capture trigger input to capture register (CR001, CR011) of 16-bit timer/event counter 01		P75/TO01/BUZ

(2) Non-port Pins (2/2)

Pin Name	Input/Output	Function	After Reset	Alternate Function
TI011	Input	Capture trigger input to capture register (CR001) of 16-bit timer/event counter 01	Input	P74/PCL
TI50		External count clock input to 8-bit timer/event counter 50		P72/TO50
TI51		External count clock input to 8-bit timer/event counter 51		P73/TO51
TO00	Output	16-bit timer/event counter 00 output	Input	P70/TI000
TO01		16-bit timer/event counter 01 output		P75/TI001/BUZ
TO50		8-bit timer/event counter 50 output	Input	P72/TI50
TO51		8-bit timer/event counter 51 output		P73/TI51
PCL	Output	Clock output (for main system clock and subsystem clock trimming)	Input	P74/TI011
BUZ	Output	Buzzer output	Input	P75/TI001/TO01
AD0 to AD7	Input/Output	Lower-order address/data bus when expanding external memory	Input	P40 to P47
A8 to A15	Output	High-order address bus when expanding external memory	Input	P50 to P57
$\overline{RD}$	Output	Strobe signal output for read operation from external memory	Input	P64
$\overline{WR}$		Strobe signal output for write operation from external memory		P65
$\overline{WAIT}$	Input	Wait insertion when accessing external memory	Input	P66
ASTB	Output	Strobe output externally latching address information output to ports 4, 5 to access external memory	Input	P67
ANI0 to ANI7	Input	A/D converter analog input	Input	P10 to P17
ADTRG	Input	A/D converter trigger signal input	Input	P03/INTP3
AV_{REF}	Input	A/D converter reference voltage input and analog power supply	—	—
AV_{SS}	—	A/D converter ground potential. Connect to V_{SS0} or V_{SS1}	—	—
$\overline{RESET}$	Input	System reset input	Input	—
X1	Input	Crystal connection for main system clock oscillation	—	—
X2	—		—	—
XT1	Input	Crystal connection for subsystem clock oscillation	—	—
XT2	—		—	—
V_{DD0}	—	Positive power supply	—	—
V_{DD1}	—	Positive power supply other than port	—	—
V_{SS0}	—	Ground potential	—	—
V_{SS1}	—	Ground potential other than port	—	—
IC	—	Internally connected. Connect directly to V_{SS0} or V_{SS1}	—	—
V_{PP}	—	High-voltage application for program write/verify. Connect directly to V_{SS0} or V_{SS1} in normal operating mode.	—	—

3.2 Description of Pin Functions

3.2.1 P00 to P03 (Port 0)

These are 4-bit input/output ports. Besides serving as input/output ports, they function as an external interrupt input, and A/D converter external trigger input.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 4-bits input/output ports.

P00 to P03 can be specified for input or output ports bit-wise with a port mode register 0 (PM0). On-chip pull-up resistors can be used for them by defining the pull-up resistor option register 0 (PU0).

(2) Control mode

In this mode, these ports function as an external interrupt request input, and A/D converter external trigger input.

(a) INTP0 to INTP3

INTP0 to INTP3 are external interrupt request input pins which can specify valid edges (rising edge, falling edge, and both rising and falling edges).

(b) ADTRG

A/D converter external trigger input.

Caution **When P03 is used as an A/D converter external trigger input, specify the valid edge by bits 1, 2 (EGA00, EGA01) of A/D converter mode register (ADM0) and set interrupt mask flag (PMK3) to 1.**

3.2.2 P10 to P17 (Port 1)

These are 8-bit input only ports. Besides serving as input ports, they function as an A/D converter analog input.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 8-bit input only ports.

(2) Control mode

These ports function as A/D converter analog input pins (ANI0 to ANI7).

3.2.3 P20 to P25 (Port 2)

These are 6-bit input/output ports. Besides serving as input/output ports, they function as data input/output to/from the serial interface and clock input/output.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 6-bit input/output ports. They can be specified bit-wise as input or output ports with port mode register 2 (PM2). On-chip pull-up resistors can be used for them by setting pull-up resistor option register 2 (PU2).

(2) Control mode

These ports function as serial interface data input/output and clock input/output functions.

(a) SI1 and SO1

Serial interface serial data input/output pins.

(b) SCK1

Serial interface serial clock input/output pin.

(c) RxD0, TxD0

Asynchronous serial interface serial data input/output pins.

(d) ASCK0

Asynchronous serial interface serial clock input pin.

3.2.4 P30 to P36 (Port 3)

These are 7-bit input/output ports. Beside serving as input/output ports, they function as serial interface data input/output and clock input/output.

P30 to P33 can drive LEDs directly.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 7-bit input/output ports. They can be specified bit-wise as input or output ports with port mode register 3 (PM3). P30 to P33 are N-ch open drain input/output port. On-chip pull-up resistor can be used by mask option. (Mask ROM version only) On-chip pull-up resistors of P34 to P36 can be used by defining the pull-up resistor option register 3 (PU3).

(2) Control mode

These ports function as serial interface data input/output and clock input/output.

(a) SI3 and SO3

Serial interface serial data input/output pins.

(b) $\overline{\text{SCK3}}$

Serial interface serial clock input/output pin.

(c) RxD2 and TxD2

Asynchronous serial interface serial data input/output pins.

(d) ASCK2

Asynchronous serial interface serial clock input pin.

3.2.5 P40 to P47 (Port 4)

These are 8-bit input/output ports. Besides serving as input/output ports, they function as an address/data bus.

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

The following operating mode can be specified bit-wise.

Caution When using the falling edge detection interrupt (INTKR), be sure to set the memory expansion mode register (MEM) to 01H.

(1) Port mode

These ports function as 8-bit input/output ports. They can be specified in 1-bit units for input or output ports by using port mode register 4 (PM4). On-chip pull-up resistors can be used by setting pull-up resistor option register 4 (PU4).

(2) Control mode

These ports function as low-order address/data bus pins (AD0 to AD7) in external memory expansion mode.

3.2.6 P50 to P57 (Port 5)

These are 8-bit input/output ports. Besides serving as input/output ports, they function as an address bus.

Port 5 can drive LEDs directly.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 8-bit input/output ports. They can be specified bit-wise as input/output ports with port mode register 5 (PM5). On-chip pull-up resistors can be used by setting pull-up resistor option register 5 (PU5).

(2) Control mode

These ports function as high-order address bus pins (A8 to A15) in external memory expansion mode.

3.2.7 P64 to P67 (Port 6)

These are 4-bit input/output ports. Besides serving as input/output ports, they are used for control in external memory expansion mode.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 4-bit input/output ports. They can be specified bit-wise as input or output ports with port mode register 6 (PM6).

On-chip pull-up resistors can be used by setting pull-up resistor option register 6 (PU6).

(2) Control mode

These ports function as control signal output pins ($\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WAIT}}$, ASTB) in external memory expansion mode.

Caution When external wait is not used in external memory expansion mode, P66 can be used as an input/output port.

3.2.8 P70 to P75 (Port 7)

These are 6-bit input/output ports. Besides serving as input/output ports, they function as a timer input/output, clock output, and buzzer output.

The following operating modes can be specified bit-wise.

(1) Port mode

Port 7 functions as a 6-bit input/output port. Bit-wise specification as an input port or output port is possible by means of port mode register 7 (PM7). On-chip pull-up resistors can be used by defining the pull-up resistor option register 7 (PU7). P70 and P71 are also 16-bit timer/event counter capture trigger signal input pins with a valid edge input.

(2) Control mode

Port 7 functions as timer input/output, clock output and buzzer output.

(a) TI000

External count clock input pin to 16-bit timer/event counter 00 and capture trigger signal input pin to 16-bit timer/event counter 00 capture register (CR000, CR010).

(b) TI001

External count clock input pin to 16-bit timer/event counter 01 and capture trigger signal input pin to 16-bit timer/event counter 01 capture register (CR001, CR011).

(c) TI010

Capture trigger signal input pin to 16-bit timer/event counter 00 capture register (CR000).

(d) TI011

Capture trigger signal input pin to 16-bit timer/event counter 01 capture register (CR001).

(e) TI50 and TI51

8-bit timer/event counter external count clock input pins.

(f) TO00, TO01, TO50, and TO51

Timer output pins.

(g) PCL

Clock output pin.

(h) BUZ

Buzzer output pin.

3.2.9 P80 (Port 8)

This is a 1-bit input/output port. Besides serving as input/output ports, it functions as chip select input of serial interface. The following operating modes can be specified bit-wise.

(1) Port mode

This port functions as a 1-bit input/output port. It can be specified bit-wise as input or output port with port mode register 8 (PM8).

On-chip pull-up resistors can be used by setting pull-up resistor option register 8 (PU8).

(2) Control mode

This port functions as chip select input ($\overline{\text{SS1}}$) of serial interface.

3.2.10 AV$_{\text{REF}}$

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

When no A/D converter is used, connect this pin to V$_{\text{SS0}}$.

3.2.11 AV$_{\text{SS}}$

This is a ground voltage pin of A/D converter. Use the same voltage as that of the V$_{\text{SS0}}$ pin or V$_{\text{SS1}}$ pin even when no A/D converter is used.

3.2.12 $\overline{\text{RESET}}$

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

3.2.13 X1 and X2

Crystal resonator connect pins for main system clock oscillation. For external clock supply, input the clock signal to X1 and its inverted signal to X2.

3.2.14 XT1 and XT2

Crystal resonator connect pins for subsystem clock oscillation.

For external clock supply, input the clock signal to XT1 and its inverted signal to XT2.

3.2.15 V$_{\text{DD0}}$ and V$_{\text{DD1}}$

V$_{\text{DD0}}$ is a positive power supply port pin.

V$_{\text{DD1}}$ is a positive power supply pin other than port pin.

3.2.16 V$_{\text{SS0}}$ and V$_{\text{SS1}}$

V$_{\text{SS0}}$ is a ground potential port pin.

V$_{\text{SS1}}$ is a ground potential pin other than port pin.

3.2.17 V$_{\text{PP}}$ (flash memory versions only)

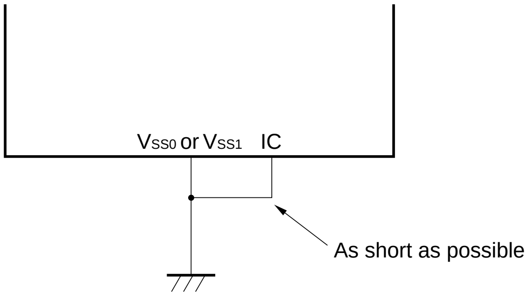
High-voltage apply pin for flash memory programming mode setting and program write/verify. Connect directly to V$_{\text{SS0}}$ or V$_{\text{SS1}}$ in the normal operating mode.

3.2.18 IC (mask ROM version only)

The IC (Internally Connected) pin is provided to set the test mode to check the μPD780078 Subseries at delivery. Connect it directly to the V_{SS0} or V_{SS1} with the shortest possible wire in the normal operating 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 pins to V_{SS0} pins or V_{SS1} pins directly.**



3.3 Pin Input/output Circuits and Recommended Connection of Unused Pins

Table 3-1 shows the types of pin input/output circuit and the recommended connections of unused pins. Refer to Figure 3-1 for the configuration of the input/output circuit of each type.

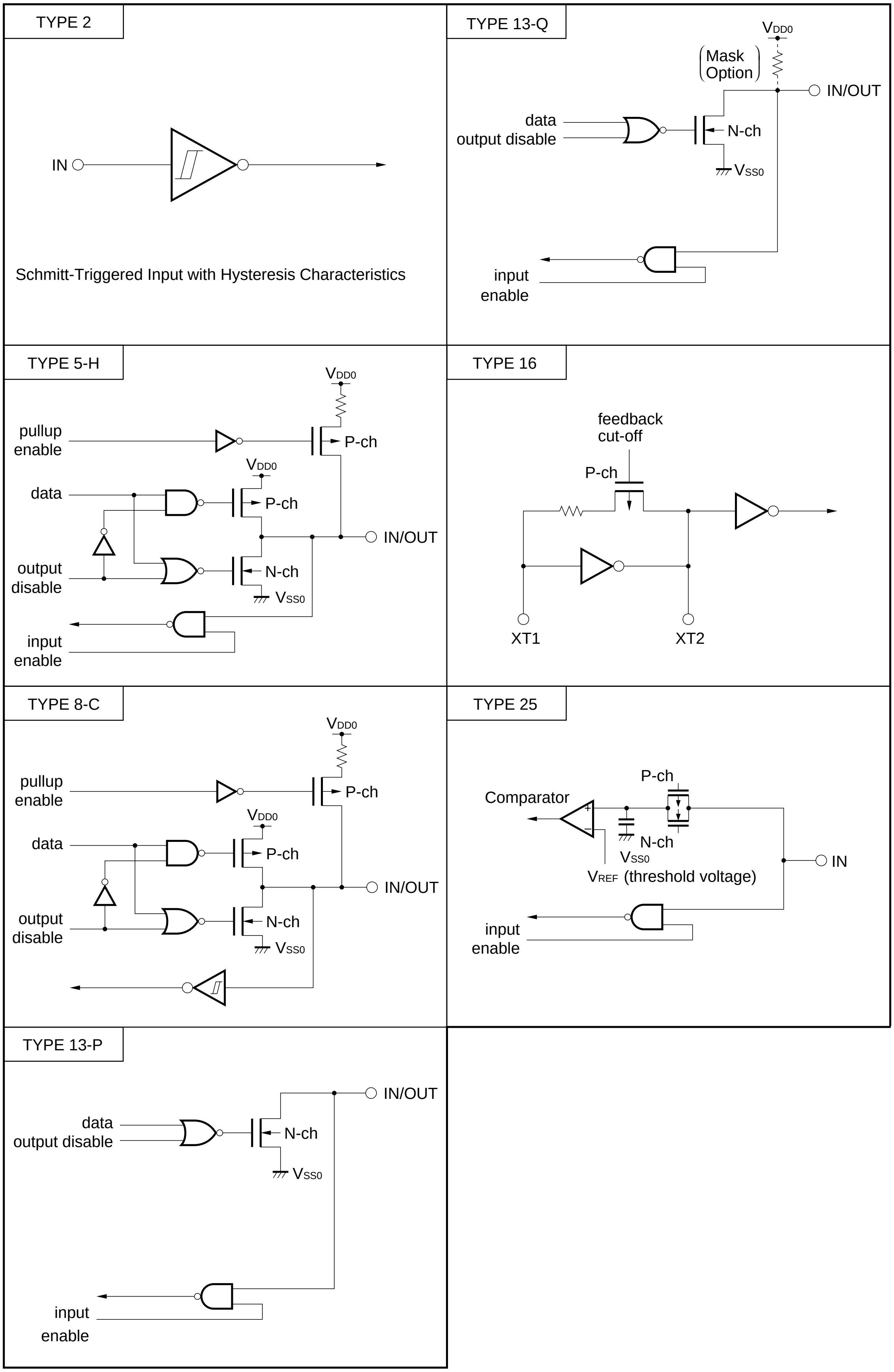
Table 3-1. Pin Input/Output Circuit Types (1/2)

Pin Name	Input/Output Circuit Type	Input/Output	Recommended Connection of Unused Pins
P00/INTP0 to P02/INTP2	8-C	Input/output	Input: Independently connect to V_{SS0} via a resistor. Output: Leave unconnected.
P03/INTP3/ADTRG			
P10/ANI0 to P17/ANI7	25	Input	Connect to V_{DD0} or V_{SS0}.
P20/SI1	8-C	Input/output	Input: Independently connect to V_{DD0} or V_{SS0} via a resistor. Output: Leave unconnected.
P21/SO1	5-H		
P22/SCK1	8-C		
P23/RxD0			
P24/TxD0	5-H		
P25/ASCK0	8-C		
P30-P33 (for mask ROM version)	13-Q	Input/output	Input: Independently connect to V_{DD0} via a resistor. Output: Leave unconnected.
P30-P33 (for flash memory version)	13-P		
P34/SI3/TxD2	8-C		Input: Independently connect to V_{DD0} or V_{SS0} via a resistor. Output: Leave unconnected.
P35/SO3/RxD2			
P36/$\overline{\text{SCK3}}$/ASCK2			
P40/AD0 to P47/AD7	5-H	Input/output	Input: Independently connect to V_{DD0} via a resistor. Output: Leave unconnected.

Table 3-1. Pin Input/Output Circuit Types (2/2)

Pin Name	Input/Output Circuit Type	Input/Output	Recommended Connection of Unused Pins
P50/A8 to P57/A15	5-H	Input/output	Input: Independently connect to V_{DD0} or V_{SS0} via a resistor. Output: Leave unconnected.
P64/$\overline{\text{RD}}$		Input/output	
P65/$\overline{\text{WR}}$			
P66/$\overline{\text{WAIT}}$			
P67/ASTB			
P70/TI000/TO00	8-C		
P71/TI010			
P72/TI50/TO50			
P73/TI51/TO51			
P74/TI011/PCL			
P75/TI001/TO01/BUZ			
P80/$\overline{\text{SS1}}$			Connect to V_{SS0} via a resistor.
$\overline{\text{RESET}}$	2	Input	—
XT1	16		Connect to V_{DD0}.
XT2		—	Leave unconnected.
AV_{REF}	—		Connect to V_{SS0}.
AV_{SS}			Connect to V_{SS0} or V_{SS1}.
IC (for mask ROM version)			Connect directly to V_{SS0} or V_{SS1}.
V_{PP} (for flash memory version)			

Figure 3-1. Pin Input/Output Circuit List



4.1 Pin Function List

(1) Port Pins (1/2)

Pin Name	Input/Output	Function		At Reset	Alternate Function
P00	Input/Output	Port 0 4-bit input/output port Input/output mode can be specified bit-wise. An on-chip pull-up resistor can be used by software.		Input	INTP0
P01					INTP1
P02					INTP2
P03					INTP3/ADTRG
P10 to P17	Input	Port 1 8-bit input only port.		Input	ANI0 to ANI7
P20	Input/Output	Port 2 6-bit input/output port Input/output mode can be specified bit-wise. An on-chip pull-up resistor can be used by software.		Input	SI1
P21					SO1
P22					SCK1
P23					RxD0
P24					TxD0
P25					ASCK0
P30	Input/Output	Port 3 7-bit input/output port Input/output mode can be specified bit-wise.	N-ch open-drain input/output port On-chip pull-up resistor can be specified by mask option. (P30 and P31 are Mask ROM version only.) LEDs can be driven directly.	Input	—
P31					
P32					SDA0
P33					SCL0
P34			An on-chip pull-up resistor can be used by software.		SI3/TxD2
P35					SO3/RxD2
P36					$\overline{\text{SCK3}}$/ASCK2
P40 to P47	Input/Output	Port 4 8-bit input/output port Input/output mode can be specified bit-wise. An on-chip pull-up resistor can be used by software. Interrupt request flag (KRIF) is set to 1 by falling edge detection.		Input	AD0 to AD7
P50 to P57	Input/Output	Port 5 8-bit input/output port LEDs can be driven directly. Input/output mode can be specified bit-wise. An on-chip pull-up resistor can be used by software.		Input	A8 to A15
P64	Input/Output	Port 6 4-bit input/output port Input/output mode can be specified bit-wise. An on-chip pull-up resistor can be used by software.		Input	$\overline{\text{RD}}$
P65					$\overline{\text{WR}}$
P66					$\overline{\text{WAIT}}$
P67					ASTB

(1) Port Pins (2/2)

Pin Name	Input/Output	Function	At Reset	Alternate Function
P70	Input/Output	Port 7 6-bit input/output port Input/output mode can be specified bit-wise. An on-chip pull-up resistor can be used by software.	Input	TI000/TO00
P71				TI010
P72				TI50/TO50
P73				TI51/TO51
P74				TI011/PCL
P75				TI001/TO01/BUZ
P80		Port 8 1-bit input/output port Input/output mode can be specified bit-wise. An on-chip pull-up resistor can be used by software.		$\overline{\text{SS1}}$

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

Pin Name	Input/Output	Function	At Reset	Alternate Function
INTP0	Input	External interrupt request input with specifiable valid edges (rising edge, falling edge, both rising and falling edges)	Input	P00
INTP1				P01
INTP2				P02
INTP3				P03/ADTRG
SI3	Input	Serial interface serial data input	Input	P20
SO1	Output	Serial interface serial data output	Input	P21
SDA0	Input/Output	Serial interface serial data input/output	Input	P32
SCK1	Input/Output	Serial interface serial clock input/output	Input	P22
$\overline{\text{SCK3}}$				P30/ASCK2
SCL0				P33
$\overline{\text{SS1}}$	Input	Serial interface chip select input	Input	P80
RxD0	Input	Asynchronous serial interface serial data input	Input	P23
RxD2				P35/SO3
TxD0	Output	Asynchronous serial interface serial data output	Input	P24
TxD2				P34/SI3
ASCK0	Input	Asynchronous serial interface serial clock input	Input	P25
ASCK2				P36/$\overline{\text{SCK3}}$
TI000	Input	External count clock input to 16-bit timer/event counter 00. Capture trigger input to capture register (CR000, CR010) of 16-bit timer/event counter 00	Input	P70/TO00
TI010		Capture trigger input to capture register (CR000) of 16-bit timer/event counter 00		P71
TI001		External count clock input to 16-bit timer/event counter 01. Capture trigger input to capture register (CR001, CR011) of 16-bit timer/event counter 01		P75/TO01/BUZ

(2) Non-port Pins (2/2)

Pin Name	Input/Output	Function	After Reset	Alternate Function
TI011	Input	Capture trigger input to capture register (CR001) of 16-bit timer/event counter 01	Input	P74/PCL
TI50		External count clock input to 8-bit timer/event counter 50		P72/TO50
TI51		External count clock input to 8-bit timer/event counter 51		P73/TO51
TO00	Output	16-bit timer/event counter 00 output	Input	P70/TI00
TO01		16-bit timer/event counter 01 output		P75/TI001/BUZ
TO50		8-bit timer/event counter 50 output	Input	P72/TI50
TO51		8-bit timer/event counter 51 output		P73/TI51
PCL	Output	Clock output (for main system clock and subsystem clock trimming)	Input	P74/TI011
BUZ	Output	Buzzer output	Input	P75/TI001/TO01
AD0 to AD7	Input/Output	Lower-order address/data bus when expanding external memory	Input	P40 to P47
A8 to A15	Output	High-order address bus when expanding external memory	Input	P50 to P57
$\overline{\text{RD}}$	Output	Strobe signal output for read operation from external memory	Input	P64
$\overline{\text{WR}}$		Strobe signal output for write operation from external memory		P65
$\overline{\text{WAIT}}$	Input	Wait insertion when accessing external memory	Input	P66
ASTB	Output	Strobe output externally latching address information output to ports 4, 5 to access external memory	Input	P67
ANI0 to ANI7	Input	A/D converter analog input	Input	P10 to P17
ADTRG	Input	A/D converter trigger signal input	Input	P03/INTP3
AV$_{REF}$	Input	A/D converter reference voltage input and analog power supply	—	—
AV$_{SS}$	—	A/D converter ground potential. Connect to V$_{SS0}$ or V$_{SS1}$	—	—
$\overline{\text{RESET}}$	Input	System reset input	Input	—
X1	Input	Crystal connection for main system clock oscillation	—	—
X2	—		—	—
XT1	Input	Crystal connection for subsystem clock oscillation	—	—
XT2	—		—	—
V$_{DD0}$	—	Positive power supply	—	—
V$_{DD1}$	—	Positive power supply other than port	—	—
V$_{SS0}$	—	Ground potential	—	—
V$_{SS1}$	—	Ground potential other than port	—	—
IC	—	Internally connected. Connect directly to V$_{SS0}$ or V$_{SS1}$	—	—
V$_{PP}$	—	High-voltage application for program write/verify. Connect directly to V$_{SS0}$ or V$_{SS1}$ in normal operating mode.	—	—

4.2 Description of Pin Functions

4.2.1 P00 to P03 (Port 0)

These are 4-bit input/output ports. Besides serving as input/output ports, they function as an external interrupt input, and A/D converter external trigger input pins.

The following operating modes can be specified bit-wise.

(1) Port mode

In this mode, these ports function as 4-bit input/output ports.

P00 to P03 can be specified for input or output ports bit-wise with a port mode register 0 (PM0). On-chip pull-up resistors can be used for them by setting pull-up resistor option register 0 (PU0).

(2) Control mode

In this mode, these ports function as an external interrupt request input, and A/D converter external trigger input pins.

(a) INTP0 to INTP3

INTP0 to INTP3 are external interrupt request input pins which can specify valid edges (rising edge, falling edge, and both rising and falling edges).

(b) ADTRG

A/D converter external trigger input pin.

Caution **When P03 is used as an A/D converter external trigger input, specify the valid edge by bits 1, 2 (EGA00, EGA01) of A/D converter mode register 0 (ADM0) and set interrupt mask flag (PMK3) to 1.**

4.2.2 P10 to P17 (Port 1)

These are 8-bit input only ports. Besides serving as input ports, they function as an A/D converter analog input.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 8-bit input only ports.

(2) Control mode

These ports function as A/D converter analog input pins (ANI0 to ANI7).

4.2.3 P20 to P25 (Port 2)

These are 6-bit input/output ports. Besides serving as input/output ports, they function as data input/output to/from the serial interface and clock input/output functions.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 6-bit input/output ports. They can be specified bit-wise as input or output ports with port mode register 2 (PM2). On-chip pull-up resistors can be used for them by setting pull-up resistor option register 2 (PU2).

(2) Control mode

These ports function as serial interface data input/output and clock input/output.

(a) SI1 and SO1

Serial interface serial data input/output pins.

(b) SCK1

Serial interface serial clock input/output pin.

(c) RxD0, TxD0

Asynchronous serial interface serial data input/output pins.

(d) ASCK0

Asynchronous serial interface serial clock input pin.

4.2.4 P30 to P36 (Port 3)

These are 7-bit input/output ports. Beside serving as input/output ports, they function as serial interface data input/output and clock input/output.

P30 to P33 (Port 3) can drive LEDs directly.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 7-bit input/output ports. They can be specified bit-wise as input or output ports with port mode register 3 (PM3). P30 to P33 are N-ch open drain output. Mask ROM version can contain pull-up resistors in P30 and P31 with the mask option. On-chip pull-up resistors of P34 to P36 can be used by defining the pull-up resistor option register 3 (PU3).

(2) Control mode

These ports function as serial interface serial data input/output and clock input/output.

(a) SI3, SO3, and SDA0

Serial interface serial data input/output pins.

(b) $\overline{\text{SCK3}}$ and SCL0

Serial interface serial clock input/output pins.

(c) RxD2 and TxD2

Asynchronous serial interface serial data input/output pins.

(d) ASCK2

Asynchronous serial interface serial clock input pin.

4.2.5 P40 to P47 (Port 4)

These are 8-bit input/output ports. Besides serving as input/output ports, they function as an address/data bus.

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

The following operating mode can be specified bit-wise.

Caution When using the falling edge detection interrupt (INTKR), be sure to set the memory expansion mode register (MEM) to 01H.

(1) Port mode

These ports function as 8-bit input/output ports. They can be specified bit-wise for input or output ports by using the port mode register 4 (PM4). On-chip pull-up resistors can be used by setting pull-up resistor option register 4 (PU4).

(2) Control mode

These ports function as low-order address/data bus pins (AD0 to AD7) in external memory expansion mode.

4.2.6 P50 to P57 (Port 5)

These are 8-bit input/output ports. Besides serving as input/output ports, they function as an address bus.

Port 5 can drive LEDs directly.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 8-bit input/output ports. They can be specified bit-wise as input/output ports with port mode register 5 (PM5). On-chip pull-up resistors can be used by setting pull-up resistor option register 5 (PU5).

(2) Control mode

These ports function as high-order address bus pins (A8 to A15) in external memory expansion mode.

4.2.7 P64 to P67 (Port 6)

These are 4-bit input/output ports. Besides serving as input/output ports, they are used for control in external memory expansion mode.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 4-bit input/output ports. They can be specified bit-wise as input or output ports with port mode register 6 (PM6).

On-chip pull-up resistors can be used by setting pull-up resistor option register 6 (PU6).

(2) Control mode

These ports function as control signal output pins ($\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WAIT}}$, $\overline{\text{ASTB}}$) in external memory expansion mode.

Caution When external wait is not used in external memory expansion mode, P66 can be used as an input/output port.

4.2.8 P70 to P75 (Port 7)

These are 6-bit input/output ports. Besides serving as input/output ports, they function as a timer input/output, clock output, and buzzer output.

The following operating modes can be specified bit-wise.

(1) Port mode

Port 7 functions as a 6-bit input/output port. Bit-wise specification as an input port or output port is possible by means of port mode register 7 (PM7). On-chip pull-up resistors can be used by setting pull-up resistor option register 7 (PU7). P70 and P71 also become a 16-bit timer/event counter capture trigger signal input pin with a valid edge input.

(2) Control mode

Port 7 functions as timer input/output, clock output and buzzer output.

(a) TI000

External count clock input pin to 16-bit timer/event counter 00 and capture trigger signal input pin to 16-bit timer/event counter 00 capture register (CR000, CR010).

(b) TI001

External count clock input pin to 16-bit timer/event counter 01 and capture trigger signal input pin to 16-bit timer/event counter 01 capture register (CR001, CR011).

(c) TI010

Capture trigger signal input pin to 16-bit timer/event counter 00 capture register (CR000).

(d) TI011

Capture trigger signal input pin to 16-bit timer/event counter 01 capture register (CR001).

(e) TI50, TI51

8-bit timer/event counter external count clock input pins.

(f) TO00, TO01, TO50, TO51

Timer output pins.

(g) PCL

Clock output pin.

(h) BUZ

Buzzer output pin.

4.2.9 P80 (Port 8)

This is a 1-bit input/output port. Besides serving as input/output ports, it functions as chip select input of serial interface. The following operating modes can be specified bit-wise.

(1) Port mode

This port functions as a 1-bit input/output port. It can be specified bit-wise as input or output port with port mode register 8 (PM8).

On-chip pull-up resistors can be used by setting pull-up resistor option register 8 (PU8).

(2) Control mode

This port functions as chip select input ($\overline{\text{SS1}}$) of serial interface.

4.2.10 AV$_{\text{REF}}$

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

When no A/D converter is used, connect this pin to V$_{\text{SS0}}$.

4.2.11 AV$_{\text{SS}}$

This is a ground voltage pin of A/D converter. Use the same voltage as that of the V$_{\text{SS0}}$ pin or V$_{\text{SS1}}$ pin even when no A/D converter is used.

4.2.12 $\overline{\text{RESET}}$

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

4.2.13 X1 and X2

Crystal resonator connect pins for main system clock oscillation. For external clock supply, input the clock signal to X1 and its inverted signal to X2.

4.2.14 XT1 and XT2

Crystal resonator connect pins for subsystem clock oscillation.

For external clock supply, input the clock signal to XT1 and its inverted signal to XT2.

4.2.15 V$_{\text{DD0}}$ and V$_{\text{DD1}}$

V$_{\text{DD0}}$ is a positive power supply pin.

V$_{\text{DD1}}$ is a positive power supply pin other than port pin.

4.2.16 V$_{\text{SS0}}$ and V$_{\text{SS1}}$

V$_{\text{SS0}}$ is a ground potential port pin.

V$_{\text{SS1}}$ is a ground potential pin other than port pin.

4.2.17 V$_{\text{PP}}$ (flash memory versions only)

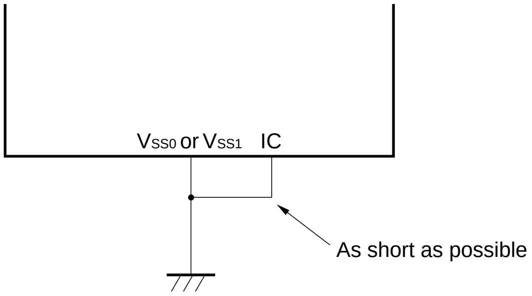
High-voltage apply pin for flash memory programming mode setting and program write/verify. Connect directly to V$_{\text{SS0}}$ or V$_{\text{SS1}}$ in normal operating mode.

4.2.18 IC (Mask ROM version only)

The IC (Internally Connected) pin is provided to set the test mode to check the μPD780078Y Subseries at delivery. Connect it directly to the V_{SS0} or V_{SS1} with the shortest possible wire in the normal operating 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 pins to V_{SS0} pins or V_{SS1} pins directly.**



4.3 Pin Input/output Circuits and Recommended Connection of Unused Pins

Table 4-1 shows the types of pin input/output circuit and the recommended connections of unused pins. Refer to Figure 4-1 for the configuration of the input/output circuit of each type.

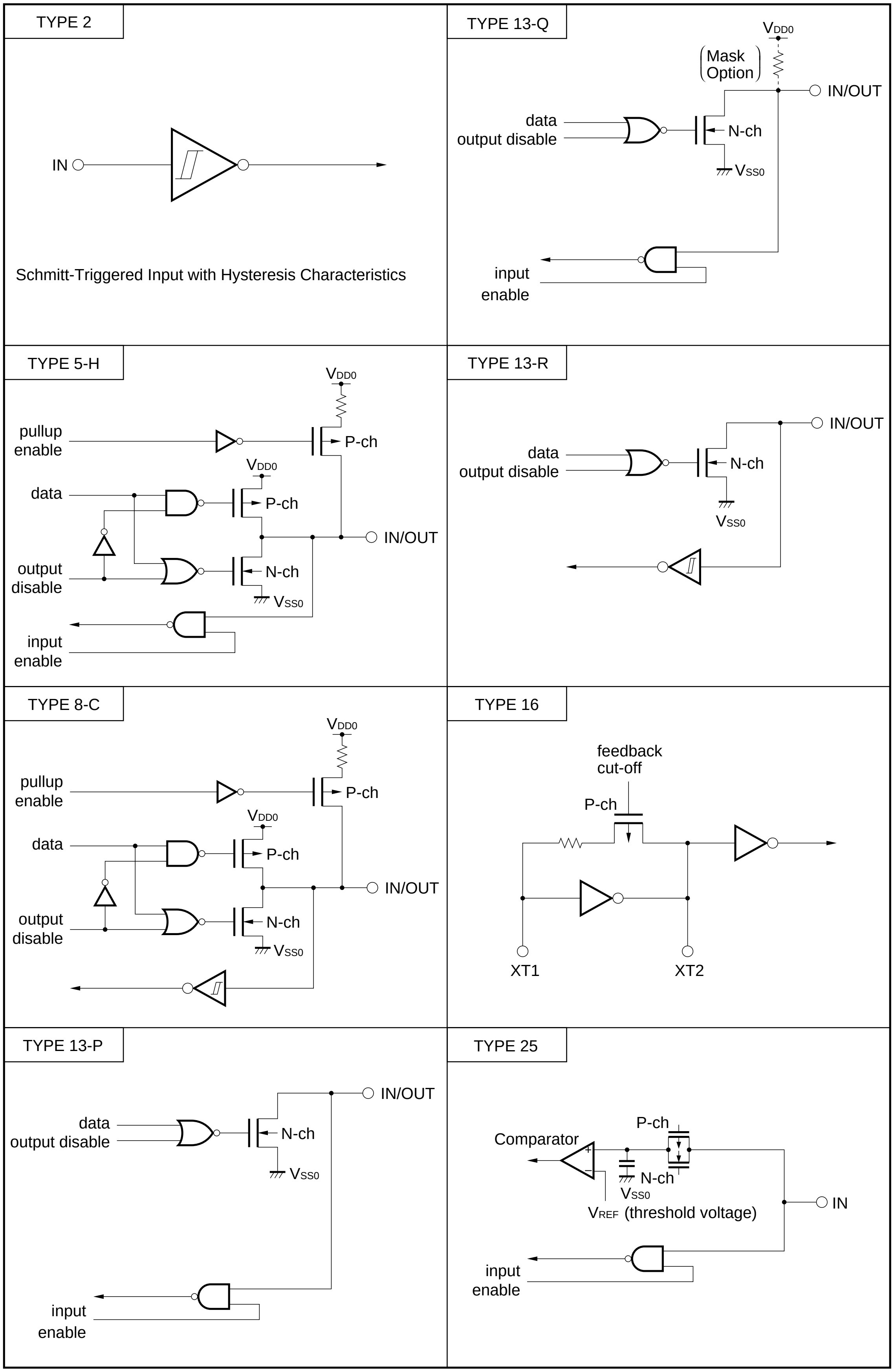
Table 4-1. Pin Input/Output Circuit Types (1/2)

Pin Name	Input/Output Circuit Type	Input/Output	Recommended Connection of Unused Pins
P00/INTP0 to P02/INTP2	8-C	Input/output	Input: Independently connect to V_{SS0} via a resistor. Output: Leave unconnected.
P03/INTP3/ADTRG			
P10/ANI0 to P17/ANI7	25	Input	Connect to V_{DD0} or V_{SS0}.
P20/SI1	8-C	Input/output	Input: Independently connect to V_{DD0} or V_{SS0} via a resistor. Output: Leave unconnected.
P21/SO1	5-H		
P22/SCK1	8-C		
P23/RxD0			
P24/TxD0	5-H		
P25/ASCK0	8-C		
P30, P31 (for mask ROM version)	13-Q	Input/output	Input: Independently connect to V_{SS0} via a resistor. Output: Leave unconnected.
P30, P31 (for flash memory version)	13-P		
P32/SDA0	13-R		
P33/SCL0			
P34/SI3/TxD2	8-C		Input: Independently connect to V_{DD0} or V_{SS0} via a resistor. Output: Leave unconnected.
P35/SO3/RxD2			
P36/$\overline{\text{SCK3}}$/ASCK2			
P40/AD0 to P47/AD7	5-H	Input/output	Input: Independently connect to V_{SS0} via a resistor. Output: Leave unconnected.

Table 4-1. Pin Input/Output Circuit Types (2/2)

Pin Name	Input/Output Circuit Type	Input/Output	Recommended Connection of Unused Pins
P50/A8 to P57/A15	5-H	Input/output	Input: Independently connect to V_{DD0} or V_{SS0} via a resistor. Output: Leave unconnected.
P64/$\overline{\text{RD}}$		Input/output	
P65/$\overline{\text{WR}}$			
P66/$\overline{\text{WAIT}}$			
P67/ASTB			
P70/TI000/TO00	8-C		
P71/TI010			
P72/TI50/TO50			
P73/TI51/TO51			
P74/TI011/PCL			
P75/TI001/TO01/BUZ			
P80/$\overline{\text{SS1}}$			Connect to V_{SS0} via a resistor.
$\overline{\text{RESET}}$	2	Input	—
XT1	16		Connect to V_{DD0}.
XT2		—	Leave unconnected.
AV_{REF}			Connect to V_{SS0}.
AV_{SS}			Connect to V_{SS0} or V_{SS1}.
IC (for mask ROM version)			Connect directly to V_{SS0} or V_{SS1}.
V_{PP} (for flash memory version)			

Figure 4-1. Pin Input/Output Circuit List



CHAPTER 5 CPU ARCHITECTURE

5.1 Memory Spaces

μPD780078, 780078Y Subseries can access 64-Kbyte memory space respectively.

Figures 5-1 to 5-3 show memory maps.

Caution The initial value of memory size switching register (IMS) and internal expansion RAM size switching register (IXS) of all products (μPD780078 and 780078Y Subseries) is fixed (IMS = CFH, IXS = 0CH). Therefore, set the value corresponding to each products indicated below.

	Value of IMS	Value of IXS
μPD780076, 780076Y	CCH	0AH
μPD780078, 780078Y	CFH	
μPD78F0078, 78F0078Y	Value corresponding to mask ROM version	

Figure 5-1. Memory Map (μPD780076, 780076Y)

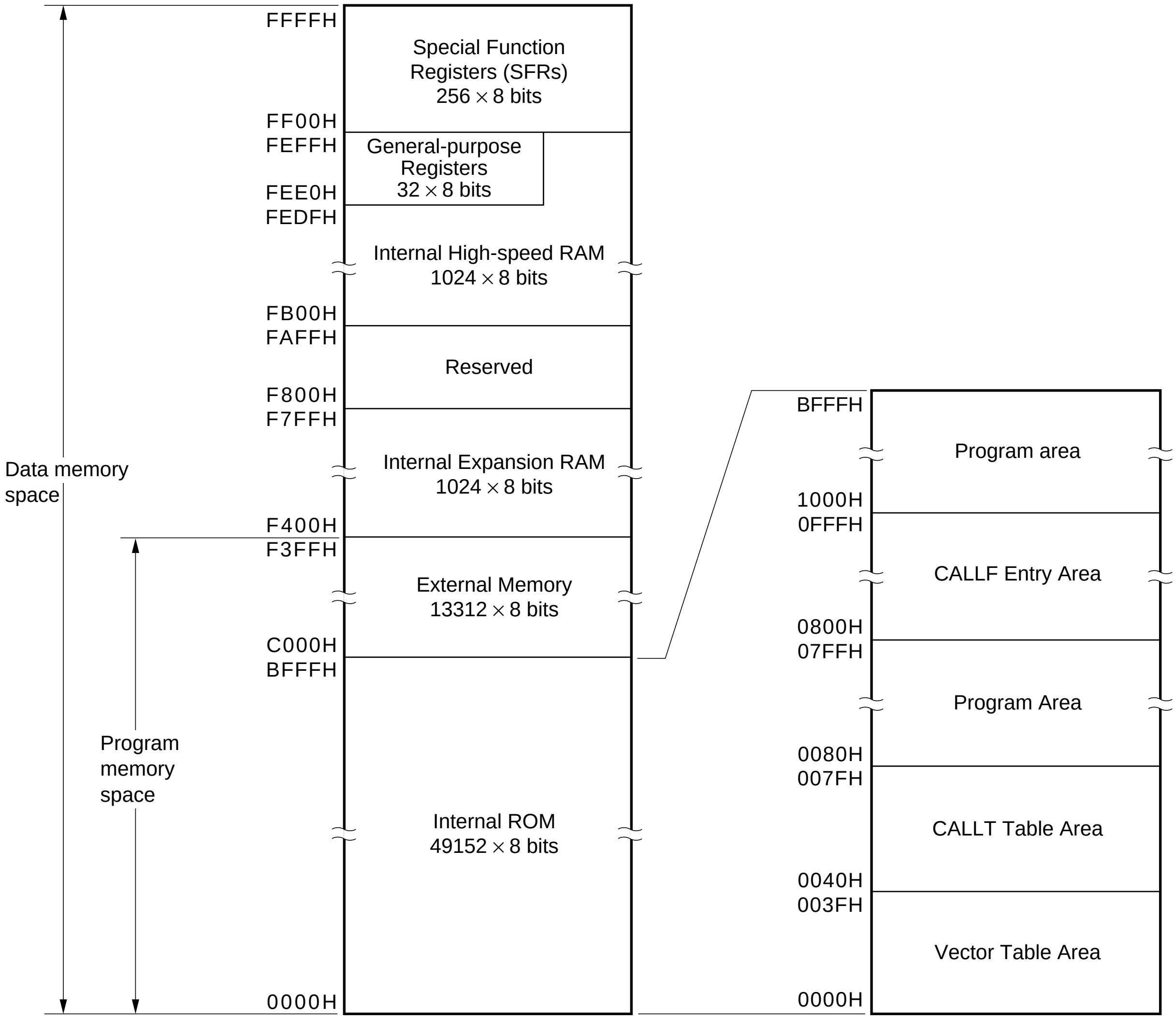


Figure 5-2. Memory Map (μPD780078, 780078Y)

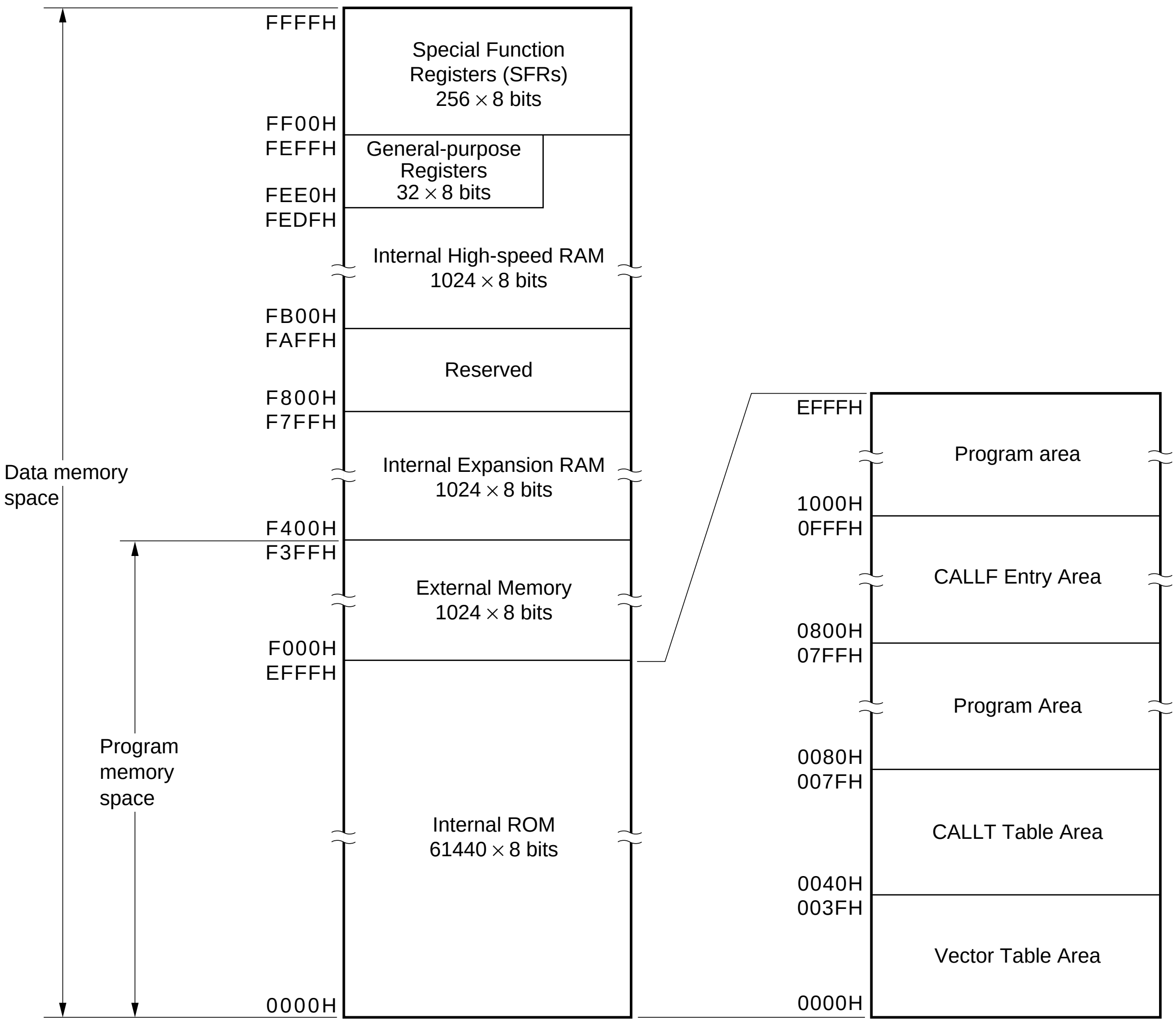
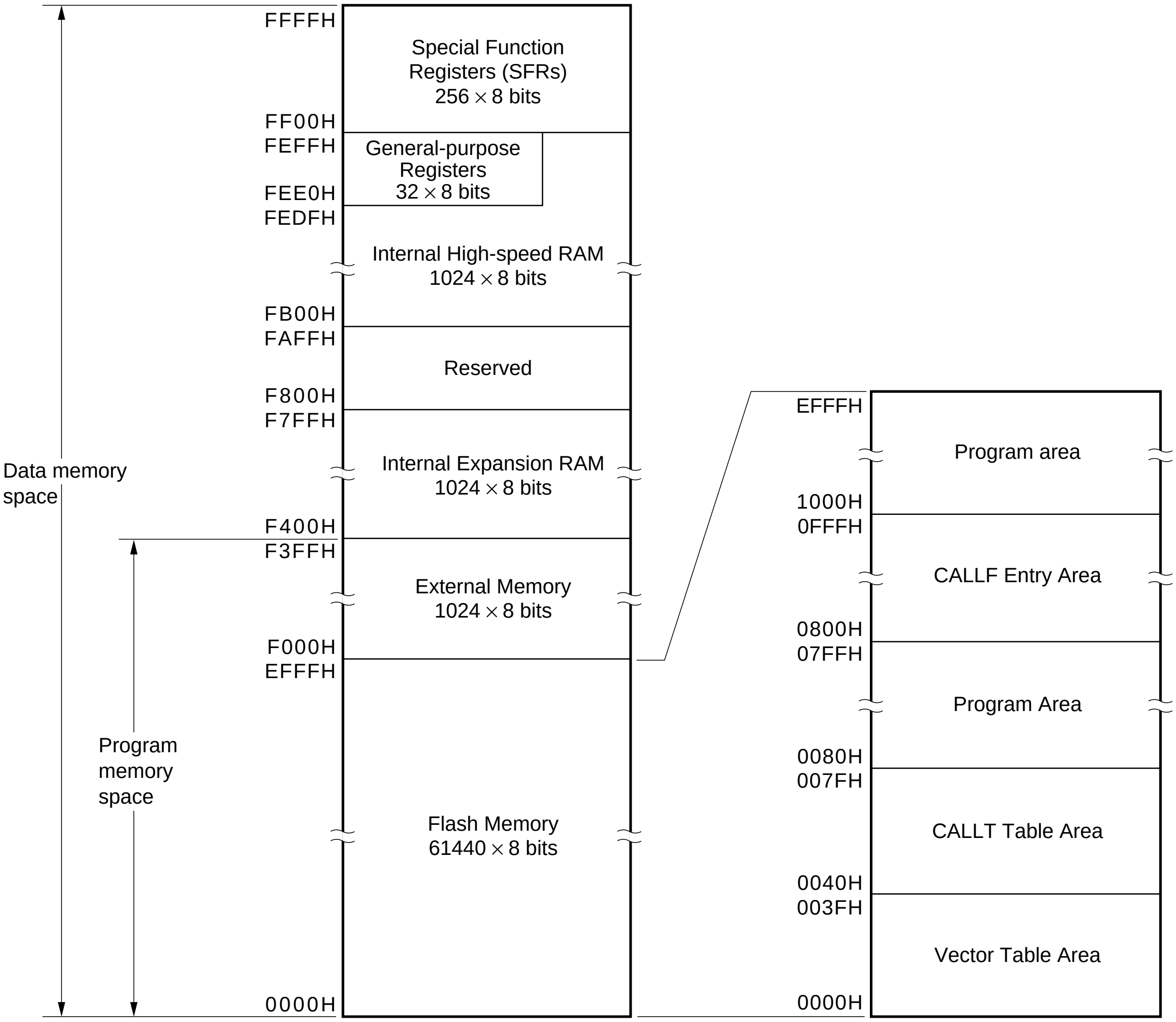


Figure 5-3. Memory Map (μPD78F0078, 78F0078Y)



5.1.1 Internal program memory space

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

The μPD780078, and 780078Y Subseries products incorporate an on-chip ROM (or flash memory), as listed below.

Table 5-1. Internal ROM Capacity

Part Number	Internal ROM	
	Type	Capacity
μPD780076, 780076Y	Mask ROM	49152 × 8 bits (0000H to BFFFH)
μPD780078, 780078Y		61440 × 8 bits (0000H to EFFFH)
μPD78F0078, 78F0078Y	Flash Memory	61440 × 8 bits (0000H to EFFFH)

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 $\overline{\text{RESET}}$ input and program start addresses for branch upon generation of each interrupt request are stored in the vector table area. Of the 16-bit address, low-order 8 bits are stored at even addresses and high-order 8 bits are stored at odd addresses.

Table 5-2. Vector Table

Vector Table Address	Interrupt Source	Vector Table Address	Interrupt Source
0000H	$\overline{\text{RESET}}$ input	001CH	INTTM000
0004H	INTWDT	001EH	INTTM010
0006H	INTP0	0020H	INTTM50
0008H	INTP1	0022H	INTTM51
000AH	INTP2	0024H	INTAD0
000CH	INTP3	0026H	INTWT
000EH	INTSER0	0028H	INTKR
0010H	INTSR0	002AH	INTSER2
0012H	INTST0	002CH	INTSR2
0014H	INTCSI1	002EH	INTST2
0016H	INTCSI3	0030H	INTTM001
0018H	INTIIC0[Note]	0032H	INTTM011
001AH	INTWTI	003EH	BRK

Note μPD780078Y Subseries only

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

5.1.2 Internal data memory space

The μPD780078 and 780078Y Subseries products incorporate the following on-chip high-speed RAMs.

(1) Internal high-speed RAM

It consists of 1024 × 8 bits, FB00H to FEFFH. The 32-byte area FEE0H to FEFFH is allocated to four general-purpose register banks composed of eight 8-bit registers.

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

(2) Internal expansion RAM

The 1024-byte area F400H to F7FFH is allocated to the internal expansion RAM.

5.1.3 Special Function Register (SFR) area

An on-chip peripheral hardware special function register (SFR) is allocated in the area FF00H to FFFFH. (Refer to **5.2.3 Special Function Register (SFR) Table 5-3. Special Function Register List**.)

Caution Do not access addresses where the SFR is not assigned.

5.1.4 External memory space

The external memory space is accessible with memory expansion mode register. External memory space can store program, table data, etc., and allocate peripheral devices.

5.1.5 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 an instruction to be executed next is addressed by the program counter (PC) (for details, see **5.3 Instruction Address Addressing**).

Several addressing modes are provided for addressing the memory relevant to the execution of instructions for the μPD780078 and 780078Y 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 5-4 to 5-6. For the details of each addressing mode, see **5.4 Operand Address Addressing**.

Figure 5-4. Data Memory Addressing (μPD780076, 780076Y)

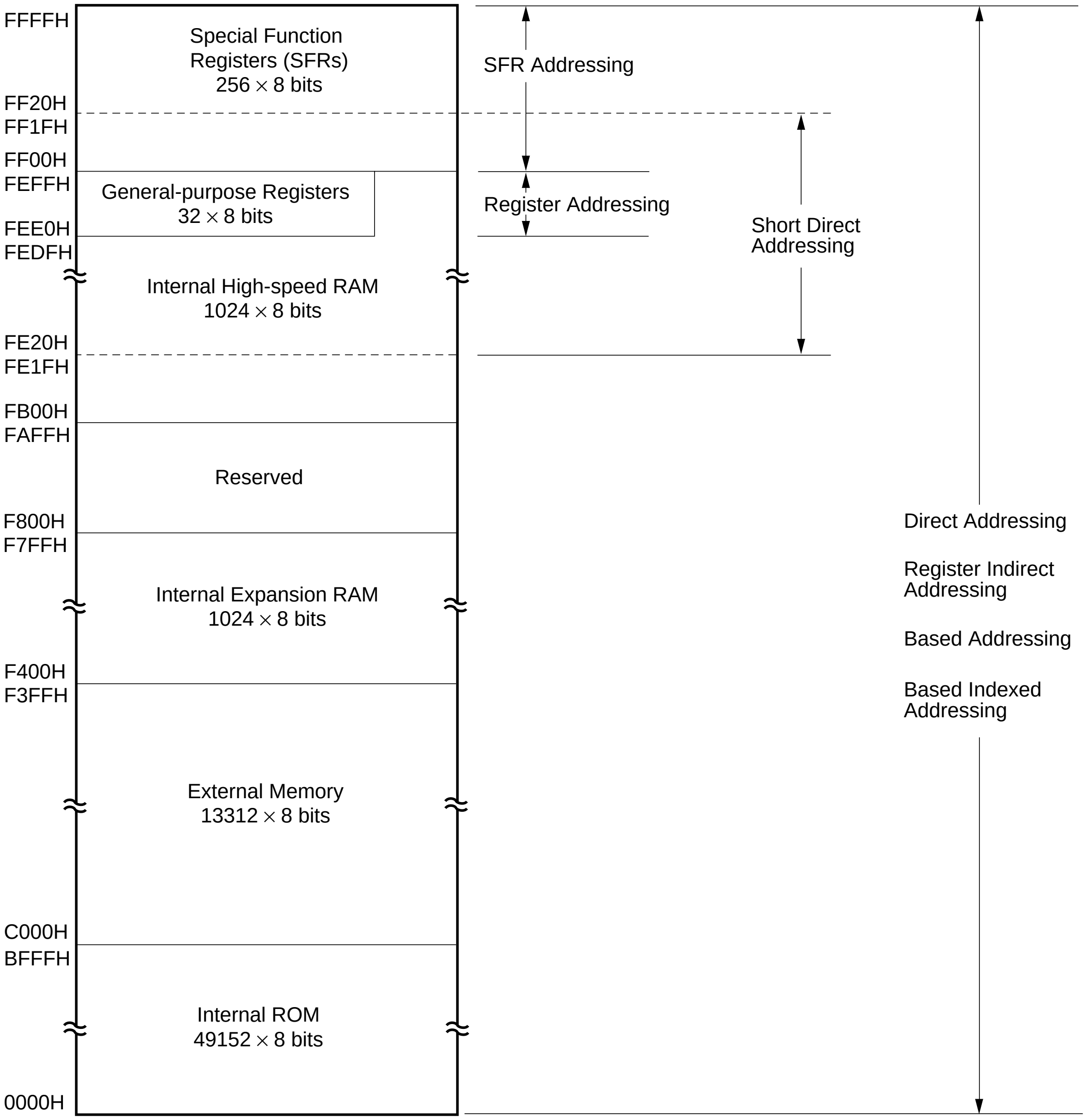

FFFFH
Special Function Registers (SFRs) 256 × 8 bits
SFR Addressing
FF20H
FF1FH
FF00H
FEFFH
General-purpose Registers 32 × 8 bits
Register Addressing
Short Direct Addressing
FEE0H
FEDFH
Internal High-speed RAM 1024 × 8 bits
FE20H
FE1FH
FB00H
FAFFH
Reserved
F800H
F7FFH
Internal Expansion RAM 1024 × 8 bits
F400H
F3FFH
External Memory 13312 × 8 bits
C000H
BFFFH
Internal ROM 49152 × 8 bits
0000H
Direct Addressing
Register Indirect Addressing
Based Addressing
Based Indexed Addressing

Figure 5-5. Data Memory Addressing (μPD780078, 780078Y)

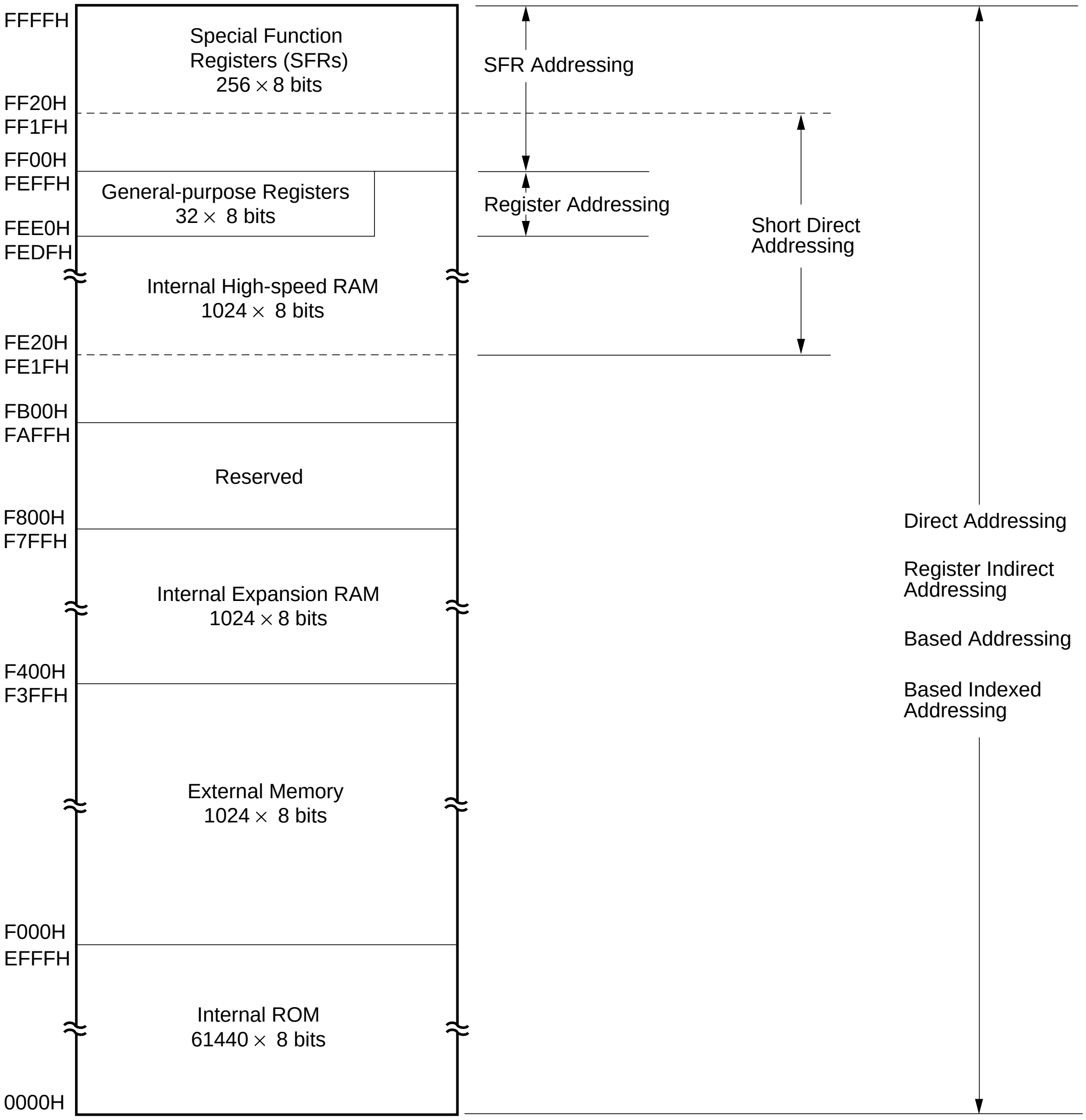
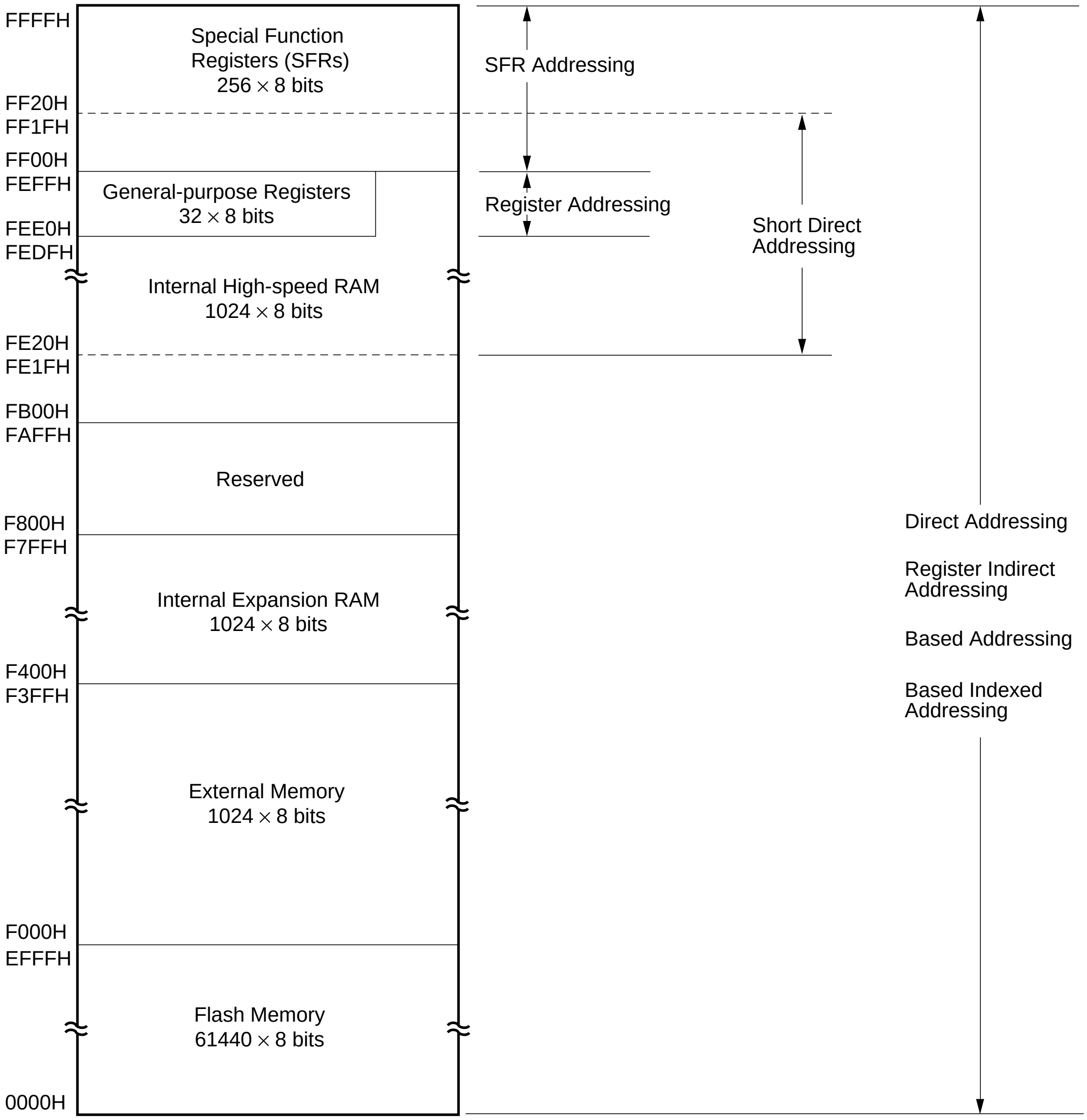

FFFFH
FF20H
FF1FH
FF00H
FEFFH
FEE0H
FEDFH
FE20H
FE1FH
FB00H
FAFFH
F800H
F7FFH
F400H
F3FFH
F000H
EFFFH
0000H
Special Function Registers (SFRs) 256 × 8 bits
General-purpose Registers 32 × 8 bits
Internal High-speed RAM 1024 × 8 bits
Reserved
Internal Expansion RAM 1024 × 8 bits
External Memory 1024 × 8 bits
Internal ROM 61440 × 8 bits
SFR Addressing
Register Addressing
Short Direct Addressing
Direct Addressing
Register Indirect Addressing
Based Addressing
Based Indexed Addressing

Figure 5-6. Data Memory Addressing (μPD78F0078, 78F0078Y)


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
F800H
F7FFH
Internal Expansion RAM 1024 × 8 bits
F400H
F3FFH
External Memory 1024 × 8 bits
F000H
EFFFH
Flash Memory 61440 × 8 bits
0000H
Direct Addressing
Register Indirect Addressing
Based Addressing
Based Indexed Addressing

5.2 Processor Registers

The μPD780078, 780078Y Subseries products incorporate the following processor registers.

5.2.1 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 reset vector table values at addresses 0000H and 0001H to the program counter.

Figure 5-7. Program Counter Format

	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 5-8. Program Status Word Format

	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 request becomes 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 with an in-service priority flag (ISP), an interrupt mask flag for various interrupt sources and a priority specification flag.

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

(b) Zero flag (Z)

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

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

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

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

(d) Auxiliary carry flag (AC)

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

(e) In-service priority flag (ISP)

This flag manages the priority of acknowledgeable maskable vectored interrupts. When this flag is 0, low-level vectored interrupts request specified with a priority specification flag register (PR0L, PR0H, PR1L) (refer to **19.3 (3) Priority specification flag registers (PR0L, PR0H, PR1L)**) are disabled for acknowledgement. When it is 1, all interrupts are acknowledgeable. Actual request acknowledgement is controlled with 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. The internal high-speed RAM areas of each product are as follows.

Figure 5-9. 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 (reset) from the stack memory.

Each stack operation saves/resets data as shown in Figures 5-10 and 5-11.

Caution Since $\overline{\text{RESET}}$ input makes SP contents undefined, be sure to initialize the SP before instruction execution.

Figure 5-10. Data To Be Saved to Stack Memory

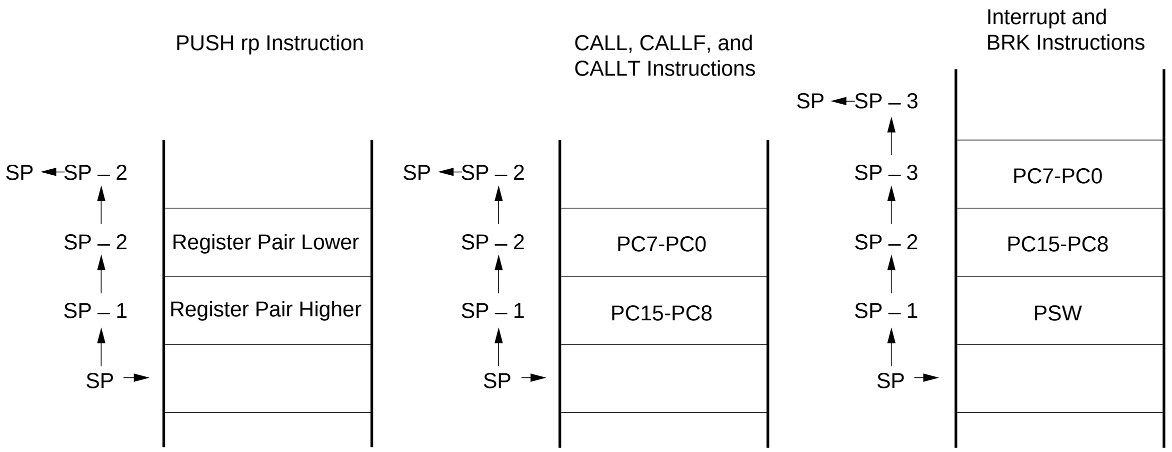
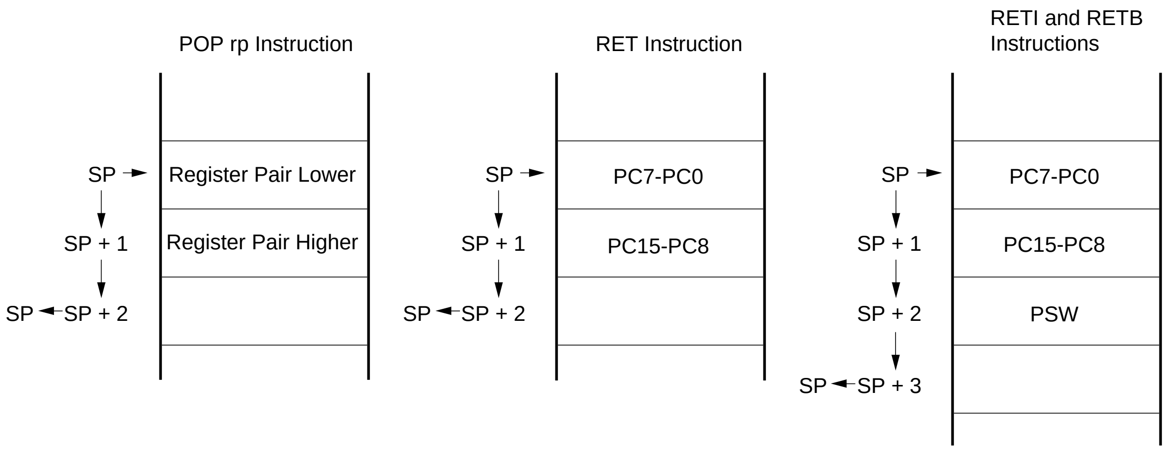


Figure 5-11. Data To Be Restored from Stack Memory



5.2.2 General-purpose registers

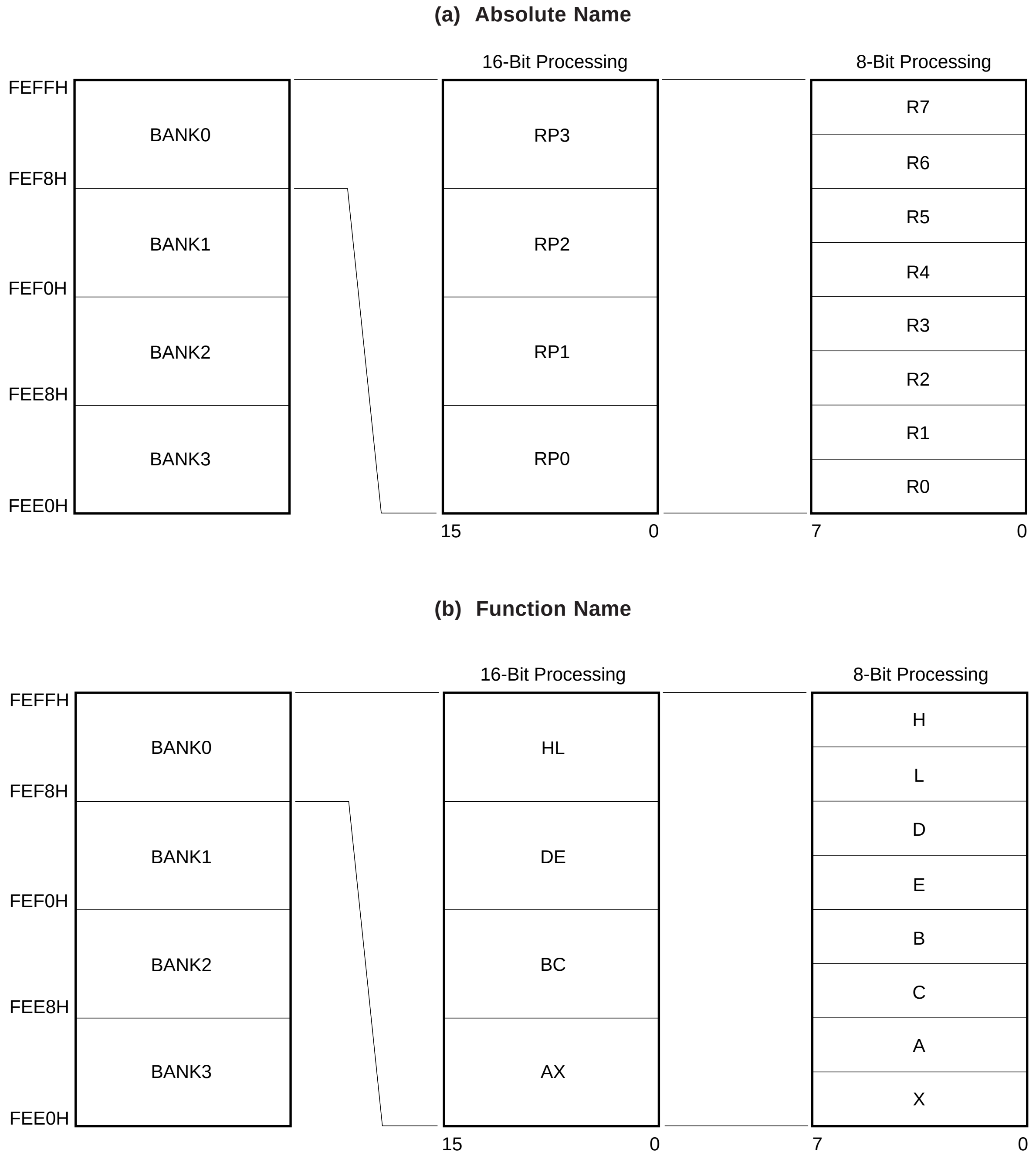
A general-purpose register is mapped at particular addresses (FEE0H to FEFFH) of the data memory. It consists of 4 banks, each bank consisting of eight 8-bit registers (X, A, C, B, E, D, L, and H).

Each register can also be used as an 8-bit register. Two 8-bit registers can 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 with 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 5-12. General-purpose Register Configuration



(a) Absolute Name

(b) Function Name

5.2.3 Special Function Register (SFR)

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

It is allocated in the FF00H to FFFFH area.

The special function register can be manipulated like the general-purpose register, with the operation, transfer and bit manipulation instructions. Manipulatable bit units, 1, 8, and 16, depend on the special function register type.

Each manipulation bit unit can be specified as follows.

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

Table 5-3 gives a list of special function registers. The meaning of items in the table is as follows.

- Symbol
  Symbol indicating the address of a special function register. It is a reserved word in the RA78K/0, and is defined via the header file “sfrbit.h” in the CC78K/0. When using the RA78K/0, 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 enable
  R : Read only
  W : Write only
- Manipulatable bit units
  Indicates the manipulatable bit unit (1, 8, or 16). “–” indicates a bit unit for which manipulation is not possible.
- After reset
  Indicates each register status upon $\overline{\text{RESET}}$ input.

Table 5-3. 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
FF01H	Port 1	P1	R	√	√	—	
FF02H	Port 2	P2	R/W	√	√	—	
FF03H	Port 3	P3		√	√	—	
FF04H	Port 4	P4		√	√	—	
FF05H	Port 5	P5		√	√	—	
FF06H	Port 6	P6		√	√	—	
FF07H	Port 7	P7		√	√	—	
FF08H	Port 8	P8		√	√	—	
FF0AH	16-bit timer capture/compare register 000	CR000		—	—	√	Undefined
FF0BH							
FF0CH	16-bit timer capture/compare register 010	CR010		—	—	√	
FF0DH							
FF0EH	16-bit timer counter 00	TM00	R	—	—	√	0000H
FF0FH							
FF10H	8-bit timer compare register 50	CR50	R/W	—	√	—	Undefined
FF11H	8-bit timer compare register 51	CR51		—	√	—	
FF12H	8-bit timer counter 50	TM5 TM50	R	—	√	√	00H
FF13H	8-bit timer counter 51	TM51		—	√		
FF14H	Transmit buffer register 2	TXB2	R/W	—	√	—	FFH
FF15H	Receive buffer register 2	RXB2	R	—	√	—	FFH
FF16H	A/D conversion result register 0	ADCR0		—	—	√	00H
FF17H				—	—		
FF18H	Transmit shift register 0	TXS0	W	—	√	—	FFH
	Receive buffer register 0	RXB0	R	—	√	—	
FF19H	Transmit buffer register 1	SOTB1	R/W	—	√	—	Undefined
FF1AH	Serial I/O shift register 1	SIO1	R	—	√	—	
FF1BH	Serial I/O shift register 3	SIO3	R/W	—	√	—	
FF1FH	IIC shift register 0[Note]	IIC0		—	√	—	00H
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		√	√	—	

Note μPD780078Y Subseries only

Table 5-3. 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	
FF30H	Pull-up resistor option register 0	PU0	R/W	√	√	—	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		√	√	—	
FF36H	Pull-up resistor option register 6	PU6		√	√	—	
FF37H	Pull-up resistor option register 7	PU7		√	√	—	
FF38H	Pull-up resistor option register 8	PU8		√	√	—	
FF40H	Clock output select register	CKS		√	√	—	
FF41H	Watch timer operation mode control register	WTM		√	√	—	
FF42H	Watchdog timer clock select 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		√	√	—	
FF64H	16-bit timer mode control register 01	TMC01		√	√	—	
FF65H	Prescaler mode register 01	PRM01		—	√	—	
FF66H	Capture/compare control register 01	CRC01		√	√	—	
FF67H	16-bit timer output control register 01	TOC01		√	√	—	
FF68H	16-bit timer capture/compare register 001	CR001		—	—	√	Undefined
FF69H							
FF6AH	16-bit timer capture/compare register 011	CR011		—	—	√	
FF6BH							
FF6CH	16-bit timer counter 01	TM01	R	—	—	√	0000H
FF6DH							
FF70H	8-bit timer mode control register 50	TMC50	R/W	√	√	—	
FF71H	Timer clock select register 50	TCL50		—	√	—	
FF78H	8-bit timer mode control register 51	TMC51		√	√	—	
FF79H	Timer clock select register 51	TCL51		—	√	—	
FF80H	A/D converter mode register 0	ADM0		√	√	—	
FF81H	Analog input channel specification register 0	ADS0		—	√	—	
FF90H	Asynchronous serial interface mode register 2	ASIM2		√	√	—	
FF91H	Transfer mode specification register 2	TRMC2		√	√	—	02H
FF92H	Clock select register 2	CKSEL2		—	√	—	00H
FF93H	Baud rate generator control register 2	BRGC2		—	√	—	

Table 5-3. 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	
FF94H	Asynchronous serial interface status register 2	ASIS2		R	—	√	—	00H
FF95H	Asynchronous serial interface transmit status register 2	ASIF2			—	√	—	
FFA0H	Asynchronous serial interface mode register 0	ASIM0			√	√	—	
FFA1H	Asynchronous serial interface status register 0	ASIS0			—	√	—	
FFA2H	Baud rate generator control register 0	BRGC0		R/W	—	√	—	
FFA8H	IIC control register 0[Note 1]	IICC0			√	√	—	
FFA9H	IIC status register 0[Note 1]	IICS0		R	√	√	—	
FFAAH	IIC transfer clock select register 0[Note 1]	IICCL0		R/W	√	√	—	
FFABH	Slave address register 0[Note 1]	SVA0			—	√	—	
FFB0H	Serial operation mode register 1	CSIM1			√	√	—	
FFB1H	Serial clock select register 1	CSIC1			√	√	—	10H
FFB8H	Serial operation mode register 3	CSIM3			√	√	—	00H
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	IF1L			√	√	—	
FFE4H	Interrupt mask flag register 0L	MK0	MK0L		√	√	√	FFH
FFE5H	Interrupt mask flag register 0H		MK0H		√	√		
FFE6H	Interrupt mask flag register 1L	MK1L			√	√	—	
FFE8H	Priority level specification flag register 0L	PR0	PR0L		√	√	√	
FFE9H	Priority level specification flag register 0H		PR0H		√	√		
FFEAH	Priority level specification flag register 1L	PR1L			√	√	—	
FFF0H	Memory size switching register	IMS			—	√	—	CFH[Note 3]
FFF4H	Internal expansion RAM size switching register	IXS			—	√	—	0CH[Note 4]
FFF8H	Memory expansion wait setting register	MM			√	√	—	10H
FFF9H	Watchdog timer mode register	WDTM			√	√	—	00H
FFFAH	Oscillation stabilization time select register	OSTS			—	√	—	04H
FFFBH	Processor clock control register	PCC			√	√	—	

Notes 1. μPD780078Y Subseries only

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

3. Although the default value of this register is CFH, set the value corresponding to each respective product as indicated below.

μPD780076, 780076Y : CCH
μPD780078, 780078Y : CFH
μPD78F0078, 78F0078Y : Value for mask ROM version

4. Although the default value of this register is 0CH, initialize this register to 0AH.

5.3 Instruction Address Addressing

An instruction address is determined by 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 **78K/0 User's Manual - Instructions (U12326E)**.

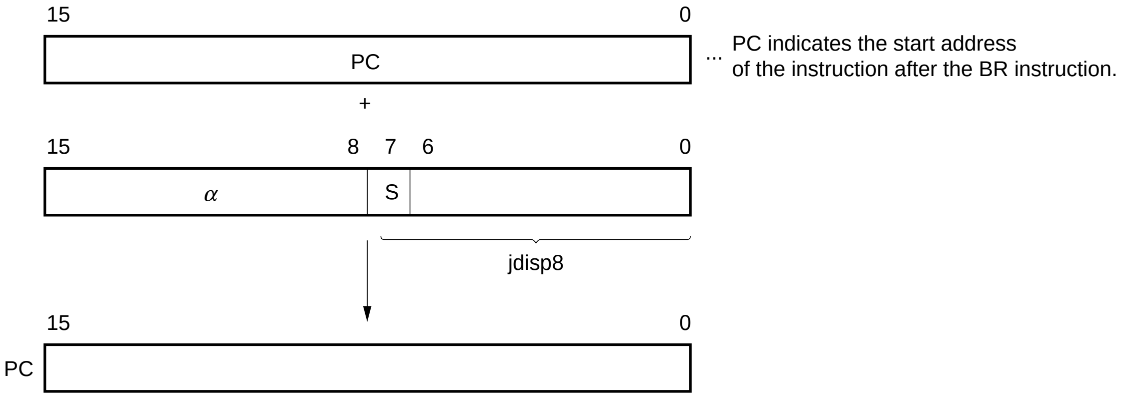
5.3.1 Relative addressing

[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 in 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.

5.3.2 Immediate addressing

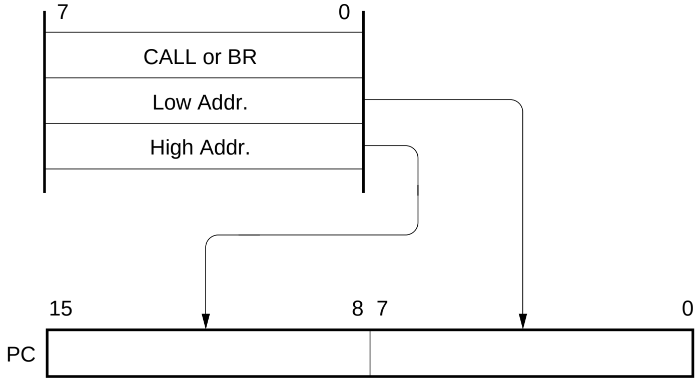
[Function]

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

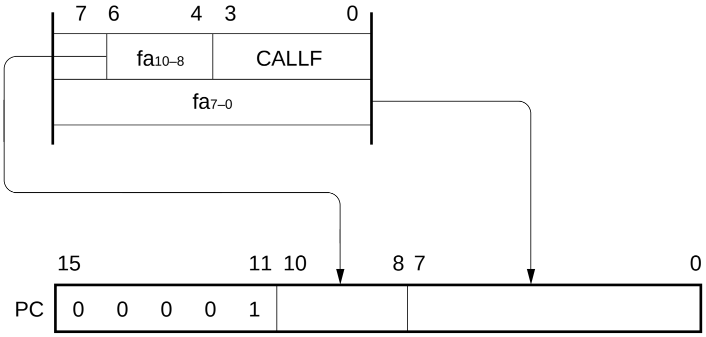
This function is carried out when the CALL !addr16 or BR !addr16 or CALLF !addr11 instruction is executed. CALL !addr16 and BR !addr16 instructions can be branched to the entire memory space. The CALLF !addr11 instruction is branched to the 0800H to 0FFFH area.

[Illustration]

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



In the case of CALLF !addr11 instruction



5.3.3 Table indirect addressing

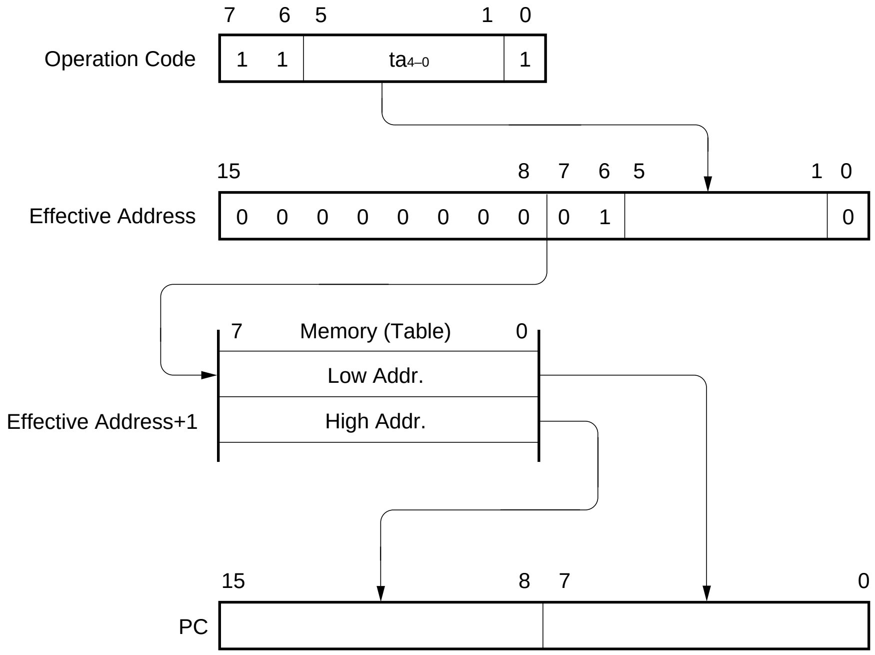
[Function]

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

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

This instruction references the address stored in the memory table from 40H to 7FH, and allows branching to the entire memory space.

[Illustration]



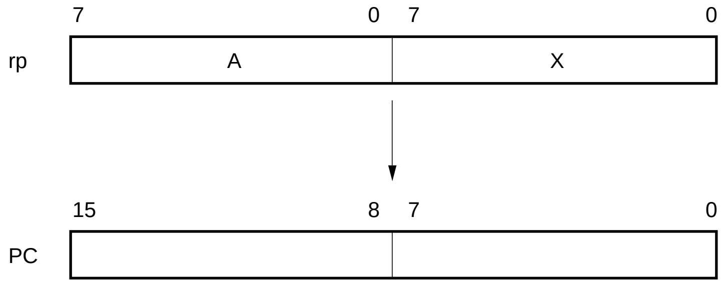
5.3.4 Register addressing

[Function]

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

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

[Illustration]



5.4 Operand Address Addressing

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

5.4.1 Implied addressing

[Function]

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

Of the μPD780078 and 780078Y 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 which become decimal correction targets
ROR4/ROL4	A register for storage of digit data which undergoes digit rotation

[Operand format]

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

[Description example]

In the case of MULU X

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

5.4.2 Register addressing

[Function]

The general-purpose register to be specified is accessed as an operand with the register specify code (Rn and RPn) of an instruction word in the registered bank specified with the register bank select 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 with 3 bits in the operation code.

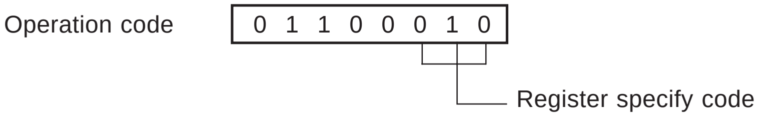
[Operand format]

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

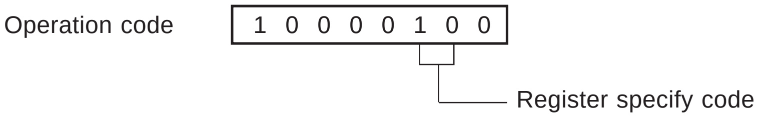
'r' and 'rp' can be described with absolute names (R0 to R7 and RP0 to RP3) as well as function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL).

[Description example]

MOV A, C; when selecting C register as r



INCW DE; when selecting DE register pair as rp



5.4.3 Direct addressing

[Function]

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

[Operand format]

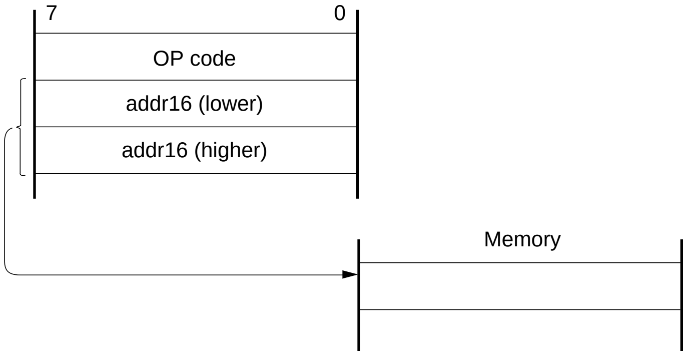
Identifier	Description
addr16	Label or 16-bit immediate data

[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]



5.4.4 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. An internal RAM and a special function register (SFR) are mapped at FE20H to FEFFH and FF00H to FF1FH, respectively.

If the SFR area (FF00H to FF1FH) where short direct addressing is applied, ports which are frequently accessed in a program and a compare register of the timer/event counter and a capture register of the timer/event counter are mapped 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]

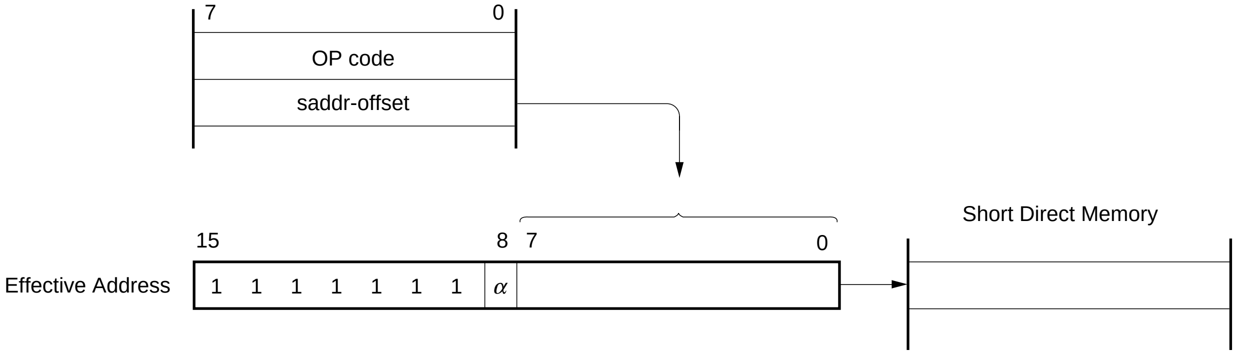
Identifier	Description
saddr	Label of FE20H to FF1FH immediate data
saddrp	Label of FE20H to FF1FH immediate data (even address only)

[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, $\alpha = 0$
When 8-bit immediate data is 00H to 1FH, $\alpha = 1$

5.4.5 Special function register (SFR) addressing

[Function]

The memory-mapped special function register (SFR) is addressed with 8-bit immediate data in an instruction word.

This addressing is applied to the 240-byte spaces FF00H to FFCFH and FFE0H to FFFFH. However, the SFR mapped at FF00H to FF1FH can be accessed with short direct addressing.

[Operand format]

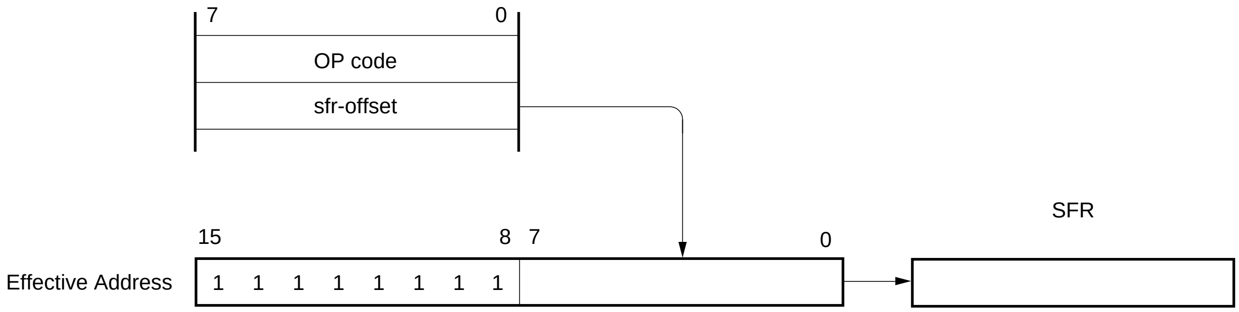
Identifier	Description
sfr	Special function register name
sfrp	16-bit manipulatable special function register name (even address only)

[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]



5.4.6 Register indirect addressing

[Function]

Register pair contents specified with a register pair specify code in an instruction word of the register bank specified with a register bank select 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]

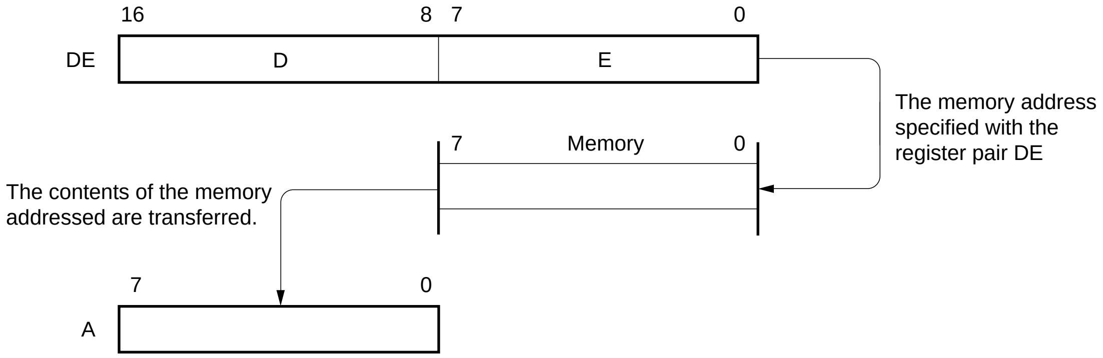
Identifier	Description
—	[DE], [HL]

[Description example]

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

Operation code 1 0 0 0 0 1 0 1

[Illustration]


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

5.4.7 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 with the register bank select flag (RBS0 and RBS1) and the sum is used to address the memory. Addition is performed by expanding the offset data as a positive number to 16 bits. A carry from the 16th bit is ignored. This addressing can be carried out for all 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

5.4.8 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 with the register bank select flag (RBS0 and RBS1) and the sum is used to address the memory.

Addition is performed by expanding the offset data as a positive number to 16 bits. A carry from the 16th bit is ignored. This addressing can be carried out for all the memory spaces.

[Operand format]

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

[Description example]

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

Operation code 1 0 1 0 1 0 1 1

5.4.9 Stack addressing

[Function]

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

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

Stack addressing enables to address the internal high-speed RAM area only.

[Description example]

In the case of PUSH DE

Operation code 1 0 1 1 0 1 0 1

[MEMO]

CHAPTER 6 PORT FUNCTIONS

6.1 Port Functions

The μPD780078 and 780078Y Subseries products incorporate eight input ports and 44 input/output ports. Figure 6-1 shows the port configuration. Every port is capable of 1-bit and 8-bit manipulations and can carry out considerably varied control operations. Besides port functions, the ports can also serve as on-chip hardware input/output pins.

Figure 6-1. Port Types

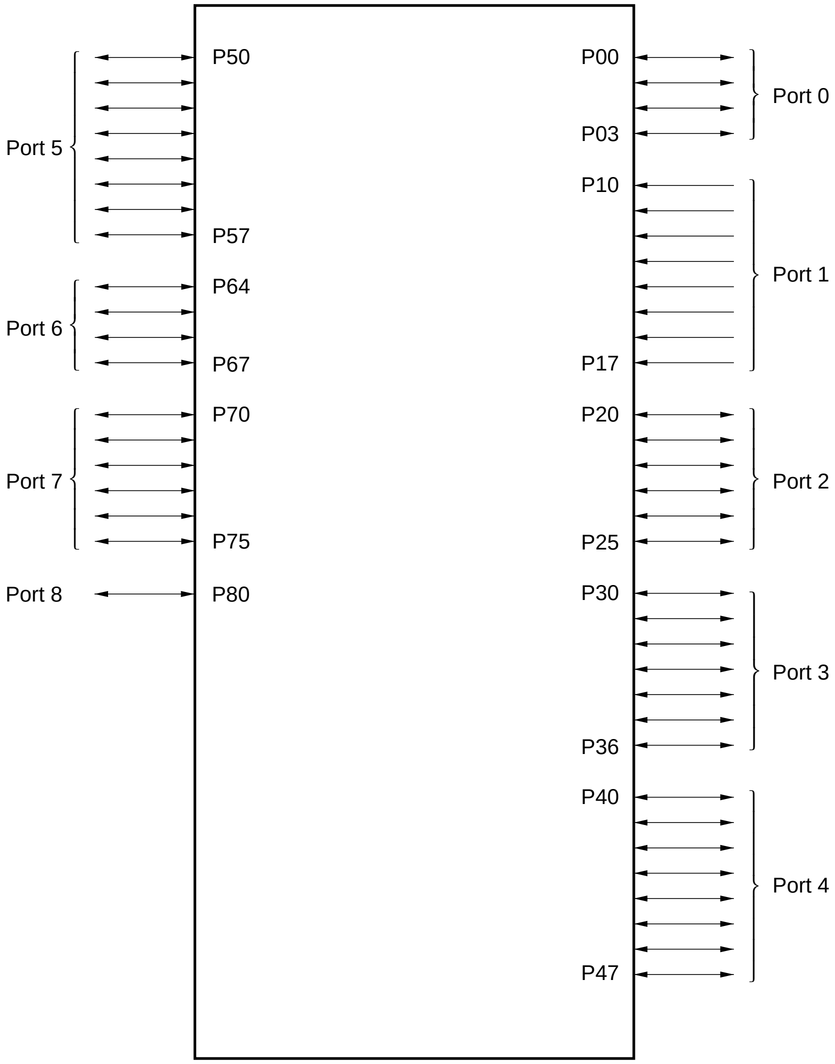


Table 6-1. Port Functions (μPD780078 Subseries)

Pin Name	Function		Alternate Function
P00	Port 0 4-bit input/output port. Input/output mode can be specified bit-wise. An on-chip pull-up resistor can be used by software.		INTP0
P01			INTP1
P02			INTP2
P03			INTP3/ADTRG
P10 to P17	Port 1 8-bit input only port.		ANI0 to ANI7
P20	Port 2 6-bit input/output port. Input/output mode can be specified bit-wise. An on-chip pull-up resistor can be used by software.		SI1
P21			SO1
P22			SCK1
P23			RxD0
P24			TxD0
P25			ASCK0
P30	Port 3 7-bit input/output port. Input/output mode can be specified bit-wise.	N-ch open-drain input/output port. On-chip pull-up resistor can be specified by mask option (Mask ROM version only). LEDs can be driven directly.	—
P31			
P32			
P33			
P34		An on-chip pull-up resistor can be specified by software.	SI3/TxD2
P35			SO3/RxD2
P36			$\overline{\text{SCK3}}$/ASCK2
P40 to P47	Port 4 8-bit input/output port. Input/output mode can be specified bit-wise. An on-chip pull-up resistor can be specified by software. Interrupt request flag (KRIF) is set to 1 by falling edge detection.		AD0 to AD7
P50 to P57	Port 5 8-bit input/output port. LEDs can be driven directly. Input/output mode can be specified bit-wise. An on-chip pull-up resistor can be used by software.		A8 to A15
P64	Port 6 4-bit input/output port. Input/output mode can be specified bit-wise. An on-chip pull-up resistor can be used by software.		$\overline{\text{RD}}$
P65			$\overline{\text{WR}}$
P66			$\overline{\text{WAIT}}$
P67			ASTB
P70	Port 7 6-bit input/output port. Input/output mode can be specified bit-wise. An on-chip pull-up resistor can be used by software.		TI000/TO00
P71			TI010
P72			TI50/TO50
P73			TI51/TO51
P74			TI011/PCL
P75			TI001/TO01/BUZ
P80	Port 8 1-bit input/output port. Input/output mode can be specified bit-wise. An on-chip pull-up resistor can be used by software.		$\overline{\text{SS1}}$

Table 6-2. Port Functions (μPD780078Y Subseries)

Pin Name	Function		Alternate Function
P00	Port 0 4-bit input/output port. Input/output mode can be specified bit-wise. An on-chip pull-up resistor can be used by software.		INTP0
P01			INTP1
P02			INTP2
0P3			INTP3/ADTRG
P10 to P17	Port 1 8-bit input only port.		ANI0 to ANI7
P20	Port 2 6-bit input/output port Input/output mode can be specified bit-wise. An on-chip pull-up resistor can be used by software.		SI1
P21			SO1
P22			SCK1
P23			RxD0
P24			TxD0
P25			ASCK0
P30	Port 3 7-bit input/output port. Input/output mode can be specified bit-wise.	N-ch open-drain input/output port. On-chip pull-up resistor can be specified by mask option (P30 and P31 are Mask ROM version only). LEDs can be driven directly.	—
P31			
P32			SDA0
P33			SCL0
P34		An on-chip pull-up resistor can be used by software.	SI3/TxD2
P35			SO3/RxD2
P36			$\overline{\text{SCK3}}$/ASCK2
P40 to P47	Port 4 8-bit input/output port. Input/output mode can be specified bit-wise. An on-chip pull-up resistor can be used by software. Interrupt request flag (KRIF) is set to 1 by falling edge detection.		AD0 to AD7
P50 to P57	Port 5 8-bit input/output port. LEDs can be driven directly. Input/output mode can be specified bit-wise. An on-chip pull-up resistor can be used by software.		A8 to A15
P64	Port 6 4-bit input/output port. Input/output mode can be specified bit-wise. An on-chip pull-up resistor can be used by software.		$\overline{\text{RD}}$
P65			$\overline{\text{WR}}$
P66			$\overline{\text{WAIT}}$
P67			ASTB
P70	Port 7 6-bit input/output port. Input/output mode can be specified bit-wise. An on-chip pull-up resistor can be connected by software.		TI000/TO00
P71			TI010
P72			TI50/TO50
P73			TI51/TO51
P74			TI011/PCL
P75			TI001/TO01/BUZ
P80	Port 8 1-bit input/output port. Input/output mode can be specified bit-wise. An on-chip pull-up resistor can be used by software.		$\overline{\text{SS1}}$

6.2 Port Configuration

A port consists of the following hardware:

Table 6-3. Port Configuration

Item	Configuration
Control register	Port mode register (PMm: m = 0, 2 to 8) Pull-up resistor option register (PUm,: m = 0, 2 to 8)
Port	Total: 52 ports (8 inputs, 44 inputs/outputs)
Pull-up resistor	• Mask ROM version Total: 44 (software specifiable: 40, mask option: 4[Note]) • Flash memory version Total: 40

Note Two mask options for the μPD780078Y subseries.

6.2.1 Port 0

Port 0 is a 4-bit input/output port with output latch. P00 to P03 pins can specify the input mode/output mode in 1-bit units with the port mode register 0 (PM0). An on-chip pull-up resistor of P00 to P03 pins can be used to them in 6-bit units with a pull-up resistor option register 0 (PU0).

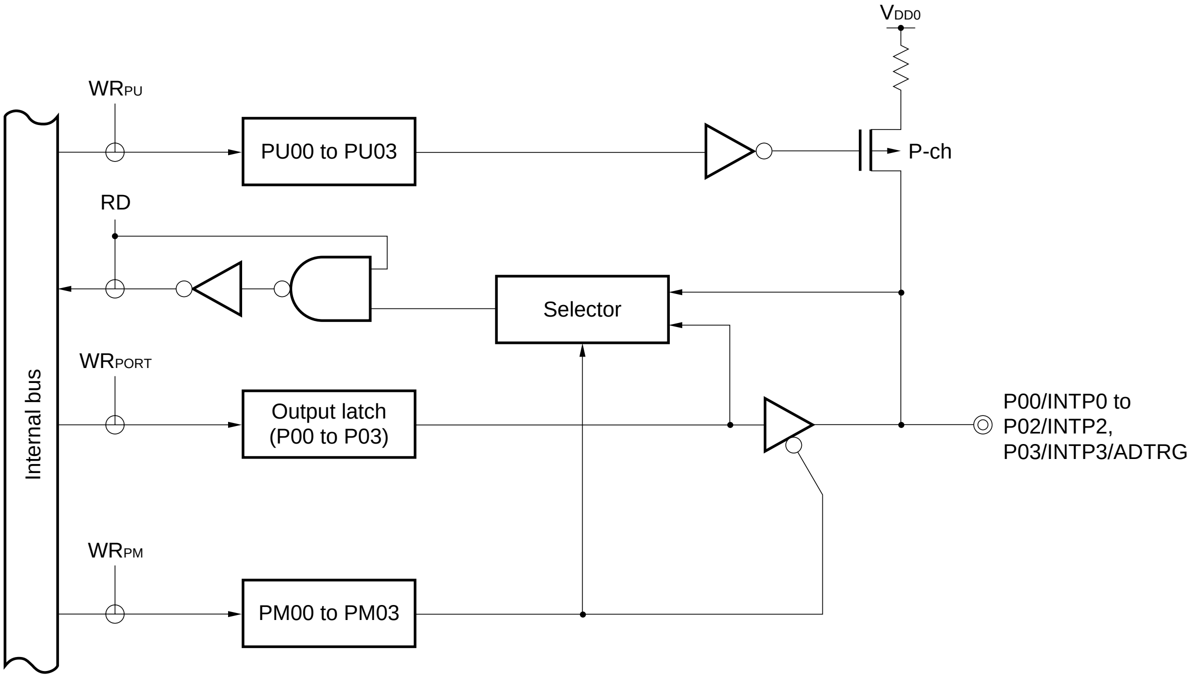
This port can also be used as an external interrupt request input, and A/D converter external trigger input.

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

Figure 6-2 shows a block diagram of port 0.

Caution Because port 0 also serves 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 6-2. P00 to P03 Block Diagram



PU : Pull-up resistor option register

PM : Port mode register

RD : Port 0 read signal

WR : Port 0 write signal

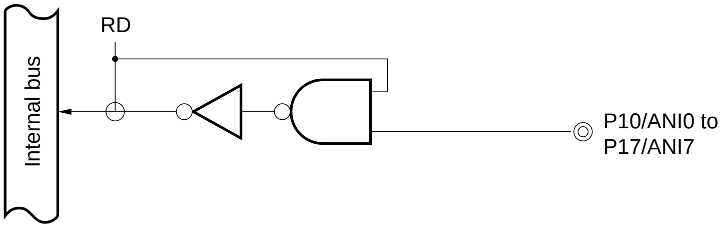
6.2.2 Port 1

Port 1 is an 8-bit input-only port.

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

Figure 6-3 shows a block diagram of port 1.

Figure 6-3. P10 to P17 Block Diagram



RD : Port 1 read signal

6.2.3 Port 2

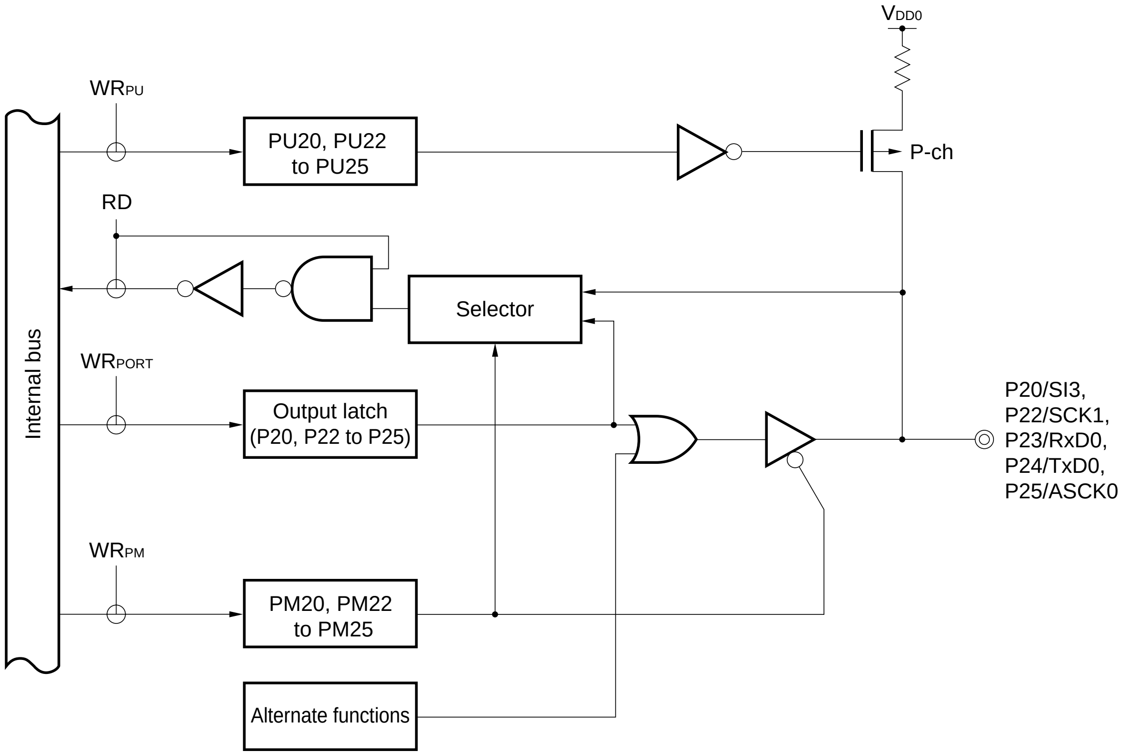
Port 2 is a 6-bit input/output port with output latch. P20 to P25 pins can specify the input mode/output mode in 1-bit units with the port mode register 2 (PM2). An on-chip pull-up resistor of P20 to P25 pins can be used for them in 1-bit units with a pull-up resistor option register 2 (PU2).

This port has also alternate functions as serial interface data input/output and clock input/output.

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

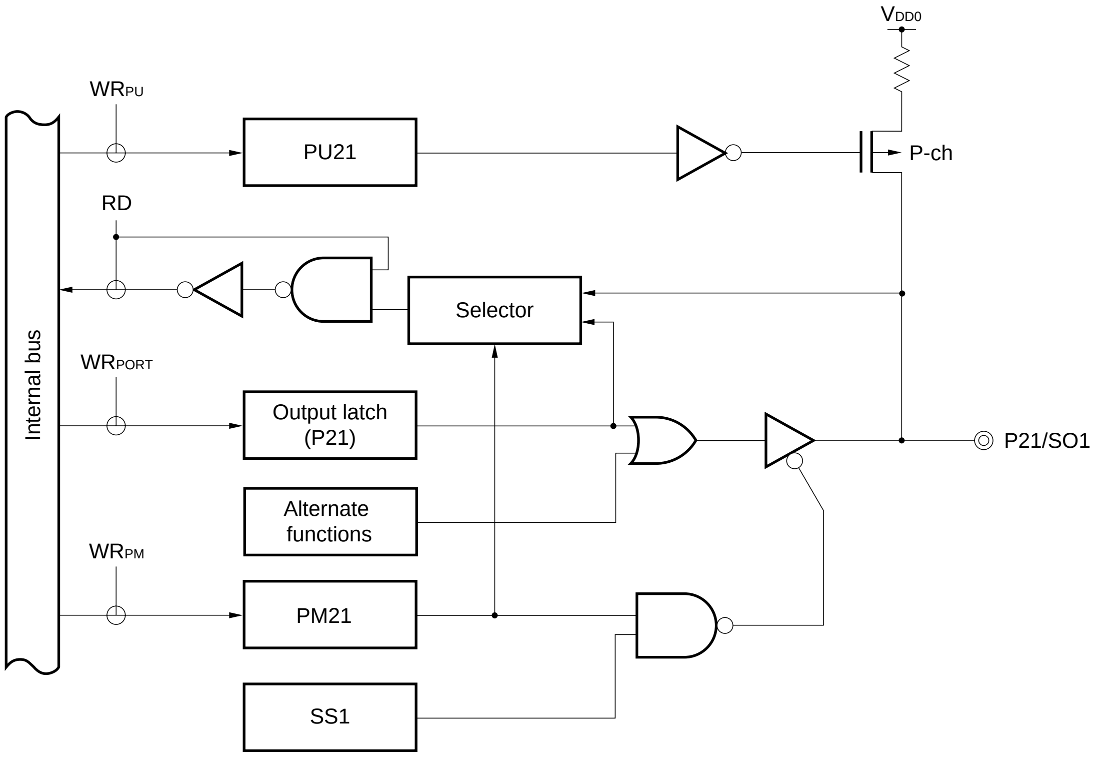
Figures 6-4 and 6-5 show a block diagram of port 2.

Figure 6-4. P20, P22 to P25 Block Diagram



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

Figure 6-5. P21 Block Diagram



PU : Pull-up resistor option register
PM : Port mode register
RD : Port 2 read signal
SS1: 3-wire SIO chip select signal
WR : Port 2 write signal

6.2.4 Port 3 (μPD780078 Subseries)

Port 2 is a 7-bit input/output port with output latch. P30 to P36 pins can specify the input mode/output mode in 1-bit units with port mode register 3 (PM3).

This port has the following functions for pull-up resistors. These functions differ depending on the board's high-order 3-bit/low-order 4-bit, and whether the products is a Mask ROM version or a flash memory version.

Table 6-4. Pull-Up Resistor of Port 3 (μPD780078 Subseries)

	High-Order 3-Bit (P34 to P36 pins)	Low-Order 4-Bit (P30 to P33 pins)
Mask ROM version	An on-chip pull-up resistor can be used bit-wise by PU3	On-chip pull-up resistor can be specified bit-wise by mask option
Flash memory version		On-chip pull-up resistor is not provided

PU3: Pull-up resistor option register 3

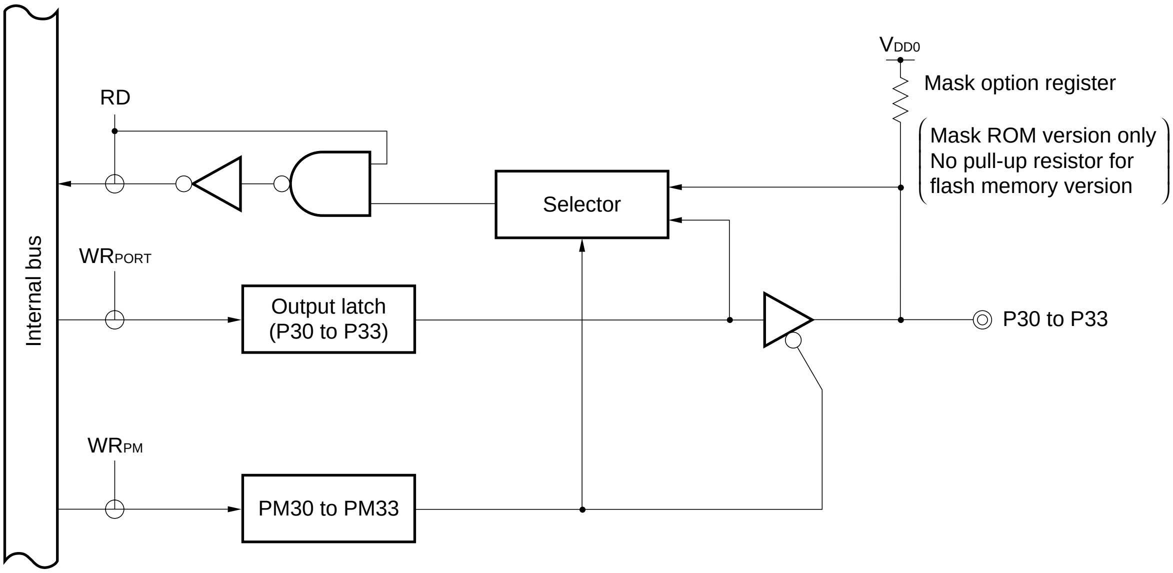
The P30 to P33 pins can drive LEDs directly.

The P34 to P36 pins can also be used for serial interface data input/output and clock input/output.

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

Figures 6-6 and 6-7 show block diagrams of port 3.

Figure 6-6. P30 to P33 Block Diagram (μPD780078 Subseries)

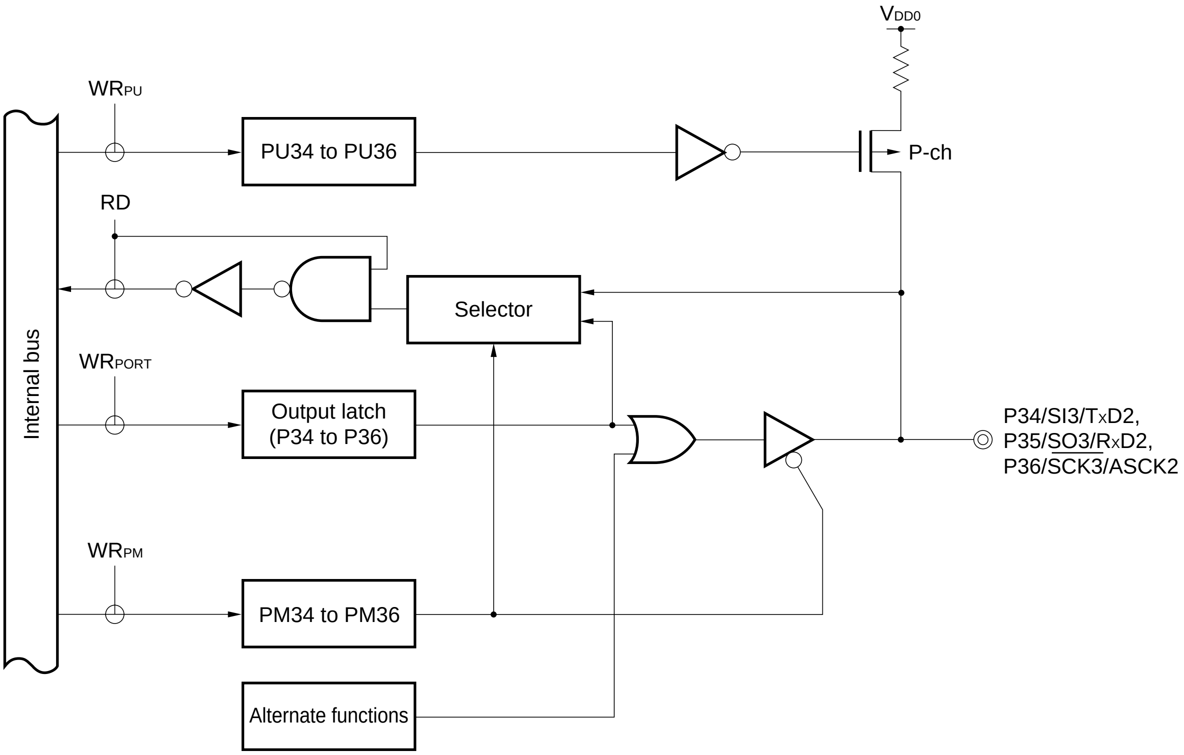


PM : Port mode register

RD : Port 3 read signal

WR : Port 3 write signal

Figure 6-7. P34 to P36 Block Diagram (μPD780078 Subseries)



PU : Pull-up resistor option register

PM : Port mode register

RD : Port 3 read signal

WR : Port 3 write signal

6.2.5 Port 3 (μPD780078Y Subseries)

Port 3 is a 7-bit input/output port with output latch. P30 to P36 pins can specify the input mode/output mode in 1-bit units with port mode register 3 (PM3).

This port has the following functions for pull-up resistors.

These functions differ depending on bits location and mask ROM version/flash memory version.

Table 6-5. Pull-Up Resistor of Port 3 (μPD780078Y Subseries)

	P34 to P36 Pins	P30, P31 Pins
Mask ROM version connected bit-wise by PU3	A pull-up resistor can be bit-wise by mask option	On-chip pull-up resistor can be specified
Flash memory version		On-chip pull-up resistor is not be specified

PU3: Pull-up resistor option register 3

Caution P32 and P33 pins have no pull-up resistor.

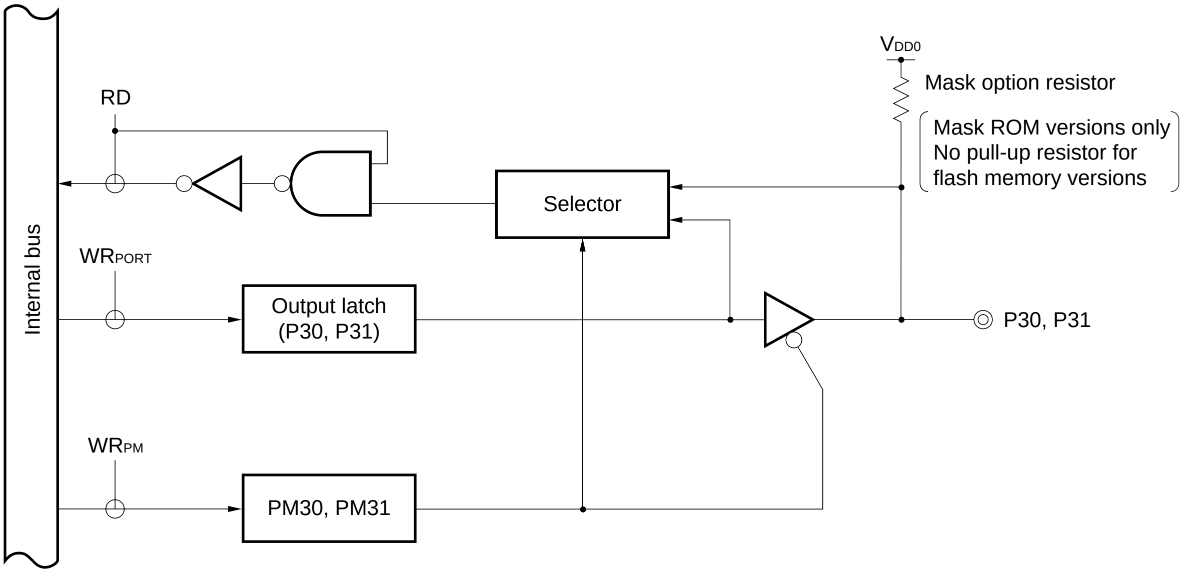
The P30 to P33 pins can drive LEDs directly.

The P32 to P36 pins can also be used for interface data input/output and clock input/output.

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

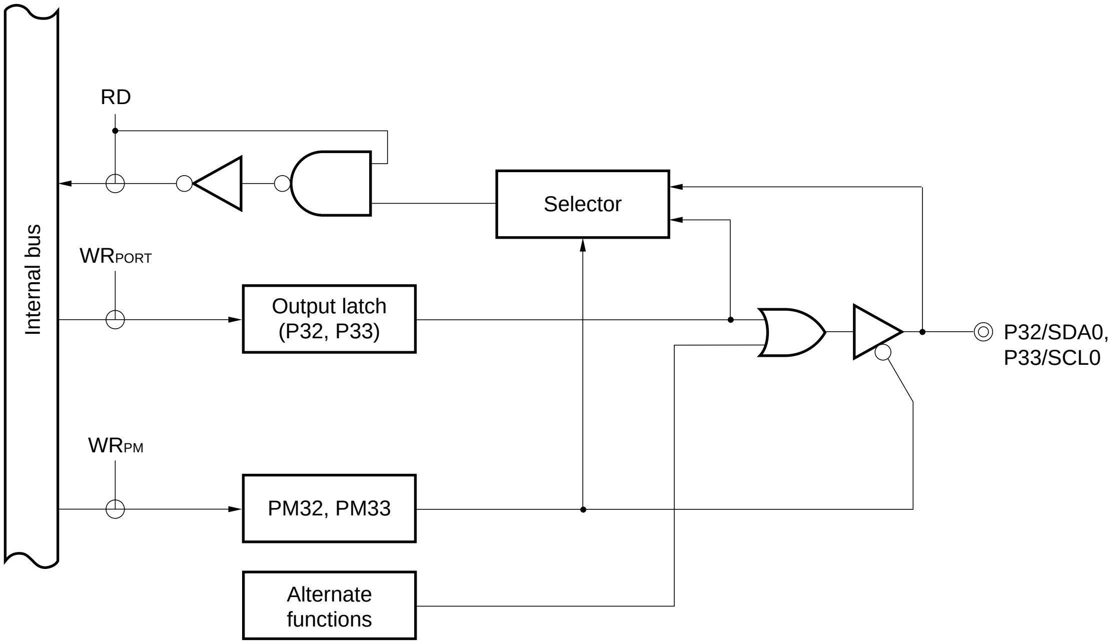
Figures 6-8 through 6-10 show block diagrams of port 3.

Figure 6-8. P30 and P31 Block Diagram (μPD780078Y Subseries)



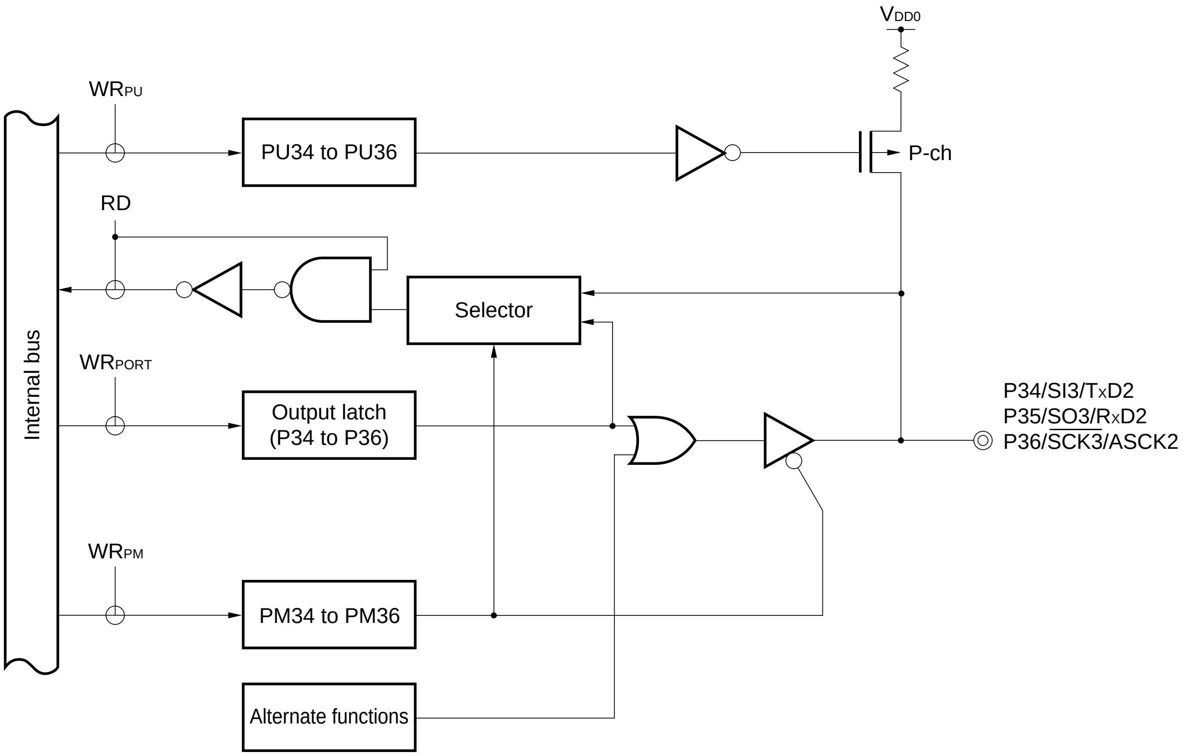
PM : Port mode register
RD : Port 3 read signal
WR : Port 3 write signal

Figure 6-9. P32 and P33 Block Diagram (μPD780078Y Subseries)



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

Figure 6-10. P34 to P36 Block Diagram (μPD780078Y Subseries)



PU : Pull-up resistor option register

PM : Port mode register

RD : Port 6 read signal

WR : Port 6 write signal

6.2.6 Port 4

Port 4 is an 8-bit input/output port with output latch. The P40 to P47 pins can specify the input mode/output mode in 1-bit units with port mode register 4 (PM4). When the P40 to P47 pins are used as input ports, a pull-up resistor can be connected to them in 1-bit units with pull-up resistor option register 4 (PU4).

The interrupt request flag (KRIF) can be set to 1 by detecting falling edges.

This port can also be used as an address/data bus in external memory expansion mode.

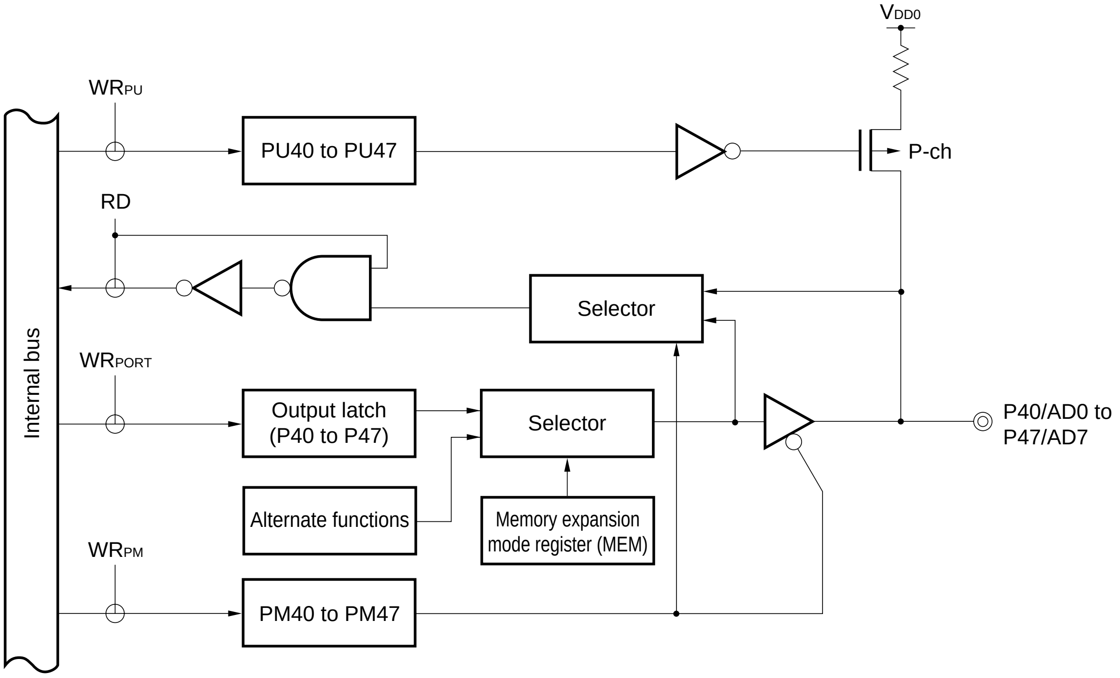
$\overline{\text{RESET}}$ input sets port 4 to input mode.

Figures 6-11 and 6-12 show a block diagram of port 4 and block diagram of the falling edge detection circuit, respectively.

Cautions 1. The internal pull-up resistor is not disconnected even if the external memory expansion mode is set when PU4n = 1 (n = 0-7).

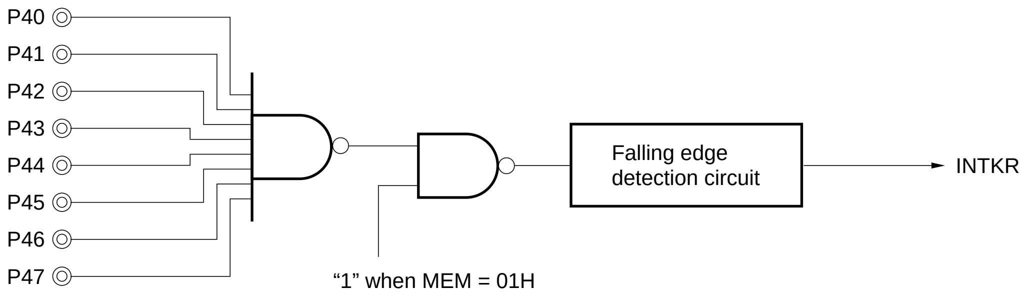
2. When using the falling edge detection interrupt (INTKR), be sure to set the memory expansion mode register (MEM) to 01H.

Figure 6-11. P40 to P47 Block Diagram



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

Figure 6-12. Falling Edge Detection Circuit Block Diagram



6.2.7 Port 5

Port 5 is an 8-bit input/output port with output latch. The P50 to P57 pins can specify the input mode/output mode in 1-bit units with port mode register 5 (PM5). An on-chip pull-up resistor of P50 to P57 pins can be used for them in 1-bit units with pull-up resistor option register 5 (PU5).

Port 5 can drive LEDs directly.

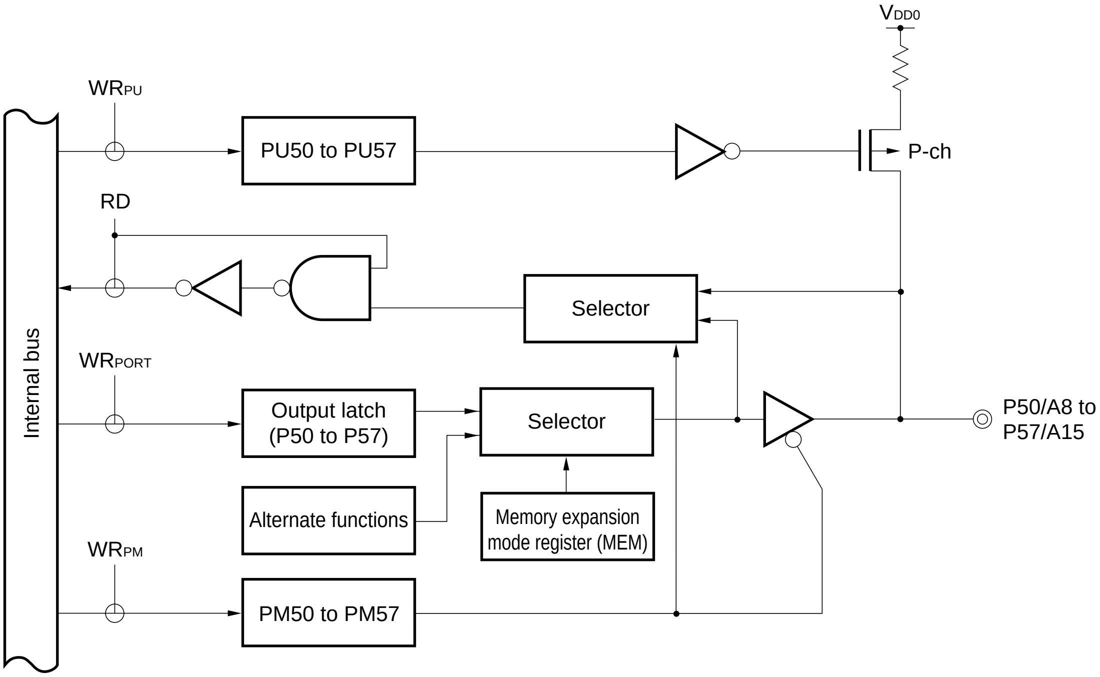
This port can also be used as an address bus in external memory expansion mode.

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

Figure 6-13 shows a block diagram of port 5.

Caution The internal pull-up resistor is not disconnected even if the external memory expansion mode is set when PU5n = 1 (n = 0-7).

Figure 6-13. P50 to P57 Block Diagram



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

6.2.8 Port 6

Port 6 is a 4-bit input/output port with output latch. The P64 to P67 pins can specify the input mode/output mode in 1-bit units with port mode register 6 (PM6).

An on-chip pull-up resistor of P64 to P67 pins can be used for them in 1-bit units with pull-up resistor option register 6 (PU6).

This port can also be used as a control signal output in external memory expansion mode.

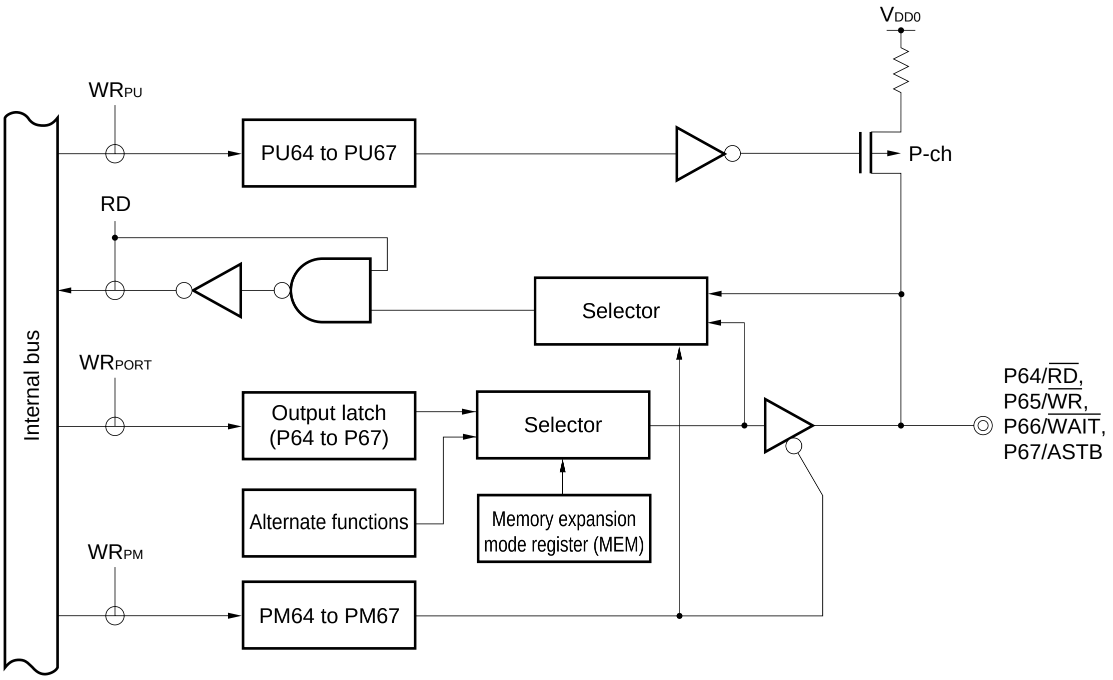
$\overline{\text{RESET}}$ input sets port 6 to input mode.

Figure 6-14 shows a block diagram of port 6.

Cautions 1. The internal pull-up resistor is not disconnected even if the external memory expansion mode is set when PU6n = 1 (n = 4-7).

2. When external wait is not used in external memory expansion mode, P66 can be used as an input/output port.

Figure 6-14. P64 to P67 Block Diagram



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

6.2.9 Port 7

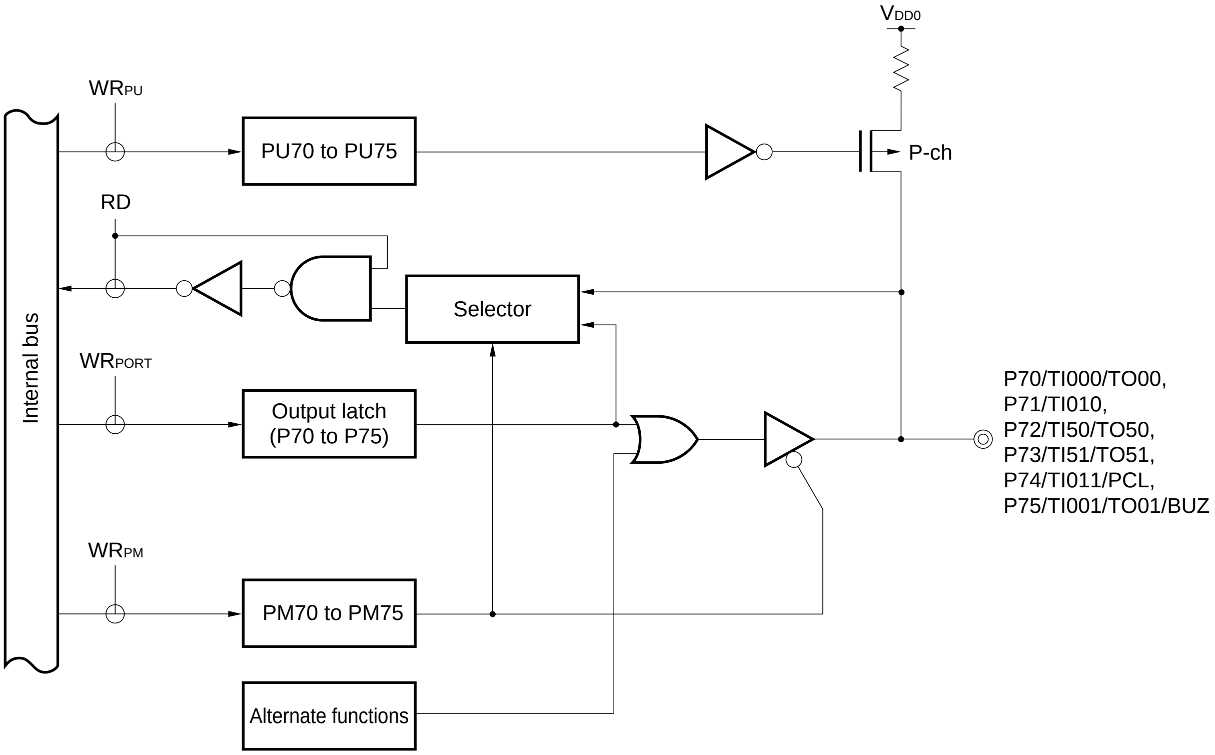
This is a 6-bit input/output port with output latches. Input mode/output mode can be specified bit-wise by means of port mode register 7 (PM7). An on-chip pull-up resistor of P70 to P75 pins can be used as a 1-bit unit by means of pull-up resistor option register 7 (PU7).

This port can also be used as a timer input/output, clock output, and buzzer output.

$\overline{\text{RESET}}$ input sets the input mode.

Figure 6-15 shows a block diagram of port 7.

Figure 6-15. P70 to P75 Block Diagram



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

6.2.10 Port 8

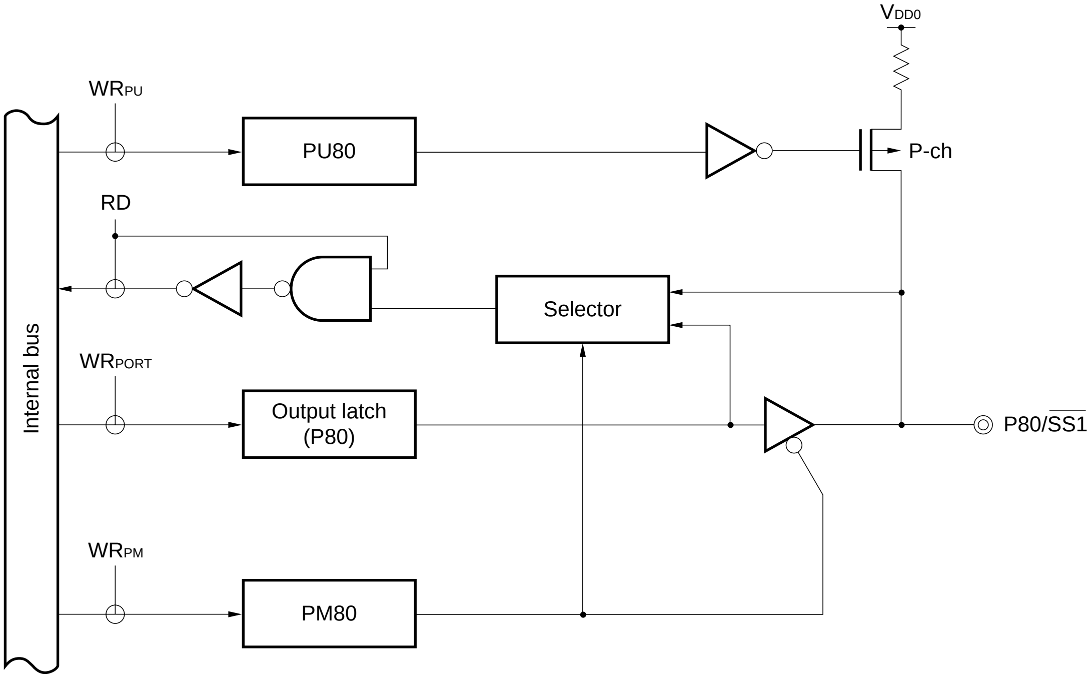
Port 8 is an 8-bit input/output port with output latch. The P80 pin can specify the input mode/output mode in 1-bit units with port mode register 8 (PM8). An on-chip pull-up resistor of P80 pin can be used for them in 1-bit units with pull-up resistor option register 8 (PU8).

This port can also be used as serial interface chip select input.

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

Figure 6-16 shows a block diagram of port 8.

Figure 6-16. P80 Block Diagram



PU : Pull-up resistor option register
PM : Port mode register
RD : Port 8 read signal
WR : Port 8 write signal

6.3 Port Function Control Registers

The following two types of registers control the ports.

- Port mode registers (PM0, PM2 to PM8)
- Pull-up resistor option register (PU0, PU2 to PU8)

(1) Port mode registers (PM0, PM2 to PM8)

These registers are used to set port input/output in 1-bit units.
PM0 and PM2 to PM8 are independently set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets the values of PM0, PM2 to PM8 to FFH.

Cautions 1. Pins P10 and P17 are input-only pins.

2. 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 set to 1 beforehand.

3. If a port has an alternate function pin and it is used as an alternate output function, set the output latches (P0 and P2 through P8) to 0.

Figure 6-17. Port Mode Register (PM0, PM2 to PM8) Format

Address: FF20H At Reset : FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	1	1	1	1	PM03	PM02	PM01	PM00

Address: FF22H At Reset : FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM2	1	1	PM25	PM24	PM23	PM22	PM21	PM20

Address: FF23H At Reset : FFH R/W

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

Address: FF24H At Reset : FFH R/W

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

Address: FF25H At Reset : FFH R/W

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

Address: FF26H At Reset : FFH R/W

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

Address: FF27H At Reset : FFH R/W

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

Address: FF28H At Reset : FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM8	1	1	1	1	1	1	1	PM80

PMmn	Pmn Pin Input/Output Mode Selection (m = 0, 2 to 8: n = 0 to 7)
0	Output mode (Output buffer on)
1	Input mode (Output buffer off)

(2) Pull-up resistor option register (PU0, PU2 to PU8)

This register is used to set whether to use an internal pull-up resistor at each port or not. By setting PU0 and PU2 through PU8, the on-chip pull-up resistors of the port pins corresponding to the bits in PU0 and PU2 through PU8 can be used.

PU0 and PU2 to PU8 are set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the values of PU0, PU2 and PU8 to 00H.

Cautions 1. The P10 and P17 pins do not incorporate a pull-up resistor.

2. Pins P30 to P33 (in μPD780078Y Subseries, P30 and P31 pins) can be used with pull-up resistor by mask option only for mask ROM version.

3. When PUm is set to 1, the on-chip pull-up resistor is connected irrespective of the input/output mode. When using in output mode, therefore, set the bit of PUm to 0 (m = 0, 2 to 8).

Figure 6-18. Pull-Up Resistor Option Register (PU0, PU2 to PU8) Format

Address: FF30H At Reset : 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU0	0	0	0	0	PU03	PU02	PU01	PU00

Address: FF32H At Reset : 00H R/W

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

Address: FF33H At Reset : 00H R/W

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

Address: FF34H At Reset : 00H R/W

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

Address: FF35H At Reset : 00H R/W

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

Address: FF36H At Reset : 00H R/W

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

Address: FF37H At Reset : 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU7	0	0	PU75	PU74	PU73	PU72	PU71	PU70

Address: FF38H At Reset : 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU8	0	0	0	0	0	0	0	PU80

PUmn	Pmn Pin Internal Pull-Up Resistor Selection (m = 0, 2 to 8: n = 0 to 7)
0	On-chip pull-up resistor not used
1	On-chip pull-up resistor used

6.4 Port Function Operations

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

6.4.1 Writing to input/output 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 1-bit memory manipulation instruction, although a single bit is manipulated the port is accessed as an 8-bit unit. Therefore, on 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.

6.4.2 Reading from input/output 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.

6.4.3 Operations on input/output 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 1-bit memory manipulation instruction, although a single bit is manipulated, the port is accessed as an 8-bit unit. Therefore, on 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.

6.5 Selection of Mask Option

The following mask option is provided in the mask ROM version. The flash memory versions have no mask options.

Table 6-6. Comparison between Mask ROM Version and Flash Memory Version

Pin Name	Mask ROM Version	Flash Memory Version
Mask option for pins P30 to P33[Note]	Bit-wise specifiable on-chip pull-up resistors	Cannot specify an on-chip pull-up resistor

Note For μPD780078Y Subseries products, only the P30 and P31 pins can incorporate a pull-up resistor.

CHAPTER 7 CLOCK GENERATOR

7.1 Clock Generator Functions

The clock generator generates the clock to be supplied to the CPU and peripheral hardware. The following two types of system clock oscillators are available.

(1) Main system clock oscillator

This circuit oscillates at frequencies of 1 to 8.38 MHz. Oscillation can be stopped by executing the STOP instruction or setting the processor clock control register (PCC).

(2) Subsystem clock oscillator

The circuit oscillates at a frequency of 32.768 kHz. Oscillation cannot be stopped. If the subsystem clock oscillator is not used, the internal feedback resistor can be disabled by the processor clock control register (PCC). This enables to reduce the power consumption in the STOP mode.

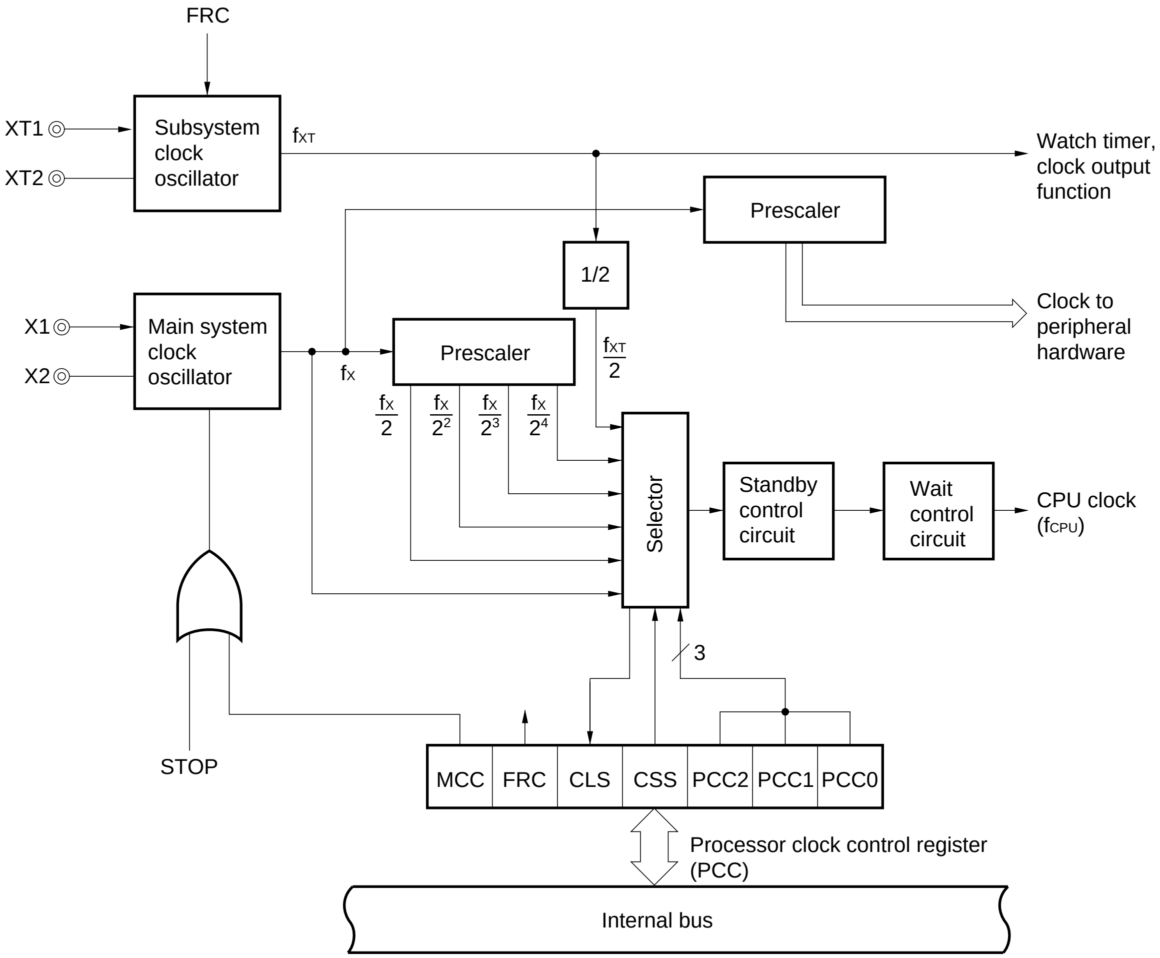
7.2 Clock Generator Configuration

The clock generator consists of the following hardware.

Table 7-1. Clock Generator Configuration

Item	Configuration
Control register	Processor clock control register (PCC)
Oscillators	Main system clock oscillator Subsystem clock oscillator

Figure 7-1. Clock Generator Block Diagram



7.3 Clock Generator Control Register

The clock generator is controlled by the processor clock control register (PCC).

The PCC sets whether to use CPU clock selection, the ratio of division, main system clock oscillator operation/ stop and subsystem clock oscillator internal feedback resistor.

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

$\overline{\text{RESET}}$ input sets the value of PCC to 04H.

Figure 7-2. Subsystem Clock Feedback Resistor

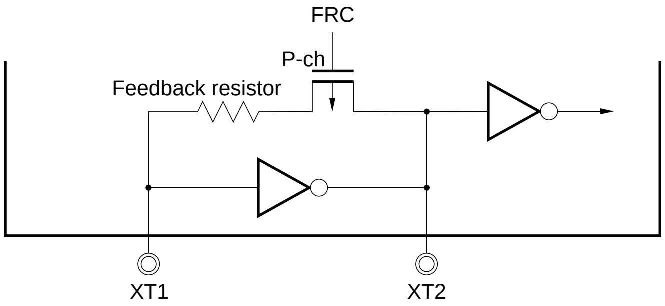


Figure 7-3. Processor Clock Control Register (PCC) Format

Address: FFFBH At Reset: 04H R/W[Note 1]

Symbol	$\boxed{7}$	$\boxed{6}$	$\boxed{5}$	$\boxed{4}$	3	2	1	0
PCC	MCC	FRC	CLS	CSS	0	PCC2	PCC1	PCC0

MCC	Main System Clock Oscillation Control [Note 2]
0	Oscillation possible
1	Oscillation stopped

FRC	Subsystem Clock Feedback Resistor Selection
0	Internal feedback resistor used
1	Internal feedback resistor not used

CLS	CPU Clock Status
0	Main system clock
1	Subsystem clock

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

Notes 1. Bit 5 is Read Only.

2. When the CPU is operating on the subsystem clock, MCC should be used to stop the main system clock oscillation. A STOP instruction should not be used.

Cautions 1. Be sure to set bit 3 to 0.

2. When the external clock is input, MCC should not be set.
This is because the X2 pin is connected to V_{DD1} via a pull-up resistor.

Remarks 1. f_X : Main system clock oscillation frequency

2. f_{XT} : Subsystem clock oscillation frequency

The fastest instructions of μPD780078 and 780078Y Subseries are carried out in two CPU clocks. The relationship of CPU clock (f_{CPU}) and minimum instruction execution time is shown in table 7-2.

Table 7-2. Relationship of CPU Clock and Min. Instruction Execution Time

CPU Clock (f_{CPU})	Min. Instruction Execution Time: 2/(f_{CPU})
f_X	0.24 μs
f_X2	0.48 μs
f_X2^2	0.95 μs
f_X2^3	1.91 μs
f_X2^4	3.81 μs
$f_{XT}2$	122 μs

f_X = 8.38 MHz, f_{XT} = 32.768 kHz

f_X: Main system clock oscillation frequency

f_{XT}: Subsystem clock oscillation frequency

7.4 System Clock Oscillator

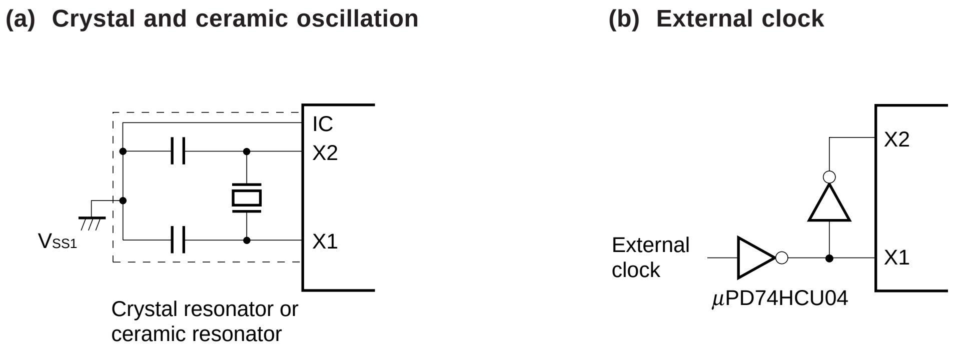
7.4.1 Main system clock oscillator

The main system clock oscillator oscillates with a crystal resonator or a ceramic resonator (8.38 MHz TYP.) connected to the X1 and X2 pins.

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

Figure 7-4 shows an external circuit of the main system clock oscillator.

Figure 7-4. External Circuit of Main System Clock Oscillator



(a) Crystal and ceramic oscillation

(b) External clock

Caution Do not execute the STOP instruction and do not set MCC (bit 7 of processor clock control register (PCC)) to 1 if an external clock is input. This is because when the STOP instruction or MCC is set to 1, the main system clock operation stops and the X2 pin is connected to V_{DD1} via a pull-up resistor.

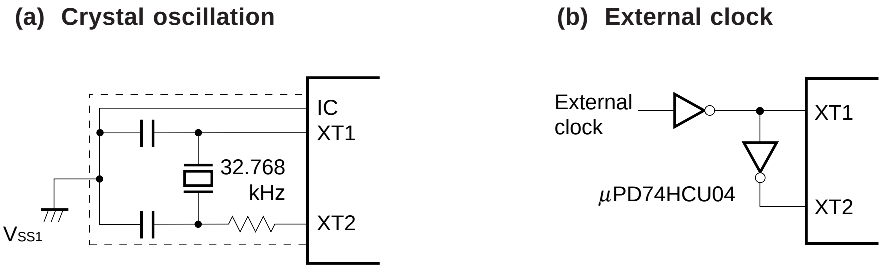
7.4.2 Subsystem clock oscillator

The subsystem clock oscillator oscillates with a crystal resonator (32.768 kHz TYP.) connected to the XT1 and XT2 pins.

External clocks can be input to the main system clock oscillator. In this case, input a clock signal to the XT1 pin and an inversed-phase clock signal to the XT2 pin.

Figure 7-5 shows an external circuit of the subsystem clock oscillator.

Figure 7-5. External Circuit of Subsystem Clock Oscillator


(a) Crystal oscillation

(b) External clock

Cautions are listed on the next page.

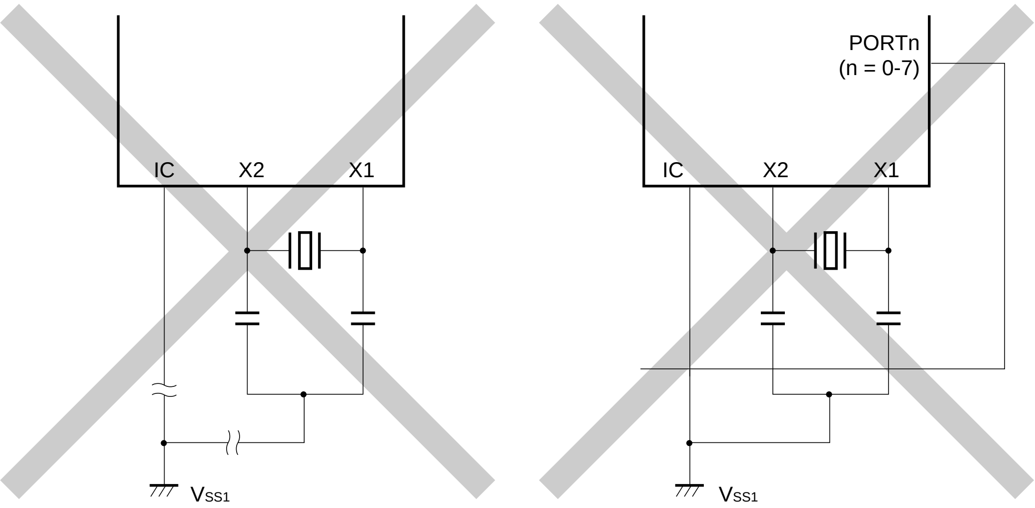
Cautions 1. When using the main system clock oscillator and a subsystem clock oscillator, carry out wiring in the broken line area in Figures 7-4 and 7-5 to prevent any effects from wiring capacitance.

- **Minimize the wiring length.**
- **Do not allow wiring to intersect with other signal lines. Do not route the wiring in the vicinity of a line through which a high-fluctuating current flows.**
- **Always keep the ground of the capacitor of the oscillator at the same potential as V_{SS1}. Do not ground a capacitor to a ground pattern where high-current flows.**
- **Do not fetch signals from the oscillator.**

Take special note of the fact that the subsystem clock oscillator is a circuit with low-level amplification so that current consumption is maintained at low levels.
Figure 7-6 shows examples of incorrect oscillator connection.

Figure 7-6. Examples of Incorrect Oscillator Connection (1/2)

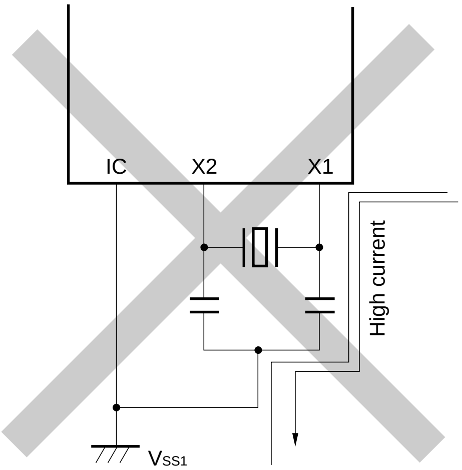
(a) Too long wiring **(b) Crossed signal line**



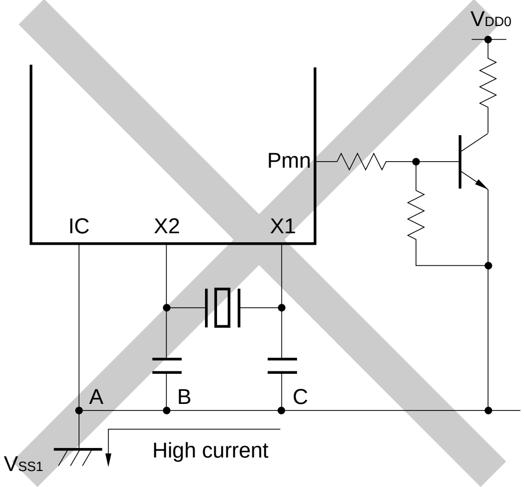
Remark When using a subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Further, insert resistors in series on the side of XT2.

Figure 7-6. Examples of Incorrect Oscillator Connection (2/2)

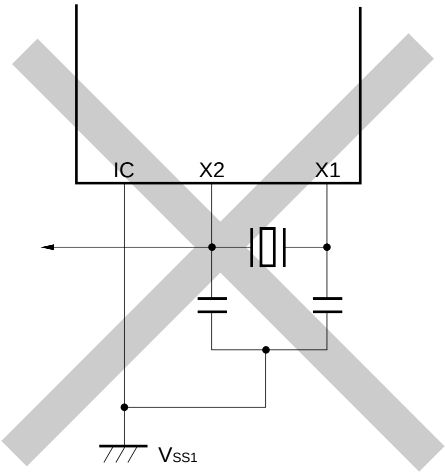
(c) Wiring near high alternating current



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



(e) Signals are fetched



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

Cautions 2. When X2 and XT1 are wired in parallel, the cross-talk noise of X2 may increase with XT1, resulting in malfunctioning.

To prevent that from occurring, it is recommended to wire X2 and XT1 so that they are not in parallel, and to connect the IC pin between X2 and XT1 directly to V_{SS1}.

7.4.3 Scaler

The scaler divides the main system clock oscillator output (f_X) and generates various clocks.

7.4.4 When no subsystem clocks are used

If it is not necessary to use subsystem clocks for low power consumption operations and clock operations, connect the XT1 and XT2 pins as follows.

XT1 : Connect to V_{DD0}
XT2 : Open

In this state, however, some current may leak via the internal feedback resistor of the subsystem clock oscillator when the main system clock stops. To minimize leakage current, the above internal feedback resistor can be removed with bit 6 (FRC) of the processor clock control register (PCC). In this case also, connect the XT1 and XT2 pins as described above.

7.5 Clock Generator Operations

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

- Main system clock f_X
- Subsystem clock f_{XT}
- CPU clock f_{CPU}
- Clock to peripheral hardware

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

(a) Upon generation of $\overline{\text{RESET}}$ signal, the lowest speed mode of the main system clock (3.81 μs @ 8.38-MHz oscillation) is selected (PCC = 04H). Main system clock oscillation stops while low level is applied to $\overline{\text{RESET}}$ pin.

(b) With the main system clock selected, one of the five CPU clock types (0.24 μs. 0.48 μs, 0.95 μs, 1.91 μs, 3.81 μs, @ 8.38-MHz operation) can be selected by setting the PCC.

(c) With the main system clock selected, two standby modes, the STOP and HALT modes, are available. To reduce current consumption in the STOP mode, the subsystem clock feedback resistor can be disconnected to stop the subsystem clock.

(d) The PCC can be used to select the subsystem clock and to operate the system with low-current consumption (122 μs @ 32.768-kHz operation).

(e) With the subsystem clock selected, main system clock oscillation can be stopped with the PCC. The HALT mode can be used. However, the STOP mode cannot be used. (Subsystem clock oscillation cannot be stopped.)

(f) The main system clock is divided and supplied to the peripheral hardware. The subsystem clock is supplied to the watch timer and clock output functions only. Thus the watch function and the clock output function can also be continued in the standby state. However, since all other peripheral hardware operate with the main system clock, the peripheral hardware also stops if the main system clock is stopped. (Except external input clock operation)

7.5.1 Main system clock operations

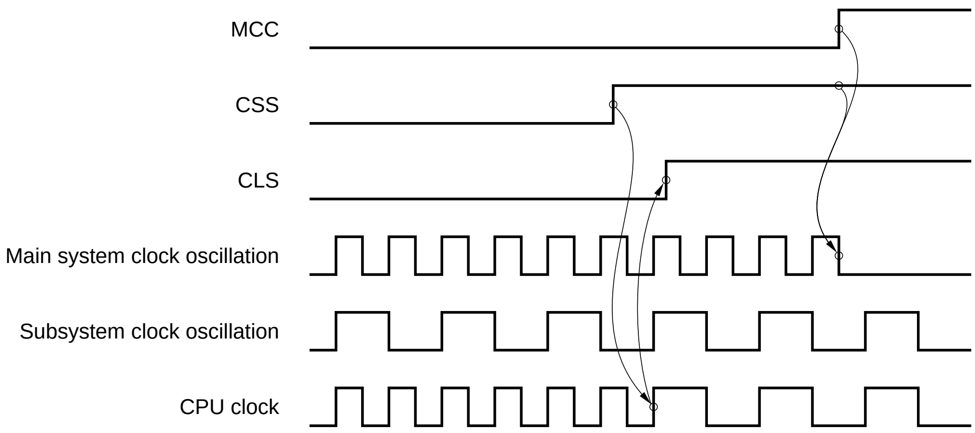
When operated with the main system clock (with bit 5 (CLS) of the processor clock control register (PCC) set to 0), the following operations are carried out by PCC setting.

(a) Because the operation guarantee instruction execution speed depends on the power supply voltage, the minimum instruction execution time can be changed by bits 0 to 2 (PCC0 to PCC2) of the PCC.

(b) If bit 7 (MCC) of the PCC is set to 1 when operated with the main system clock, the main system clock oscillation does not stop. When bit 4 (CSS) of the PCC is set to 1 and the operation is switched to subsystem clock operation (CLS = 1) after that, the main system clock oscillation stops (see **Figure 7-7**).

Figure 7-7. Main System Clock Stop Function (1/2)

(a) Operation when MCC is set after setting CSS with main system clock operation



(b) Operation when MCC is set in case of main system clock operation

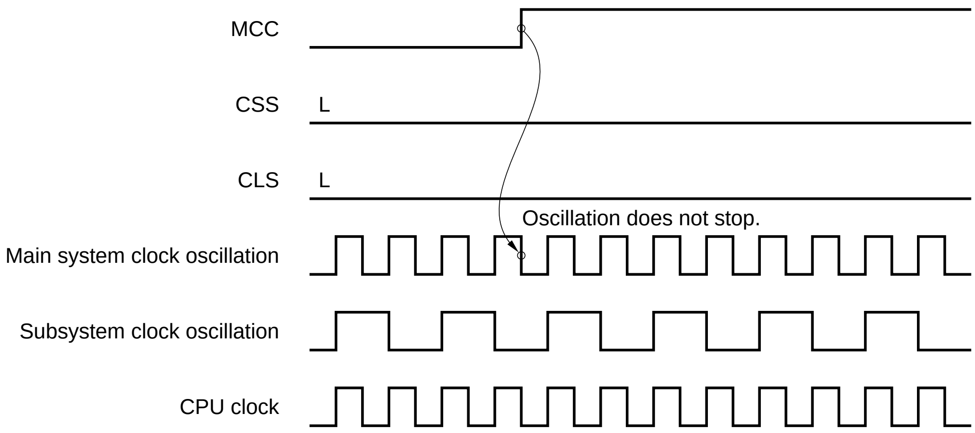
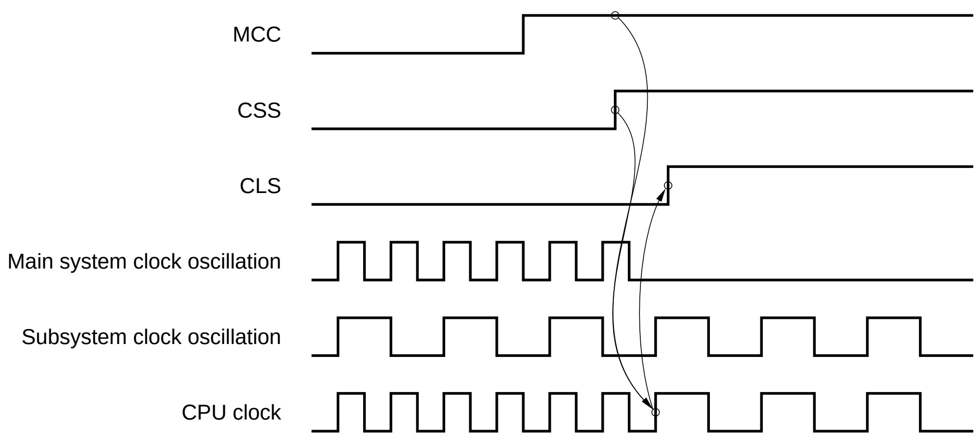

MCC
CSS
L
CLS
L
Oscillation does not stop.
Main system clock oscillation
Subsystem clock oscillation
CPU clock

Figure 7-7. Main System Clock Stop Function (2/2)

(c) Operation when CSS is set after setting MCC with main system clock operation



7.5.2 Subsystem clock operations

When operated with the subsystem clock (with bit 5 (CLS) of the processor clock control register (PCC) set to 1), the following operations are carried out.

(a) The minimum instruction execution time remains constant (122 μs @ 32.768-kHz operation) irrespective of bits 0 to 2 (PCC0 to PCC2) of the PCC.

(b) Watchdog timer counting stops.

Caution Do not execute the STOP instruction while the subsystem clock is in operation.

7.6 Changing System Clock and CPU Clock Settings

7.6.1 Time required for switchover between system clock and CPU clock

The system clock and CPU clock can be switched over by means of bits 0 to 2 (PCC0 to PCC2) and bit 4 (CSS) of the processor clock control register (PCC).

The actual switchover operation is not performed directly after writing to the PCC, but operation continues on the pre-switchover clock for several instructions (see **Table 7-3**).

Determination as to whether the system is operating on the main system clock or the subsystem clock is performed by bit 5 (CLS) of the PCC register.

Table 7-3. Maximum Time Required for CPU Clock Switchover

Set Value before Switchover				Set Value after Switchover					
CSS	PCC2	PCC1	PCC0	CSS PCC2 PCC1 PCC0 0 0 0 0	CSS PCC2 PCC1 PCC0 0 0 0 1	CSS PCC2 PCC1 PCC0 0 0 1 0	CSS PCC2 PCC1 PCC0 0 0 1 1	CSS PCC2 PCC1 PCC0 0 1 0 0	CSS PCC2 PCC1 PCC0 1 × × ×
0	0	0	0		16 instructions	16 instructions	16 instructions	16 instructions	$f_X/2f_{XT}$ instruction (128 instructions)
	0	0	1	8 instructions		8 instructions	8 instructions	8 instructions	$f_X/4f_{XT}$ instruction (64 instructions)
	0	1	0	4 instructions	4 instructions		4 instructions	4 instructions	$f_X/8f_{XT}$ instruction (32 instructions)
	0	1	1	2 instructions	2 instructions	2 instructions		2 instructions	$f_X/16f_{XT}$ instruction (16 instructions)
	1	0	0	1 instruction	1 instruction	1 instruction	1 instruction		$f_X/32f_{XT}$ instruction (8 instructions)
1	×	×	×	1 instruction	1 instruction	1 instruction	1 instruction	1 instruction	

Remarks 1. One instruction is the minimum instruction execution time with the pre-switchover CPU clock.

2. Figures in parentheses are for operation with f_X = 8.38 MHz and f_{XT} = 32.768 kHz.

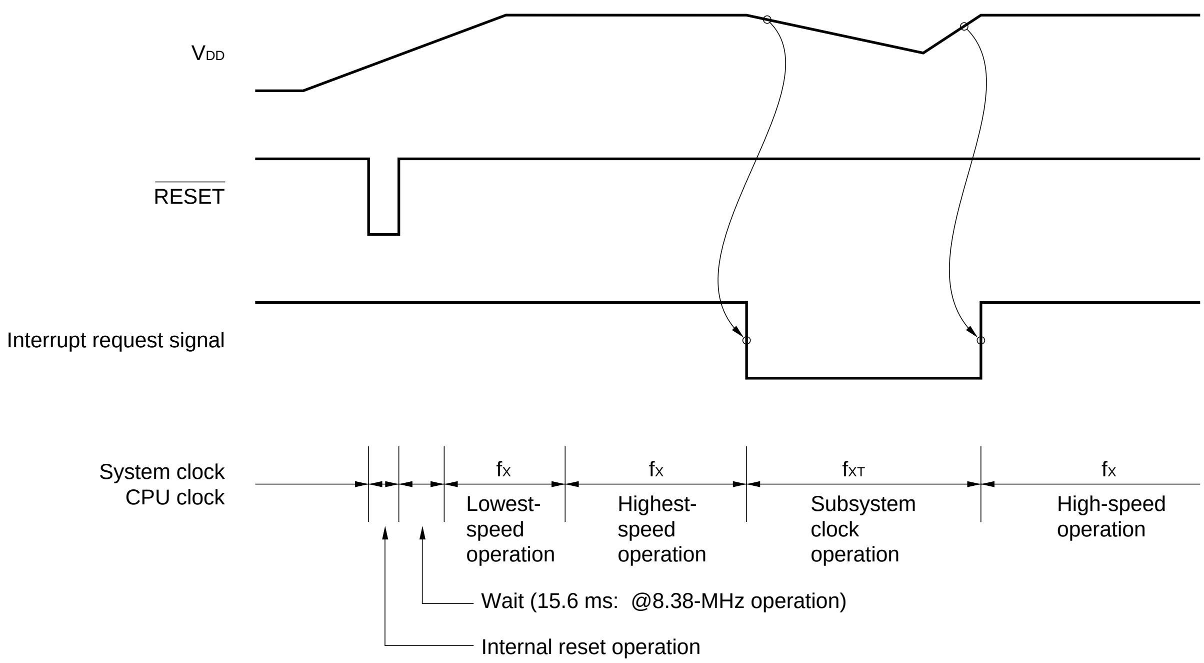
Caution Selection of the CPU clock cycle scaling factor (PCC0 to PCC2) and switchover from the main system clock to the subsystem clock (changing CSS from 0 to 1) should not be set simultaneously.

Simultaneous setting is possible, however, for selection of the CPU clock cycle scaling factor (PCC0 to PCC2) and switch over from the subsystem clock to the main system clock (changing CSS from 1 to 0).

7.6.2 System clock and CPU clock switching procedure

This section describes switching procedure between the system clock and CPU clock.

Figure 7-8. System Clock and 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, main system clock starts oscillation. At this time, oscillation stabilization time ($2^{17}/f_X$) is secured automatically.
After that, the CPU starts executing the instruction at the minimum speed of the main system clock (3.81 μs @ 8.38-MHz operation).

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

<3> Upon detection of a decrease of the V_{DD} voltage due to an interrupt request signal, the main system clock is switched to the subsystem clock (which must be in an oscillation stable state).

<4> Upon detection of V_{DD} voltage reset due to an interrupt, 0 is set to the MCC and oscillation of the main system clock is started. After the lapse of time required for stabilization of oscillation, the PCC is rewritten and the maximum-speed operation is resumed.

Caution When subsystem clock is being operated while the main system clock is stopped, if switching to the main system clock is done again, be sure to switch after securing oscillation stabilization time by software.

CHAPTER 8 16-BIT TIMER/EVENT COUNTERS 00, 01

8.1 Outline of 16-bit Timer/Event Counters 00, 01

The TM0 can be used as an interval timer, PPG output, pulse widths measurement (infrared ray remote control receive function), external event counter, square wave output of any frequency, or one-shot pulse output.

8.2 16-Bit Timer/Event Counters 00, 01 Functions

The 16-bit timer/event counters 00, 01 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 00, 01 generate interrupt request at the preset time interval.

(2) PPG output

The 16-bit timer/event counters 00, 01 can output a square wave whose frequency and output pulse can be set freely.

(3) Pulse width measurement

The 16-bit timer/event counters 00, 01 can measure the pulse width of an externally input signal.

(4) External event counter

The 16-bit timer/event counters 00, 01 can measure the number of pulses of an externally input signal.

(5) Square-wave output

The 16-bit timer/event counters 00, 01 can output a square wave with any selected frequency.

(6) One-shot pulse output

The 16-bit timer/event counters 00, 01 are able to output one-shot pulse which can set any width of output pulse.

8.3 16-Bit Timer/Event Counters 00, 01 Configuration

16-bit timer/event counters 00, 01 consists of the following hardware.

Table 8-1. 16-Bit Timer/Event Counters 00, 01 Configuration

Item	Configuration
Timer/counter	16 bits × 2 (TM0n)
Register	16-bit capture/compare register: 16 bits × 4 (CR00n, CR01n)
Timer output	2 (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 register 7 (PM7)[Note]

Note See **Figure 6-15 P70 to P75 Block Diagram.**

Remark n = 0, 1

Figures 8-1 and 8-2 show a block diagram.

Figure 8-1. 16-Bit Timer/Event Counter 00 Block Diagram

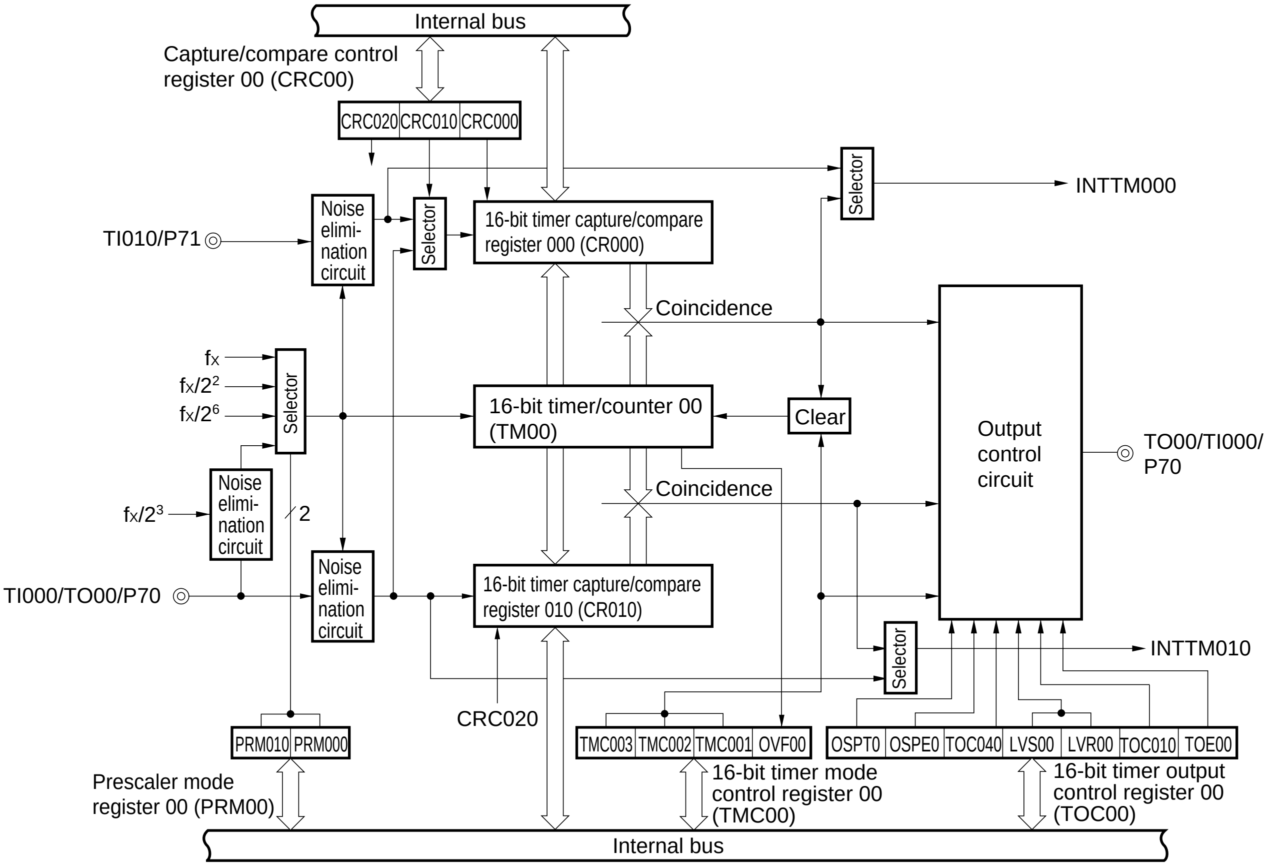
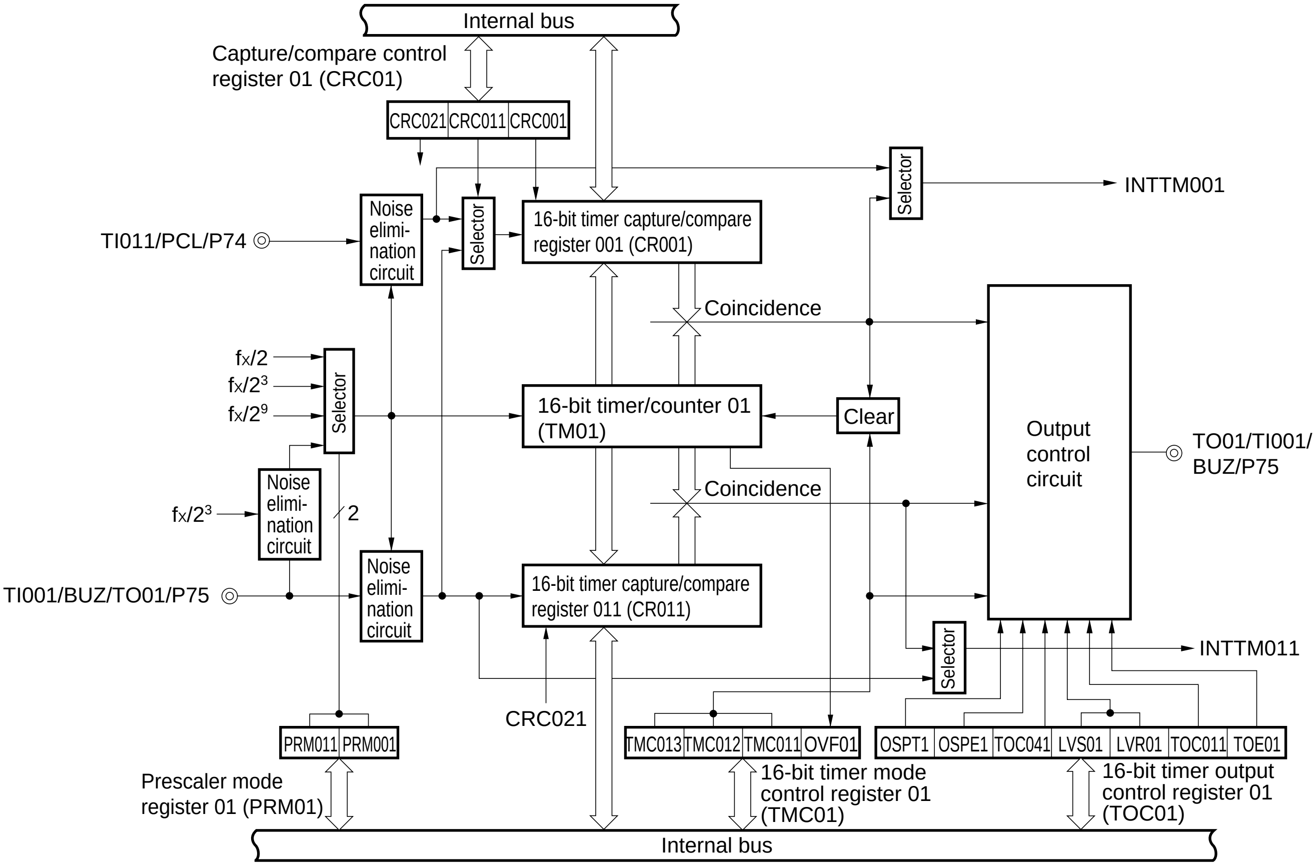

Internal bus
Capture/compare control register 00 (CRC00)
CRC020
CRC010
CRC000
Selector
INTTM000
TI010/P71
Noise elimi-nation circuit
Selector
16-bit timer capture/compare register 000 (CR000)
Coincidence
fx
fx/2^2
fx/2^6
Selector
16-bit timer/counter 00 (TM00)
Clear
Output control circuit
TO00/TI000/P70
Noise elimi-nation circuit
fx/2^3
2
Coincidence
TI000/TO00/P70
Noise elimi-nation circuit
16-bit timer capture/compare register 010 (CR010)
Selector
INTTM010
CRC020
PRM010
PRM000
TMC003
TMC002
TMC001
OVF00
OSPT0
OSPE0
TOC040
LVS00
LVR00
TOC010
TOE00
Prescaler mode register 00 (PRM00)
16-bit timer mode control register 00 (TMC00)
16-bit timer output control register 00 (TOC00)
Internal bus

Figure 8-2. 16-Bit Timer/Event Counter 01 Block Diagram


Internal bus
Capture/compare control register 01 (CRC01)
CRC021
CRC011
CRC001
Selector
INTTM001
Noise elimination circuit
Selector
16-bit timer capture/compare register 001 (CR001)
TI011/PCL/P74
Coincidence
fx/2
fx/2^3
fx/2^9
Selector
16-bit timer/counter 01 (TM01)
Clear
Output control circuit
TO01/TI001/BUZ/P75
Noise elimination circuit
fx/2^3
2
Coincidence
Noise elimination circuit
TI001/BUZ/TO01/P75
16-bit timer capture/compare register 011 (CR011)
Selector
INTTM011
CRC021
PRM011
PRM001
TMC013
TMC012
TMC011
OVF01
OSPT1
OSPE1
TOC041
LVS01
LVR01
TOC011
TOE01
Prescaler mode register 01 (PRM01)
16-bit timer mode control register 01 (TMC01)
16-bit timer output control register 01 (TOC01)
Internal bus

(1) 16-bit timer counter 0n (TM0n)

TM0n is a 16-bit read-only register that counts count pulses.

The counter is incremented in synchronization with the rising edge of an 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 valid edge of TI00n is input in the clear & start mode by inputting valid edge of TI00n
<4> If TM0n and CR00n coincide with each other in the clear & start mode on coincidence between TM0n and CR00n
<5> If OSPTn is set in the one-shot pulse output mode

(2) 16-bit timer capture/compare register 00n (CR00n)

CR00n is a 16-bit register which has the functions of both a capture register and a compare register. Whether it is used as a capture register or as a compare register is set by bit 0n (CRC00n) of capture/compare control register 0 (CRC0n).

- **When CR00n is used as a compare register**

  The value set in the 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 which holds the interval time then TM0n is set to interval timer operation, and as the register which sets the pulse width in the PWM operating mode.

- **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 means of prescaler mode register 0n (PRM0n).

  If CR00n is specified as a capture register and capture trigger is specified to be the valid edge of the TI00n pin, the situation is as shown in Table 8-2. On the other hand, when capture trigger is specified to be the valid edge of the TI01n pin, the situation is as shown in Table 8-3.

Table 8-2. TI00n Pin Valid Edge and CR00n, CR01n Capture Trigger

ES01n	ES00n	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

Table 8-3. TI01n Pin Valid Edge and CR00n Capture Trigger

ES11n	ES10n	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

Remark n = 0, 1

CR00n is set by a 16-bit memory manipulation instruction.
After $\overline{\text{RESET}}$ input, the value of CR00n is undefined.

Cautions 1. Set a value other than 0000H in CR00n. This means 1-pulse count operation cannot be performed when CR00n is used as an event counter. However, in the free-running mode and in the clear mode using the valid edge of TI00n, if 0000H is set to CR00n, an interrupt request (INTTM00n) is generated following overflow (FFFFH).

2. If the new value of CR00n is less than the value of 16-bit timer counter n (TM0n), TM0n continues counting, overflows, and then start counting from 0 again. If the new value of CR00n is less than the old value, therefore, the timer must be restarted after the value of CR00n is changed.

3. When P70 is used as the valid edge of TI000, it cannot be used as timer output (TO00). Moreover, when P70 is used as TO00, it cannot be used as the valid edge of TI000.

4. When P75 is used as the valid edge of TI001, it cannot be used as timer output (TO01). Moreover, when P75 is used as TO01, it cannot be used as the valid edge of TI001.

(3) 16-bit capture/compare register 01n (CR01n)

CR01n is a 16-bit register which has 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 (CRC02n) 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). (Refer to **Table 8-3**).

CR01n is set by a 16-bit memory manipulation instruction.
After $\overline{\text{RESET}}$ input, the value of CR01n is undefined.

Caution Set other than 0000H to CR01n. This means 1-pulse count operation cannot be performed when CR01n is used as the event counter. However, in the free-running mode and in the clear mode using the valid edge of TI00n, if 0000H is set to CR01n, an interrupt request (INTTM01n) is generated following overflow (FFFFH).

Remark n = 0, 1

8.4 Registers to Control 16-Bit Timer/Event Counters 00, 01

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

- 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 register 7 (PM7)

(1) 16-bit timer mode control register 0n (TMC0n: n = 0, 1)

This register sets the 16-bit timer operating mode, the 16-bit timer counter 0n (TM0n) clear mode, and output timing, and detects an overflow.

TMC0n is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the value of TMC0n to 00H.

Caution **The 16-bit timer counter 0n (TM0n) starts operation at the moment a value other than 0, 0 (operation stop mode) is set in TMC02n to TMC03n, respectively. Set 0, 0 in TMC02n to TMC03n to stop the operation.**

Figure 8-3. 16-Bit Timer Mode Control Register 00 (TMC00) Format

Address FF60H At Reset: 00H R/W

Symbol	7	6	5	4	$\boxed{3}$	$\boxed{2}$	1	$\boxed{0}$
TMC00	0	0	0	0	TMC030	TMC020	TMC010	OVF00

TMC003	TMC002	TMC001	Operating Mode and Clear Mode Selection	TO00 Output Timing Selection	Interrupt Request Generation
0	0	0	Operation stop (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	—	
1	0	1			
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	16-Bit Timer Counter 00 (TM00) Overflow Detection
0	Overflow not detected
1	Overflow detected

Cautions 1. Timer operation must be stopped before writing to bits other than the OVF00 flag.

2. Set the valid edge of the TI000/TO00/P70 pin with prescaler mode register 00 (PRM00).

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

Remarks 1. TO00 : 16-bit timer/event counter output pin

2. TI000 : 16-bit timer/event counter input pin

3. TM00 : 16-bit timer counter 0

4. CR000 : 16-bit timer capture/compare register 000

5. CR010 : 16-bit timer capture/compare register 010

Figure 8-4. 16-Bit Timer Mode Control Register 01 (TMC01) Format

Address FF64H At Reset: 00H R/W

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

TMC013	TMC012	TMC011	Operating Mode and Clear Mode Selection	TO01 Output Timing Selection	Interrupt Request Generation
0	0	0	Operation stop (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	—	
1	0	1			
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	16-Bit Timer Counter 01 (TM01) Overflow Detection
0	Overflow not detected
1	Overflow detected

Cautions 1. Timer operation must be stopped before writing to bits other than the OVF01 flag.

2. Set the valid edge of the TI001/TO01/BUZ/P75 pin with prescaler mode register 01 (PRM01).

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

Remarks 1. TO01 : 16-bit timer/event counter output pin

2. TI001 : 16-bit timer/event counter input pin

3. TM01 : 16-bit timer counter 01

4. CR001 : 16-bit timer capture/compare register 001

5. CR011 : 16-bit timer capture/compare register 011

(2) Capture/compare control register 0n (CRC0n: n = 0, 1)

This register controls the operation of the 16-bit capture/compare registers (CR00n, CR01n).

CRC0n is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the value of CRC0 to 00H.

Figure 8-5. Capture/Compare Control Register 00 (CRC00) Format

Address: FF62H At Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CRC00	0	0	0	0	0	CRC020	CRC010	CRC000

CRC020	CR010 Operating Mode Selection
0	Operates as compare register
1	Operates as capture register

CRC010	CR000 Capture Trigger Selection
0	Captures on valid edge of TI010
1	Captures on valid edge of TI000 by reverse phase

CRC000	CR000 Operating 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 on a match between TM00 and CR000 is selected with the 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 have been selected as the valid edges of TI000, capture is not performed.

4. To ensure the reliability of the capture operation, the capture trigger requires a pulse two times longer than the count clock selected by prescaler mode register 00 (PRM00).

Figure 8-6. Capture/Compare Control Register 01 (CRC01) Format

Address: FF66H At Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CRC01	0	0	0	0	0	CRC021	CRC011	CRC001

CRC021	CR011 Operating 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 valid edge of TI001 by reverse phase

CRC001	CR001 Operating 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 on a match between TM01 and CR001 is selected with the 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 have been selected as the valid edges of TI001, capture is not performed.**
4. **To ensure the reliability of the capture operation, the capture trigger requires a pulse two times longer than the count clock selected by prescaler mode register 01 (PRM01).**

(3) 16-bit timer output control register 0n (TOC0n: n = 0, 1)

This register controls the operation of the 16-bit timer/event counter output control circuit. It sets R-S type flip-flop (LV0n) setting/resetting, output inversion enabling/disabling, 16-bit timer/event counter timer output enabling/disabling, one-shot pulse output operation enabling/disabling, and output trigger for a one-shot pulse by software.

TOC0n is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the value of TOC0 to 00H.

Figure 8-7. 16-Bit Timer Output Control Register 00 (TOC00) Format

Address: FF63H At Reset: 00H R/W

Symbol	7	$\boxed{6}$	$\boxed{5}$	4	$\boxed{3}$	$\boxed{2}$	1	$\boxed{0}$
TOC00	0	OSPT0	OSPE0	TOC040	LVS00	LVR00	TOC010	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 mode
1	One-shot pulse output mode**Note**

TOC040	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 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

TOC010	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 Output Control
0	Output disabled (Output set to level 0)
1	Output enabled

Note One-shot pulse output mode operates normally only in the free-running mode.

Cautions
1. **Timer operation must be stopped before setting TOC00.**
2. **If LVS00 and LVR00 are read after data is set, they will be 0.**
3. **OSPT0 is cleared automatically after data setting, and will therefore be 0 if read.**
4. **Do not set OSPT0 to 1 in any mode other than one-shot pulse output.**
5. **The one-shot pulse output by the external trigger cannot be used because the input pin of the external trigger is multiplexed with the output pin for the one-shot pulse.**

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

Address: FF67H At Reset: 00H R/W

Symbol	7	$\boxed{6}$	$\boxed{5}$	4	$\boxed{3}$	$\boxed{2}$	1	$\boxed{0}$
TOC01	0	OSPT1	OSPE1	TOC041	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 mode
1	One-shot pulse output mode[Note]

TOC041	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 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 Output Control
0	Output disabled (Output set to level 0)
1	Output enabled

Note One-shot pulse output mode operates normally only in the free-running mode.

Cautions 1. Timer operation must be stopped before setting TOC01.

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

3. OSPT1 is cleared automatically after data setting, and will therefore be 0 if read.

4. Do not set OSPT1 to 1 in any mode other than one-shot pulse output.

5. The one-shot pulse output by the external trigger cannot be used because the input pin of the external trigger is multiplexed with the output pin for the one-shot pulse.

(4) Prescaler mode register 0n (PRM0n: n = 0, 1)

This register is used to set 16-bit timer counter 0n (TM0n) count clock and TI00n, TI01n input valid edges.

PRM0n is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the value of PRM0 to 00H.

Figure 8-9. Prescaler Mode Register 00 (PRM00) Format

Address: FF61H At Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PRM00	ES110	ES100	ES010	ES000	0	0	PRM010	PRM000

ES110	ES100	TI010 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES010	ES000	TI000 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

PRM010	PRM000	Count Clock Selection
0	0	f_X (8.38 MHz)
0	1	$f_X/2^2$ (2.09 MHz)
1	0	$f_X/2^6$ (131 kHz)
1	1	TI000 valid edge[Note]

Note The external clock requires a pulse two times longer than internal count clock ($f_X/2^3$).

Cautions 1. If the valid edge of TI000 is to be set to the count clock, do not set the clear/start mode and the capture trigger at the valid edge of TI000.
Moreover, do not use the P70/TI000/TO00 pins as timer outputs (TO00).

2. Always set data to PRM00 after stopping the timer operation.

3. If the TI000 or TI010 pin is high level immediately after system reset, the rising edge is immediately detected after the rising edge or both the rising and falling edges are set as the valid edge(s) of the TI000 pin or TI010 pin to enable the operation of the 16-bit timer counter 00 (TM00). Please be careful when pulling up the TI000 pin or the TI010 pin. However, when re-enabling operation after the operation has been stopped once, the rising edge is not detected.

Remarks 1. f_X: Main system clock oscillation frequency

2. TI000, TI010: 16-bit timer/event counter input pin

3. Figures in parentheses are for operation with f_X = 8.38 MHz.

Figure 8-10. Prescaler Mode Register 01 (PRM01) Format

Address: FF65H At Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PRM01	ES111	ES101	ES011	ES001	0	0	PRM011	PRM001

ES111	ES101	TI011 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES011	ES001	TI001 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

PRM011	PRM001	Count Clock Selection
0	0	$f_X/2$ (4.19 MHz)
0	1	$f_X/2^3$ (1.05 MHz)
1	0	$f_X/2^9$ (16.37 kHz)
1	1	TI001 valid edge[Note]

Note The external clock requires a pulse two times longer than internal count clock ($f_X/2^3$).

Cautions 1. If the valid edge of TI001 is to be set to the count clock, do not set the clear/start mode and the capture trigger at the valid edge of TI001.
Moreover, do not use the P75/TI001/TO01/BUZ pins as timer outputs (TO01).

2. Always set data to PRM0 after stopping the timer operation.

3. If the TI001 or TI011 pin is high level immediately after system reset, the rising edge is immediately detected after the rising edge or both the rising and falling edges are set as the valid edge(s) of the TI001 pin or TI011 pin to enable the operation of the 16-bit timer counter 01 (TM01). Please be careful when pulling up the TI001 pin or the TI011 pin. However, when re-enabling operation after the operation has been stopped once, the rising edge is not detected.

Remarks 1. f_X: Main system clock oscillation frequency

2. TI001, TI011: 16-bit timer/event counter input pin

3. Figures in parentheses are for operation with f_X = 8.38 MHz.

(5) Port mode register 7 (PM7)

This register sets port 7 input/output in 1-bit units.

When using the P70/TO00/TI000 or P75/TO01/TI001/BUZ pin for timer output, set PM70 or PM75 and the output latch of P70 or P75 to 0.

PM7 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the value of PM7 to FFH.

Figure 8-11. Port Mode Register 7 (PM7) Format

Address: FF27H At 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 Input/Output Mode Selection (n = 0 to 5)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

8.5 16-Bit Timer/Event Counters 00, 01 Operations

8.5.1 Interval timer operation

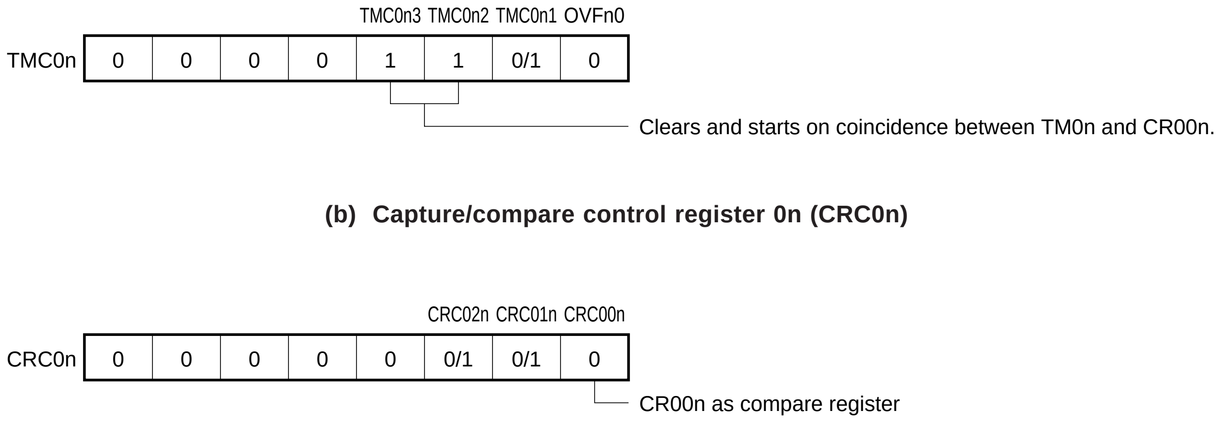
Setting the 16-bit timer mode control register 0n (TMC0n) and capture/compare control register 0n (CRC0n) as shown in Figure 8-12 allows operation as an interval timer. Interrupt request is generated repeatedly using the count value set in 16-bit timer capture/compare register 00n (CR00n) beforehand as the interval.

When the count value of the 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.

Count clock of the 16-bit timer/event counter can be selected with bits 0 to 1 (PRM00n, PRM01n) of the prescaler mode register 0n (PRM0n).

See **8.6 16-Bit Timer/Event Counters 00, 01 Operating Precautions (2) 16-bit compare register setting** about the operation when the compare register value is changed during timer count operation.

Figure 8-12. Control Register Settings for Interval Timer Operation



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

					TMC0n3	TMC0n2	TMC0n1	OVFn0
TMC0n	0	0	0	0	1	1	0/1	0

Clears and starts on coincidence between TM0n and CR00n.

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

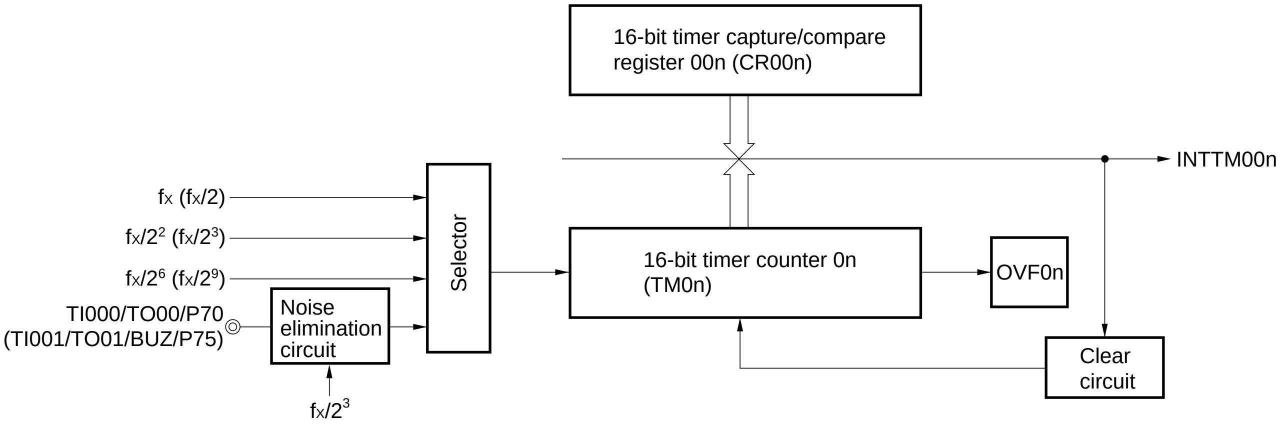
						CRC02n	CRC01n	CRC00n
CRC0n	0	0	0	0	0	0/1	0/1	0

CR00n as compare register

Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with the interval timer. See **Figures 8-3** to **8-6**.

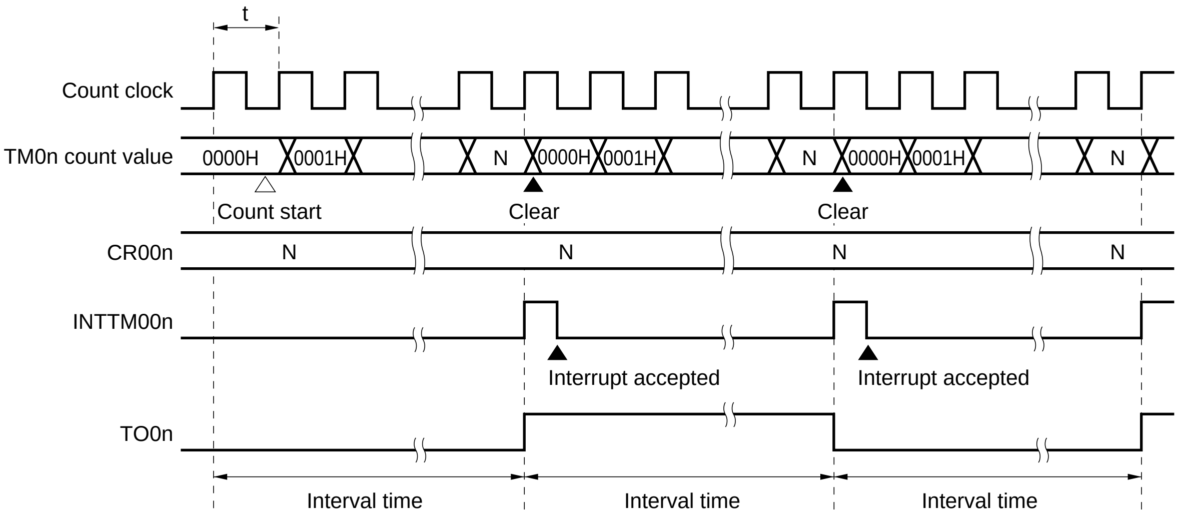
n = 0, 1

Figure 8-13. Interval Timer Configuration Diagram



Remark Values outside parentheses apply to the 16-bit timer/event counter 00, and the values in parentheses apply to the 16-bit timer/event counter 01.

Figure 8-14. Timing of Interval Timer Operation



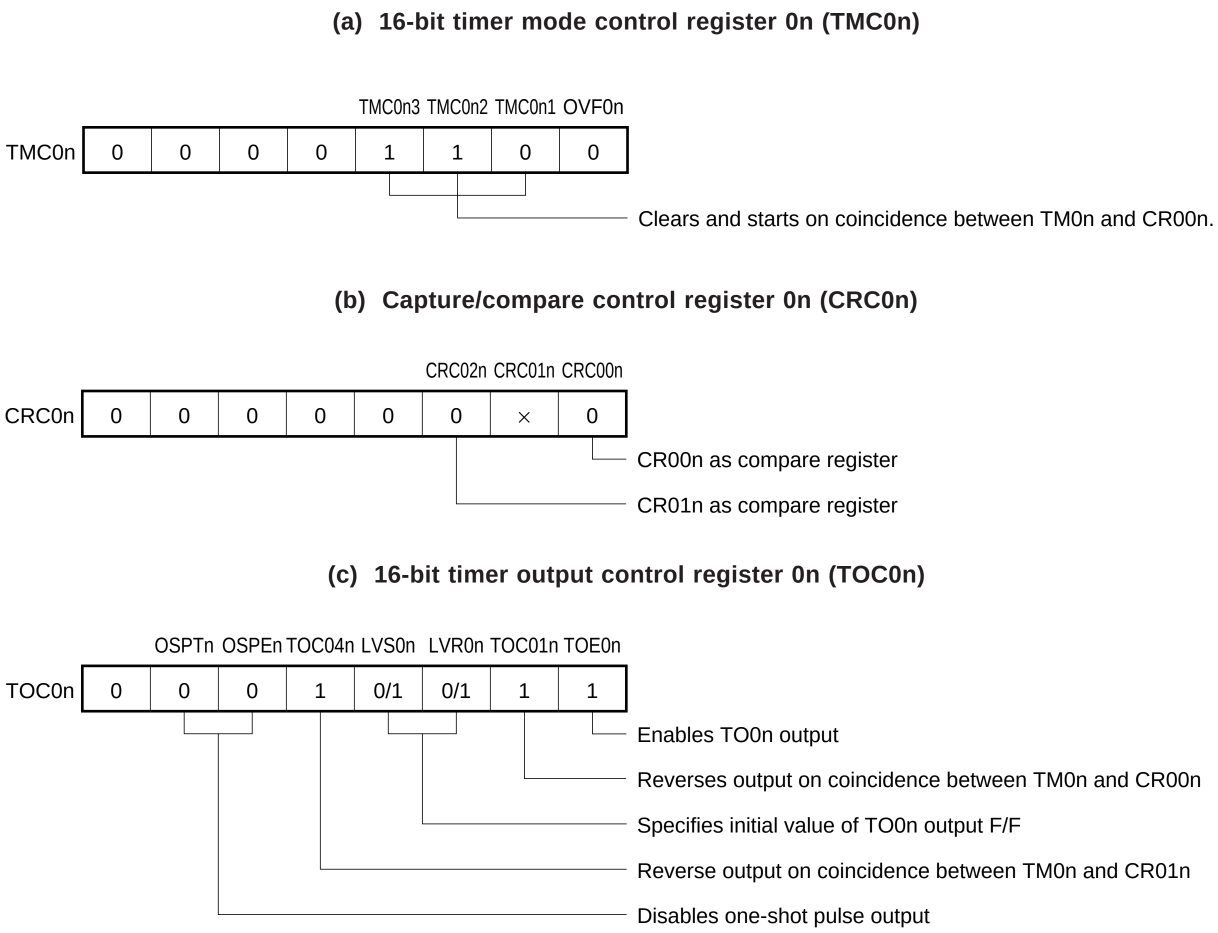
Remark Interval time = (N + 1) × t
N = 0000H to FFFFH
n = 0, 1

8.5.2 PPG output operation

Setting the 16-bit timer mode control register 0n (TMC0n) and capture/compare control register 0n (CRC0n) as shown in Figure 8-10 allows operation as PPG (Programmable Pulse Generator) output.

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

Figure 8-15. 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 coincidence between TM0n and CR00n.

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

						CRC02n	CRC01n	CRC00n
CRC0n	0	0	0	0	0	0	×	0

CR00n as compare register
CR01n as compare register

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

		OSPTn	OSPEn	TOC04n	LVS0n	LVR0n	TOC01n	TOE0n
TOC0n	0	0	0	1	0/1	0/1	1	1

Enables TO0n output
Reverses output on coincidence between TM0n and CR00n
Specifies initial value of TO0n output F/F
Reverse output on coincidence between TM0n and CR01n
Disables one-shot pulse output

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

2. The cycle of the pulse generated through PPG output (CR00n setting value + 1) has a duty of (CR01n setting value + 1)/(CR00n setting value + 1).

Remark × : Don't care
n = 0, 1

8.5.3 Pulse width measurement operations

It is possible to measure the pulse width of the signals input to the TI00n pin and TI01n pin using the 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.

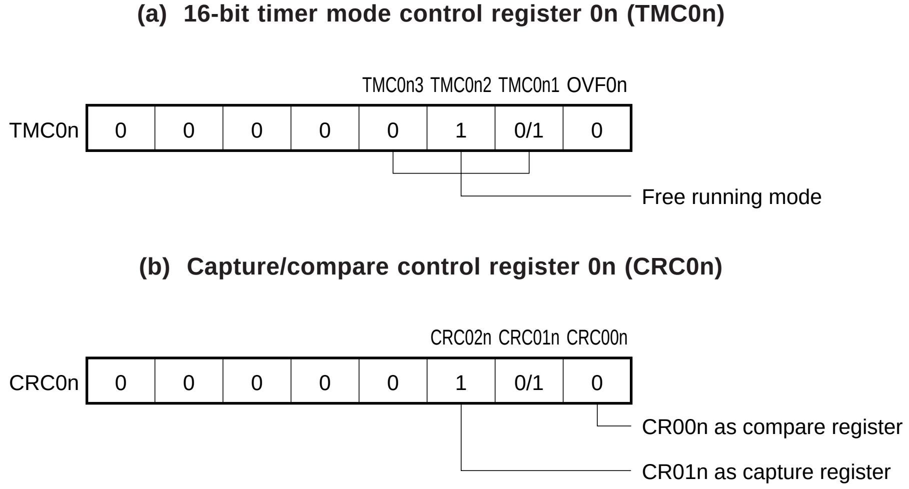
(1) Pulse width measurement with free-running counter and one capture register

When the 16-bit timer counter 0n (TM0n) is operated in free-running mode (see register settings in **Figure 8-16**), 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 timer capture/compare register 01n (CR01n) and an external interrupt request signal (INTTM01n) is set.

Any of three edge can be selected—rising, falling, or both edges—specified by means of bits 4 and 5 (ES00n and ES01n) of PRM0n.

For valid edge detection, sampling is performed at the count clock selected by PRM0n), and a capture operation is only performed when a valid level is detected twice, thus eliminating noise with a short pulse width.

Figure 8-16. Control Register Settings for Pulse Width Measurement with Free-Running Counter and One Capture Register



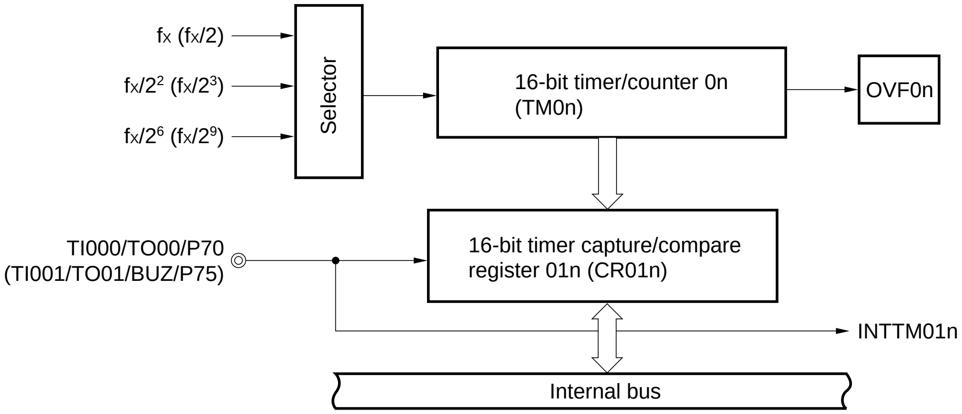
(a) 16-bit timer mode control register 0n (TMC0n)

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

Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. See **Figures 8-3** to **8-6**.

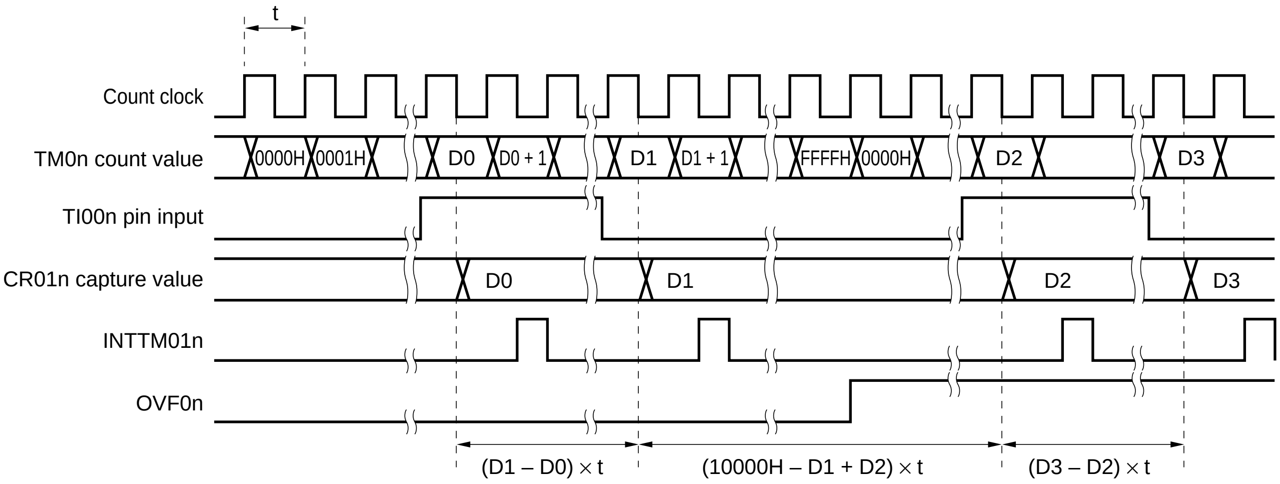
n = 0, 1

Figure 8-17. Configuration Diagram for Pulse Width Measurement by Free-Running Counter



Remark Values outside parentheses apply to the 16-bit timer/event counter 00, and the values in parentheses apply to the 16-bit timer/event counter 01.

Figure 8-18. 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 the 16-bit timer counter 0n (TM0n) is operated in free-running mode (see register settings in **Figure 8-19**), it is possible to simultaneously measure the pulse widths of the two signals input to the TI00n pin and the TI01n pin.

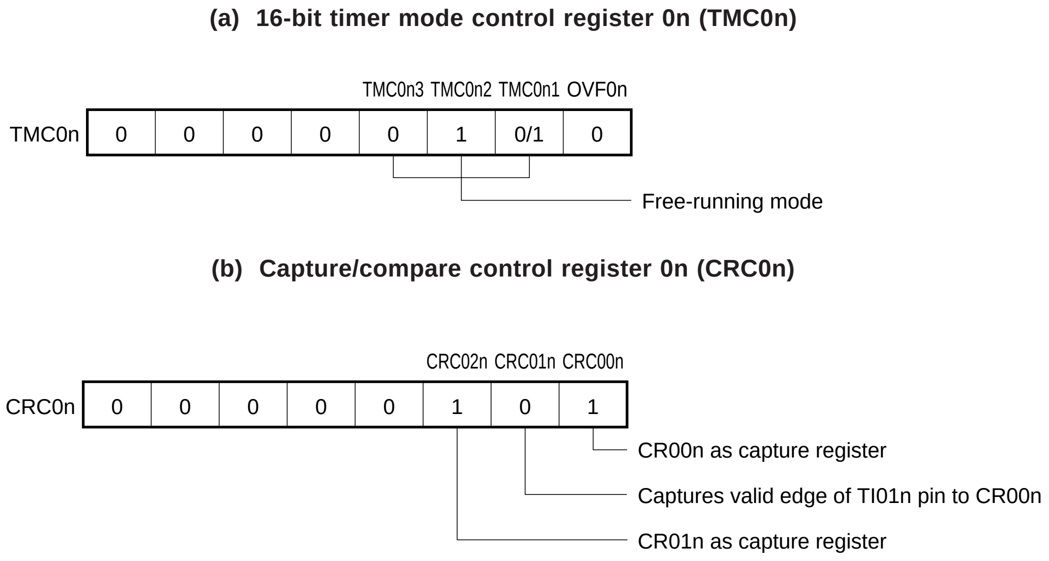
When the edge specified by bits 4 and 5 (ES00n and ES01n) of prescaler mode register 0n (PRM0n) is input to the TI00n pin, the value of TM0n is taken into 16-bit timer capture/compare register 01n (CR01n) and an interrupt request signal (INTTM01n) is set.

Also, when the edge specified by bits 6 and 7 (ES10n and ES11n) of PRM0n is input to the TI01n pin, the value of TM0n is taken into 16-bit timer capture/compare register 00n (CR00n) and an external interrupt request signal (INTTM00n) is set.

Any of three edge can be selected—rising, falling, or both edges—as the valid edges for the TI00n pin and the TI01n pin specified by means of bits 4 and 5 (ES00n and ES01n) and bits 6 and 7 (ES10n and ES11n) of PRM0n, respectively.

For valid edge detection of TI00n and TI01n pins, sampling is performed at the interval selected by means of the prescaler mode register 0n (PRM0n), and a capture operation is only performed when a valid level is detected twice, thus eliminating noise with a short pulse width.

Figure 8-19. 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)

TMC0n3 TMC0n2 TMC0n1 OVF0n
TMC0n
0 0 0 0 0 1 0/1 0
Free-running mode
CRC02n CRC01n CRC00n
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

Remark 0/1: When these bits are reset to 0 or set to 1, the other functions can be used along with the pulse width measurement function. For details, see **Figures 8-3** and **8-4**.

n = 0, 1

- **Capture operation (Free-Running mode)**

  Capture register operation in capture trigger input is shown.

Figure 8-20. CR01n Capture Operation with Rising Edge Specified

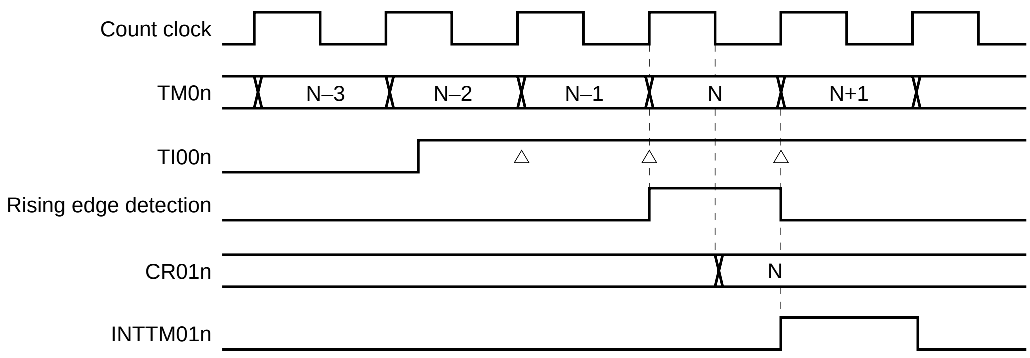
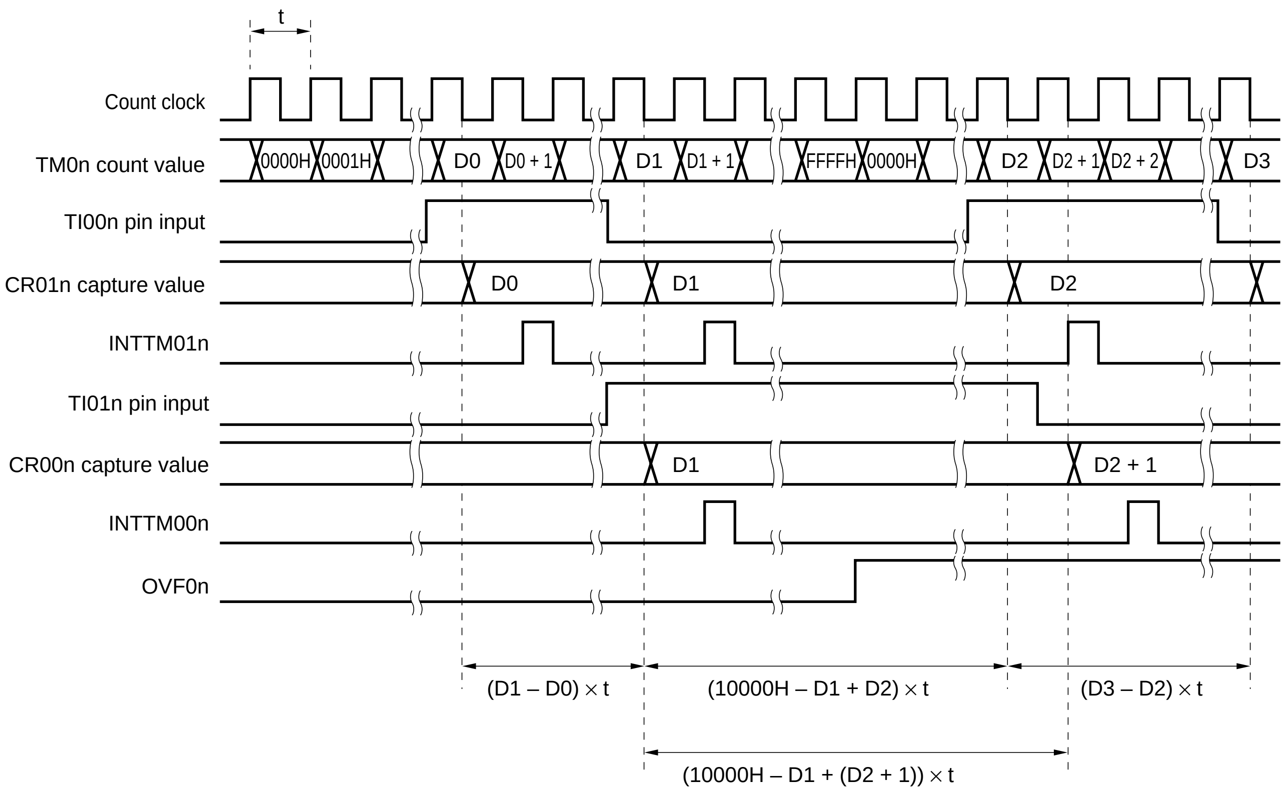


Figure 8-21. Timing of Pulse Width Measurement Operation with Free-Running Counter (with Both Edges Specified)



Remark n = 0, 1

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

When the 16-bit timer counter 0n (TM0n) is operated in free-running mode (see register settings in **Figure 8-22**), 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 (ES00n and ES01n) of prescaler mode register 0n (PRM0n) is input to the TI00n pin, the value of TM0n is taken into 16-bit timer capture/compare register 01n (CR01n) and an interrupt request signal (INTTM01n) is set.

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

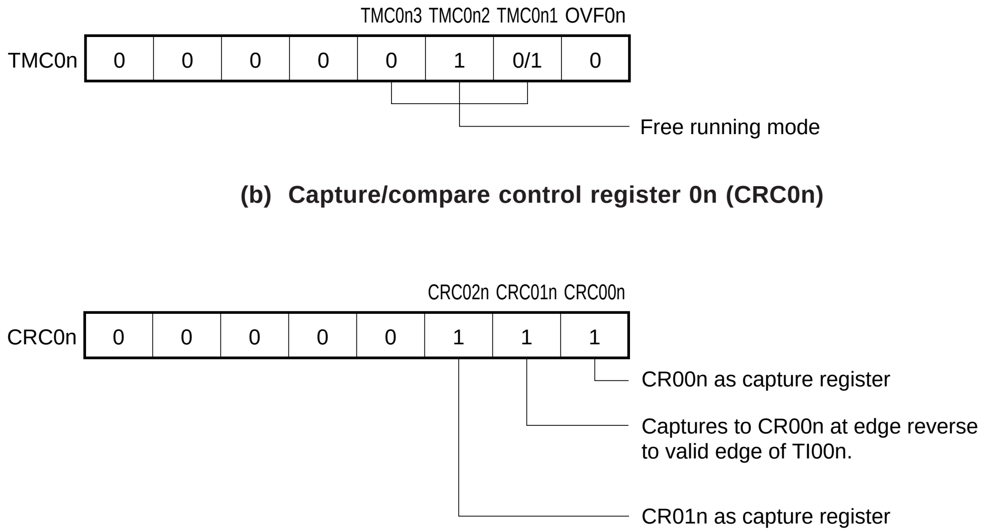
Either of two edge can be selected—rising or falling—as the valid edges for the TI00n pin specified by means of bits 4 and 5 (ES00n and ES01n) of prescaler mode register 0n (PRM0n).

For TI00n pin valid edge detection, sampling is performed at the interval selected by means of the prescaler mode register 0n (PRM0n), and a capture operation is only performed when a valid level is detected twice, thus eliminating noise with a short pulse width.

Caution If the valid edge of TI00n is specified to be both rising and falling edge, 16-bit timer capture/compare register 00n (CR00n) cannot perform the capture operation.

Figure 8-22. Control Register Settings for Pulse Width Measurement with Free-Running Counter and Two Capture Registers

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

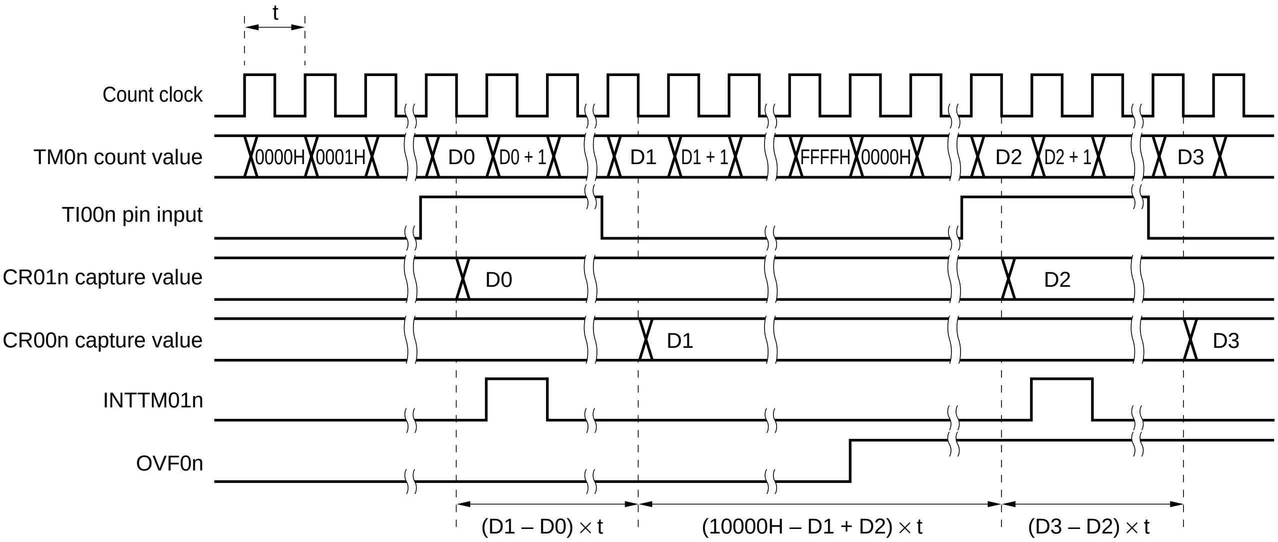

TMC0n3 TMC0n2 TMC0n1 OVF0n
TMC0n
0 0 0 0 0 1 0/1 0
Free running mode
CRC02n CRC01n CRC00n
CRC0n
0 0 0 0 0 1 1 1
CR00n as capture register
Captures to CR00n at edge reverse to valid edge of TI00n.
CR01n as capture register

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

Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. See the description of the respective control registers for details.

n = 0, 1

Figure 8-23. Timing of Pulse Width Measurement Operation by Free-Running Counter and Two Capture Registers (with Rising Edge Specified)



(4) Pulse width measurement by means of restart

When input of a valid edge to the TI00n pin is detected, the count value of the 16-bit timer/counter 0n (TM0n) is taken into 16-bit timer 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 8-24**).

The edge specification can be selected from two types, rising and falling edges by bits 4 and 5 (ES00n and ES01n) of the prescaler mode register 0n (PRM0n)

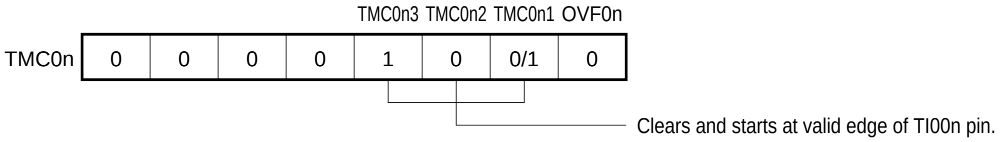
In a valid edge detection, the sampling is performed by a cycle selected by the prescaler mode register 0n (PRM0n) and a capture operation is only performed when a valid level is detected twice, thus eliminating noise with a short pulse width.

Caution **If the valid edge of TI00n is specified to be both rising and falling edges, the 16-bit timer capture/compare register 00n (CR00n) cannot perform the capture operation.**

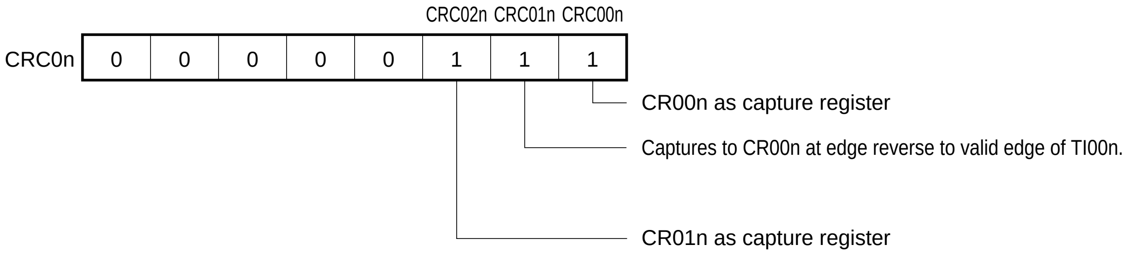
Remark n = 0, 1

Figure 8-24. Control Register Settings for Pulse Width Measurement by Means of Restart

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

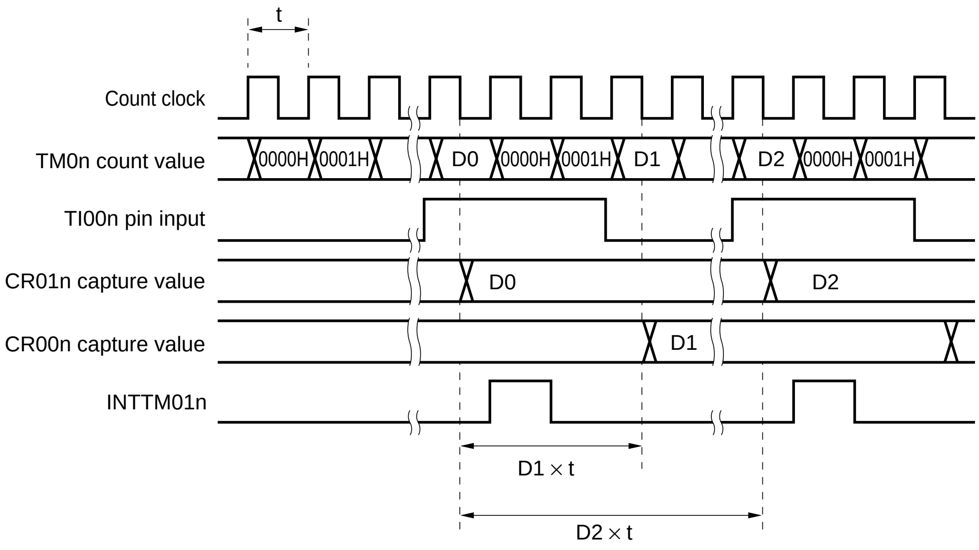


(b) Capture/compare control register 0n (CRC0n)


CRC02n
CRC01n
CRC00n
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

Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. See **Figures 8-3** and **8-4**.

Figure 8-25. Timing of Pulse Width Measurement Operation by Means of Restart (with Rising Edge Specified)



Remark n = 0, 1

8.5.4 External event counter operation

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

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

When the TM0n counted value matches the 16-bit timer capture/compare register 00n (CR00n) value, TM0 is cleared to 0 and the interrupt request signal (INTTM00n) is generated.

Input the value except 0000H to CR00n. (Count operation with a pulse cannot be carried out.)

The rising edge, the falling edge, or both edges can be selected with bits 4 and 5 (ES00n and ES01n) of prescaler mode register 0n (PRM0n).

Because operation is carried out only after the valid edge is detected twice by sampling with the internal clock ($f_X/2^3$), noise with short pulse widths can be removed.

Caution When used as an external event counter, the P70/TI000/TO00 or P75/TI001/TO01/BUZ pin cannot be used as timer output (TO00, TO01).

Figure 8-26. 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 coincidence between TM0n and CR00n.

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

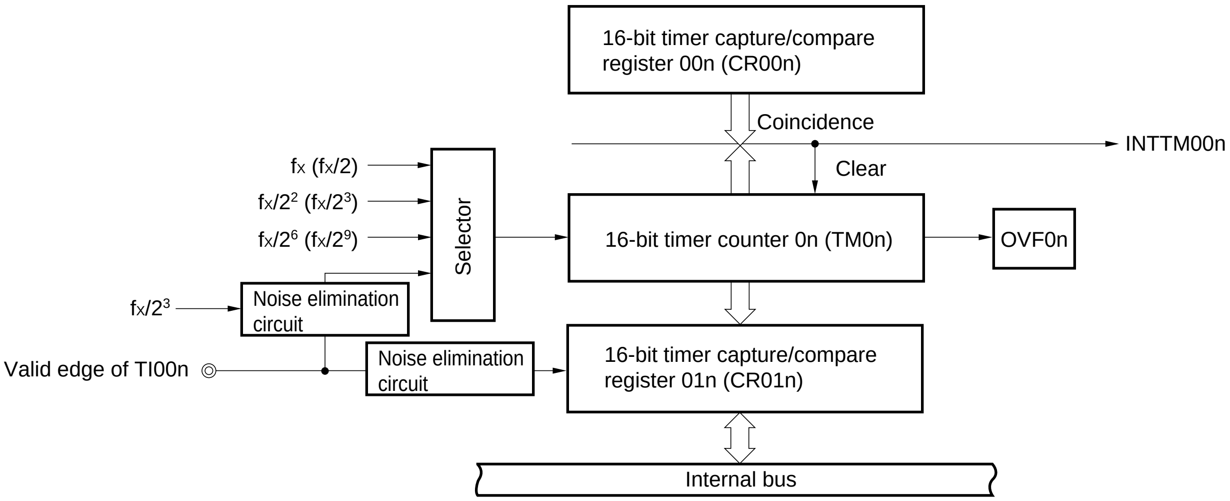
						CRC02n	CRC01n	CRC00n
CRC0n	0	0	0	0	0	0/1	0/1	0

CR00n as compare register

Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with the external event counter. See **Figures 8-3** to **8-6**.

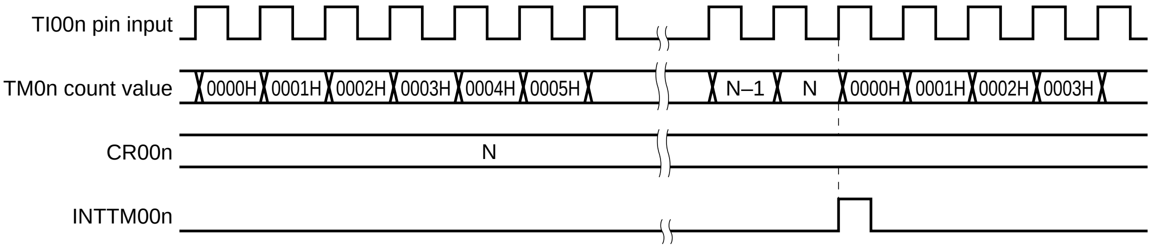
n = 0, 1

Figure 8-27. External Event Counter Configuration Diagram


16-bit timer capture/compare register 00n (CR00n)
Coincidence
INTTM00n
Clear
fX (fX/2)
fX/2² (fX/2³)
fX/2⁶ (fX/2⁹)
Selector
16-bit timer counter 0n (TM0n)
OVF0n
fX/2³
Noise elimination circuit
Valid edge of TI00n
Noise elimination circuit
16-bit timer capture/compare register 01n (CR01n)
Internal bus

Remark Values outside parentheses apply to the 16-bit timer/event counter 00, and the values in parentheses apply to the 16-bit timer/event counter 01.

Figure 8-28. External Event Counter Operation Timings (with Rising Edge Specified)



Caution When reading the external event counter count value, TM0n should be read.

8.5.5 Square-wave output operation

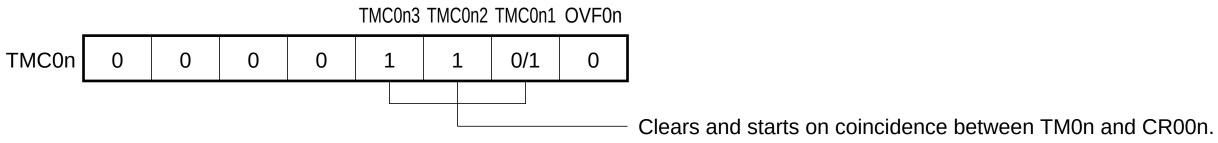
A square wave with any selected frequency to be output at intervals of the count value preset to the 16-bit timer capture/compare register 00n (CR00n) operates.

The TO0n pin output status is reversed at intervals of the count value preset to CR00n by setting bit 0 (TOE0n) and bit 1 (TOC01n) of the 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

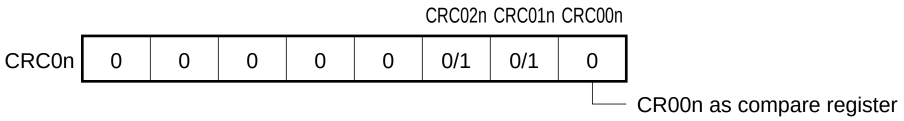
Figure 8-29. Control Register Settings in Square-Wave Output Mode

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



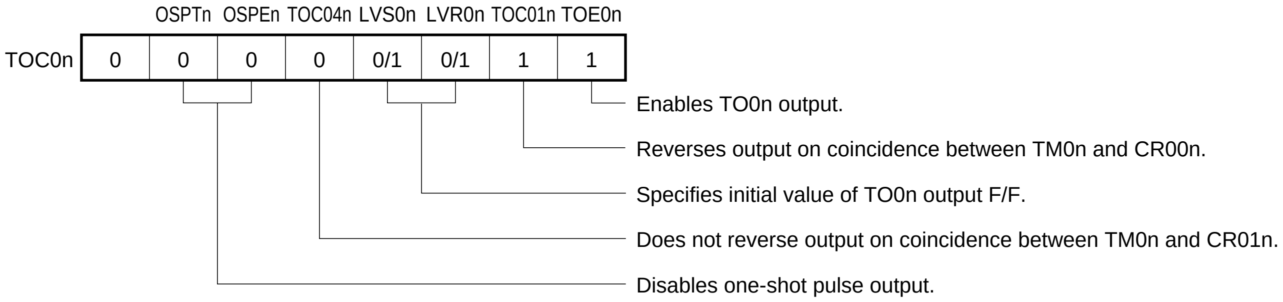
					TMC0n3	TMC0n2	TMC0n1	OVF0n
TMC0n	0	0	0	0	1	1	0/1	0

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



						CRC02n	CRC01n	CRC00n
CRC0n	0	0	0	0	0	0/1	0/1	0

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



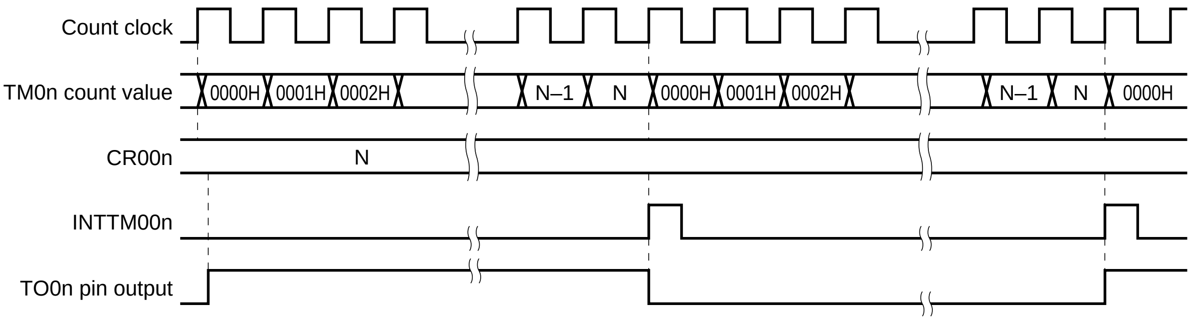
		OSPTn	OSPEn	TOC04n	LVS0n	LVR0n	TOC01n	TOE0n
TOC0n	0	0	0	0	0/1	0/1	1	1

Enables TO0n output.
Reverses output on coincidence between TM0n and CR00n.
Specifies initial value of TO0n output F/F.
Does not reverse output on coincidence between TM0n and CR01n.
Disables one-shot pulse output.

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

n = 0, 1

Figure 8-30. Square-Wave Output Operation Timing



8.5.6 One-shot pulse output operation

It is possible to output one-shot pulses by software trigger.

If the 16-bit timer mode control register 0n (TMC0n), capture/compare control register 0n (CRC0n), and the 16-bit timer output control register 0n (TOC0n) are set as shown in Figure 8-31, and 1 is set in bit 6 (OSPTn) of TOC0n by software, a one-shot pulse is output from the TO0n pin.

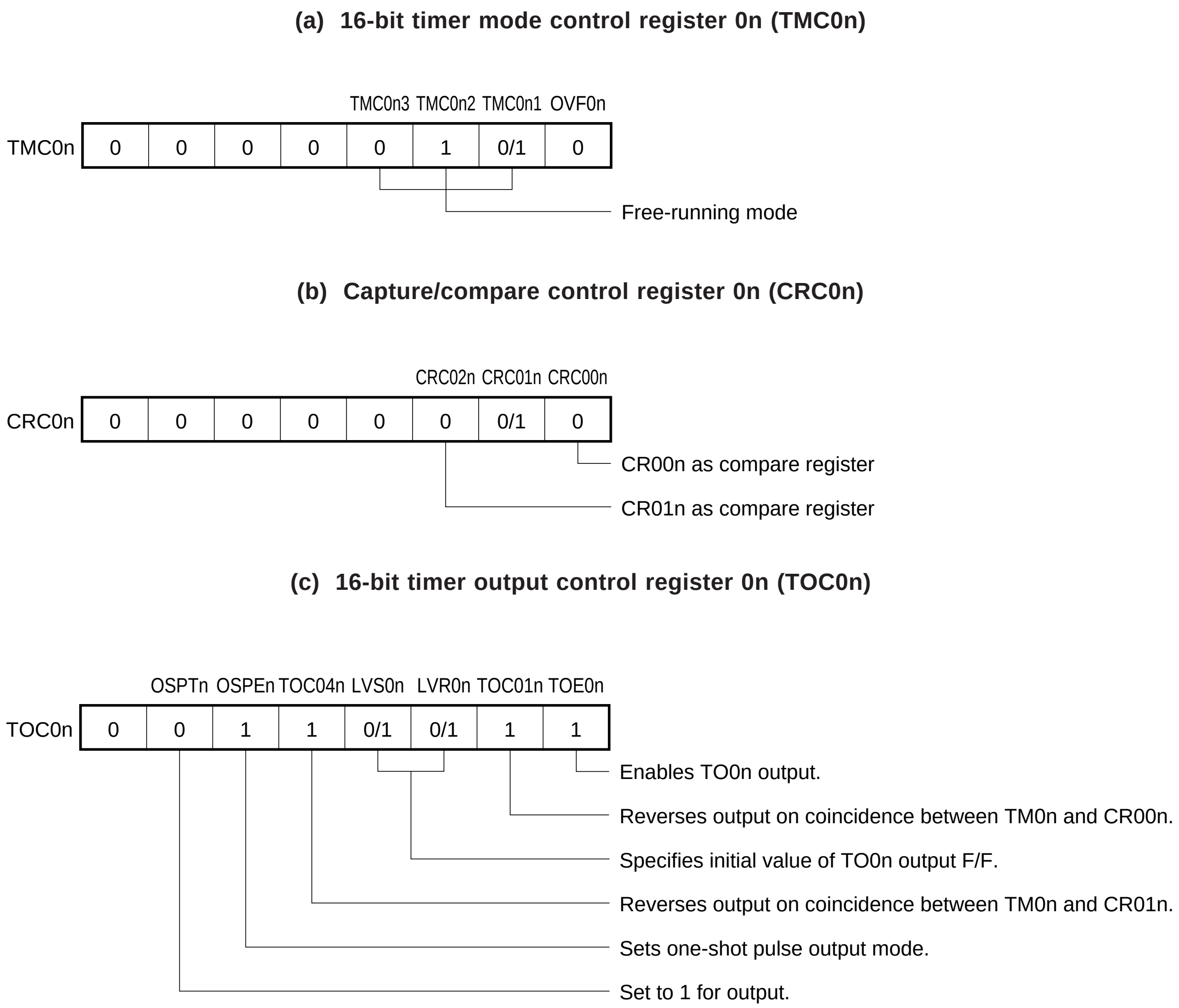
By setting 1 in OSPTn, the 16-bit timer counter 0n (TM0n) is cleared and started, and output is activated by the count value set beforehand in 16-bit timer capture/compare register 01n (CR01n). Thereafter, output is inactivated by the count value set beforehand in 16-bit timer capture/compare register 00n (CR00n).

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

Caution When outputting one-shot pulse, do not set 1 in OSPTn (bit 6 of 16-bit timer output control register 0n (TOC0n)). When outputting one-shot pulse again, do so after the INTTM00n, or interrupt match signal with CR00n, is generated.

Remark n = 0, 1

Figure 8-31. Control Register Settings for One-Shot Pulse Output Operation Using Software Trigger



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

TMC0n3 TMC0n2 TMC0n1 OVF0n
TMC0n
0 0 0 0 0 1 0/1 0
Free-running mode

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

CRC02n CRC01n CRC00n
CRC0n
0 0 0 0 0 0 0/1 0
CR00n as compare register
CR01n as compare register

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

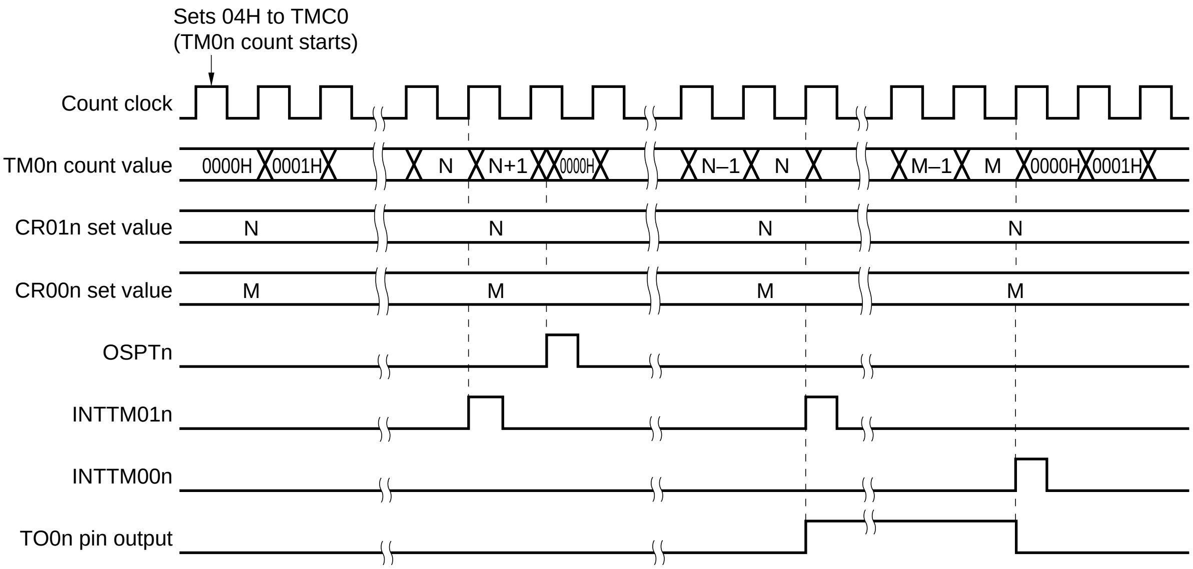
OSPTn OSPEn TOC04n LVS0n LVR0n TOC01n TOE0n
TOC0n
0 0 1 1 0/1 0/1 1 1
Enables TO0n output.
Reverses output on coincidence between TM0n and CR00n.
Specifies initial value of TO0n output F/F.
Reverses output on coincidence between TM0n and CR01n.
Sets one-shot pulse output mode.
Set to 1 for output.

Caution Values in the following range should be set in CR00n and CR01n.
0000H < CR01n < CR00n ≤ FFFFH

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

n = 0, 1

Figure 8-32. Timing of One-Shot Pulse Output Operation Using Software Trigger



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

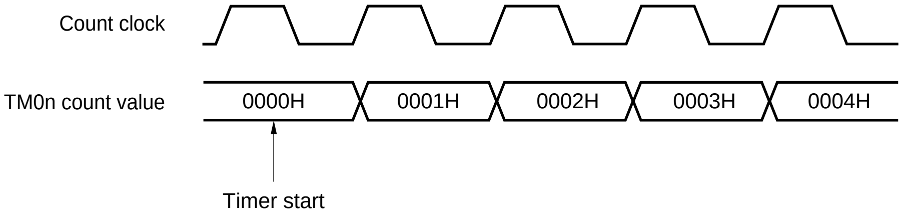
Remark n = 0, 1

8.6 16-Bit Timer/Event Counters 00, 01 Cautions

(1) Timer start errors

An error with a maximum of one clock may occur concerning the time required for a match signal to be generated after timer start. This is because the 16-bit timer counter 0n (TM0n) is started asynchronously with the count clock.

Figure 8-33. 16-Bit Timer Counter 0n (TM0n) Start Timing



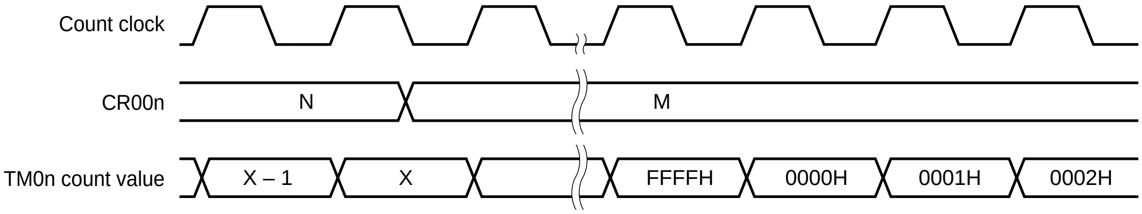
(2) 16-Bit timer compare register setting

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

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

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

Figure 8-34. Timings after Change of Compare Register during Timer Count Operation

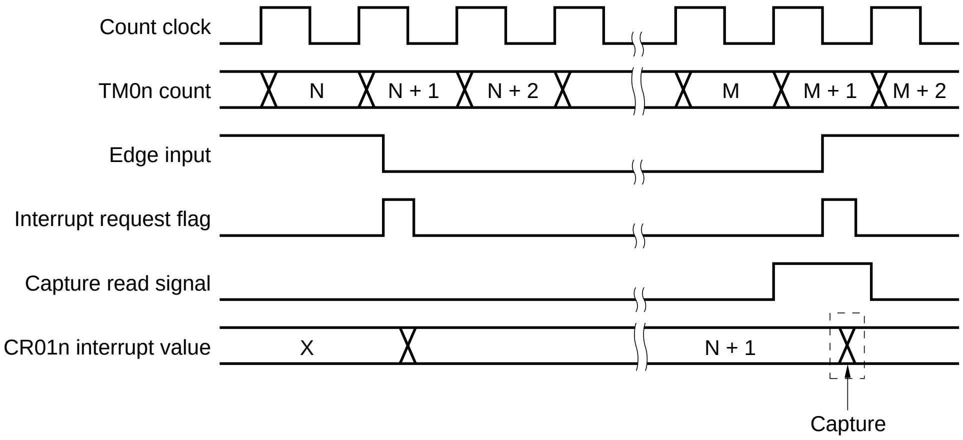


Remark N > X > M
n = 0, 1

(4) Capture register data retention timings

If the valid edge of the TI00n pin is input during 16-bit timer capture/compare register 01n (CR01n) read, CR01n holds data without carrying out capture operation. However, the interrupt request flag (TMIF01n) is set upon detection of the valid edge.

Figure 8-35. Capture Register Data Retention Timing



(5) Valid edge setting

Set the valid edge of the TI00n pin after setting bits 2 and 3 (TMC0n2 and TMC0n3) of the 16-bit timer mode control register 0n (TMC0n) to 0, 0, respectively, and then stopping timer operation. Valid edge is set with bits 4 and 5 (ES00n and ES01n) of the prescaler mode register 0n (PRM0n).

(6) Output of one-shot pulse

<1> When outputting one-shot pulse, do not set 1 in OSPTn (bit 6 of 16-bit timer output control register 0n (TOC0n)). When outputting one-shot pulse again, output it after the INTTM00n, or interrupt request match signal with CR00n, is generated.

<2> Do not set OSPTn to 1 in any mode other than one-shot pulse output.

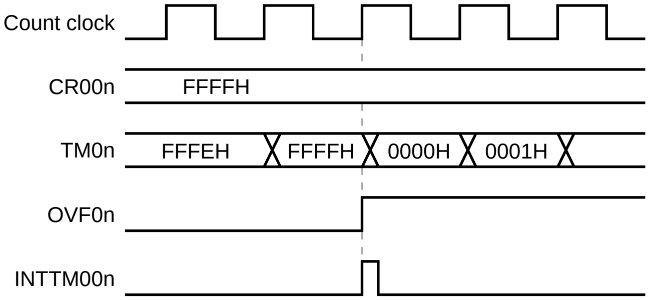
Remark n = 0, 1

(7) Operation of OVF0n flag

<1> OVF0n flag is set to 1 in the following case.

The clear & start mode on match between TM0n and CR00n is selected.
↓
CR00n is set to FFFFH.
↓
When TM0n is counted up from FFFFH to 0000H.

Figure 8-36. Operation Timing of OVF0n Flag



<2> Even if the OVF0n flag is cleared before the next count clock (before TM0n becomes 0001H) after the occurrence of TM0n overflow, the OVF0 flag is reset newly and clear is disabled.

(8) Contending operations

<1> The contending operation between the read time of 16-bit timer capture/compare register (CR00n/CR01n) and capture trigger input (CR00n/CR01n used as capture/register)
Capture/trigger input is prior to the other. The data read from CR00n/CR01n is not defined.

<2> The coincidence timing of contending operation between the write period of 16-bit timer capture/compare register (CR00n/CR01n) and 16-bit timer counter 0n (TM0n) (CR00n/CR01n used as a compare register)
The coincidence discriminant is not performed normally. Do not write any data to CR00n/CR01n near the coincidence timing.

(9) Timer operation

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

<2> Regardless of the CPU’s operation mode, when the timer stops, the input signals to pins TI00n/TI01n are not acknowledged.

<3> One-shot pulse output operates normally only the free-running mode. In the clear & start mode by TM0n and CR00n match , no overflow occurs, and therefore one-shot pulse output is not possible.

Remark n = 0, 1

(10) Capture operation

<1> If TI00n is specified as the valid edge of the count clock, capture operation by the capture register specified as the trigger for TI00n is not possible.

<2> If both the rising and falling edges are selected as the valid edges of TI00n, capture is not performed.

<3> To ensure the reliability of the capture operation, the capture trigger requires a pulse two times longer than the count clock selected by prescaler mode register 0n (PRM0n).

<4> The capture operation is performed at the fall of the count clock. An interrupt request input (INTTM0nn), however, is generated at the rise of the next count clock.

(11) Compare operation

<1> When the 16-bit timer capture/compare register (CR00n/CR01n) is overwritten during timer operation, match interrupt may be generated or clear operation may not be performed normally if that value is close to the timer value and larger than the timer value.

<2> Capture operation may not be performed for CR00n/CR01n set in compare mode even if a capture trigger has been input.

(12) Edge detection

<1> If the TI00n pin or the TI01n pin is high level immediately after system reset and rising edge or both the rising and falling edges are specified as the valid edge for the TI00n pin or TI01n pin to enable the 16-bit timer counter 0n (TM0n) operation, a rising edge is detected immediately after. Be careful when pulling up the TI00n pin or the TI01n pin. However, the rising edge is not detected at restart after the operation has been stopped once.

<2> The sampling clock used to remove noise differs when a TI00n valid edge is used as count clock and when it is used as a capture trigger. In the former case, the count clock is $f_X/2^3$, and in the latter case the count clock is selected by prescaler mode register 0n (PRM0n). To detect a valid edge, the capture operation is not performed until the valid edge is sampled and the valid level is detected two times. Therefore, noise with a short pulse width can be rejected.

Remark n = 0, 1

[MEMO]

CHAPTER 9 8-BIT TIMER/EVENT COUNTERS 50, 51

9.1 Outline of 8-bit Timer/Event Counters 50, 51

8-bit timer/event counters 50, 51 can be used to serve as an interval timer, an external event counter, to output square wave output with any selected frequency, and PWM output. Two 8-bit timer/event counters can be used as one 16-bit timer/event counter.

9.2 8-Bit Timer/Event Counter Functions

8-bit timer/event counters 50, 51 (TM50, TM51) has the following two modes.

- Mode using 8-bit timer/event counters 50, 51 alone (individual mode)
- Mode using the cascade connection (16-bit resolution: cascade connection mode)

These two modes are described next.

(1) Mode using 8-bit timer/event counters 50, 51 alone (individual mode)

The timer operates as 8-bit timer/event counter 50 or 51.
It has the following functions.

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

(2) Mode using the cascade connection (16-bit resolution: cascade connection mode)

The timer operates as a 16-bit timer/event counters 50, 51 by connecting in cascade.
It has the following functions.

- Interval timer with 16-bit resolution
- External event counter with 16-bit resolution
- Square wave output with 16-bit resolution

Figures 9-1 and 9-2 show 8-bit timer/event counters 50, 51 block diagrams.

Figure 9-1. 8-Bit Timer/Event Counter 50 Block Diagram

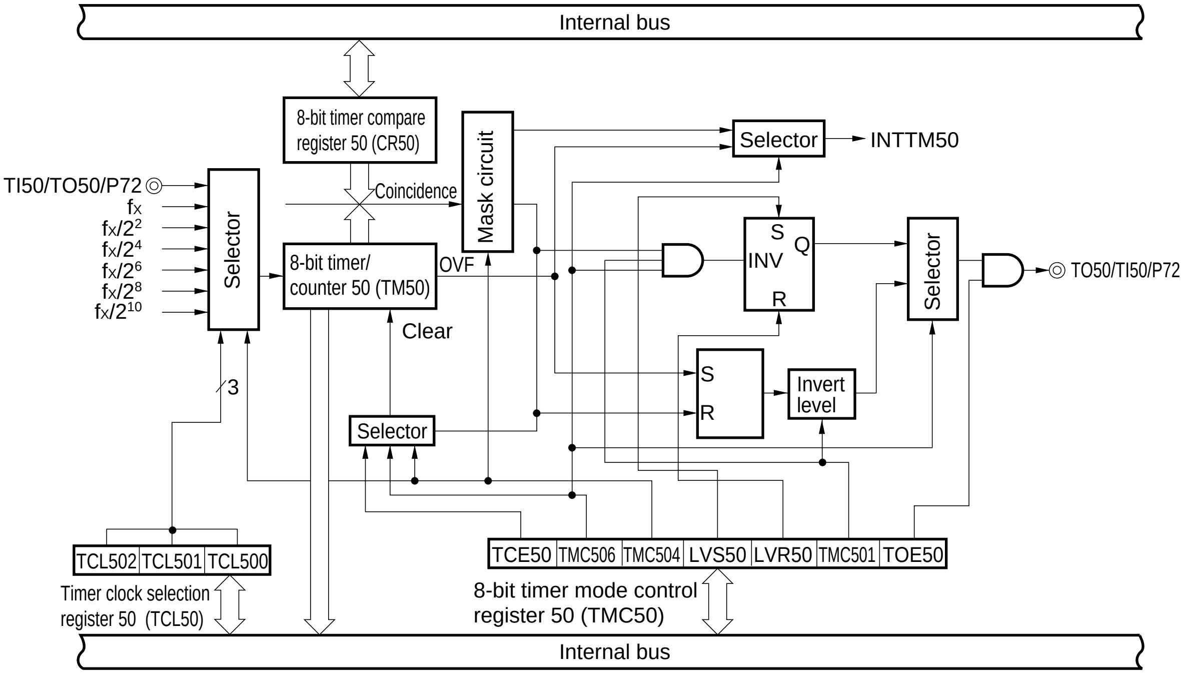
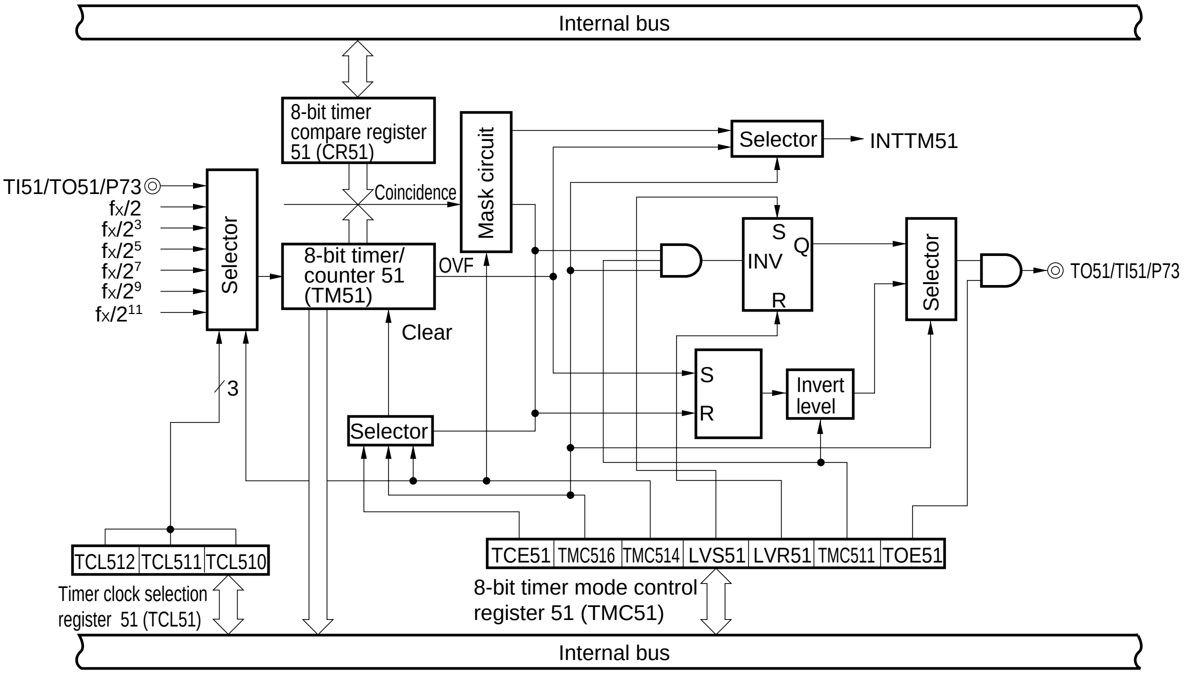

Internal bus
8-bit timer compare register 50 (CR50)
Mask circuit
Selector
INTTM50
TI50/TO50/P72
fX
fX/2^2
fX/2^4
fX/2^6
fX/2^8
fX/2^10
Selector
Coincidence
8-bit timer/ counter 50 (TM50)
OVF
S
INV
Q
R
Selector
TO50/TI50/P72
Clear
S
R
Invert level
3
Selector
TCL502 TCL501 TCL500
Timer clock selection register 50 (TCL50)
TCE50 TMC506 TMC504 LVS50 LVR50 TMC501 TOE50
8-bit timer mode control register 50 (TMC50)
Internal bus

Figure 9-2. 8-Bit Timer/Event Counter 51 Block Diagram


Internal bus
8-bit timer compare register 51 (CR51)
Mask circuit
Selector
INTTM51
TI51/TO51/P73
fX/2
fX/2^3
fX/2^5
fX/2^7
fX/2^9
fX/2^11
Selector
Coincidence
8-bit timer/ counter 51 (TM51)
OVF
S
INV
Q
R
Selector
TO51/TI51/P73
Clear
S
R
Invert level
3
Selector
TCL512 TCL511 TCL510
Timer clock selection register 51 (TCL51)
TCE51 TMC516 TMC514 LVS51 LVR51 TMC511 TOE51
8-bit timer mode control register 51 (TMC51)
Internal bus

9.3 8-Bit Timer/Event Counters 50, 51 Configurations

8-bit timer/event counters 50, 51 consists of the following hardware.

Table 9-1. 8-Bit Timer/Event Counters 50, 51 Configuration

Item	Configuration
Timer register	8-bit timer counter 5n (TM5n)
Register	8-bit timer compare register 5n (CR5n)
Timer output	2 (TO5n)
Control registers	Timer clock select register 5n (TCL5n) 8-bit timer mode control register 5n (TMC5n) Port mode register 7 (PM7)[Note]

Note See **Figure 6-14 P70 to P75 Block Diagram.**

Remark n = 0, 1

(1) 8-bit timer counter 5n (TM5n: n = 0,1)

TM5n is an 8-bit read-only register which counts the count pulses.

When count clock starts, a counter is incremented. TM50 and TM51 can be connected in cascade and used as a 16-bit timer.

When TM50 and TM51 can be connected in cascade and used as a 16-bit timer, they can be read by a 16-bit memory operation instruction. However, since they are connected by an internal 8-bit bus, TM50 and TM51 are read separately in two times. Thus, take read during count change into consideration and compare them in two times reading. When count value is read during operation, count clock input is temporary stopped, and then the count value is read. In the following situations, count value is set to 00H.

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

<2> When TCE5n is cleared

<3> When TM5n and CR5n match in clear & start mode if this mode was entered upon match of TM5n and CR5n values.

Caution In cascade connection mode, the count value is reset to 00H when the lowest timer TCE5n is cleared.

Remark n = 0, 1

(2) 8-bit timer compare register 5n (CR5n: n = 0, 1)

When CR5n is used as a compare resistor, the value set in CR5n is constantly compared with the 8-bit timer/counter (TM5n) count value, and an interrupt request (INTTM5n) is generated if they match. (Except PWM mode). It is possible to rewrite the value of CR5n within 00H to FFH during count operation.

When TM50 and TM51 can be connected in cascade and used as a 16-bit timer, CR50 and CR51 operate as the 16-bit compare register. It compares count value with register value, and if the values are matched, interrupt request (INTTM50) are generated. INTTM51 interrupt request is also generated at this time. Thus, when TM50 and TM51 are used as cascade connection, mask INTTM51 interrupt request.

Caution In cascade connection mode, stop the timer operation before setting the data.

Remark n = 0, 1

9.4 Registers to Control 8-Bit Timer/Event Counters 50, 51

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

- Timer clock select register 5n (TCL5n)
- 8-bit timer mode control register 5n (TMC5n)
- Port mode register 7 (PM7)

n = 0, 1

(1) Timer clock select register 5n (TCL5n: n = 0, 1)

This register sets count clocks of 8-bit timer/event counter 5n and the valid edge of TI50, TI51 input.

TCL5n is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the value of TCL5n to 00H.

Figure 9-3. Timer Clock Select Register 50 (TCL50) Format

Address: FF71H At Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TCL50	0	0	0	0	0	TCL502	TCL501	TCL500

TCL502	TCL501	TCL500	Count Clock Selection
0	0	0	TI50 falling edge
0	0	1	TI50 rising edge
0	1	0	f_X (8.38 MHz)
0	1	1	$f_X/2^2$ (2.09 MHz)
1	0	0	$f_X/2^4$ (523 kHz)
1	0	1	$f_X/2^6$ (131 kHz)
1	1	0	$f_X/2^8$ (32.7 kHz)
1	1	1	$f_X/2^{10}$ (8.18 kHz)

Cautions 1. When rewriting TCL50 to other data, stop the timer operation beforehand.

2. Be sure to set bits 3 through 7 to 0.

Remarks 1. When cascade connection is used, the settings of TCL5n0 to TCL5n2 (n = 0, 1) are valid only for the lowermost timer.

2. f_X: Main system clock oscillation frequency

3. Figures in parentheses are for operation with f_X = 8.38 MHz

Figure 9-4. Timer Clock Select Register 51 (TCL51) Format

Address: FF79H At Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TCL51	0	0	0	0	0	TCL512	TCL511	TCL510

TCL512	TCL511	TCL510	Count Clock Selection
0	0	0	TI51 falling edge
0	0	1	TI51 rising edge
0	1	0	$f_X/2$ (4.19 MHz)
0	1	1	$f_X/2^3$ (1.04 MHz)
1	0	0	$f_X/2^5$ (261 kHz)
1	0	1	$f_X/2^7$ (65.4 kHz)
1	1	0	$f_X/2^9$ (16.3 kHz)
1	1	1	$f_X/2^{11}$ (4.09 kHz)

Cautions 1. When rewriting TCL51 to other data, stop the timer operation beforehand.

2. Be sure to set bits 3 through 7 to 0.

Remarks 1. When cascade connection is used, the settings of TCL5n0 to TCL5n2 (n = 0, 1) are valid only for the lowermost timer.

2. f_X: Main system clock oscillation frequency

3. Figures in parentheses are for operation with f_X = 8.38 MHz

(2) 8-bit timer mode control register 5n (TMC5n: n = 0, 1)

TMC5n is a register which sets up the following six types.

<1> 8-bit timer counter 5n (TM5n) count operation control
<2> 8-bit timer counter 5n (TM5n) operating mode selection
<3> Single mode/cascade connection mode selection
<4> Timer output F/F (flip flop) status setting
<5> Active level selection in timer F/F control or PWM (free-running) mode.
<6> Timer output control

TMC5n is set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets to 00H.

Figure 9-5. 8-Bit Timer Mode Control Register 50 (TMC50) Format

Address: FF70H At Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TMC50	TCE50	TMC506	0	TMC504	LVS50	LVR50	TMC501	TOE50

TCE50	TM50 Count Operation Control
0	After cleaning to 0, count operation disabled (prescaler disabled)
1	Count operation start

TMC506	TM50 Operating Mode Selection
0	Clear and start mode by matching between TM50 and CR50
1	PWM (Free-running) mode

TMC504	Single Mode/Cascade Connection Mode Selection
0	Single mode (use the lowest timer)
1	Cascade connection mode (connect to lower timer)

LVS50	LVR50	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

TMC501	In Other Modes (TMC506 = 0)	In PWM Mode (TMC506 = 1)
	Timer F/F control	Active level selection
0	Inversion operation disabled	Active high
1	Inversion operation enabled	Active low

TOE50	Timer Output Control
0	Output disabled (Port mode)
1	Output enabled

Caution **Before clearing TCE50 to 0, set the interrupt mask flag (TMMK50) to 1. This is because the interrupt may occur after TCE50 has been cleared. Clear TCE50 in 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.
     ⋮
```

Remarks 1. In PWM mode, PWM output will be inactive because of TCE50 = 0.

2. If LVS50 and LVR50 are read after data is set, 0 is read.

Figure 9-6. 8-Bit Timer Mode Control Register 51 (TMC51) Format

Address: FF78H At Reset: 00H R/W

Symbol	$\boxed{7}$	6	5	4	$\boxed{3}$	$\boxed{2}$	1	$\boxed{0}$
TMC51	TCE51	TMC516	0	TMC514	LVS51	LVR51	TMC511	TOE51

TCE51	TM51 Count Operation Control
0	After cleaning to 0, count operation disabled (prescaler disabled)
1	Count operation start

TMC516	TM51 Operating Mode Selection
0	Clear and start mode by matching between TM51 and CR51
1	PWM (Free-running) mode

TMC514	Single Mode/Cascade Connection Mode Selection
0	Single mode (use the lowest timer)
1	Cascade connection mode (connect to lower timer)

LVS51	LVR51	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

TMC511	In Other Modes (TMC516 = 0)	In PWM Mode (TMC516 = 1)
	Timer F/F control	Active level selection
0	Inversion operation disabled	Active high
1	Inversion operation enabled	Active low

TOE51	Timer Output Control
0	Output disabled (Port mode)
1	Output enabled

Caution Before setting TCE51 to 1, set the interrupt mask flag (TMMK51) to 1. This is because the interrupt may occur after TCE51 has been cleared. Clear TCE51 in 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. In PWM mode, PWM output will be inactive because of TCE51 = 0.

2. If LVS51 and LVR51 are read after data is set, 0 is read.

(3) Port mode register 7 (PM7)

This register sets port 7 input/output in 1-bit units.

When using the P72/TO50/TI50 and P73/TI51/TO51 pins for timer output, set PM72, PM73, and output latches of P72 and P73 to 0.

PM7 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the value of PM7 to FFH.

Figure 9-7. Port Mode Register 7 (PM7) Format

Address: FF27H At 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 Input/Output Mode Selection (n = 0 to 5)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

9.5 8-Bit Timer/Event Counters 50, 51 Operations

9.5.1 8-bit interval timer operation

The 8-bit timer/event counters operate as interval timers which generate interrupt requests repeatedly at intervals of the count value preset to 8-bit timer compare registers 5n (CR5n).

When the count values of the 8-bit timer counter 5n (TM5n) match the values set to CR5n, counting continues with the TM5n values cleared to 0 and the interrupt request signals (INTTM5n) are generated.

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

See **9.6 8-Bit Timer/Event Counters 50, 51 Cautions (2) Operation after compare register transition during timer count operation** about the operation when the compare register value is changed during timer count operation.

[Setting]

<1> Set the registers.

- TCL5n : Select count clock.
- CR5n : Compare value
- TMC5n : Clear and Start mode by match of TM5n and CR5n.
  (TMC5n = 0000×××0B × = don’t care)

<2> After TCE5n = 1 is set, count operation starts.

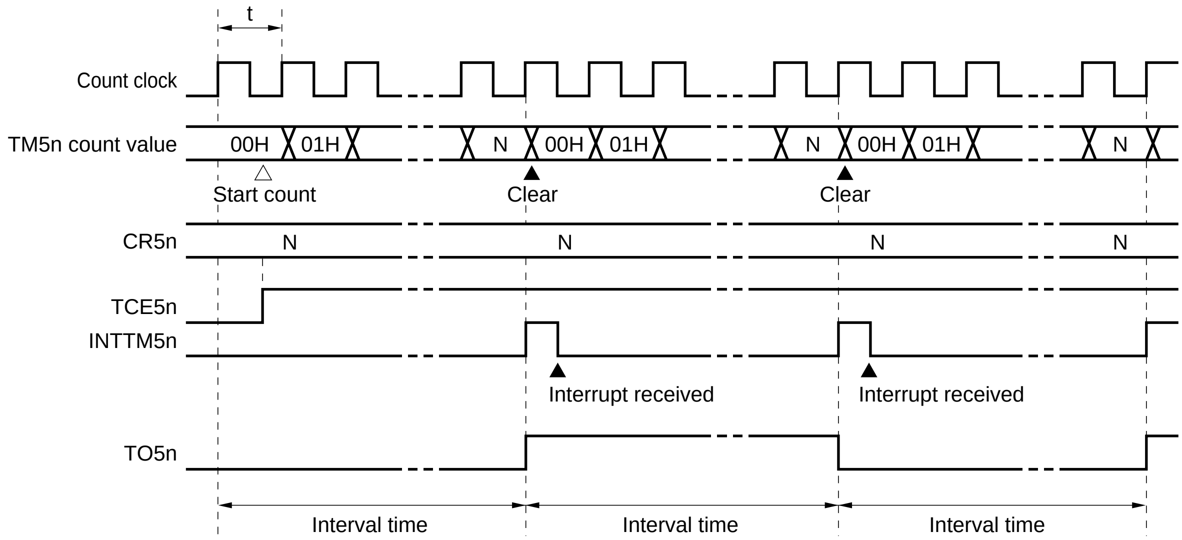
<3> If the values of TM5n and CR5n match, the timer output flip-flop inverts. Also, INTTM5n is generated and TM5n is cleared to 00H.

<4> INTTM5n generates repeatedly at the same interval.
Set TCE5n to 0 to stop count operation.

Remark n = 0, 1

Figure 9-8. Interval Timer Operation Timings (1/3)

(a) Basic operation

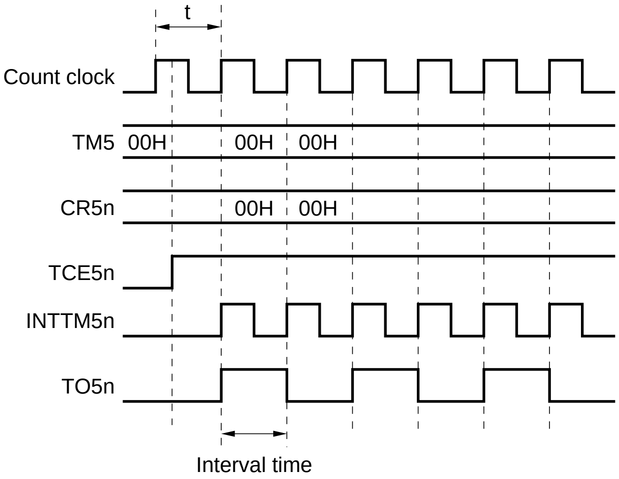


Remarks 1. Interval time = (N + 1) × t
N = 00H to FFH

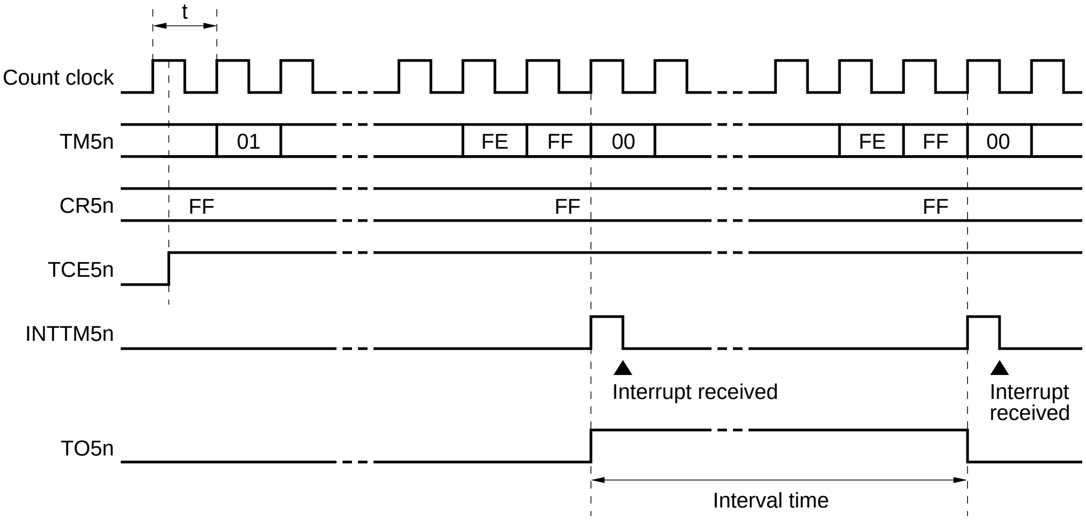
2. n = 0, 1

Figure 9-8. Interval Timer Operation Timings (2/3)

(b) When CR5n = 00H



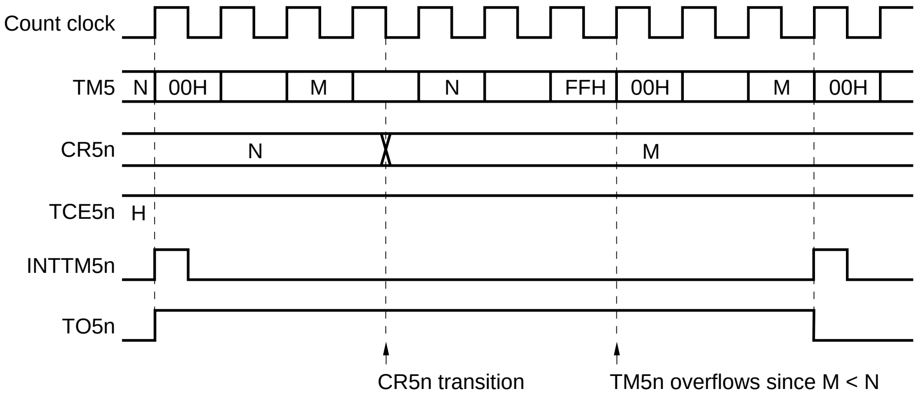
(c) When CR5n = FFH



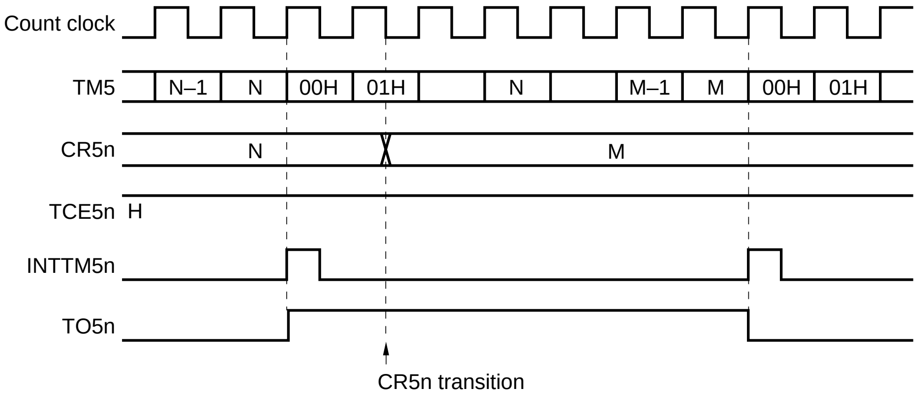
n = 0, 1

Figure 9-8. Interval Timer Operation Timings (3/3)

(d) Operated by CR5n transition (M < N)



(e) Operated by CR5n transition (M > N)



n = 0, 1

9.5.2 External event counter operation

The external event counter counts the number of external clock pulses to be input to the TI5n by the 8-bit timer counter 5n (TM5n).

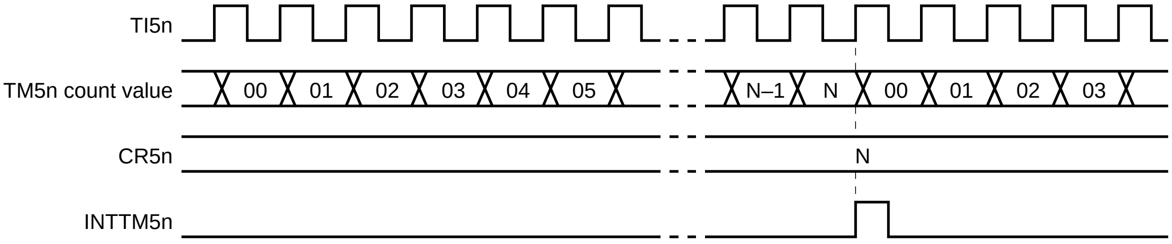
TM5n is incremented each time the valid edge specified with the timer clock select register 5n (TCL5n) is input. Either the rising or falling edge can be selected.

When the TM5n counted values match the values of 8-bit timer compare register 5n (CR5n), TM5n is cleared to 0 and the interrupt request signal (INTTM5n) are generated.

Whenever the TM5n counted value matches the value of CR5n, INTTM5n is generated.

Remark n = 0, 1

Figure 9-9. External Event Counter Operation Timing (with Rising Edge Specified)



n = 0, 1

9.5.3 Square-wave output (8-bit resolution) operation

A square wave with any selected frequency is output at intervals of the value preset to the 8-bit timer compare register 5n (CR5n).

TO5n pin output status is reversed at intervals of the count value preset to CR5n by setting bit 0 (TOE5n) of 8-bit timer mode control register 5n (TMC5n) to 1. This enables a square wave with any selected frequency to be output (duty = 50%).

[Setting]

<1> Set each register

- Set port latch and port mode register to 0.
- TCL5n: Select count clock
- CR5n: Compared value
- TMC5n: Clear and Start mode by match of TM5n and CR5n

LVS5n	LVR5n	Timer Output F/F Status Setting
1	0	High-level output
0	1	Low-level output

Timer output F/F reverse enable
Timer output enable → TOE5n = 1

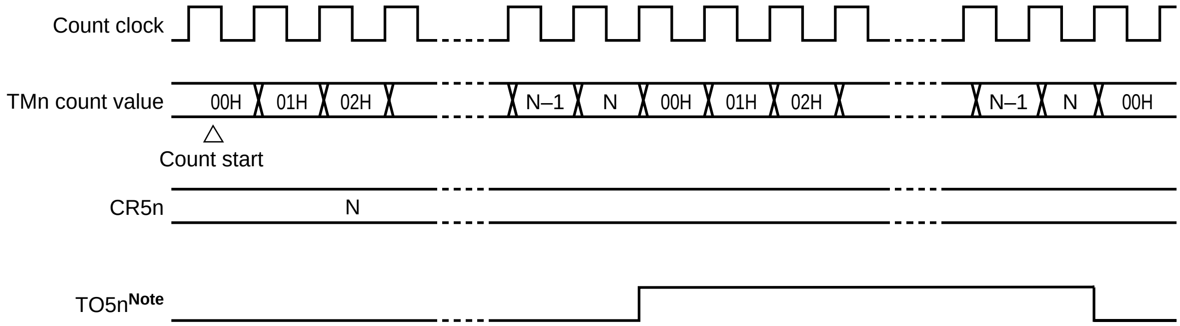
<2> After TCE5n=1 is set, count operation starts

<3> Timer output F/F is reversed by match of TM5n and CR5n. After INTTM5n is generated, TM5n is cleared to 00H

<4> Timer output F/F is reversed at the same interval and square wave is output from TO5n

Remark n = 0,1

Figure 9-10. Square-Wave Output Operation Timing



Note TO5n output initial value can be set by bits 2 and 3 (LVR5n, LVS5n) of the 8-bit timer mode control register 5n (TMC5n)

Remark n = 0,1

9.5.4 8-bit PWM output operation

8-bit timer/event counter operates as PWM output when bit 6 (TMC5n6) of 8-bit timer mode control register 5n (TMC5n) is set to 1.

The duty rate pulse determined by the value set to 8-bit timer compare register 5n (CR5n).

Set the active level width of PWM pulse to CR5n, and the active level can be selected with bit 1 of TMC5n (TMC5n1).

Count clock can be selected with bit 0 to 2 (TCL5n0 to TCL5n2) of timer clock select register 5n (TCL5n).

Enable/disable for PWM output can be selected with bit 0 of TMC5n (TOE5n).

Caution Rewrite of CR5n in PWM mode is allowed only once in a cycle.

Remark n = 0, 1

(1) PWM output basic operation

[Setting]

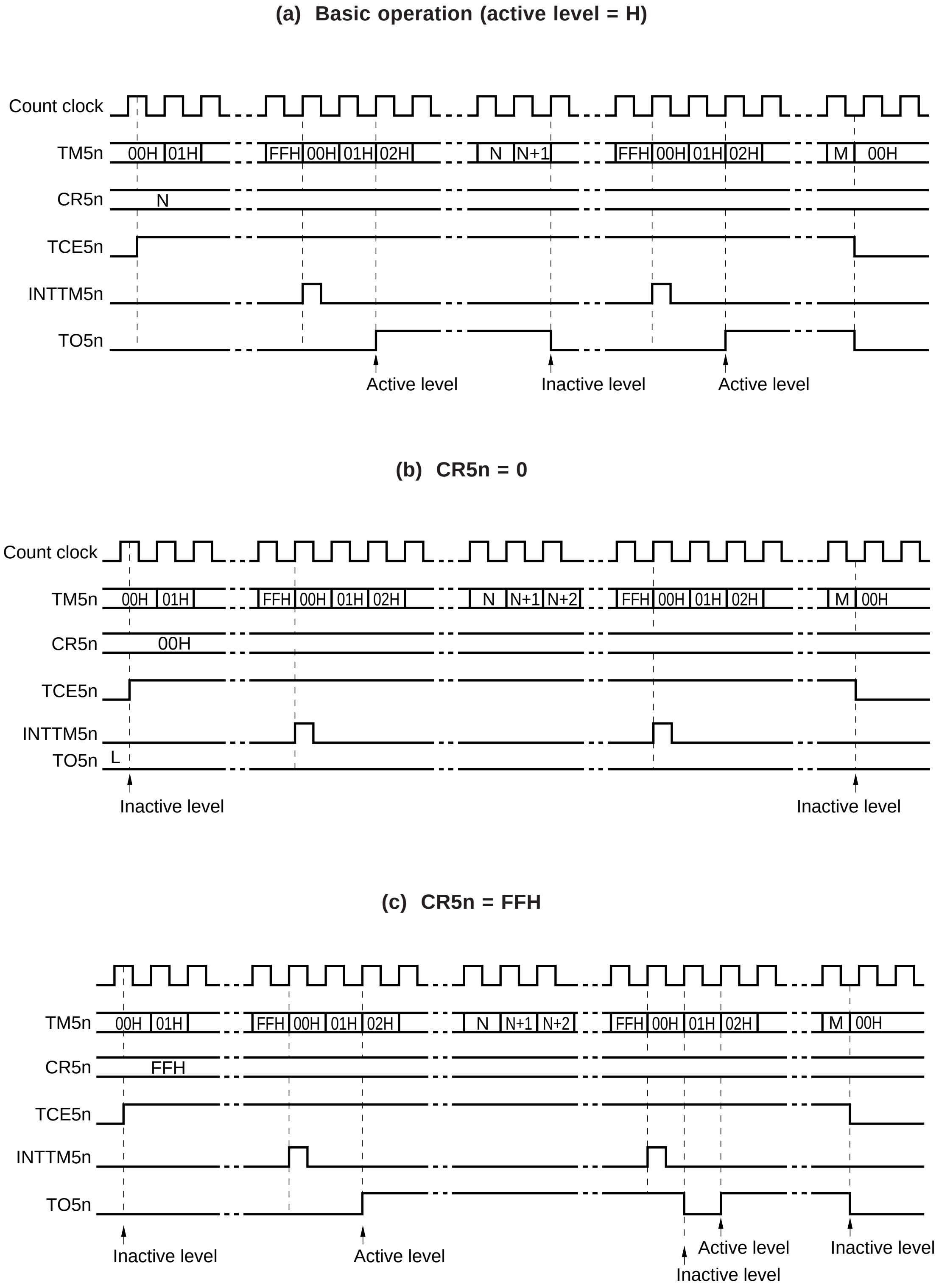
<1> Set port latch (P72, 73) and port mode register 7 (PM72, PM73) to 0.
<2> Set active level width with 8-bit timer compare register (CR5n).
<3> Select count clock with timer clock select register 5n (TCL5n).
<4> Set active level with bit 1 of TMC5n (TMC5n1).
<5> Count operation starts when bit 7 of TMC5n is set to 1.
Set TCE5n to 0 to stop count operation.

[PWM output operation]

<1> PWM output (output from TO5n) outputs inactive level after count operation starts until overflow is generated.
<2> When overflow is generated, the active level set in <1> of setting is output.
The active level is output until CR5n matches the count value of 8-bit timer counter 5n (TM5n).
<3> After the CR5n matches the count value, PWM output outputs the inactive level again until overflow is generated.
<4> Operations <2> and <3> are repeated until the count operation stops.
<5> When the count operation is stopped with TCE5n = 0, PWM output comes to inactive level.

Remark n = 0, 1

Figure 9-11. PWM Output Operation Timing



(a) Basic operation (active level = H)

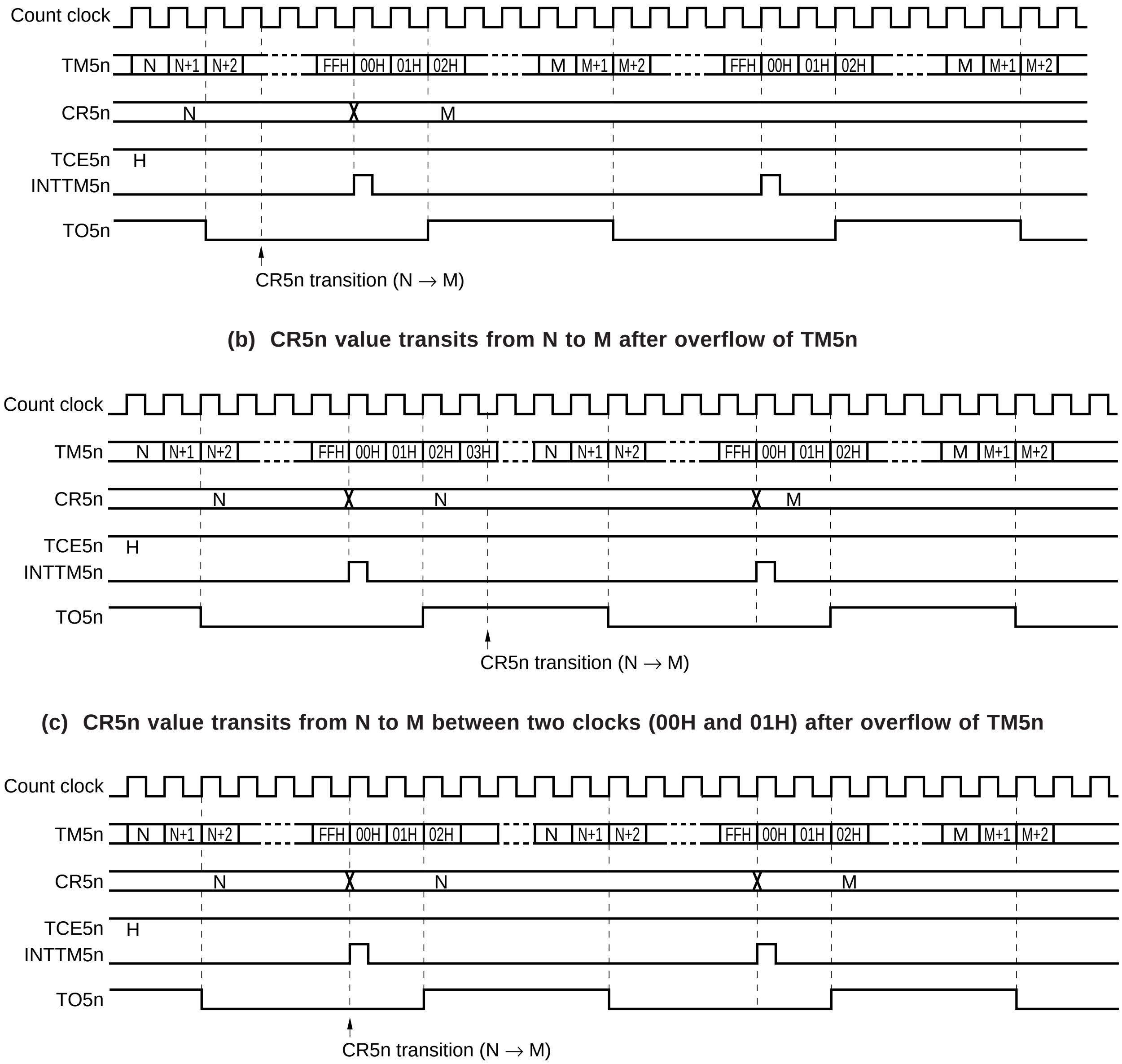
(b) CR5n = 0

(c) CR5n = FFH

n = 0, 1

(2) Operated by CR5n transition

Figure 9-12. Timing of Operation by Change of CR5n



(a) CR5n value transits from N to M before overflow of TM5n

CR5n transition (N → M)

(b) CR5n value transits from N to M after overflow of TM5n

CR5n transition (N → M)

(c) CR5n value transits from N to M between two clocks (00H and 01H) after overflow of TM5n

CR5n transition (N → M)

n = 0, 1

9.5.5 Interval timer (16-bit) operations

When “1” is set in bit 4 (TMC514) of 8-bit timer mode control register 51 (TMC51), the 16-bit resolution timer/counter mode is entered.

The 8-bit timer/event counter operates as an interval timer which generates interrupt requests repeatedly at intervals of the count value preset to the 8-bit timer compare registers (CR50, CR51).

[Setting]

<1> Set each register

TCL50 : Select count clock in TM50.
Cascade-connected TM51 need not be selected.

CR50, CR51 : Compared value (each value can be set at 00H-FFH)

TMC50, TMC51 : Select the clear & start mode by match of TM50 and CR50 (TM51 and CR51).

(TM50 → TMC50 = 0000×××0B ×: don’t care
TM51 → TMC51 = 0001×××0B ×: don’t care)

<2> When TMC51 is set to TCE51 = 1 and then, TCE50 is set to TCE50 = 1, count operation starts.

<3> When the values of TM50 and CR50 of cascade-connected timer match, INTTM50 of TM50 is generated. (TM50 and TM51 are cleared to 00H)

<4> INTTM5n generates repeatedly at the same interval.

Cautions 1. Stop timer operation without fail before setting compare register (CR50, CR51).

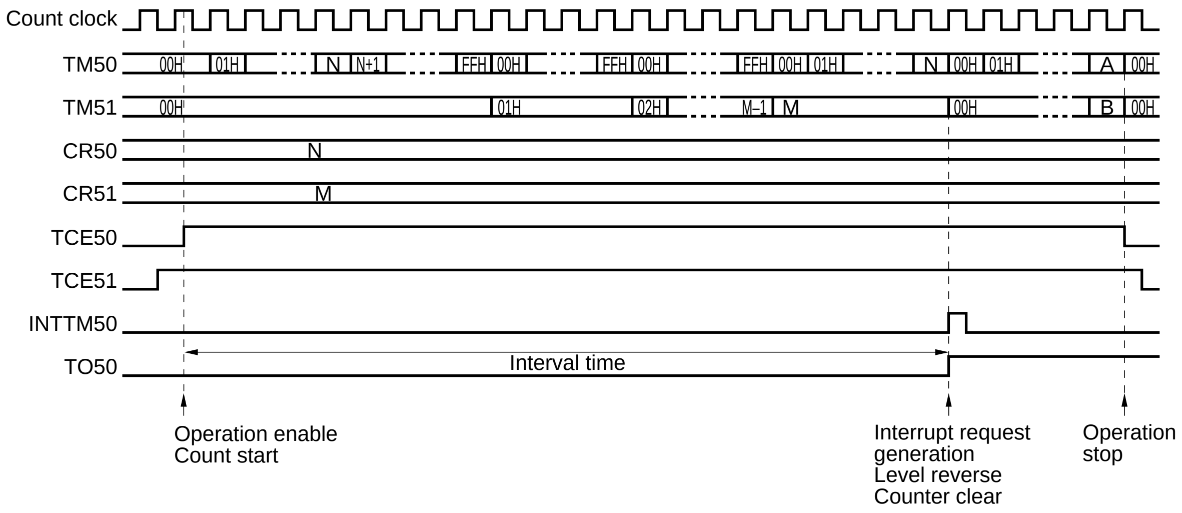
2. INTTM51 of TM51 is generated when TM51 count value matches CR51, even if cascade connection is used. Ensure to mask TM51 to prohibit interrupt.

3. Set TCE50 and TCE51 in a sequential order of TM51 and TM50.

4. Count restart/stop can only be controlled by setting TCE50 of TM50 to 1/0.

Figure 9-13 shows an example of 16-bit resolution cascade connection mode timing.

Figure 9-13. 16-Bit Resolution Cascade Connection Mode

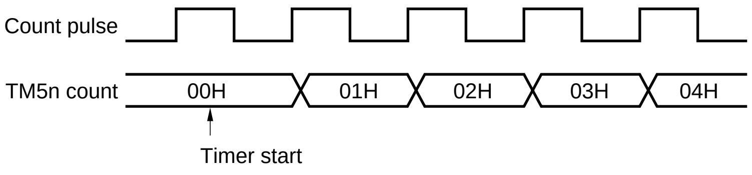


9.6 8-Bit Timer/Event Counters 50, 51 Cautions

(1) Timer start errors

An error with the maximum of one clock may occur concerning the time required for a match signal to be generated after timer start. This is because the 8-bit timer counter 5n (TM5n) is started asynchronously with the count pulse.

Figure 9-14. 8-Bit Timer/Counter Start Timing

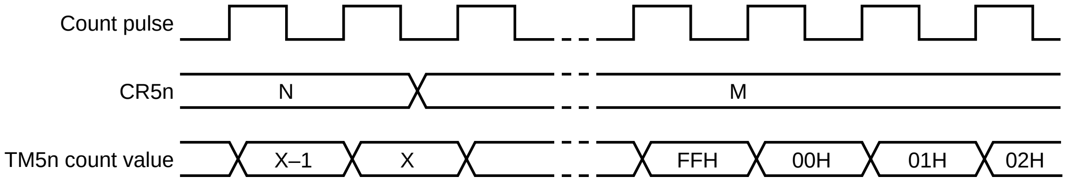


n = 0, 1

(2) Operation after compare register transition during timer count operation

If the values after the 8-bit timer compare register 5n (CR5n) is transmitted is smaller than the value of 8-bit timer counter 5n (TM5n), TM5n continues counting, overflows and then restarts counting from 0. Thus, if the value (M) after CR5n is smaller than value (N) before transition, it is necessary to restart the timer after transitting CR5n.

Figure 9-15. Timing after Compare Register Transition during Timer Count Operation



Caution Except when the TI5n input is selected, always set TCE5n = 0 before setting the stop state.

Remarks 1. N > X > M

2. n = 0, 1

(3) TM5n (n = 0, 1) reading during timer operation

When reading TM5n during operation, select count clock having high/low level wave form longer than two cycles of CPU clock because count clock stops temporary. For example, in the case where CPU clock (f_{CPU}) is fx, when the selected count clock is fx/4 or below, it can be read.

Remark n = 0, 1

10.1 Outline of Watch Time

This timer can set a flag every 0.5 sec. or 0.25 sec. and simultaneously generate an interrupt request at the preset time intervals.

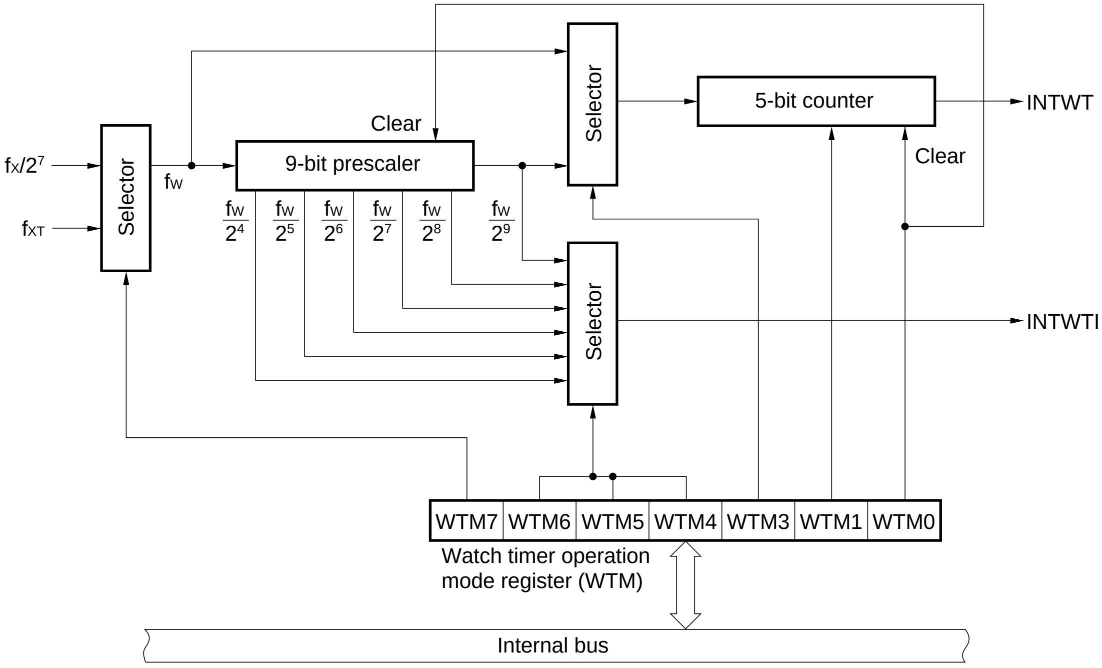
10.2 Watch Timer Functions

The watch timer has the following functions.

- Watch timer
- Interval timer

The watch timer and the interval timer can be used simultaneously.
Figure 10-1 shows the watch timer block diagram.

Figure 10-1. Watch Timer Block Diagram



Remark fX : Main system clock oscillation frequency
fXT : Subsystem clock oscillation frequency

(1) Watch timer

When the main system clock or subsystem clock is used, interrupt requests (INTWT) are generated at $2^{14}/f_W$ seconds or $2^5/f_W$ seconds intervals.

Remark f_W : Watch timer clock frequency ($f_X/2^7$ or f_{XT})
f_X : Main system clock oscillation frequency
f_{XT}: Subsystem clock oscillation frequency

(2) Interval timer

Interrupt requests (INTWTI) are generated at the preset time interval.

Table 10-1. Interval Timer Interval Time

Interval Time		When Operated at f_X = 8.38 MHz	When Operated at f_X = 4.19 MHz	When Operated at f_{XT} = 32.768 kHz
$2^{11} \times 1/f_X$	$2^4 \times 1/f_{XT}$	244 μs	489 μs	488 μs
$2^{12} \times 1/f_X$	$2^5 \times 1/f_{XT}$	489 μs	978 μs	977 μs
$2^{13} \times 1/f_X$	$2^6 \times 1/f_{XT}$	978 μs	1.96 ms	1.95 ms
$2^{14} \times 1/f_X$	$2^7 \times 1/f_{XT}$	1.96 ms	3.91 ms	3.91 ms
$2^{15} \times 1/f_X$	$2^8 \times 1/f_{XT}$	3.91 ms	7.82 ms	7.81 ms
$2^{16} \times 1/f_X$	$2^9 \times 1/f_{XT}$	7.82 ms	15.6 ms	15.6 ms

Remark f_X : Main system clock oscillation frequency
f_{XT} : Subsystem clock oscillation frequency

10.3 Watch Timer Configuration

The watch timer consists of the following hardware.

Table 10-2. Watch Timer Configuration

Item	Configuration
Counter	5 bits × 1
Prescaler	9 bits × 1
Control register	Watch timer operation mode register (WTM)

10.4 Register to Control Watch Timer

Watch timer operation mode register (WTM) is a register to control watch timer.

- **Watch timer operation mode register (WTM)**

  This register sets the watch timer count clock, enables/disables operation, prescaler interval time, 5-bit counter operation control, and watch timer interrupt time selection.
  WTM is set by an 8-bit memory manipulation instruction.
  $\overline{\text{RESET}}$ input sets the value of WTM to 00H.

Figure 10-2. Watch Timer Operation Mode Register (WTM) Format

Address: FF41H At Reset: 00H R/W

Symbol	7	6	5	4	3	2	$\boxed{1}$	$\boxed{0}$
WTM	WTM7	WTM6	WTM5	WTM4	WTM3	0	WTM1	WTM0

WTM7	Watch Timer Count Clock Selection
0	$f_X/2^7$ (65.4 kHz)
1	f_{XT} (32.768 kHz)

WTM6	WTM5	WTM4	Prescaler Interval Time Selection
0	0	0	$2^4/f_W$
0	0	1	$2^5/f_W$
0	1	0	$2^6/f_W$
0	1	1	$2^7/f_W$
1	0	0	$2^8/f_W$
1	0	1	$2^9/f_W$
Other than above			Setting prohibited

WTM3	Selects Interrupt Request Time of Watch Timer.
0	$2^{14}/f_W$
1	$2^5/f_W$

WTM1	5-Bit Counter Operation Control
0	Clear after operation stop
1	Start

WTM0	Watch Timer Enables Operation
0	Operation stop (clear both prescaler and timer)
1	Operation enable

Caution Do not change the count clock, interval time, and interrupt request time (by using bits 3 to 7 (WTM3 to WTM7) of WTM) while the watch timer is operating.

Remarks 1. f_W : Watch timer clock frequency ($f_X/2^7$ or f_{XT})

2. f_X : Main system clock oscillation frequency

3. f_{XT} : Subsystem clock oscillation frequency

4. Figures in parentheses apply to operation with f_X = 8.38 MHz, f_{XT} = 32.768 kHz.

10.5 Watch Timer Operations

10.5.1 Watch timer operation

The watch timer generates an interrupt request (INTWT) at specific time intervals ($2^{14}/f_W$ seconds or $2^5/f_W$ seconds) by using the main system clock or subsystem clock. The interrupt request is generated at the following time intervals (where WTM3 = 0):

- If main system clock (8.38 MHz) is selected: 0.25 seconds
- If subsystem clock (32.768 kHz) is selected: 0.5 seconds

When bit 0 (WTM0) and bit 1 (WTM1) of the watch timer operation mode register (WTM) is set to 1, the 5-bit counter is cleared and the count operation stops.

When the interval timer is simultaneously operated, zero-second start can be achieved only for the watch timer by setting WTM1 to 0. In this case, however, the 9-bit prescaler is not cleared.

Therefore, an error up to $2^9 \times 1/f_W$ seconds occurs in the first overflow (INTWT) after zero-second start.

Remark f_W : Watch timer clock frequency

10.5.2 Interval timer operation

The watch timer operates as interval timer which generates interrupt requests (INTWTI) repeatedly at an interval of the preset count value.

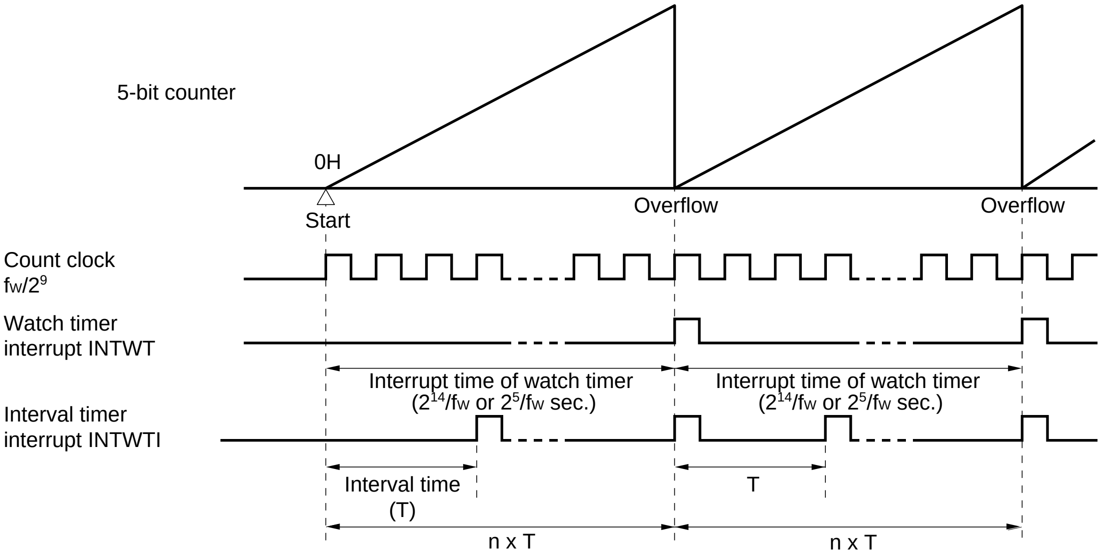
The interval time can be selected with bits 4 to 6 (WTM4 to WTM6) of the watch timer operation mode register (WTM).

Table 10-3. Interval Timer Interval Time

WTM6	WTM5	WTM4	Interval Time	When Operated at f_X = 8.38 MHz	When Operated at f_X = 4.19 MHz	When Operated at f_{XT} = 32.768 kHz
0	0	0	$2^4 \times 1/f_W$	244 μs	489 μs	488 μs
0	0	1	$2^5 \times 1/f_W$	489 μs	978 μs	977 μs
0	1	0	$2^6 \times 1/f_W$	978 μs	1.96 ms	1.95 ms
0	1	1	$2^7 \times 1/f_W$	1.96 ms	3.91 ms	3.91 ms
1	0	0	$2^8 \times 1/f_W$	3.91 ms	7.82 ms	7.81 ms
1	0	1	$2^9 \times 1/f_W$	7.82 ms	15.6 ms	15.6 ms
Other than above			Setting prohibited			

Remark f_X : Main system clock oscillation frequency
f_{XT} : Subsystem clock oscillation frequency
f_W : Watch timer clock frequency

Figure 10-3. Operation Timing of Watch Timer/Interval Timer



Remark fw : Watch timer clock frequency

n : The number of times of interval timer operations

Caution **If the watch timer and 5-bit counter are enabled by the watch timer mode control register (WTM) (by setting bits 0 (WTM0) and 1 (WTM1) of WTM to 1), the time from this setting to the occurrence of the first interrupt request (INTWT) is not exactly the value set by bit 3 (WTM3) of WTM. This is because the 5-bit counter is late by one output cycle of the 9-bit prescaler in starting counting. The second INTWT signal and those that follow are generated exactly at the set time.**

[MEMO]

11.1 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 the preset time intervals.

11.2 Watchdog Timer Functions

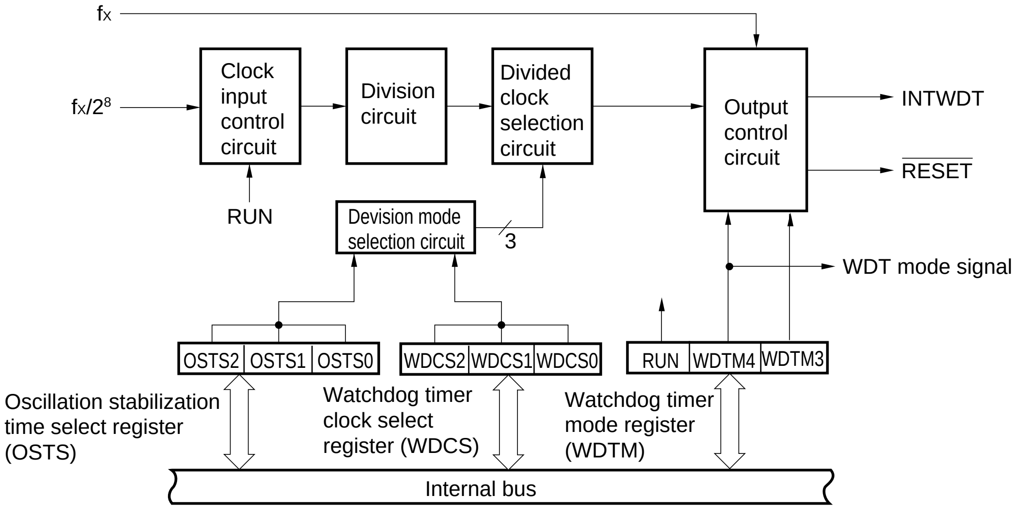
The watchdog timer has the following functions.

- Watchdog timer
- Interval timer
- Oscillation stabilization time selection

Caution Select the watchdog timer mode or the interval timer mode with the watchdog timer mode register (WDTM). (The watchdog timer and the interval timer cannot be used simultaneously.)

Figure 11-1 shows a block diagram of the watchdog timer.

Figure 11-1. Watchdog Timer Block Diagram


fX
fX/2^8
Clock input control circuit
Division circuit
Divided clock selection circuit
Output control circuit
INTWDT
RESET
RUN
Devision mode selection circuit
3
WDT mode signal
OSTS2
OSTS1
OSTS0
WDCS2
WDCS1
WDCS0
RUN
WDTM4
WDTM3
Oscillation stabilization time select register (OSTS)
Watchdog timer clock select register (WDCS)
Watchdog timer mode register (WDTM)
Internal bus

(1) Watchdog timer mode

A runaway is detected. Upon detection of the runaway, a non-maskable interrupt request or $\overline{\text{RESET}}$ can be generated.

Table 11-1. Watchdog Timer Runaway Detection Time

Runaway Detection Time
$2^{12} \times 1/f_X$ (489 μs)
$2^{13} \times 1/f_X$ (978 μs)
$2^{14} \times 1/f_X$ (1.96 ms)
$2^{15} \times 1/f_X$ (3.91 ms)
$2^{16} \times 1/f_X$ (7.82 ms)
$2^{17} \times 1/f_X$ (15.6 ms)
$2^{18} \times 1/f_X$ (31.3 ms)
$2^{20} \times 1/f_X$ (125 ms)

Remarks 1. f_X: Main system clock oscillation frequency

2. Figures in parentheses are for operation with f_X = 8.38 MHz

(2) Interval timer mode

Interrupt requests are generated at the preset time intervals.

Table 11-2. Interval Time

Interval Time
$2^{12} \times 1/f_X$ (489 μs)
$2^{13} \times 1/f_X$ (978 μs)
$2^{14} \times 1/f_X$ (1.96 ms)
$2^{15} \times 1/f_X$ (3.91 ms)
$2^{16} \times 1/f_X$ (7.82 ms)
$2^{17} \times 1/f_X$ (15.6 ms)
$2^{18} \times 1/f_X$ (31.3 ms)
$2^{20} \times 1/f_X$ (125 ms)

Remarks 1. f_X: Main system clock oscillation frequency

2. Figures in parentheses are for operation with f_X = 8.38 MHz

11.3 Watchdog Timer Configuration

The watchdog timer consists of the following hardware.

Table 11-3. Watchdog Timer Configuration

Item	Configuration
Control registers	Watchdog timer clock select register (WDCS) Watchdog timer mode register (WDTM) Oscillation stabilization time select register (OSTS)

11.4 Registers to Control the Watchdog Timer

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

- Watchdog timer clock select register (WDCS)
- Watchdog timer mode register (WDTM)
- Oscillation stabilization time select register (OSTS)

(1) Watchdog timer clock select register (WDCS)

This register sets overflow time of the watchdog timer and the interval timer.

WDCS is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the value of WDCS to 00H.

Figure 11-2. Watchdog Timer Clock Select Register (WDCS) Format

Address: FF42H At 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	$2^{12}/f_X$ (489 μs)
0	0	1	$2^{13}/f_X$ (978 μs)
0	1	0	$2^{14}/f_X$ (1.96 ms)
0	1	1	$2^{15}/f_X$ (3.91 ms)
1	0	0	$2^{16}/f_X$ (7.82 ms)
1	0	1	$2^{17}/f_X$ (15.6 ms)
1	1	0	$2^{18}/f_X$ (31.3 ms)
1	1	1	$2^{20}/f_X$ (125 ms)

Remarks 1. f_X: Main system clock oscillation frequency

2. Figures in parentheses are for operation with f_X = 8.38 MHz

(2) Watchdog timer mode register (WDTM)

This register sets the watchdog timer operating mode and enables/disables counting.

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

$\overline{\text{RESET}}$ input sets the value of WDTM to 00H.

Figure 11-3. Watchdog Timer Mode Register (WDTM) Format

Address: FFF9H At 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 stop
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 an overflow)
1	0	Watchdog timer mode 1 (Non-maskable interrupt request occurs upon generation of an overflow)
1	1	Watchdog timer mode 2 (Reset operation is activated upon generation of an 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 operations as the interval timer when 1 is set to RUN.

Caution When 1 is set to RUN so that the watchdog timer is cleared, the actual overflow time is up to 0.5% shorter than the time set by watchdog timer clock select register (WDCS).

Remark × : Don’t care

(3) Oscillation stabilization time select register (OSTS)

A register to select oscillation stabilization time from reset time or STOP mode released time to the time when oscillation is stabilized.

OSTS is set by an 8-bit memory manipulation instruction.

By $\overline{\text{RESET}}$ input, it is turned into 04H. Thus, when releasing the STOP mode by $\overline{\text{RESET}}$ input, the time required to release is 2^{17}/fx.

Figure 11-4. Oscillation Stabilization Time Select Register (OSTS) Format

Address: FFFAH At 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	$2^{12}/f_X$ (488 μs)
0	0	1	$2^{14}/f_X$ (1.95 ms)
0	1	0	$2^{15}/f_X$ (3.91 ms)
0	1	1	$2^{16}/f_X$ (7.81 ms)
1	0	0	$2^{17}/f_X$ (15.6 ms)
Other than the above			Setting prohibited

Remarks 1. f_X: Main system clock oscillation frequency

2. Figures in parentheses are for operation with f_X = 8.38 MHz

11.5 Watchdog Timer Operations

11.5.1 Watchdog timer operation

When bit 4 (WDTM4) of the watchdog timer mode register (WDTM) is set to 1, the watchdog timer is operated to detect any runaway.

The runaway detection time interval is selected with bits 0 to 2 (WDCS0 to WDCS2) of the watchdog timer clock select register (WDCS).

Watchdog timer starts by setting bit 7 (RUN) of WDTM to 1. After the watchdog timer is started, set RUN to 1 within the set runaway time interval. The watchdog timer can be cleared and counting is started by setting RUN to 1. If RUN is not set to 1 and the runaway detection time is exceeded, system reset or a non-maskable interrupt request is generated according to WDTM bit 3 (WDTM3) value.

The watchdog timer continues operating in the HALT mode but it stops in the STOP mode. Thus, set RUN to 1 before the STOP mode is set, clear the watchdog timer and then execute the STOP instruction.

Cautions 1. The actual runaway detection time may be shorter than the set time by a maximum of 0.5%.

2. When the subsystem clock is selected for CPU clock, watchdog timer count operation is stopped.

Table 11-4. Watchdog Timer Runaway Detection Time

Runaway Detection Time
$2^{12} \times 1/f_X$ (489 μs)
$2^{13} \times 1/f_X$ (978 μs)
$2^{14} \times 1/f_X$ (1.96 ms)
$2^{15} \times 1/f_X$ (3.91 ms)
$2^{16} \times 1/f_X$ (7.82 ms)
$2^{17} \times 1/f_X$ (15.6 ms)
$2^{18} \times 1/f_X$ (31.3 ms)
$2^{20} \times 1/f_X$ (125 ms)

Remarks 1. f_X: Main system clock oscillation frequency

2. Figures in parentheses are for operation with f_X = 8.38 MHz.

11.5.2 Interval timer operation

The watchdog timer operates as an interval timer which generates interrupt requests repeatedly at an interval of the 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 with bits 0 to 2 (WDCS0 to WDCS2) of the watchdog timer clock select register (WDCS). When 1 is set to bit 7 (RUN) of WDTM, the watchdog timer operates as the interval timer.

When the watchdog timer operated as the interval timer, the interrupt mask flag (WDTMK) and priority specify flag (WDTPR) are validated and the maskable interrupt request (INTWDT) can be generated. Among maskable interrupts, INTWDT has the highest priority at default.

The interval timer continues operating in the HALT mode but it stops in STOP mode. Thus, set RUN to 1 before the STOP mode is set, 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 $\overline{\text{RESET}}$ input is applied.

2. The interval time just after setting by WDTM may be shorter than the set time by a maximum of 0.5%.

3. When the subsystem clock is selected for CPU clock, watchdog timer count operation is stopped.

Table 11-5. Interval Timer Interval Time

Interval Time
$2^{12} \times 1/f_X$ (489 μs)
$2^{13} \times 1/f_X$ (978 μs)
$2^{14} \times 1/f_X$ (1.96 ms)
$2^{15} \times 1/f_X$ (3.91 ms)
$2^{16} \times 1/f_X$ (7.82 ms)
$2^{17} \times 1/f_X$ (15.6 ms)
$2^{18} \times 1/f_X$ (31.3 ms)
$2^{20} \times 1/f_X$ (125 ms)

Remarks 1. f_X: Main system clock oscillation frequency

2. Figures in parentheses are for operation with f_X = 8.38 MHz.

12.1 Outline of Clock Output/Buzzer

The clock output circuit supplies other devices with the divided main system clock and the subsystem clock, and buzzer output supplies the buzzer frequency with the divided main system clock.

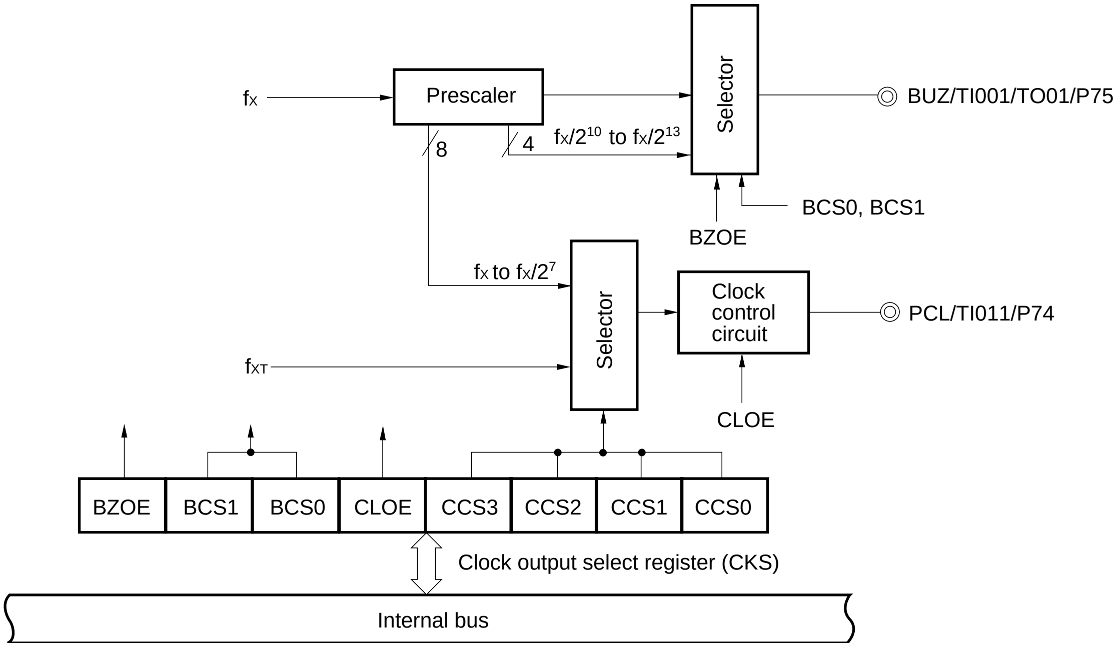
12.2 Clock Output/Buzzer Output Control Circuit Functions

The clock output control circuit is intended for carrier output during remote controlled transmission and clock output for supply to peripheral LSIs. The clock selected with the clock output selection register (CKS) is output.

In addition, the buzzer output is intended for square wave output of buzzer frequency selected with CKS.

Figure 12-1 shows the block diagram of clock output/buzzer output control circuits.

Figure 12-1. Clock Output/Buzzer Output Control Circuit Block Diagram



12.3 Clock Output/Buzzer Output Control Circuit Configuration

The clock output/buzzer output control circuits consists of the following hardware.

Table 12-1. Clock Output/Buzzer Output Control Circuits Configuration

Item	Configuration
Control registers	Clock output select register (CKS) Port mode register (PM7)[Note]

Note See **Figure 6-15 P70 to P75 Block Diagram**.

12.4 Register to Control Clock Output/Buzzer Output Control Circuit

The following two types of registers are used to control the clock output/buzzer output control circuits.

- Clock output select register (CKS)
- Port mode register (PM7)

(1) Clock output select register (CKS)

This register sets output enable/disable for clock output (PCL) and for the buzzer frequency output (BUZ), and sets the output clock.

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

$\overline{\text{RESET}}$ input sets the value of CKS to 00H.

Figure 12-2. Clock Output Select Register (CKS) Format

Address: FF40H At Reset: 00H R/W

Symbol	$\boxed{7}$	6	5	$\boxed{4}$	3	2	1	0
CKS	BZOE	BCS1	BCS0	CLOE	CCS3	CCS2	CCS1	CCS0

BZOE	BUZ Output Enable/Disable Specification
0	Stop clock division circuit operation. BUZ fixed to low level.
1	Enable clock division circuit operation. BUZ output enabled.

BCS1	BCS0	BUZ Output Clock Selection
0	0	$f_X/2^{10}$ (8.18 kHz)
0	1	$f_X/2^{11}$ (4.09 kHz)
1	0	$f_X/2^{12}$ (2.04 kHz)
1	1	$f_X/2^{13}$ (1.02 kHz)

CLOE	PCL Output Enable/Disable Setting
0	Stop clock division circuit operation. PCL fixed to low level
1	Enable clock division circuit operation. PCL output enabled.

CCS3	CCS2	CCS1	CCS0	PCL Output Clock Selection
0	0	0	0	f_X (8.38 MHz)
0	0	0	1	$f_X/2$ (4.19 MHz)
0	0	1	0	$f_X/2^2$ (2.09 MHz)
0	0	1	1	$f_X/2^3$ (1.04 MHz)
0	1	0	0	$f_X/2^4$ (524 kHz)
0	1	0	1	$f_X/2^5$ (262 kHz)
0	1	1	0	$f_X/2^6$ (131 kHz)
0	1	1	1	$f_X/2^7$ (65.5 kHz)
1	0	0	0	f_{XT} (32.768 kHz)
Other than above				Setting prohibited

Remarks 1. f_X : Main system clock oscillation frequency

2. f_{XT} : Subsystem clock oscillation frequency

3. Figures in parentheses are for operation with f_X = 8.38 MHz or f_{XT} = 32.768 kHz.

(2) Port mode register (PM7)

This register sets port 7 input/output in 1-bit units.

When using the P74/PCL/TI011 pin for clock output and the P75/BUZ/TI001/TO01 pin for buzzer output, set PM74, PM75 and the output latch of P74, P75 to 0.

PM7 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the value of PM7 to FFH.

Figure 12-3. Port Mode Register 7 (PM7) Format

Address: FF27H At 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 Input/Output Mode Selection (n = 0 to 5)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

12.5 Clock Output/Buzzer Output Control Circuit Operations

12.5.1 Operation as clock output

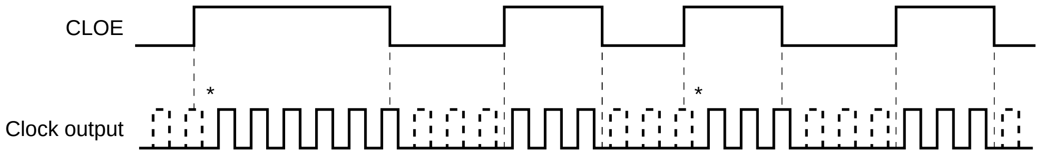
The clock pulse is output as the following procedure.

<1> Select the clock pulse output frequency with bits 0 to 3 (CCS0 to CCS3) of the clock output select register (CKS) (clock pulse output in disabled status).

<2> Set bit 4 (CLOE) of CKS to 1, and enable clock output.

Remark The clock output control circuit is designed not to output pulses with a small width during output enable/disable switching of the clock output. As shown in Figure 12-4, be sure to start output from the low period of the clock (marked with ***** in the figure). When stopping output, do so after securing high level of the clock.

Figure 12-4. Remote Control Output Application Example



12.5.2 Operation as buzzer output

The buzzer frequency is output as the following procedure.

<1> Select the buzzer output frequency with bits 5 and 6 (BCS0, BCS1) of the clock output select register (CKS) (buzzer output in disabled status).

<2> Set bit 7 (BZOE) of CKS to 1 to enable buzzer output.

[MEMO]

13.1 A/D Converter Functions

A/D converter is a 10-bit resolution converter that converts analog inputs into digital signals. It can control up to 8 analog input channels (ANI0 to ANI7).

(1) Hardware start

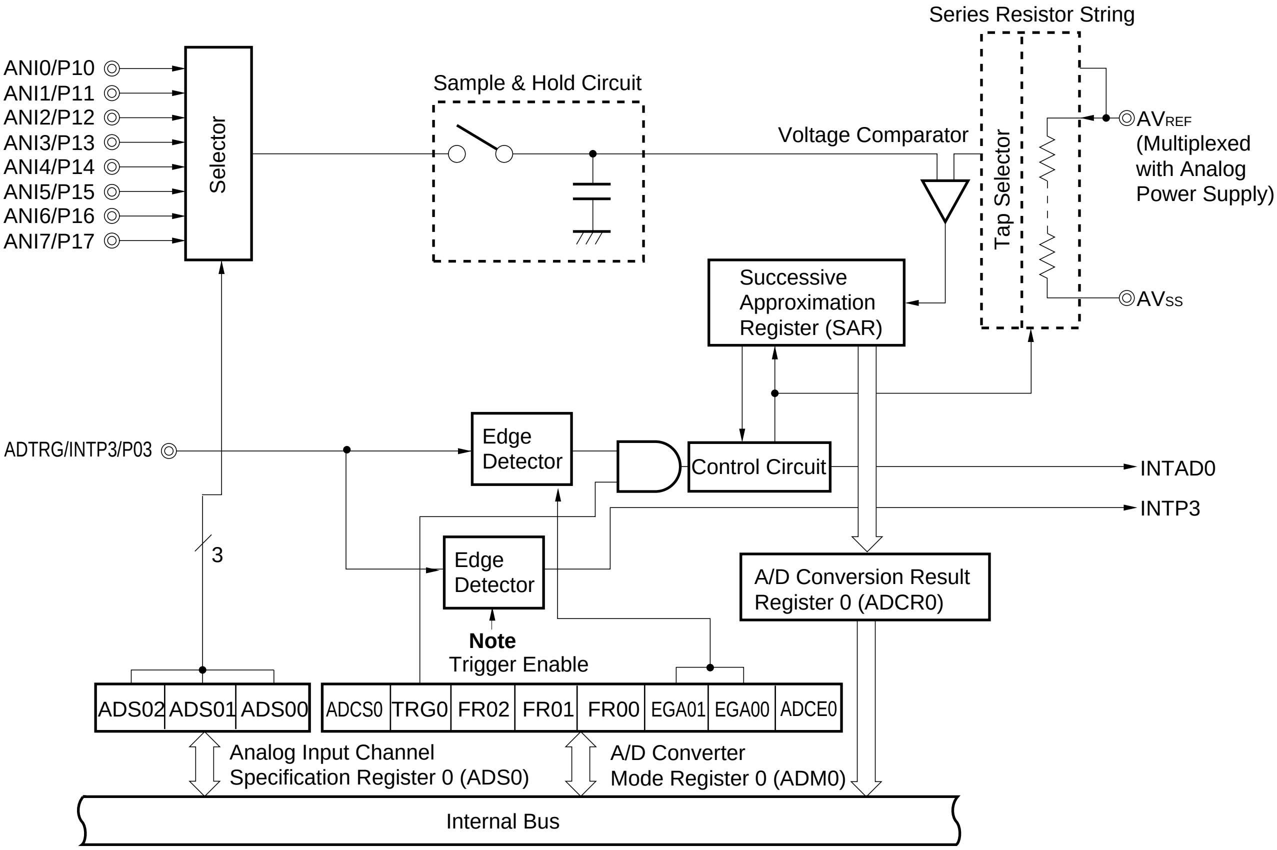
Conversion is started by trigger input (ADTRG: rising edge, falling edge, or both rising and falling edges can be specified).

(2) Software start

Conversion is started by setting the A/D converter mode register 0 (ADM0).

Select one channel for analog input from ANI0 to ANI7 to start A/D conversion. In the case of hardware start, the A/D converter stops when A/D conversion is completed, and an interrupt request (INTAD0) is generated. In the case of software start, A/D conversion is repeated. Each time as A/D conversion operation ends, an interrupt request (INTAD0) is generated.

Figure 13-1. 10-Bit A/D Converter Block Diagram



Note The valid edge is specified by bit 3 of the EGP and EGN registers (See **Figure 19-5 External Interrupt Rising Edge Enable Register (EGP), External Interrupt Falling Edge Enable Register (EGN) Format**.)

13.2 A/D Converter Configuration

A/D converter consists of the following hardware.

Table 13-1. A/D Converter Configuration

Item	Configuration
Analog input	8 channels (ANI0 to ANI7)
Registers	Successive approximation register (SAR) A/D conversion result register 0 (ADCR0)
Control registers	A/D converter mode register 0 (ADM0) Analog input channel specification register 0 (ADS0) External interrupt rising edge enable register (EGP) External interrupt falling edge enable register (EGN)

(1) Successive approximation register (SAR)

This register compares the analog input voltage value to the voltage tap (compare value) 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 hold (end of A/D conversion), the SAR contents are transferred to the A/D conversion result register 0 (ADCR0).

(2) A/D conversion result register 0 (ADCR0)

This is a 16-bit register which stores the A/D conversion results. Lower 6-bit is fixed to 0. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register (SAR) and held by this register. ADCR0 is read by a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ADCR0 to 00H.

Caution When writing is performed to the A/D converter mode register 0 (ADM0) and analog input channel specification register 0 (ADS0), the contents of ADCR0 may become undefined. Read the conversion result following conversion completion before writing to ADM0, ADS0. 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_{REF} and AV_{SS}, and generates a voltage to be compared to the analog input.

(6) ANI0 to ANI7 pins

These are eight analog input pins to input analog signals to undergo A/D conversion to the A/D converter. ANI0 to ANI7 are dual-function pins that can also be used for digital input.

Cautions 1. Use ANI0 to ANI7 input voltages within the specification range. If a voltage higher than AV_{REF} or lower than AV_{SS} 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 (ANI0 to ANI7) pins are alternate function pins that can also be used as input port (P10 to P17) pins. When A/D conversion is performed by selecting any one of ANI0 through ANI7, do not execute any input instruction to port 1 during conversion. It may cause the lower conversion resolution.

When a digital pulse is applied to a pin adjacent to the pin in the process of A/D conversion, A/D conversion values may not be obtained as expected due to coupling noise. Thus, do not apply any pulse to a pin adjacent to the pin in the process of A/D conversion.

(7) AV_{REF} pin

This is an A/D converter reference voltage pin and also a analog power supply pin.

It converts signals input to ANI0 to ANI7 into digital signals according to the voltage applied between AV_{REF} and AV_{SS}.

Caution A series resistor string is connected between the AV_{REF} and AV_{SS} pins. Therefore, when output impedance of the reference voltage is too high, it seems as if the AV_{REF} pin and the series resistor string are connected in series. This may cause a greater reference voltage error.

(8) AV_{SS} pin

This is the GND potential pin of the A/D converter. Always keep it at the same potential as the V_{SS0} pin when not using the A/D converter.

13.3 Registers to Control A/D Converter

The following 4 types of registers are used to control A/D Converter.

- A/D converter mode register 0 (ADM0)
- Analog input channel specification register 0 (ADS0)
- External interrupt rising edge enable register (EGP)
- External interrupt falling edge enable register (EGN)

(1) A/D converter mode register 0 (ADM0)

This register sets the conversion time for analog input to be A/D converted, conversion start/stop, and external trigger.

ADM0 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the value of ADM0 to 00H.

Figure 13-2. A/D Converter Mode Register 0 (ADM0) Format

Address: FF80H At Reset: 00H R/W

Symbol	$\boxed{7}$	$\boxed{6}$	5	4	3	2	1	$\boxed{0}$
ADM0	ADCS0	TRG0	FR02	FR01	FR00	EGA01	EGA00	ADCE0

ADCS0	A/D Conversion Operation Control
0	Stop conversion operation.
1	Enable conversion operation.

TRG0	Software Start/Hardware Start Selection
0	Software start
1	Hardware start

FR02	FR01	FR00	Conversion Time Selection[Note 1]
0	0	0	144/f_X (17.1 μs)
1	0	1	120/f_X (14.3 μs)
0	1	0	96/f_X (Setting prohibited[Note 2])
1	0	0	72/f_X (Setting prohibited[Note 2])
1	0	1	60/f_X (Setting prohibited[Note 2])
1	1	0	48/f_X (Setting prohibited[Note 2])
Other than above			Setting prohibited

EGA01	EGA00	External Trigger Signal, Edge Specification
0	0	No edge detection
0	1	Falling edge detection
1	0	Rising edge detection
1	1	Both falling and rising edge detection

ADCE0	Voltage Booster for A/D Converter Circuit Control[Note 3].
0	Stops operation.
1	Enables operation.

Notes 1. Set so that the A/D conversion time is 14 μs or more.

2. Setting prohibited because A/D conversion time is less than 14 μs.

3. Before executing A/D conversion (ADCS0 = 1), be sure to start the voltage booster (ADCE0 = 1).

Cautions 1. When rewrite FR00 to FR02 to other than the same data, stop A/D conversion operations once before performing it.

2. In software, make sure that a wait time of 14 μs (MIN.) elapses from when ADCE0 is set to when ADCS0 is set.

3. Before clearing ADCE0, clear ADCS0.

Remarks 1. f_X: Main system clock oscillation frequency

2. Figures in parentheses are for operation with f_X = 8.38 MHz.

(2) Analog input channel specification register 0 (ADS0)

This register specifies the analog voltage input port for A/D conversion.

ADS0 is set by an 8-bit memory manipulation instruction.

RESET input sets the value of ADS0 to 00H.

Figure 13-3. Analog Input Channel Specification Register 0 (ADS0) Format

Address: FF81H At Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS0	0	0	0	0	0	ADS02	ADS01	ADS00

ADS02	ADS01	ADS00	Analog Input Channel Specification
0	0	0	ANI0
0	0	1	ANI1
0	1	0	ANI2
0	1	1	ANI3
1	0	0	ANI4
1	0	1	ANI5
1	1	0	ANI6
1	1	1	ANI7

(3) External interrupt rising edge enable register (EGP), external interrupt falling edge enable register (EGN)

These registers specify the valid edge for INTP0 to INTP3.

EGP and EGN are set by a 1-bit or 8-bit memory manipulation instruction.

RESET input sets the values of EGP and EGN to 00H.

Figure 13-4. External Interrupt Rising Edge Enable Register (EGP), External Interrupt Falling Edge Enable Register (EGN) Format

Address: FF48H At Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP	0	0	0	0	EGP3	EGP2	EGP1	EGP0

Address: FF49H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN	0	0	0	0	EGN3	EGN2	ENG1	ENG0

EGPn	EGNn	INTPn Pin Valid Edge Selection (n = 0 to 3)
0	0	Interrupt disable
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

13.4 A/D Converter Operation

13.4.1 Basic operations of A/D converter

<1> Select one channel for A/D conversion with the analog input channel specification register 0 (ADS0).

<2> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<3> When sampling has been done 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 is ended.

<4> Bit 9 of the successive approximation register (SAR) is set. The series resistor string voltage tap is set to (1/2) AV_{REF} by the tap selector.

<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_{REF}, the MSB of SAR remains set. If the analog input is smaller than (1/2) AV_{REF}, 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 9, as described below.

- Bit 9 = 1: (3/4) AV_{REF}
- Bit 9 = 0: (1/4) AV_{REF}

The voltage tap and analog input voltage are compared and bit 8 of SAR is manipulated as follows.

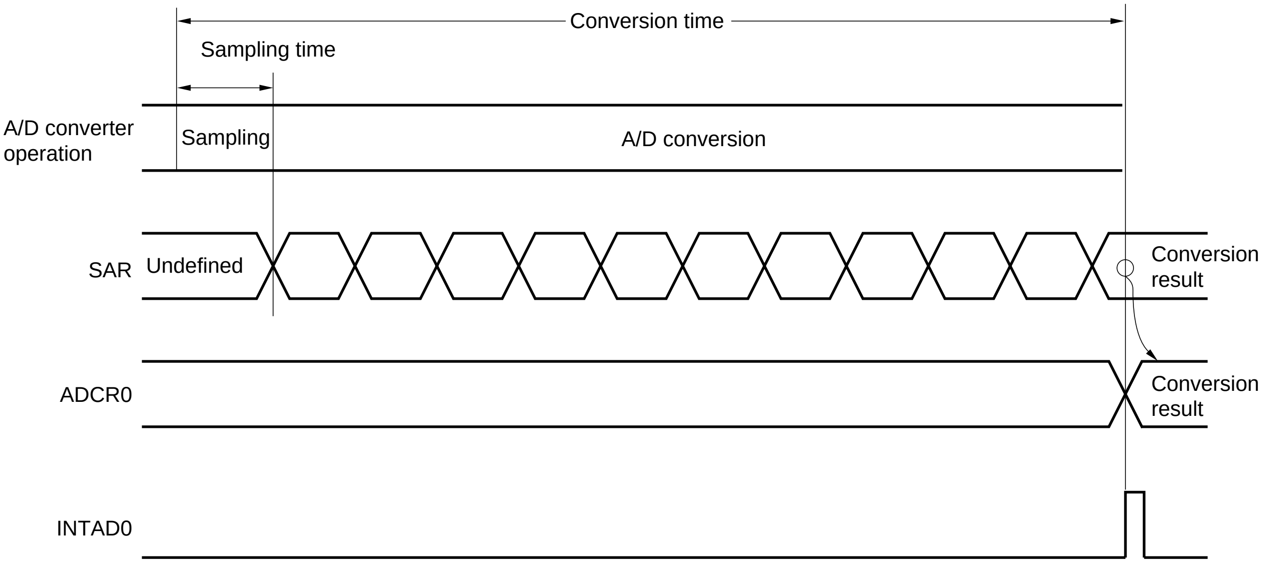
- Analog input voltage ≥ Voltage tap: Bit 8 = 1
- Analog input voltage < Voltage tap: Bit 8 = 0

<7> Comparison is continued in this way up to bit 0 of SAR.

<8> Upon completion of the comparison of 10 bits, an effective digital result value remains in SAR, and the result value is transferred to and latched in the A/D conversion result register 0 (ADCR0).
At the same time, the A/D conversion end interrupt request (INTAD0) can also be generated.

Caution The first A/D conversion value just after A/D conversion operations start may not fall within the rating.

Figure 13-5. Basic Operation of A/D Converter



A/D conversion operations are performed continuously until bit 7 (ADCS0) of the A/D converter mode register 0 (ADM0) is reset (0) by software.

If a write operation is performed to the ADM0 or the analog input channel specification register 0 (ADS0) during an A/D conversion operation, the conversion operation is initialized, and if the ADCS0 bit is set (1), conversion starts again from the beginning.

$\overline{\text{RESET}}$ input sets the A/D conversion result register 0 (ADCR0) to 00H.

13.4.2 Input voltage and conversion results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI7) and the A/D conversion result (stored in the A/D conversion result register 0 (ADCR0)) is shown by the following expression.

$$\text{ADCR0} = \text{INT}\left(\frac{V_{IN}}{AV_{REF}} \times 1024 + 0.5\right)$$

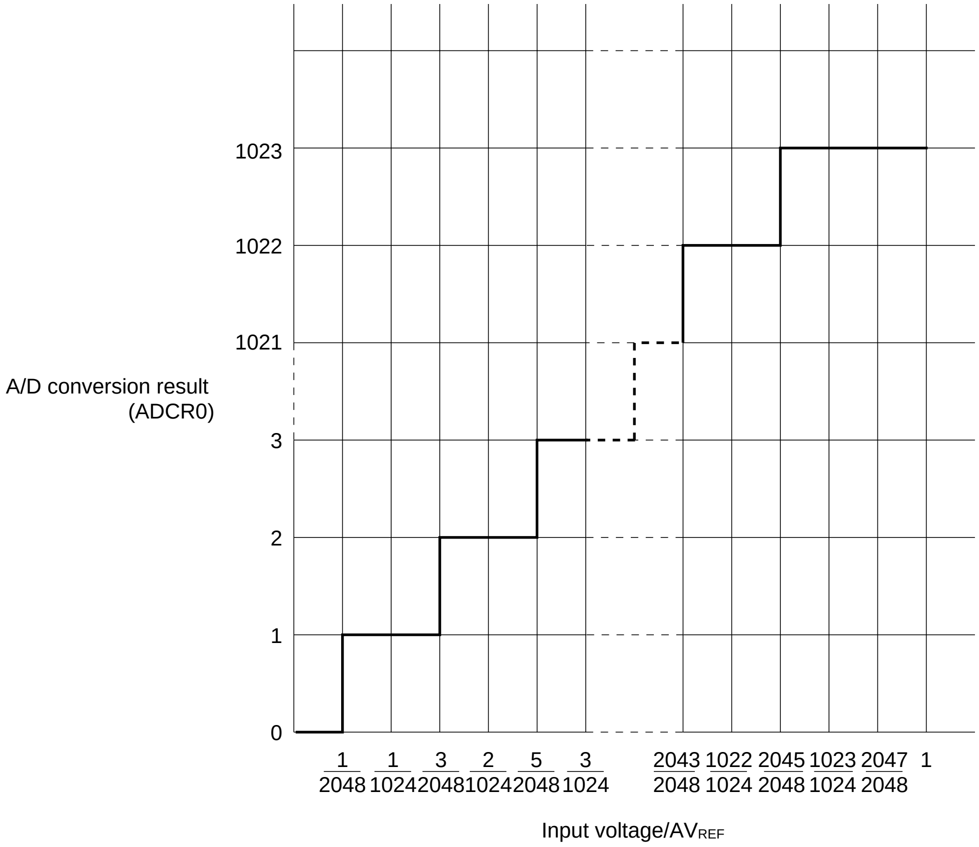
or

$$(\text{ADCR0} - 0.5) \times \frac{AV_{REF}}{1024} \le V_{IN} < (\text{ADCR0} + 0.5) \times \frac{AV_{REF}}{1024}$$

where, INT( ) : Function which returns integer part of value in parentheses
V_{IN} : Analog input voltage
AV_{REF} : AV_{REF} pin voltage
ADCR0 : A/D conversion result register 0 (ADCR0) value

Figure 13-6 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 13-6. Relationship between Analog Input Voltage and A/D Conversion Result



13.4.3 A/D converter operation mode

Select one analog input channel from among ANI0 to ANI7 by the analog input channel specification register 0 (ADS0) to start A/D conversion.

A/D conversion can be started in either of the following two ways.

- Hardware start : Conversion is started by trigger input (rising edge, falling edge, or both rising and falling edges enabled).
- Software start : Conversion is started by specifying the A/D converter mode register 0 (ADM0).

The A/D conversion result is stored in the A/D conversion result register 0 (ADCR0), and the interrupt request signal (INTAD0) is simultaneously generated.

(1) A/D conversion by hardware start

When bit 6 (TRG0) and bit 7 (ADCS0) of the A/D converter mode register 0 (ADM0) are set to 1 after bit 0 (ADCE0) is set to 1, the A/D conversion standby state is set. When the external trigger signal (ADTRG) is input, A/D conversion of the voltage applied to the analog input pin specified by the analog input channel specification register 0 (ADS0) starts.

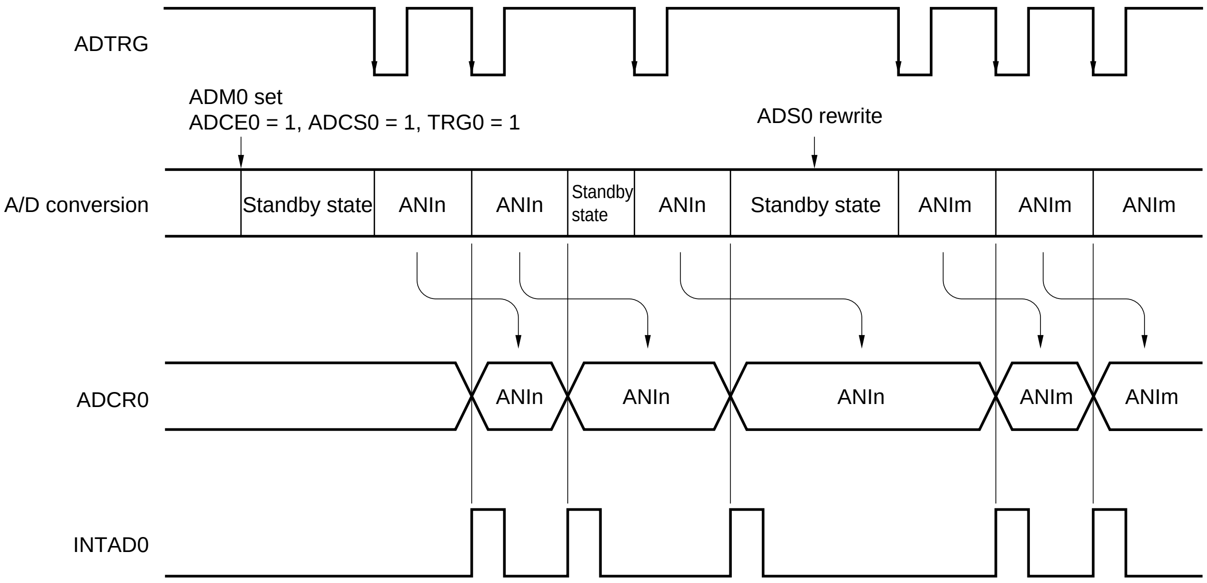
Upon the end of the A/D conversion, the conversion result is stored in the A/D conversion result register (ADCR0), and the interrupt request signal (INTAD0) is generated. After one A/D conversion operation is started and ended, the next conversion operation is not started until a new external trigger signal is input.

If ADS0 is rewritten during A/D conversion operation, the converter suspends A/D conversion and waits for a new external trigger signal to be input. When the external trigger input signal is reinput, A/D conversion is carried out from the beginning. If ADS0 is rewritten during A/D conversion waiting, A/D conversion starts when the following external trigger input signal is input.

If data with ADCS0 set to 0 is written to ADM0 during A/D conversion, the A/D conversion operation stops immediately.

Caution When P03/INTP3/ADTRG is used as the external trigger input (ADTRG), specify the valid edge by bits 1, 2 (EGA00, EGA01) of A/D converter mode register 0 (ADM0) and set the interrupt mask flag (PMK3) to 1.

Figure 13-7. A/D Conversion by Hardware Start (When Falling Edge Is Specified)



Remarks 1. n = 0, 1, ..., 7

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

(2) A/D conversion by software start

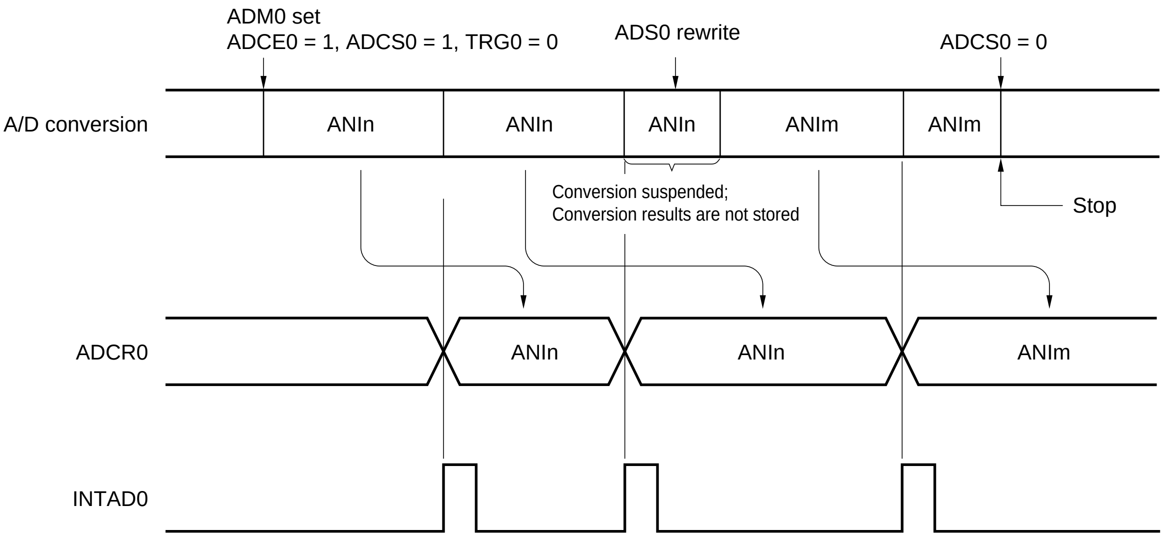
When bit 6 (TRG0) and bit 7 (ADCS0) of the A/D converter mode register 0 (ADM0) are set to 0 and 1 after bit 0 (ADCE0) is set to 1, respectively, A/D conversion of the voltage applied to the analog input pin specified by the analog input channel specification register 0 (ADS0) starts.

Upon the end of the A/D conversion, the conversion result is stored in the A/D conversion result register 0 (ADCR0), and the interrupt request signal (INTAD0) 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 ADS0.

If ADS0 is rewritten during A/D conversion, the converter suspends A/D conversion operation and A/D conversion of the new selected analog input channel starts.

If data with ADCS0 set to 0 is written to ADM0 during A/D conversion, the A/D conversion operation stops immediately.

Figure 13-8. A/D Conversion by Software Start



Remarks 1. n = 0, 1, ..., 7

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

13.5 How to Read the A/D Converter Characteristics Table

Here we will explain the special terms unique to A/D converters.

(1) Resolution

This is the minimum analog input voltage that can be identified. That is, the percentage of the analog input voltage per 1 bit of digital output is called 1LSB (Least Significant Bit). The percentage of 1LSB with respect to the full scale is expressed by %FSR (Full Scale Range).

When the resolution is 10 bits,

$$\begin{aligned} 1\text{LSB} &= 1/2^{10} = 1/1024 \\ &= 0.098\%\text{FSR} \end{aligned}$$

Accuracy has no relation to resolution, but is determined by overall error.

(2) Overall error

This shows the maximum error value between the actual measured value and the theoretical value.

Zero scale offset, full scale offset, integral linearity error, differential linearity error and errors which are combinations of these express overall error.

Furthermore, quantization error is not included in overall error in the characteristics table.

(3) Quantization error

When analog values are converted to digital values, there naturally occurs a ±1/2LSB error. In an A/D converter, an analog input voltage in a range of ±1/2LSB are converted to the same digital code, so a quantization error cannot be avoided.

Furthermore, it is not included in the overall error, zero scale offset, full scale offset, integral linearity error, and differential linearity error in the characteristics table.

Figure 13-9. Overall Error

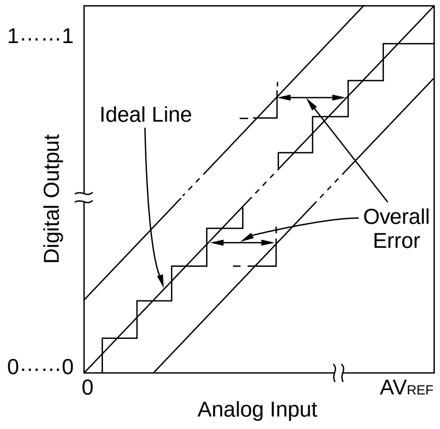
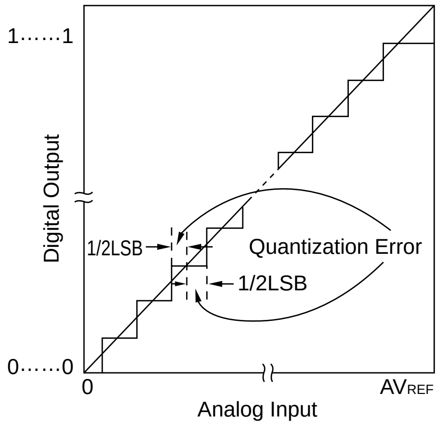


Figure 13-10. Quantization Error



(4) Zero scale offset

This shows the difference between the actual measured value of the analog input voltage and the theoretical value (1/2 LSB) when the digital output changes from 0……000 to 0……001. If the actual measured value is greater than the theoretical value, it shows the difference between the actual measured value of the analog input voltage and the theoretical value (3/2LSB) when the digital output changes from 0……001 to 0……010.

(5) Full scale offset

This shows the difference between the actual measured value of the analog input voltage and the theoretical value (full scalle –3/2 LSB) when the digital output changes from 1……110 to 1……111.

(6) Integral linearity error

This shows the degree to which the conversion characteristics deviate from the ideal linear relationship. It expresses the maximum value of the difference between the actual measured value and the ideal straight line when the zero scale offset and full scale offset are 0.

(7) Differential linearity error

Although the ideal output width for a given code is 1LSB, this value shows the difference between the actual measured value and the ideal value of the width when outputting a particular code.

Figure 13-11. Zero Scale Offset

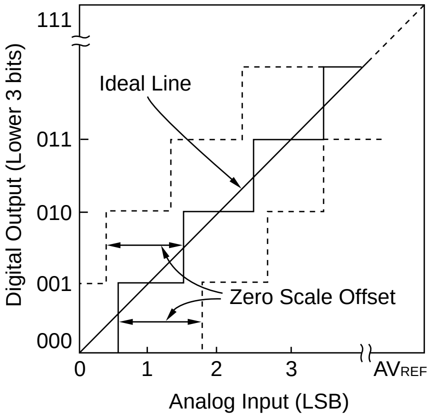


Figure 13-12. Full Scale Offset

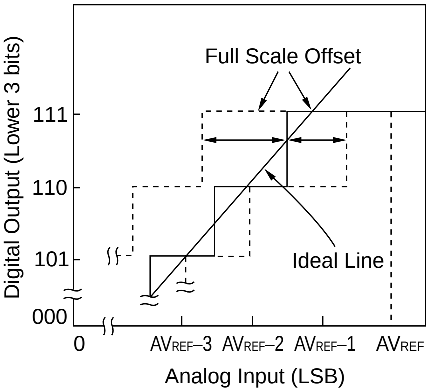


Figure 13-13. Integral Linearity Error

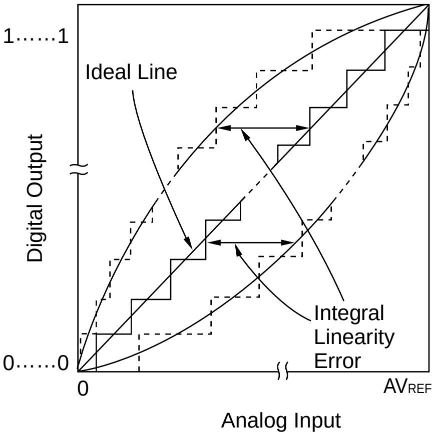
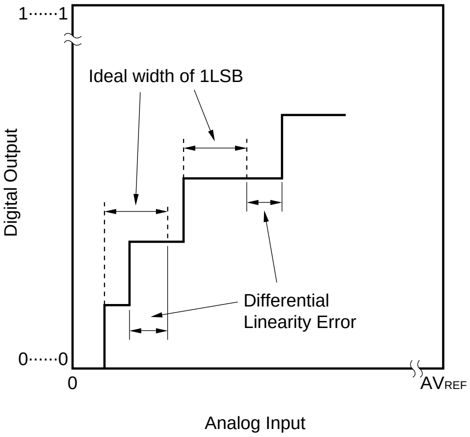


Figure 13-14. Differential Linearity Error



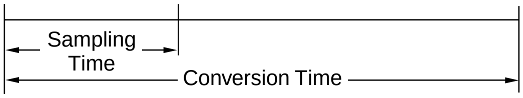
(8) Conversion time

This expresses the time from when the analog input voltage was applied to the time when the digital output was obtained.

Sampling time is included in the conversion time in the characteristics table.

(9) Sampling time

This is the time the analog switch is turned on for the analog voltage to be sampled by the sample and hold circuit.

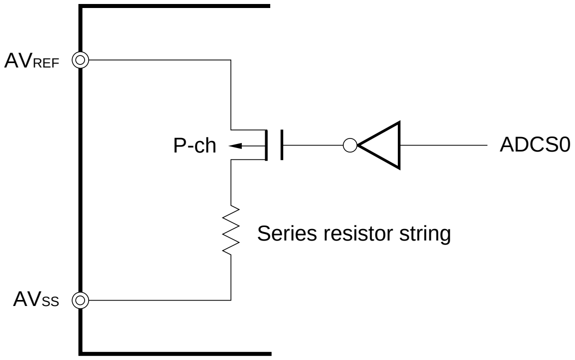


13.6 A/D Converter Cautions

(1) Current consumption in standby mode

A/D converter stops operating in the standby mode. At this time, current consumption can be reduced by stopping the conversion operation (by setting bit 7 (ADCS0) of the A/D converter mode register 0 (ADM0) to 0). Figure 13-15 shows how to reduce the current consumption in the standby mode.

Figure 13-15. Example of Method of Reducing Current Consumption in Standby Mode



(2) Input range of ANI0 to ANI7

The input voltages of ANI0 to ANI7 should be within the specification range. In particular, if a voltage higher than AV_{REF} or lower than AV_{SS} 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) Contending operations

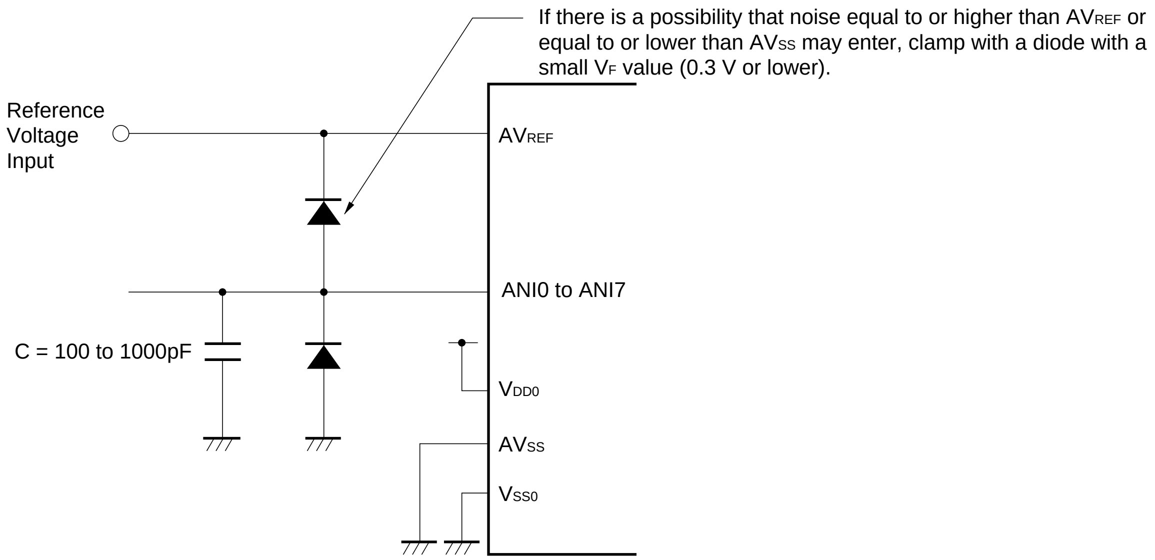
<1> Contention between A/D conversion result register 0 (ADCR0) write and ADCR0 read by instruction upon the end of conversion
ADCR0 read is given priority. After the read operation, the new conversion result is written to ADCR0.

<2> Contention between ADCR0 write and external trigger signal input upon the end of conversion
The external trigger signal is not accepted during A/D conversion. Therefore, the external trigger signal is not accepted during ADCR0 write.

<3> Contention between ADCR0 write and A/D converter mode register 0 (ADM0) write or analog input channel specification register 0 (ADS0) write
ADM0 or ADS0 write is given priority. ADCR0 write is not performed, nor is the conversion end interrupt request signal (INTAD0) generated.

<4> Noise countermeasures
To maintain the 10-bit resolution, attention must be paid to noise input to pin AV_{REF} and pins ANI0 to ANI7. 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 13-16 to reduce noise.

Figure 13-16. Analog Input Pin Connection


If there is a possibility that noise equal to or higher than AVREF or equal to or lower than AVSS may enter, clamp with a diode with a small VF value (0.3 V or lower).
Reference Voltage Input
AVREF
ANI0 to ANI7
C = 100 to 1000pF
VDD0
AVSS
VSS0

(5) ANI0 to ANI7

The analog input pins (ANI0 to ANI7) also function as input/output port pins (P10 to P17).

When A/D conversion is performed with any of pins ANI0 to ANI7 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 applied to another analog input pin are applied to a pin in the process of A/D conversion, the expected A/D conversion value may not be obtainable due to coupling noise. Therefore, avoid applying pulses to another analog input pin to a pin undergoing A/D conversion.

(6) AV_{REF} pin input impedance

A series resistor string is connected between the AV_{REF} pin and the AV_{SS} pin.

Therefore, when the output impedance of the reference voltage is too high, it seems as if the AV_{REF} pin and the series resistor string are connected in series. This may cause a greater reference voltage error.

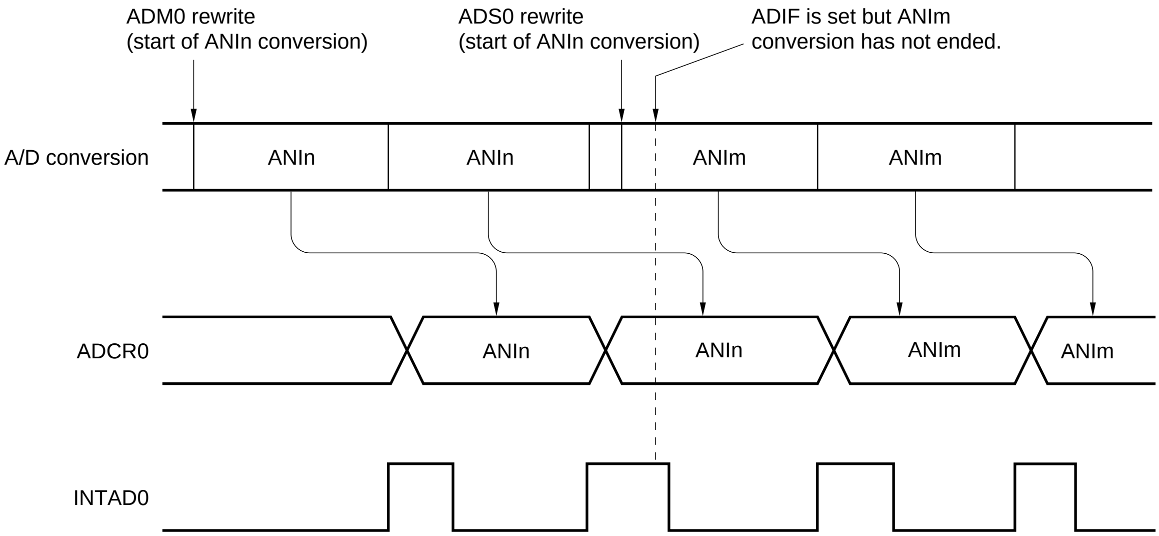
(7) Interrupt request flag (ADIF0)

The interrupt request flag (ADIF0) is not cleared even if the analog input channel specification register 0 (ADS0) 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 the ADS0 rewrite. Caution is therefore required since, at this time, when ADIF0 is read immediately just after the ADS0 rewrite, ADIF0 is set despite the fact that the A/D conversion for the post-change analog input has not ended.

When A/D conversion is restarted after it is stopped, clear ADIF0 before restart.

Figure 13-17. A/D Conversion End Interrupt Request Generation Timing



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

(8) Conversion results just after A/D conversion start

If bit 7 (ADCS0) of A/D converter mode register 0 (ADM0) is set to 1 without setting bit 0 (ADCE0) to 1, the first A/D conversion value immediately after A/D conversion has been started may not satisfy the rated value. Polling A/D conversion end interrupt request (INTAD0) and take measures such as removing the first conversion results. The same may apply if ADCS0 is set to 1 without the lapse of the wait time of 14 μs (MIN.) after ADCE0 has been set to 1. Make sure that the specified wait time elapses.

(9) A/D conversion result register 0 (ADCR0) read operation

When writing is performed to the A/D converter mode register 0 (ADM0) and analog input channel specification register 0 (ADS0), the contents of ADCR0 may become undefined. Read the conversion result following conversion completion before writing to ADM0, ADS0. Using a timing other than the above may cause an incorrect conversion result to be read.

(10) Timing at which A/D conversion result is undefined

The A/D conversion value may be undefined if the timing of completion of A/D conversion and the timing of stopping the A/D conversion conflict with each other. Therefore, read the the A/D conversion result during the A/D conversion operation. To read the conversion result after stopping the A/D conversion operation, be sure to stop the A/D conversion before the next conversion ends.

Figures 13-18 and 13-19 show the timing of reading the conversion result.

Figure 13-18. Timing of Reading Conversion Result (when conversion result is undefined)

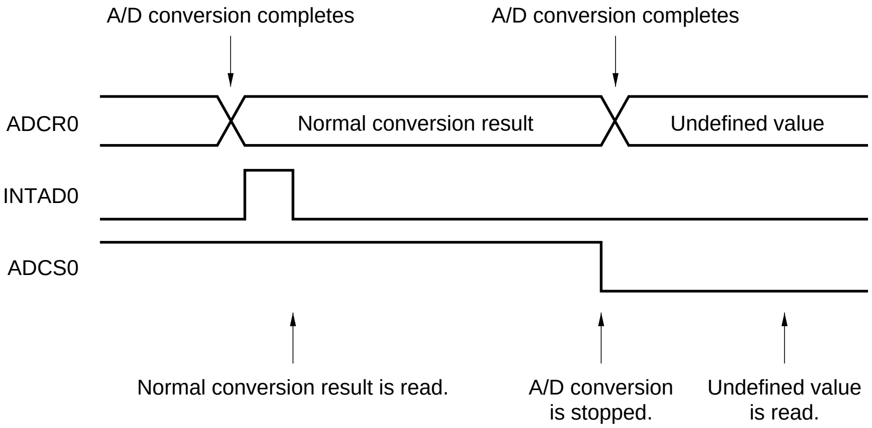
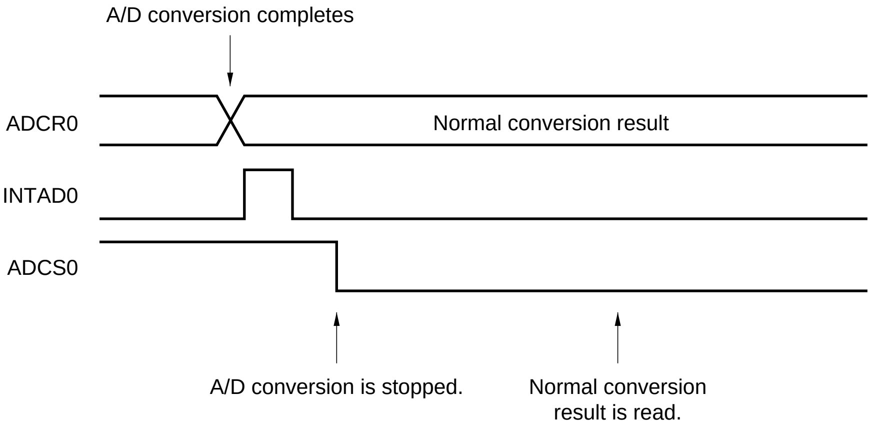


Figure 13-19. Timing of Reading Conversion Result (when conversion result is normal)



(11) Notes on board design

Locate analog circuits as far away from digital circuits as possible on the board because the analog circuits may be affected by the noise of the digital circuits. In particular, do not cross an analog signal line with a digital signal line, or wire an analog signal line in the vicinity of a digital signal line. Otherwise, the A/D conversion characteristics may be affected by the noise of the digital line.

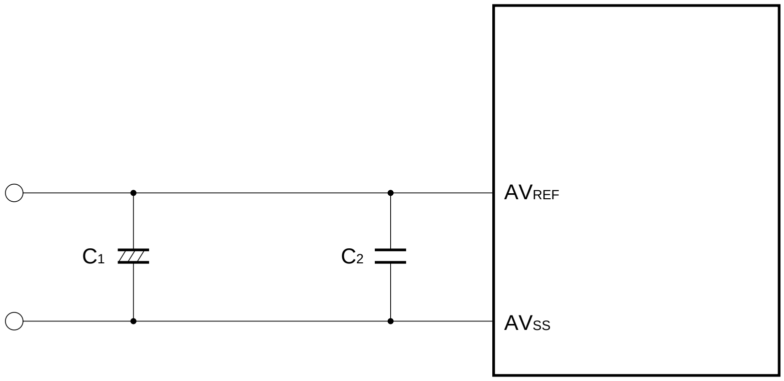
Connect AV_{SS0} and V_{SS0} at one location on the board where the voltages are stable.

(12) AV_{REF} pin

Connect a capacitor to the AV_{REF} pin to minimize conversion errors due to noise. If an A/D conversion operation has been stopped and then is started, the voltage applied to the AV_{REF} pin becomes unstable, causing the accuracy of the A/D conversion to drop. To prevent this, also connect a capacitor to the AV_{REF} pin.

Figure 13-20 shows an example of connecting a capacitor.

Figure 13-20. Example of Connecting Capacitor to AV_{REF} 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.

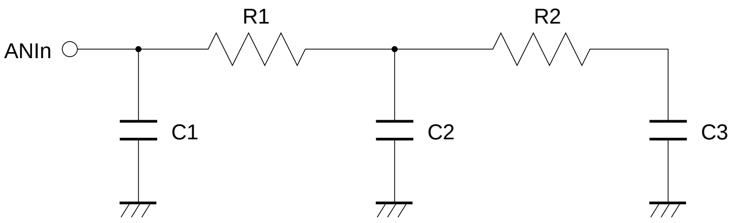
(13) Internal equivalent circuit of ANI0 to ANI7 pins and permissible signal source impedance

To complete sampling within the sampling time with sufficient A/D conversion accuracy, the impedance of the signal source such as a sensor must be sufficiently low. Figure 13-21 shows the internal equivalent circuit of the ANI0 to ANI7 pins.

If the impedance of the signal source is high, connect capacitors with a high capacitance to the pins ANI0 to ANI7. An example of this is shown in Figure 13-22. In this case, however, the microcontroller cannot follow an analog signal with a high differential coefficient because a lowpass filter is created.

To convert a high-speed analog signal or to convert an analog signal in the scan mode, insert a low-impedance buffer.

Figure 13-21. Internal Equivalent Circuit of Pins ANI0 through ANI7



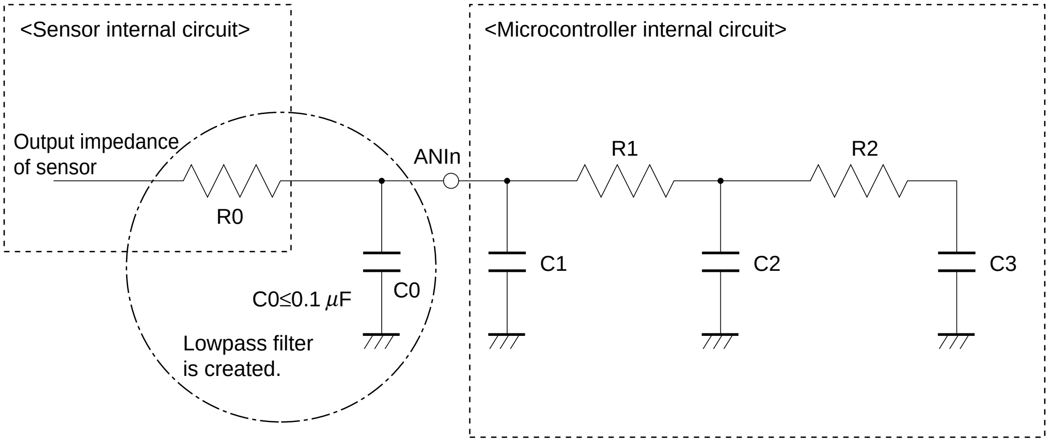
Remark n = 0 to 7

Table 13-2. Resistances and Capacitances of Equivalent Circuit (reference values)

AV_{REF}	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 resistances and capacitances in Table 13-2 are not guaranteed values.

Figure 13-22. Example of Connection If Signal Source Impedance Is High


<Sensor internal circuit>
<Microcontroller internal circuit>
Output impedance of sensor
R0
ANIn
R1
R2
C0
C0≤0.1 μF
Lowpass filter is created.
C1
C2
C3

Remark n = 0 to 7

CHAPTER 14 SERIAL INTERFACE (UART0)

14.1 Serial Interface (UART0) Functions

The serial interface (UART0) has the following three modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed to reduce power consumption.
For details, see **14.4.1 Operation stop mode**.

(2) Asynchronous serial interface (UART) mode

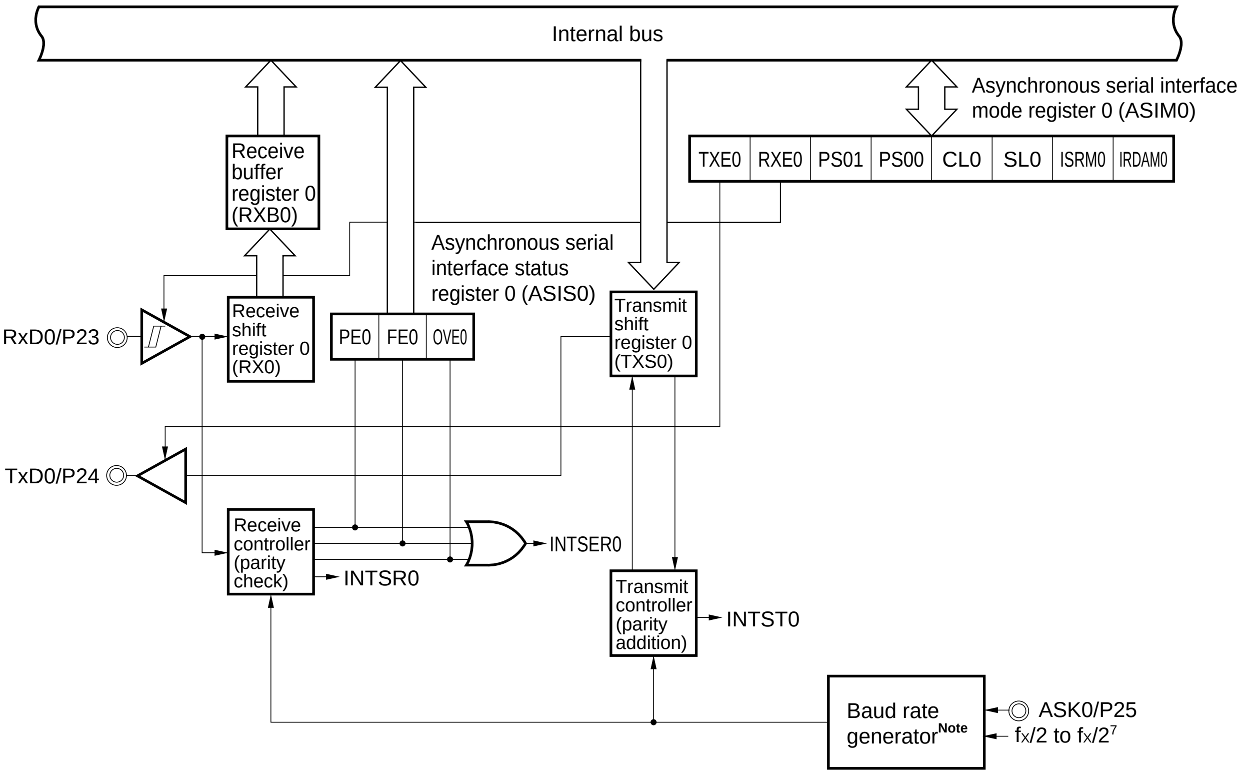
This mode enables full-duplex operation wherein one byte of data after the start bit is transmitted and received.
The on-chip baud rate generator dedicated to UART enables communications using a wide range of selectable baud rates. In addition, a baud rate can also be defined by dividing clocks input 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 **14.4.2 Asynchronous serial interface (UART) mode**.

(3) Infrared data transfer mode

For details, see **14.4.3 Infrared data transfer mode**.

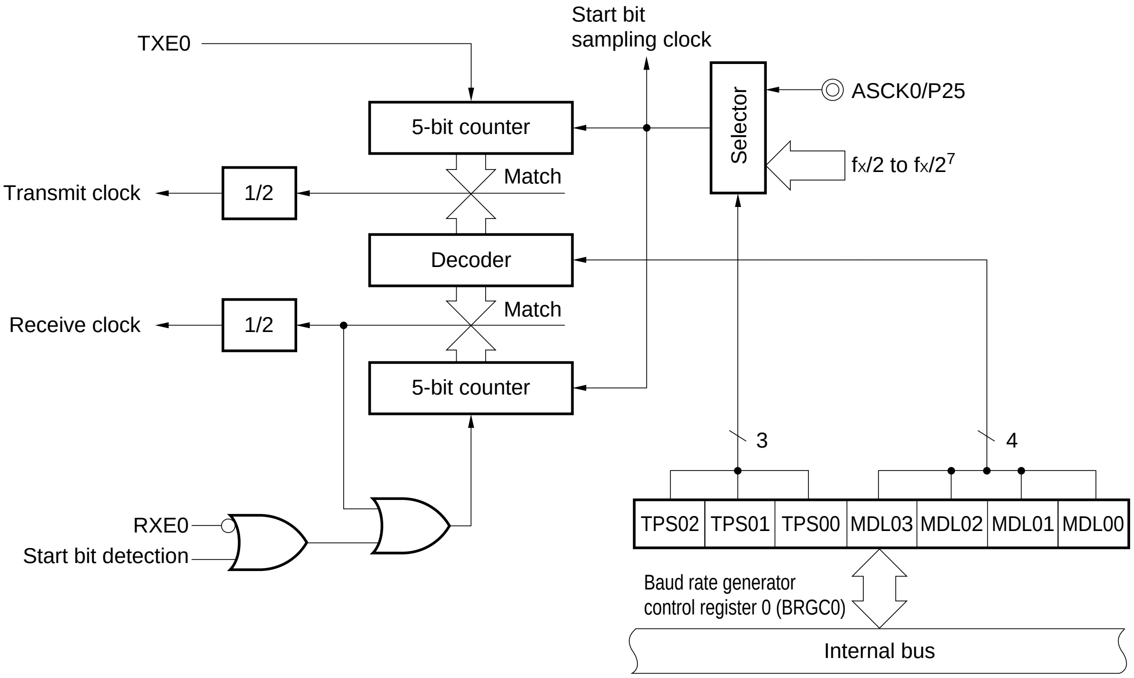
Figure 14-1 shows a block diagram of the serial interface (UART0) macro.

Figure 14-1. Serial Interface (UART0) Block Diagram


Internal bus
Asynchronous serial interface mode register 0 (ASIM0)
TXE0
RXE0
PS01
PS00
CL0
SL0
ISRM0
IRDAM0
Receive buffer register 0 (RXB0)
Asynchronous serial interface status register 0 (ASIS0)
RxD0/P23
Receive shift register 0 (RX0)
PE0
FE0
OVE0
Transmit shift register 0 (TXS0)
TxD0/P24
Receive controller (parity check)
INTSER0
INTSR0
Transmit controller (parity addition)
INTST0
Baud rate generatorNote
ASK0/P25
fx/2 to fx/2^7

Note For the configuration of the baud rate generator, refer to **Figure 14-2**.

Figure 14-2. Baud Rate Generator Block Diagram



Remark TXE0: Bit 7 of asynchronous serial interface mode register 0 (ASIM0)
RXE0: Bit 6 of asynchronous serial interface mode register 0 (ASIM0)

14.2 Serial Interface (UART0) Configuration

The serial interface (UART0) includes the following hardware.

Table 14-1. Serial Interface (UART0) Configuration

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 by an 8-bit memory manipulation instruction. It cannot be read.
$\overline{\text{RESET}}$ input sets the value of TXS0 to FFH.

Caution Do not write to TXS0 during a transmit operation.
The same address is assigned to TXS0 and the 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 the 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 the receive shift register (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 always becomes 0.
RXB0 can be read by an 8-bit memory manipulation instruction. It cannot be written to.
$\overline{\text{RESET}}$ input sets the value of RXB0 to FFH.

Caution The same address is assigned to RXB0 and the transmit shift register 0 (TXS0). During a write operation, values are written to TXS0.

(4) Transmission control circuit

The transmission control circuit controls transmit operations, such as adding a start bit, parity bit, and stop bit to data that is written to the transmit shift register 0 (TXS0), based on the values set to the asynchronous serial interface mode register 0 (ASIM0).

(5) Reception control circuit

The reception control circuit controls receive operations based on the values set to the 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 the asynchronous serial interface status register 0 (ASIS0) according to the type of error that is detected.

14.3 Registers to Control Serial Interface (UART0)

The serial interface (UART0) uses the following three types of registers for control functions.

- 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 interface (UART0)'s serial transfer operations.

ASIM0 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ASIM0 to 00H.

Figure 14-3 shows the format of ASIM0.

Caution In UART mode, set the port mode register (PMXX) as follows. Set the output latch of the port set to output mode (PMXX = 0) to 0.

- **During receive operation**
  Set P23 (RXD0) to input mode (PM23 = 1)
- **During transmit operation**
  Set P24 (TXD0) to output mode (PM24 = 0)
- **During transmit/receive operation**
  Set P23 (RXD0) to input mode, and P24 to output mode

Figure 14-3. Asynchronous Serial Interface Mode Register 0 (ASIM0) Format

Address: FFA0H At Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ASIM0	TXE0	RXE0	PS01	PS00	CL0	SL0	ISRM0	IRDAM0

TXE0	RXE0	Operation Mode	RxD0/P23 Pin Function	TxD0/P24 Pin Function
0	0	Operation stop	Port function (P23)	Port function (P24)
0	1	UART mode (receive only)	Serial function (RxD0)	
1	0	UART mode (transmit only)	Port function (P23)	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	Receive Completion Interrupt Control When Error Occurs
0	Receive completion interrupt request is issued when an error occurs
1	Receive completion interrupt request is not issued when an error occurs

IRDAM0	Operation Specified for Infrared Data Transfer Mode[Note 1]
0	UART (transmit/receive) mode
1	Infrared data transfer (transmit/receive) mode[Note 2]

Notes 1. The UART/infrared data transfer mode specification is controlled by TXE0 and RXE0.

2. When using infrared data transfer mode, be sure to set "10H" to the baud rate generator control register 0 (BRGC0).

Caution **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 during UART mode, this register indicates the type of error.

ASIS0 can be read by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ASIS0 to 00H.

Figure 14-4. Asynchronous Serial Interface Status Register 0 (ASIS0) Format

Address: FFA1H At 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[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 (RXB0))

Notes 1. Even if a stop bit length is set to two bits by setting bit 2 (SL0) in the 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 the 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 by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the value of BRGC0 to 00H.

Figure 14-5 shows the format of BRGC0.

Figure 14-5. Baud Rate Generator Control Register 0 (BRGC0) Format

Address: FFA2H At Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
BRGC0	0	TPS02	TPS01	TPS00	MDL03	MDL02	MDL01	MDL00

(f_X = 8.38 MHz)

TPS02	TPS01	TPS00	Source Clock Selection for 5-Bit Counter	n
0	0	0	P25/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 prohibit	—

Cautions 1. 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.

2. Set BRGC0 to 10H when using in infrared data transfer mode.

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 ($0 \leq k \leq 14$)

14.4 Serial Interface (UART0) Operations

This section explains the three modes of the serial interface (UART0).

14.4.1 Operation stop mode

Because serial transfer is not performed during this mode, the power consumption can be reduced.

In addition, pins can be used as ordinary ports.

(1) Register settings

Operation stop mode are set by the 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 the value of ASIM0 to 00H.

Address: FFA0H At Reset: 00H R/W

Symbol	$\boxed{7}$	$\boxed{6}$	5	4	3	2	1	0
ASIM0	TXE0	RXE0	PS01	PS00	CL0	SL0	ISRM0	IRDAM0

TXE0	RXE0	Operation Mode	RxD0/P23 Pin Function	TxD0/P24 Pin Function
0	0	Operation stop	Port function (P23)	Port function (P24)
0	1	UART mode (receive only)	Serial function (RxD0)	
1	0	UART mode (transmit only)	Port function (P23)	Serial function (TxD0)
1	1	UART mode (transmit and receive)	Serial function (RxD0)	

Caution Do not switch the operation mode until the current serial transmit/receive operation has stopped.

14.4.2 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 communications 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 the asynchronous serial interface mode register 0 (ASIM0), asynchronous serial interface status register 0 (ASIS0), and the 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 the value of ASIM0 to 00H.

Caution In UART mode, set the port mode register (PMXX) as follows. Set the output latch of the port set to output mode (PMXX = 0) to 0.

- **During receive operation**
  Set P23 (RXD0) to input mode (PM23 = 1)
- **During transmit operation**
  Set P24 (TXD0) to output mode (PM24 = 0)
- **During transmit/receive operation**
  Set P23 (RXD0) to input mode, and P24 to output mode

Address: FFA0H At Reset: 00H R/W

Symbol	$\boxed{7}$	$\boxed{6}$	5	4	3	2	1	0
ASIM0	TXE0	RXE0	PS01	PS00	CL0	SL0	ISRM0	IRDAM0

TXE0	RXE0	Operation Mode	RxD0/P23 Pin Function	TxD0/P24 Pin Function
0	0	Operation stop	Port function (P23)	Port function (P24)
0	1	UART mode (receive only)	Serial function (RxD0)	
1	0	UART mode (transmit only)	Port function (P23)	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	Receive Completion Interrupt Control When Error Occurs
0	Receive completion interrupt request is issued when an error occurs
1	Receive completion interrupt request is not issued when an error occurs

IRDAM0	Operation Specified for Infrared Data Transfer Mode[Note 1]
0	UART (transmit/receive) mode
1	Infrared data transfer (transmit/receive) mode[Note 2]

Notes 1. The UART/infrared data transfer mode specification is controlled by TXE0 and RXE0.

2. When using infrared data transfer mode, be sure to set the baud rate generator control register 0 (BRGC0) to 10H.

Caution **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 an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the value of ASIS0 to 00H.

Address: FFA1H At 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 (RXB0))

Notes 1. Even if a stop bit length is set to two bits by setting bit 2 (SL0) in the 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 the 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 by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the value of BRGC0 to 00H.

Address: FFA2H At Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
BRGC0	0	TPS02	TPS01	TPS00	MDL03	MDL02	MDL01	MDL00

(f_X = 8.38 MHz)

TPS02	TPS01	TPS00	Source Clock Selection for 5-Bit Counter	n
0	0	0	P25/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 prohibit	—

Cautions 1. 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.

2. Set 10H to BRGC0 when using infrared data transfer mode.

Remarks 1. f_{SCK} : Source clock for 5-bit counter

2. n : Value set via TPS00 to TPS02 ($0 \le n \le 7$)

3. k : Value set via MDL00 to MDL03 ($0 \le k \le 14$)

The transmit/receive clock that is used to generate the baud rate is obtained by dividing the main system clock.

- Transmit/receive clock generation for baud rate by using main system clock
  The main system clock is divided to generate the transmit/receive clock. The baud rate generated from the main system clock is determined according to the following formula.

$$[\text{Baud rate}] = \frac{f_X}{2^{n+1}(k + 16)} \quad [\text{Hz}]$$

f_X : Oscillation frequency of main system clock
When ASCK0 is selected as the source clock of the 5-bit counter, substitute the input clock frequency to ASCK0 pin for f_X in the above expression.

n : Value set via TPS00 to TPS02 ($0 \leq n \leq 7$)
For details, see **Table 14-2**.

k : Value set via MDL00 to MDL03 ($0 \leq k \leq 14$)

Table 14-2 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 14-2. Relationship between 5-Bit Counter’s Source Clock and “n” Value

TPS02	TPS01	TPS00	5-Bit Counter’s Source Clock Selected	n
0	0	0	P25/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

Remark f_X: Oscillation frequency of main system clock

- **Error tolerance range for baud rates**

  The tolerance range for baud rates depends on the number of bits per frame and the counter's division rate [1/(16 + k)].

  Table 14-3 describes the relationship between the main system clock and the baud rate and Figure 14-6 shows an example of a baud rate error tolerance range.

Table 14-3. Relationship between Main System Clock and Baud Rate

Baud Rate (bps)	f_X = 8.386 MHz		f_X = 8.000 MHz		f_X = 7.3728 MHz		f_X = 5.000 MHz		f_X = 4.1943 MHz	
	BRGC0	ERR (%)	BRGC0	ERR (%)	BRGC0	ERR (%)	BRGC0	ERR (%)	BRGC0	ERR (%)
600	–	–	–	–	–	–	–	–	7BH	1.14
1200	7BH	1.10	7AH	0.16	78H	0	70H	1.73	6BH	1.14
2400	6BH	1.10	6AH	0.16	68H	0	60H	1.73	5BH	1.14
4800	5BH	1.10	5AH	0.16	58H	0	50H	1.73	4BH	1.14
9600	4BH	1.10	4AH	0.16	48H	0	40H	1.73	3BH	1.14
19200	3BH	1.10	3AH	0.16	38H	0	30H	1.73	2BH	1.14
31250	31H	–1.3	30H	0	2DH	1.70	24H	0	21H	–1.3
38400	2BH	1.10	2AH	0.16	28H	0	20H	1.73	1BH	1.14
76800	1BH	1.10	1AH	0.16	18H	0	10H	1.73	–	–
115200	12H	1.10	11H	2.12	10H	0	–	–	–	–
Infrared data transfer mode[Note]	131031 bps		125000 bps		115200 bps		78125 bps		65536 bps	

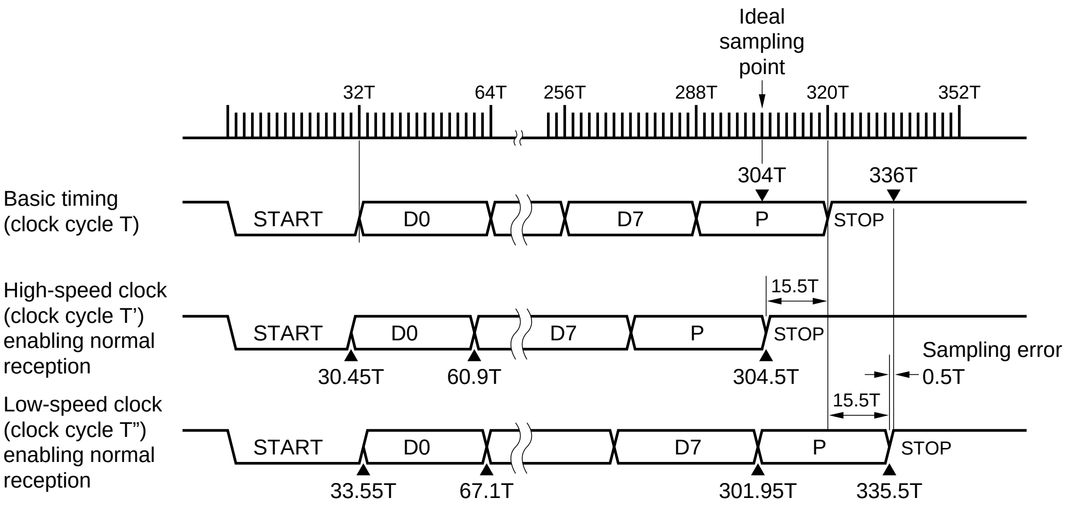
Note The UART/infrared data transfer mode specification is controlled by TXE0 and RXE0.

When using the infrared data transfer mode, be sure to set the baud rate generator control register 0 (BRGC0) as follows.

- k = 0 (MDL0 to MDL3 = 0000)
- n = 1 (TPS00 to TPS02 = 001)

Remark f_X : Oscillation frequency of main system clock
n : Value set via TPS00 to TPS02 ($0 \le n \le 7$)
k : Value set via MDL00 to MDL03 ($0 \le k \le 14$)

Figure 14-6. Error Tolerance (when k = 0), Including Sampling Errors



Remark T: 5-bit counter's source clock cycle

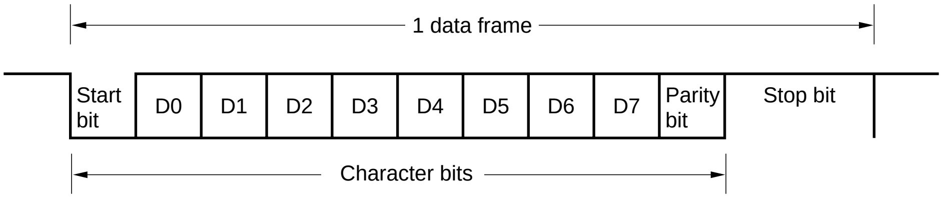
Baud rate error tolerance (when k = 0) = $\frac{\pm 15.5}{320} \times 100 = 4.8438$ (%)

(2) Communication operations

(a) Data format

Figure 14-7 shows the format of the transmit/receive data.

Figure 14-7. Format of Transmit/Receive Data in Asynchronous Serial Interface



1 data frame consists of the following bits.

- Start bit .... 1 bit
- Character bits ... 7 bits or 8 bits
- Parity bit .... Even parity, odd parity, zero parity, or no parity
- Stop bit(s) .... 1 bit or 2 bits

The 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 low-order 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”.
The ASIM0 and the 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 recognized by reading the 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

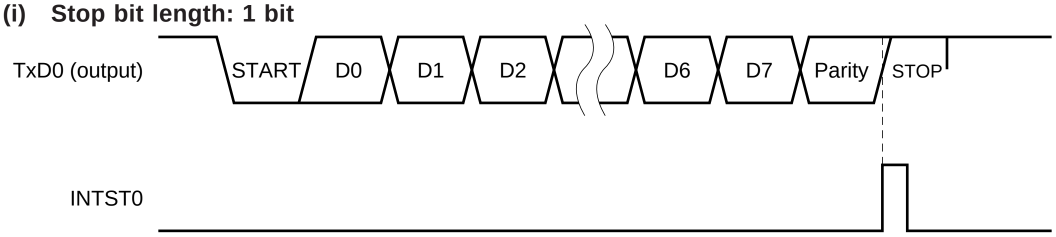
The transmit operation is enabled if bit 7 (TXE0) of asynchronous serial interface mode register 0 (ASIM0) is set to 1, the transmit operation is started when transmit data is written to the transmit shift register 0 (TXS0). A start bit, parity bit, and stop bit(s) are automatically added to the data.

Starting the transmit operation shifts out the data in TXS0, thereby emptying TXS0, after which a transmit completion interrupt request (INTST0) is issued.

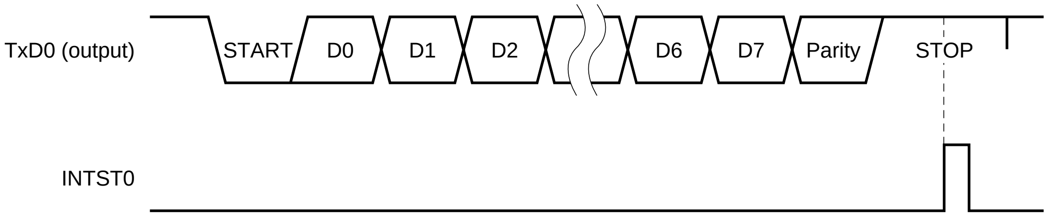
The timing of the transmit completion interrupt request is shown in Figure 14-8.

Figure 14-8. 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 the asynchronous serial interface mode register 0 (ASIM0) during a transmit operation. Rewriting ASIM0 register during a transmit operation may disable further transmit operations (in such cases, enter a $\overline{\text{RESET}}$ to restore normal operation). Whether or not a transmit operation is in progress can be determined via software using the transmit completion interrupt request (INTST0) or the interrupt request flag (STIF0) that is set by INTST0.

(d) Reception

The receive operation is enabled when "1" is set to bit 6 (RXE0) of the asynchronous serial interface mode register 0 (ASIM0), 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.

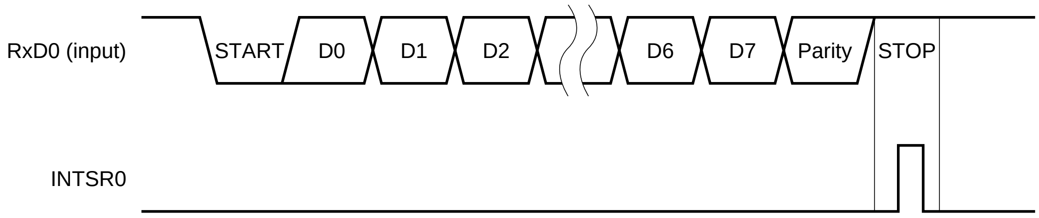
Once reception of one data frame is completed, the receive data in the shift register is transferred to the receive buffer register 0 (RXB0) and a receive completion interrupt request (INTSR0) occurs.

Even if an error has occurred, the receive data in which the error occurred is still transferred to RXB0. When ASIM0 bit 1 (ISRM0) is cleared (0) upon occurrence of an error, INTSR0 occurs (see **Figure 14-10**). When ISRM0 bit is set (1), INTSR0 does not occurr.

If the RXE0 bit is reset (to "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 14-9 shows the timing of the asynchronous serial interface receive completion interrupt request.

Figure 14-9. Timing of Asynchronous Serial Interface Receive Completion Interrupt Request



Caution **Be sure to read the contents of the 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 the asynchronous serial interface status register 0 (ASIS0), a receive error interrupt request (INTSER0) will occur. Receive error interrupts requests are generated before receiving completion interrupts request (INTSR0). Table 14-4 lists the causes behind receive errors.

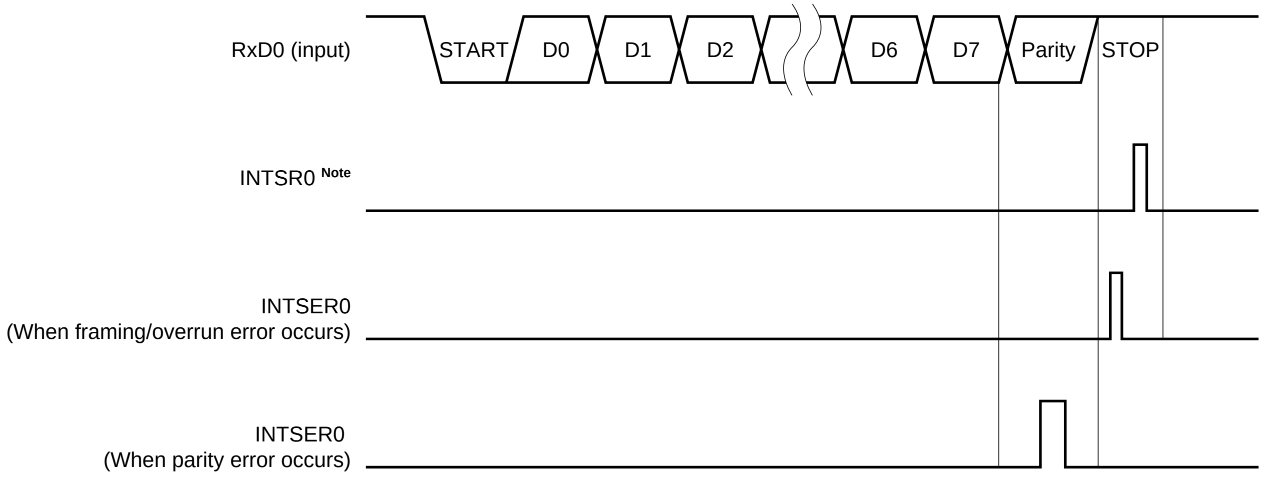
As part of receive error interrupt request (INTSER0) servicing, the contents of ASIS0 can be read to determine which type of error occurred during the receive operation (see **Table 14-4** and **Figure 14-10**).

The contents of ASIS0 are reset (to “0”) when the 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 14-4. 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 the next data was completed before data was read from the receive buffer register 0 (RXB0)	01H

Figure 14-10. Receive Error Timing



Note If a receive error occurs when ISRM0 bit has been set (1), INTSR0 does not occur.

Cautions 1. The contents of asynchronous serial interface status register 0 (ASIS0) are reset (to “0”) when the 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 the 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.

14.4.3 Infrared data transfer mode

In infrared data transfer mode, the following data format pulse output and pulse receiving are enabled. The relationship between the main system clock and baud rate is shown in Table 14-3.

(1) Data format

Figure 14-11 compares the data format used in UART mode with that used in infrared data transfer mode.

The IR (infrared) frame corresponds to the bit string of the UART frame, which consists of pulses – a start bit, eight data bits, and a stop bit.

The length of the electrical pulses that are used to transmit and receive in an IR frame is 3/16 the length of the cycle time for one bit (i.e., the “bit time”). This pulse (whose width is 3/16 the length of one bit time) rises from the middle of the bit time (see the figure below).

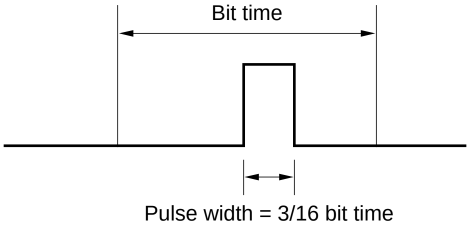
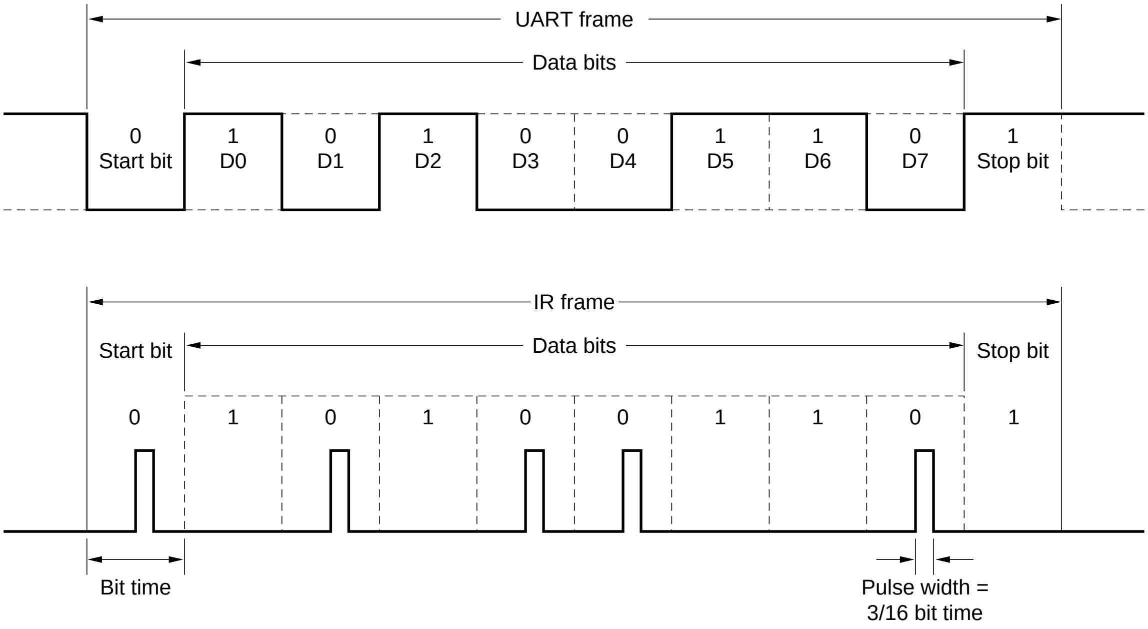


Figure 14-11. Data Format Comparison between Infrared Data Transfer Mode and UART Mode



(2) Bit rate and pulse width

Table 14-5 lists bit rates, bit rate error tolerances, and pulse width values.

Table 14-5. Bit Rate and Pulse Width Values

Bit Rate (kbits/s)	Bit Rate Error Tolerance (% of bit rate)	Pulse Width Minimum Value (µs) **Note 2**	3/16 Pulse Width <nominal value> (µs)	Maximum Pulse Width (µs)
115.2 **Note 1**	+/– 0.87	1.41	1.63	2.71

Notes 1. At the operation time with f_X = 7.3728 MHz

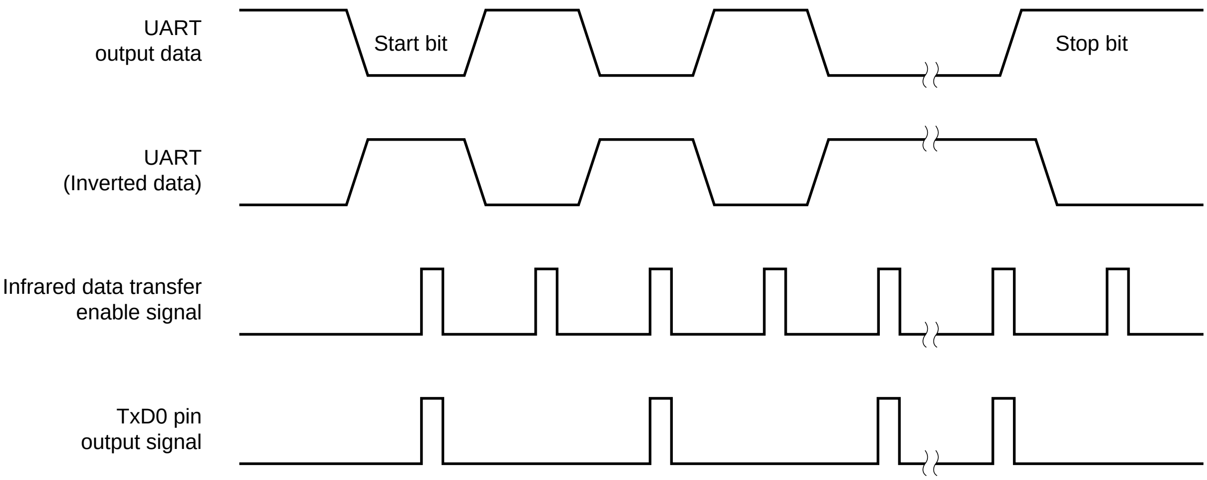
2. When a digital noise elimination circuit is used in a microcontroller operating at 1.41 MHz or above.

Caution **When using the baud rate generator control register 0 (BRGC0) in infrared data transfer mode, set 10H to it.**

Remark f_X: Main system clock oscillation frequency

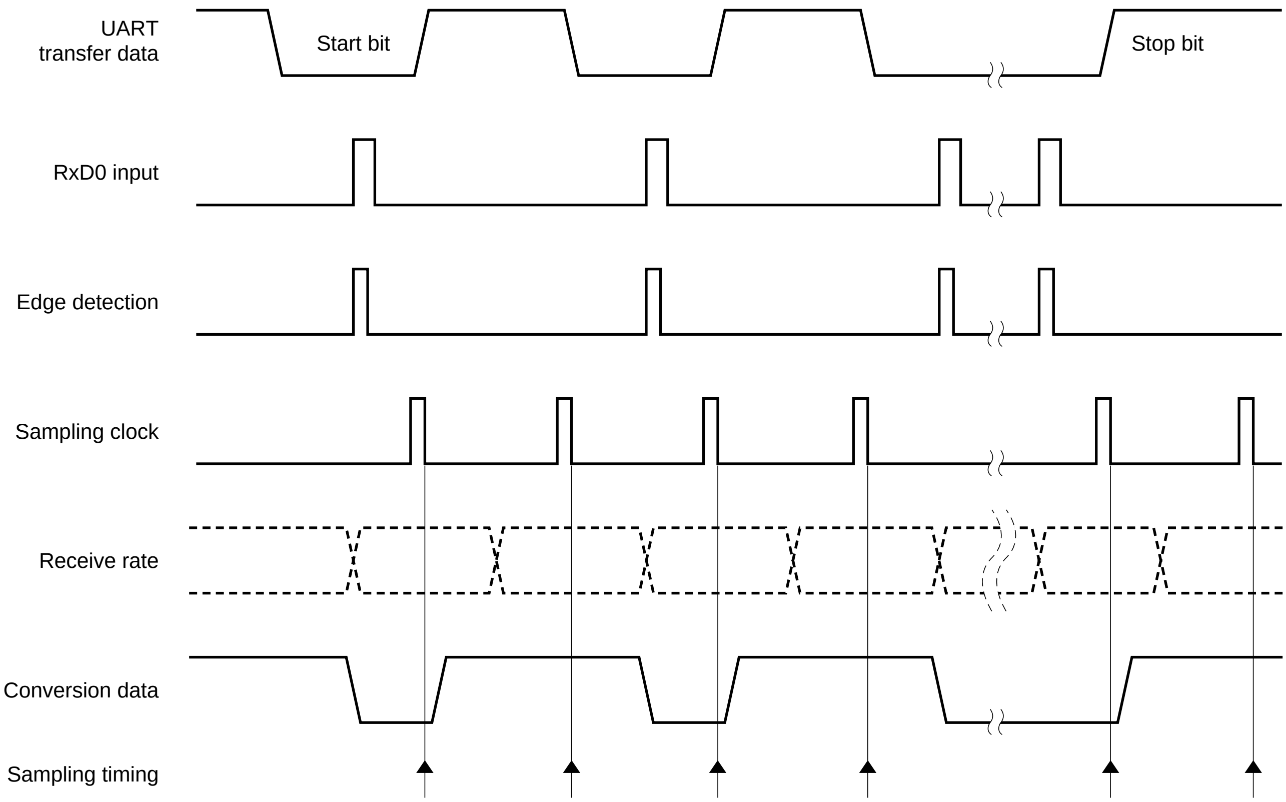
(3) Input data and internal signals

- **Transmit operation timing**



- **Receive operation timing**

  Data reception is delayed for one-half of the specified baud rate.



[MEMO]

CHAPTER 15 SERIAL INTERFACE (UART2)

The serial interface (UART2/SIO3) can be used in the asynchronous serial interface (UART) mode or 3-wire serial I/O mode.

Caution Do not enable UART2 and SIO3 at the same time.

15.1 Serial Interface (UART2) Functions

The serial interface (UART2) has the following four modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed to reduce power consumption.
For details, see **15.4.1 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 communications 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 **15.4.2 Asynchronous serial interface (UART) mode**.

(3) Multi-processor transfer mode

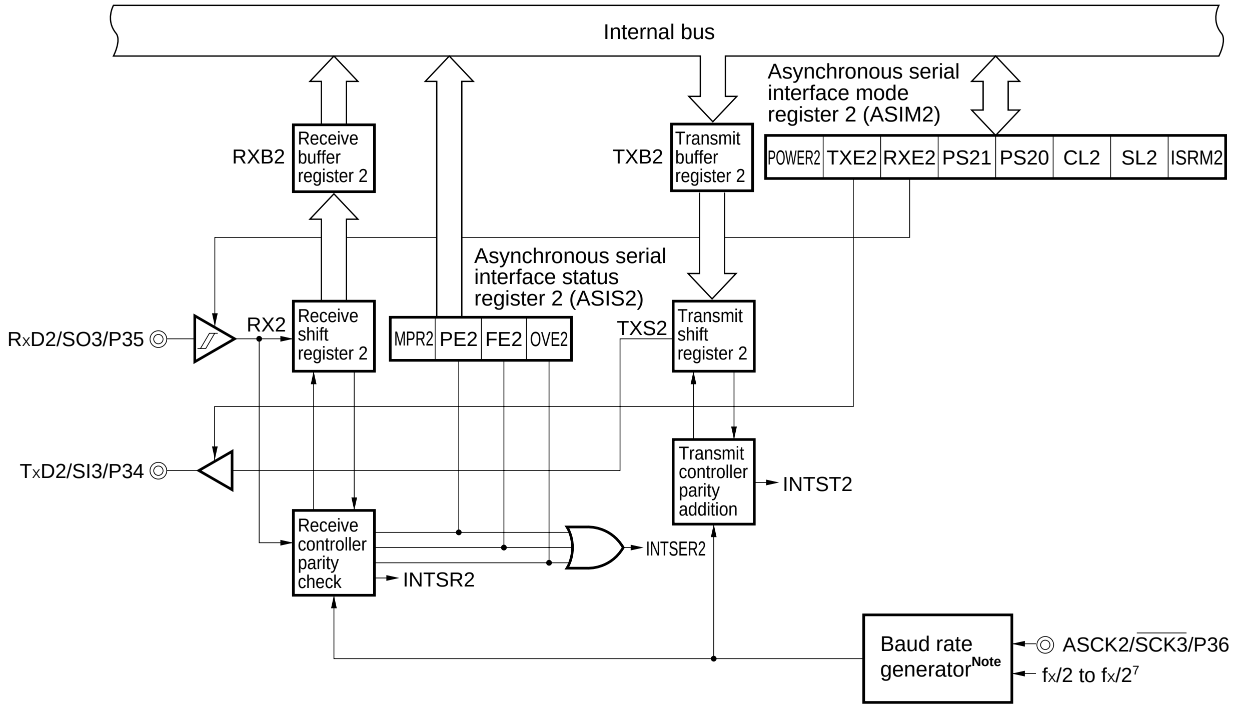
In this mode data can be transferred to or received from two or more processors. For details, refer to **15.4.3 Multi-processor transfer mode**.

(4) Infrared data transfer (IrDA) mode

In this mode, pulses can be output or received in the data format of IrDA specification. This mode can be used to transfer data with another digital device such as a personal computer.
For details, see **15.4.4 Infrared data transfer (IrDA) mode**.

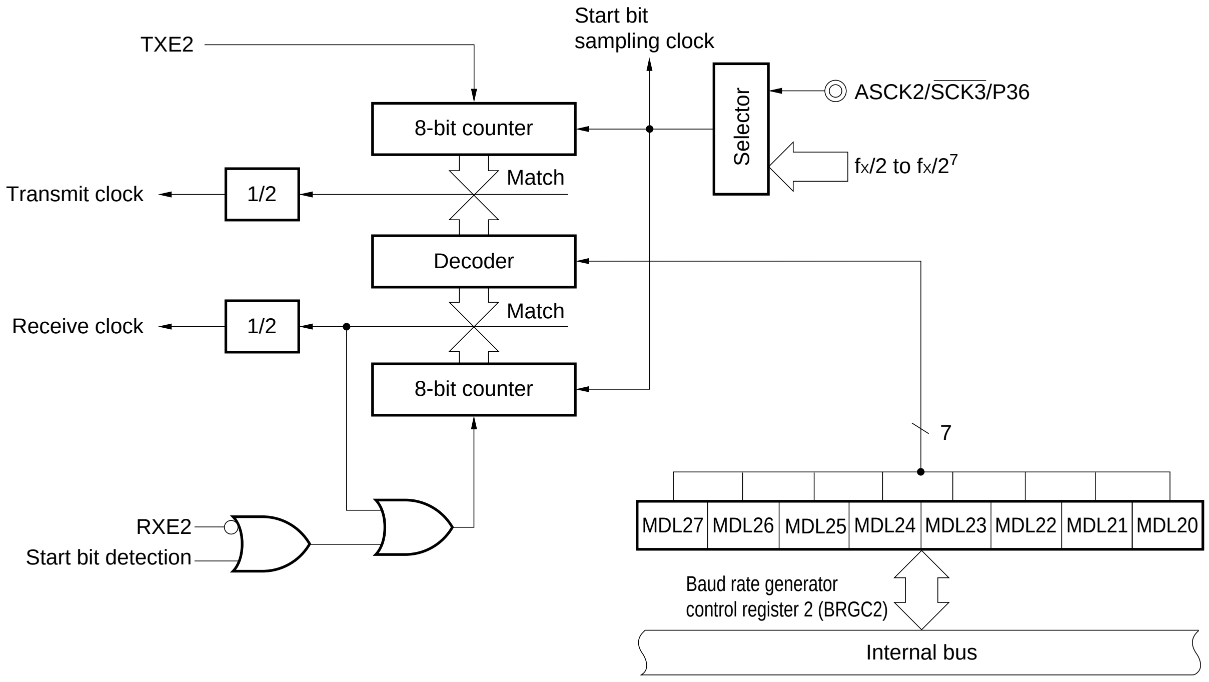
Figure 15-1 shows a block diagram of the serial interface (UART0) macro.

Figure 15-1. Serial Interface (UART2) Block Diagram


Internal bus
Asynchronous serial interface mode register 2 (ASIM2)
RXB2
Receive buffer register 2
TXB2
Transmit buffer register 2
POWER2
TXE2
RXE2
PS21
PS20
CL2
SL2
ISRM2
Asynchronous serial interface status register 2 (ASIS2)
RxD2/SO3/P35
RX2
Receive shift register 2
MPR2
PE2
FE2
OVE2
TXS2
Transmit shift register 2
TxD2/SI3/P34
Transmit controller parity addition
INTST2
Receive controller parity check
INTSER2
INTSR2
Baud rate generatorNote
ASCK2/SCK3/P36
fX/2 to fX/2^7

Note For the configuration of the baud rate generator, refer to **Figure 15-2**.

Figure 15-2. Baud Rate Generator Block Diagram



Remark TXE2: Bit 7 of asynchronous serial interface mode register 2 (ASIM2)
RXE2: Bit 6 of asynchronous serial interface mode register 2 (ASIM2)

15.2 Serial Interface (UART2) Configuration

The serial interface (UART2) includes the following hardware.

Table 15-1. Serial Interface (UART2) Configuration

Item	Configuration
Registers	Transmit shift register 2 (TXS2) Receive shift register 2 (RX2) Transmit buffer register 2 (TXB2) Receive buffer register 2 (RXB2)
Control registers	Asynchronous serial interface mode register 2 (ASIM2) Asynchronous serial interface status register 2 (ASIS2) Baud rate generator control register 2 (BRGC2) Asynchronous serial interface transmit status register 2 (ASIF2) Clock select register 2 (CKSEL2) Transfer mode specification register 2 (TRMC2)

(1) Transmit shift register 2 (TXS2)

This register transmits the data transferred from transmit buffer register 2 (TXB2), from the TXD2 pin as serial data.

The value of this register is set to FFH if bits 7 and 6 (TXE2) of asynchronous serial interface mode register 2 (ASIM2) are cleared to 0. TX2 cannot be directly manipulated in software.

(2) Receive shift register 2 (RX2)

This register converts serial data input via the RxD2 pin to parallel data. When one byte of data is received at this register, the receive data is transferred to the receive buffer register 2 (RXB2).

RX2 cannot be manipulated directly by a program.

(3) Transmit buffer register 2 (TXB2)

This register sets data to be transmitted. The data written to TXB2 is transferred to the transmit shift register 2 (RX2) and transmitted from the TXD pin as serial data.

No data can be written to TXB2 if bit 1 (TXBF) of transmit status register 2 (ASIF2) is 1.

TXB2 is manipulated by using an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the value of TXB2 to FFH.

(4) Receive buffer register 2 (RXB2)

This register holds receive data. When one byte of data is received, one byte of new receive data is transferred from the receive shift register (RX2).

When the data length is set as 7 bits, receive data is sent to bits 0 to 6 of RXB2. In this case, the MSB of RXB2 always becomes 0.

If an overrun (OVE2) occurs, however, the receive data is not transferred to RXB2 but is discarded.

RXB0 can be read by an 8-bit memory manipulation instruction. It cannot be written to.

The value of this register is also initialized to FFH at reset or by clearing bit 7 (POWER2) of asynchronous serial interface mode register 2 (ASIM2) to 0.

(5) Transmission control circuit

The transmission control circuit controls transmit operations, such as adding a start bit, parity bit, and stop bit to data that is written to the transmit shift register 2 (TXS2), based on the values set to the asynchronous serial interface mode register 2 (ASIM2).

(6) Reception control circuit

The reception control circuit controls receive operations based on the values set to the asynchronous serial interface mode register 2 (ASIM2). During a receive operation, it performs error checking, such as for parity errors, and sets various values to the asynchronous serial interface status register 2 (ASIS2) according to the type of error that is detected.

15.3 Registers to Control Serial Interface (UART2)

The serial interface (UART2) uses the following six types of registers for control functions.

- Asynchronous serial interface mode register 2 (ASIM2)
- Asynchronous serial interface status register 2 (ASIS2)
- Baud rate generator control register 2 (BRGC2)
- Asynchronous serial interface transmit status register 2 (ASIF2)
- Clock select register 2 (CKSEL2)
- Transfer mode specification register 2 (TRMC2)

(1) Asynchronous serial interface mode register 2 (ASIM2)

This is an 8-bit register that controls serial interface (UART2)'s serial transfer operations.
ASIM0 is set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets ASIM2 to 00H.
Figure 15-3 shows the format of ASIM2.

Caution In UART, IrDA, and multi-processor modes, set the port mode register (PMXX) as follows. Set the output latch of the port set to output mode (PMXX = 0) to 0.

- **During receive operation**
  Set P35 (RXD2) to input mode (PM35 = 1)
- **During transmit operation**
  Set P34 (TXD2) to output mode (PM34 = 0)
- **During transmit/receive operation**
  Set P35 (RXD2) to input mode, and P34 to output mode

Figure 15-3. Asynchronous Serial Interface Mode Register 2 (ASIM2) Format (1/2)

Address: FF90H At Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ASIM2	POWER2	TXE2	RXE2	PS21	PS20	CL2	SL2	ISRM2

POWER2	Enables/Stops Clock Operation
0	Stops clock operation. Power consumption decreases and latch in UART2 is asynchronously reset (TxD2 pin is low).
1	Enables clock operation (TxD2 pin is high).

TXE2[Note 1]	Stops/Enables Transmission
0	Stops transmission (transmission circuit is synchronously reset).
1	Enables transmission.

RXE2[Note 2]	Stops/Enables Reception
0	Stops reception (reception circuit is synchronously reset).
1	Enables reception.

Notes 1. To transmit data with UART2, first specify the clock operation (set POWER2 to 1 and then TXE2 to 1), wait for the duration of 2 clocks[Note 3], and then write the transmit data to transmit buffer register 2 (TXB2). To stop transmission by UART2, specify stopping transmission (TXE2 = 0), wait for the duration of 2 clocks[Note 3], and then stop the clock operation (POWER2 = 0).

2. To receive data with UART2, first specify the clock operation (set POWER2 to 1 and then RXE2 to 1), wait for the duration of 2 clocks[Note 3], and then start reception. To stop reception by UART2, specify stopping reception (RXT2 = 0), wait for the duration of 2 clocks[Note 3], and then stop the clock operation (POWER2 = 0).

3. The clock is the output clock of the 8-bit counter or the input clock of the baud rate generator.

Figure 15-3. Asynchronous Serial Interface Mode Register 2 (ASIM2) Format (2/2)

PS21[Note 1]	PS20[Note 1]	Parity Bit Specification	
		Transmission	Reception
0	0	Does not output parity bit.	Reception without parity
0	1	Outputs 0 parity.	Reception as 0 parity[Note 2]
1	0	Outputs odd parity.	Identifies odd parity.
1	1	Outputs even parity.	Identifies even parity.

CL2[Note 3]	Data Character Length Specification
0	7 bits
1	8 bits

SL2[Note 4]	Specification for Number of Stop Bits for Transmission
0	1 bit
1	2 bits

ISRM2[Note 5]	Reception Error Interrupt Signal Control
0	INTSR2 is generated.
1	INTSER2 is generated.

Notes **1.** To specify a parity bit, stop transmission and reception (TXE2 = 0 and RXE2 = 0) before rewriting PS1 and PS0.

2. The parity is not identified with this setting. Therefore, bit 2 (PE2) of asynchronous serial interface status register 2 (ASIS2) is not set and the error interrupt does not occur.

3. To specify a data character length, stop transmission and reception (TXE2 = 0 and RXE2 = 0) before rewriting CL2.

4. To specify the number of stop bits, stop transmission (TXE2 = 0) before rewriting SL2. Reception is always performed on the assumption that the number of stop bits is 1.

5. To specify an interrupt that occurs in case of an error, stop transmission (TXE2 = 0) before rewriting ISEM2.

(2) Asynchronous serial interface status register 2 (ASIS2)

When a receive error occurs during UART mode, this register indicates the type of error.

ASIS2 can be read by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the value of ASIS2 to 00H.

Figure 15-4. Asynchronous Serial Interface Status Register 2 (ASIS2) Format

Address: FFA1H At Reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS2	0	0	0	0	MPR2[Note 1]	PE2[Note 1]	FE2[Note 1]	OVE2[Note 1]

MPR2	ID Reception Status Flag (during reception in multi-processor transfer mode)[Note 2]
0	Multi-processor appended bit “1” is not received.
1	Multi-processor appended bit “1” is received.

PE2	Parity Error Flag
0	No parity error
1	Parity error (Incorrect parity bit detected[Note 3])

FE2	Framing Error Flag
0	No framing error
1	Framing error[Note 4] (Stop bit not detected)

OVE2	Overrun Error Flag
0	No overrun error
1	Overrun error[Note 5] (Next receive operation was completed before data was read from receive buffer register 2 (RXB2))

Notes 1. These bits are reset to 0 if bit 7 (POWER2) of asynchronous serial interface mode register 2 (ASIM2) is reset to 0.

2. This flag is affected only if the multi-processor transfer mode is selected by using bits 6 and 7 (TRM02 and TRM12) of transfer mode specification register 2 (TRMC2).

3. The operation of the parity error flag is affected by the set values of bits 3 and 4 (PS20 and PS21) of ASIM2.

4. Even if a stop bit length is set to two bits by setting bit 2 (SL2) in ASIM2, stop bit detection during a receive operation only applies to a stop bit length of 1 bit.

5. Be sure to read the contents of the receive buffer register 2 (RXB2) when an overrun error has occurred.

Until the contents of RXB2 are read, further overrun errors will occur when receiving data. The next receive data is not written to the receive buffer register 2 (RXB2) and is discarded.

(3) Baud rate generator control register 2 (BRGC2)

This register sets the serial clock for serial interface.

BRGC2 is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the value of BRGC2 to 00H.

Figure 15-5 shows the format of BRGC2.

Figure 15-5. Baud Rate Generator Control Register 2 (BRGC2) Format

Address: FF93H At Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
BRGC2	MDL27	MDL26	MDL25	MDL24	MDL23	MDL22	MDL21	MDL20

MDL27	MDL26	MDL25	MDL24	MDL23	MDL22	MDL21	MDL20	Input Clock Selection for Baud Rate Generator	k
0	0	0	0	0	×	×	×	Setting prohibited	—
0	0	0	0	1	0	0	0	f_{SCK}/8	8
0	0	0	0	1	0	0	1	f_{SCK}/9	9
0	0	0	0	1	0	1	0	f_{SCK}/10	10
0	0	0	0	1	0	1	1	f_{SCK}/11	11
0	0	0	0	1	1	0	0	f_{SCK}/12	12
0	0	0	0	1	1	0	1	f_{SCK}/13	13
0	0	0	0	1	1	1	0	f_{SCK}/14	14
0	0	0	0	1	1	1	1	f_{SCK}/15	15
0	0	0	1	0	0	0	0	f_{SCK}/16	16
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	1	1	1	1	1	1	1	f_{SCK}/255	255

Caution Writing to BRGC2 during a communication operation may cause abnormal output from the baud rate generator and disable further communication operations. Therefore, do not write to BRGC2 during a communication operation.
Before rewriting BRGC2, clear bits 5 and 6 (RXE2 and TXE2) of asynchronous serial interface mode register 2 (ASIM2) to 0.

Remarks 1. f_{SCK} : Source clock for 8-bit counter
Set by bits 4 through 6 (TPS20 through TPS22) of clock select register 2 (CKSEL2)

2. k : Value set via MDL27 to MDL20 ($8 \le k \le 255$)

(4) Asynchronous serial interface transmit status register 2 (ASIF2)

This register indicates the status of transmission.

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

$\overline{\text{RESET}}$ input sets the value of ASIF2 to 00H.

Figure 15-6. Asynchronous Serial Interface Transmit Status Register 2 (ASIF2) Format

Address: FF95H At Reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIF2	0	0	0	0	0	0	TXBF	TXSF

TXBF	Transmit Buffer Data Flag
0	• If bit 7 (POWER2) or bit 6 (TXE2) of asynchronous serial interface mode register 2 (ASIM2) is cleared to 0 • If data is transferred to transmit shift register 2 (TXS2)
1	If data is written to transmit buffer register 2 (TXB2) (if data exists in TXB2)

TXSF	Transmit Shift Register Data Flag
0	• If bit 7 (POWER2) or bit 6 (TXE2) of asynchronous serial interface mode register 2 (ASIM2) is cleared to 0 • If no more data is transferred from transmit buffer register 2 (TXB2) after completion of transfer.
1	If data is transferred from transmit buffer register 2 (TXB2) (during transmission)

Caution To transmit data successively, be sure to check the value of TXBF and write the data to transmit buffer register 2 (TXB2). Data may or may not be written to TXB2 depending on the transmission status, as indicated in the table below.

TXBF	TXSF	Transmission Status	Writing to TXB2
0	0	Initial status or completion of transmission	Possible
1	0	Transmission wait (data exists in TXB2)	Impossible
0	1	During transmission (no data in TXB2)	Possible
1	1	During transmission (data exists in TXB2)	Impossible

(5) Clock select register 2 (CKSEL2)

This 8-bit register is used to select the input clock for the baud rate of UART2 and the transmit pulse width of IrDA.

This register is set by using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the value of this register to 00H.

Figure 15-7. Clock Select Register 2 (CKSEL2) Format

Address: FF92H At Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CKSEL2	0	TPS22^{Note}	TPS21^{Note}	TPS20^{Note}	TPW23	TPW22	TPW21	TPW20

TPS22	TPS21	TPS20	Source Clock of 8-bit Counter	n
0	0	0	ASCK2/$\overline{\text{SCK3}}$/P36	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

TPW23	TPW22	TPW21	TPW20	Selects IrDA Transmit Pulse Width of 1-bit Data
0	0	1	0	Width of two f_{SCK} clocks
0	0	1	1	Width of three f_{SCK} clocks
0	1	0	0	Width of four f_{SCK} clocks
0	1	0	1	Width of five f_{SCK} clocks
0	1	1	0	Width of six f_{SCK} clocks
0	1	1	1	Width of seven f_{SCK} clocks
1	0	0	0	Width of eight f_{SCK} clocks
1	0	0	1	Width of nine f_{SCK} clocks
1	0	1	0	Width of ten f_{SCK} clocks
1	0	1	1	Width of 11 f_{SCK} clocks
1	1	0	0	Width of 12 f_{SCK} clocks
1	1	0	1	Width of 13 f_{SCK} clocks
1	1	1	0	Width of 14 f_{SCK} clocks
1	1	1	1	Width of 15 f_{SCK} clocks
0	0	0	0	Width of 16 f_{SCK} clocks
Others				Setting prohibited

Note To rewrite TPS0 through TPS2, clear bit 7 (POWER2) of the asynchronous serial interface mode register 2 (ASIM2) to 0.

Cautions 1. If data is written to CKSEL2 during a communication operation, the output of the baud rate generator is disturbed and the communication cannot be performed correctly. Therefore, do not rewrite CKSEL2 during communication.

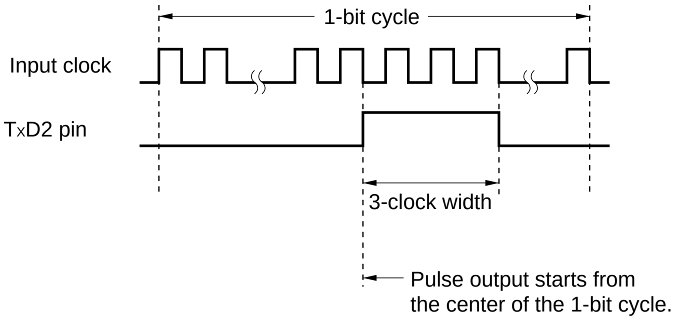
2. To transfer data in the infrared data transfer (IrDA) mode, the following conditions must be satisfied when a transmit pulse width is specified:

(Condition)

1.41 μs ≤ Transmit pulse width < Transfer rate

(Set values of bits 0 through 3 (TPW0 through TPW3) of CKSEL2) (Set values of bits 0 through 7 (MDL20 through MDL27) of BRGC2)

Example If the transmit pulse width is set to the width of three f_{SCK} clocks (TPW3 through TPW0 = 0, 0, 1, 1)



Remarks f_X : Main system clock oscillation frequency

f_{SCK} : Source clock of 8-bit counter

(6) Transfer mode specification register 2 (TRMC2)

This 8-bit register is used to specify a transfer mode, enable or disable occurrence of the reception completion interrupt in the multi-processor transfer mode, and specify a multi-processor transfer appended bit.

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

$\overline{\text{RESET}}$ input sets the value of TRMC2 to 02H.

Figure 15-8. Transfer Mode Specification Register 2 (TRMC2) Format

Address: FF91H At Reset: 02H R/W

Symbol	7	6	5	4	3	2	$\boxed{1}$	$\boxed{0}$
TRMC2	TRM12[Note 1]	TRM02[Note 1]	0	0	0	0	MPIEN2	MPS2[Note 2]

TRM12	TRM02	Transfer Mode
0	0	UART transfer mode[Note 3]
0	1	Multi-processor transfer mode[Note 3]
1	0	Infrared data transfer (IrDA) mode[Note 3]
1	1	

MPIEN2	Enables/disables reception completion interrupt in multi-processor transfer mode[Note 4]	
	Condition	Enables/disables INTSR2[Note 5]
0[Note 6]	If “0” is written to this bit	Disabled
1	• If it 7 (POWER2) or bit 6 (TXE2) of asynchronous serial interface mode register 2 (ASIM2) is cleared to 0 • If multi-processor appended data has received data of “1”	Enabled

MPS2	Setting of multi-processor transmission appended bit[Note 4]
0	Appends “0” to and transmits multi-processor appended bit (during data transmission).
1	Appends “1” to and transmits multi-processor appended bit (during ID transmission).

Notes

1. Before rewriting TRM12 and TRM02, clear bits 6 (TXE2) and 5 (RXE2) of asynchronous serial interface mode register 2 (ASIM2) to 0.

2. Before setting a value to MPS2, confirm that bit 1 (TXBF) of asynchronous serial interface transmit status register 2 (ASIF2) is cleared to 0. Before writing transmit data to the transmit buffer register 2 (TXB2), specify whether “0” or “1” is appended to the multi-processor appended bit.

3. The setting of bits 0 through 4 (ISEM2, SL2, CL2, PS20, and PS21) of ASIM2 is valid in all the transfer modes.

4. The specification by MPIEN2 and MPS2 is valid only when bit 7 (TRM12) is cleared to 0 and bit 6 (TRM02) is set to 1 (i.e., when the multi-processor transfer mode is set).

5. Enabling or disabling the occurrence of the reception completion interrupt (INTSR2) in case of an error is affected by the setting of bit 0 (ISRM2) of ASIM2.

6. Even if MPIEN is cleared to 0, reception is started if the start bit is detected, in order to detect address (ID) reception. At this time, the error in the receive data is not detected if the multi-processor appended bit is “0”. If data “1” is received by mistake, due to bit slip or other cause, when the multi-processor appended bit is detected, however, ID reception is detected. Consequently, the error in the receive data is identified, and the error interrupt signal may be generated and the error flag may be set.

Remark To receive data in the multi-processor transfer mode, the reception completion interrupt (INTSR2) occurs, regardless of the value of MPIEN2, if data with the multi-processor appended bit set to “1” is received. Usually, this receive data is an address (ID) that indicates the other party of communication. The subsequent receive data can be ignored and the occurrence of an unnecessary reception completion interrupt (INTSR2) can be disabled by comparing this received ID with the ID of the microcontroller (for which software processing is necessary) and clearing MPIEN2 if the two IDs do not match.

15.4 Serial Interface (UART2) Operations

This section explains the four modes of the serial interface (UART2).

15.4.1 Operation stop mode

Because serial transfer is not performed during this mode, the power consumption can be reduced.

In addition, pins can be used as ordinary ports.

(1) Register settings

Operation stop mode are set by the asynchronous serial interface mode register 2 (ASIM2).

ASIM2 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the value of ASIM2 to 00H.

Address: FF90H At Reset: 00H R/W

Symbol	$\boxed{7}$	$\boxed{6}$	$\boxed{5}$	4	3	2	1	0
ASIM2	POWER2	TXE2	RXE2	PS21	PS20	CL2	SL2	ISRM2

POWER2	Enables/Stops Clock Operation
0	Stops clock operation. Power consumption decreases and latch in UART2 is asynchronously reset (TxD2 pin is low).
1	Enables clock operation (TxD2 pin is high).

TXE2[Note 1]	Stops/Enables Transmission
0	Stops transmission (transmission circuit is synchronously reset).
1	Enables transmission.

RXE2[Note 1]	Stops/Enables Reception
0	Stops reception (reception circuit is synchronously reset).
1	Enables reception.

Notes 1. To stop the serial transmission/reception, wait for the duration of 2 clocks[Note 2] after specifying stopping the transmission/reception (TXT2 = 0 or RXT2 = 0), and then stop the clock operation (POWER2 = 0).

2. The clock is the output clock of the 8-bit counter or the input clock of the baud rate generator.

15.4.2 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 communications 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 the asynchronous serial interface mode register 2 (ASIM2), asynchronous serial interface status register 2 (ASIS2), the baud rate generator control register 2 (BRGC2) asynchronous serial interface transmit status register 2 (ASIF2), clock select register 2 (CKSEL2), and transfer mode specification register 2 (TRMC2).

(a) Asynchronous serial interface mode register 2 (ASIM2)

ASIM2 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the value of ASIM2 to 00H.

Caution In UART mode, set the port mode register (PMXX) as follows. Set the output latch of the port set to output mode (PMXX = 0) to 0.

- **During receive operation**
  Set P35 (RXD2) to input mode (PM35 = 1)
- **During transmit operation**
  Set P34 (TXD2) to output mode (PM34 = 0)
- **During transmit/receive operation**
  Set P35 (RXD2) to input mode, and P34 to output mode

Address: FF90H At Reset: 00H R/W

Symbol	$\boxed{7}$	$\boxed{6}$	$\boxed{5}$	4	3	2	1	0
ASIM2	POWER2	TXE2	RXE2	PS21	PS20	CL2	SL2	ISRM2

POWER2	Enables/Stops Clock Operation
0	Stops clock operation. Power consumption decreases and latch in UART2 is asynchronously reset (TxD2 pin is low).
1	Enables clock operation (TxD2 pin is high).

TXE2[Note 1]	Stops/Enables Transmission
0	Stops transmission (transmission circuit is synchronously reset).
1	Enables transmission.

RXE2[Note 2]	Stops/Enables Reception
0	Stops reception (reception circuit is synchronously reset).
1	Enables reception.

Notes 1. To transmit data with UART2, first specify the clock operation (set POWER2 to 1 and then TXE2 to 1), wait for the duration of 2 clocks[Note 3], and then write the transmit data to transmit buffer register 2 (TXB2). To stop transmission by UART2, specify stopping transmission (TXE2 = 0), wait for the duration of 2 clocks[Note 3], and then stop the clock operation (POWER2 = 0).

2. To receive data with UART2, first specify the clock operation (set POWER2 to 1 and then RXE2 to 1), wait for the duration of 2 clocks[Note 3], and then start reception. To stop reception by UART2, specify stopping reception (RXT2 = 0), wait for the duration of 2 clocks[Note 3], and then stop the clock operation (POWER2 = 0).

3. The clock is the output clock of the 8-bit counter or the input clock of the baud rate generator.

PS21[Note 1]	PS20[Note 1]	Parity Bit Specification	
		Transmission	Reception
0	0	Does not output parity bit.	Reception without parity
0	1	Outputs 0 parity.	Reception as 0 parity[Note 2]
1	0	Outputs odd parity.	Identifies odd parity.
1	1	Outputs even parity.	Identifies even parity.

CL2[Note 3]	Data Character Length Specification
0	7 bits
1	8 bits

SL2[Note 4]	Specification for Number of Stop Bits for Transmission
0	1 bit
1	2 bits

ISRM2[Note 5]	Reception Error Interrupt Signal Control
0	INTSR2 is generated.
1	INTSER2 is generated.

Notes **1.** To specify a parity bit, stop transmission and reception (TXE2 = 0 and RXE2 = 0) before rewriting PS1 and PS0.

2. The parity is not identified with this setting. Therefore, bit 2 (PE2) of asynchronous serial interface status register 2 (ASIS2) is not set and the error interrupt does not occur.

3. To specify a data character length, stop transmission and reception (TXE2 = 0 and RXE2 = 0) before rewriting CL2.

4. To specify the number of stop bits, stop transmission (TXE2 = 0) before rewriting SL2. Reception is always performed on the assumption that the number of stop bits is 1.

5. To specify an interrupt that occurs in case of an error, stop transmission (TXE2 = 0) before rewriting ISEM2.

(b) Asynchronous serial interface status register 2 (ASIS2)

When a receive error occurs during UART mode, this register indicates the type of error.

ASIS2 can be read by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the value of ASIS2 to 00H.

Address: FFA1H At Reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS2	0	0	0	0	MPR2[Note 1]	PE2[Note 1]	FE2[Note 1]	OVE2[Note 1]

MPR2	ID Reception Status Flag (during reception in multi-processor transfer mode)[Note 2]
0	Multi-processor appended bit “1” is not received.
1	Multi-processor appended bit “1” is received.

PE2	Parity Error Flag
0	No parity error
1	Parity error (Incorrect parity bit detected[Note 3])

FE2	Framing Error Flag
0	No framing error
1	Framing error[Note 4] (Stop bit not detected)

OVE2	Overrun Error Flag
0	No overrun error
1	Overrun error[Note 5] (Next receive operation was completed before data was read from receive buffer register 2 (RXB2))

Notes 1. These bits are reset to 0 if bit 7 (POWER2) of asynchronous serial interface mode register 2 (ASIM2) is reset to 0.

2. This flag is affected only if the multi-processor transfer mode is selected by using bits 6 and 7 (TRM02 and TRM12) of transfer mode specification register 2 (TRMC2).

3. The operation of the parity error flag is affected by the set values of bits 3 and 4 (PS20 and PS21) of ASIM2.

4. Even if a stop bit length is set to two bits by setting bit 2 (SL2) in ASIM2, stop bit detection during a receive operation only applies to a stop bit length of 1 bit.

5. Be sure to read the contents of the receive buffer register 2 (RXB2) when an overrun error has occurred.

Until the contents of RXB2 are read, further overrun errors will occur when receiving data. The next receive data is not written to the receive buffer register 2 (RXB2) and is discarded.

(c) Baud rate generator control register 2 (BRGC2)

This register sets the serial clock for serial interface.

BRGC2 is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the value of BRGC2 to 00H.

Address: FF93H At Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
BRGC2	MDL27	MDL26	MDL25	MDL24	MDL23	MDL22	MDL21	MDL20

MDL27	MDL26	MDL25	MDL24	MDL23	MDL22	MDL21	MDL20	Input Clock Selection for Baud Rate Generator	k
0	0	0	0	0	×	×	×	Setting prohibited	—
0	0	0	0	1	0	0	0	$f_{SCK}/8$	8
0	0	0	0	1	0	0	1	$f_{SCK}/9$	9
0	0	0	0	1	0	1	0	$f_{SCK}/10$	10
0	0	0	0	1	0	1	1	$f_{SCK}/11$	11
0	0	0	0	1	1	0	0	$f_{SCK}/12$	12
0	0	0	0	1	1	0	1	$f_{SCK}/13$	13
0	0	0	0	1	1	1	0	$f_{SCK}/14$	14
0	0	0	0	1	1	1	1	$f_{SCK}/15$	15
0	0	0	1	0	0	0	0	$f_{SCK}/16$	16
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	1	1	1	1	1	1	1	$f_{SCK}/255$	255

Caution Writing to BRGC2 during a communication operation may cause abnormal output from the baud rate generator and disable further communication operations. Therefore, do not write to BRGC2 during a communication operation.

Before rewriting BRGC2, clear bits 5 and 6 (RXE2 and TXE2) of asynchronous serial interface mode register 2 (ASIM2) to 0.

Remarks 1. f_{SCK} : Source clock for 8-bit counter
Set by bits 4 through 6 (TPS20 through TPS22) of clock select register 2 (CKSEL2)

2. k : Value set via MDL27 to MDL20 ($8 \leq k \leq 255$)

(d) Asynchronous serial interface transmit status register 2 (ASIF2)

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

$\overline{\text{RESET}}$ input sets the value of ASIF2 to 00H.

Address: FF95H At Reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIF2	0	0	0	0	0	0	TXBF	TXSF

TXBF	Transmit Buffer Data Flag
0	• If bit 7 (POWER2) or bit 6 (TXE2) of asynchronous serial interface mode register 2 (ASIM2) is cleared to 0 • If data is transferred to transmit shift register 2 (TXS2)
1	If data is written to transmit buffer register 2 (TXB2) (if data exists in TXB2)

TXSF	Transmit Shift Register Data Flag
0	• If bit 7 (POWER2) or bit 6 (TXE2) of asynchronous serial interface mode register 2 (ASIM2) is cleared to 0 • If no more data is transferred from transmit buffer register 2 (TXB2) after completion of transfer.
1	If data is transferred from transmit buffer register 2 (TXB2) (during transmission)

Caution To transmit data successively, be sure to check the value of TXBF and write the data to the transmit buffer register 2 (TXB2). Data may be or may not be written to TXB2 depending on the transmission status, as indicated in the table below.

TXBF	TXSF	Transmission Status	Writing to TXB2
0	0	Initial status or completion of transmission	Possible
1	0	Transmission wait (data exists in TXB2)	Impossible
0	1	During transmission (no data in TXB2)	Possible
1	1	During transmission (data exists in TXB2)	Impossible

(e) Clock select register 2 (CKSEL2)

This register is set by using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the value of CKSEL2 to 00H.

Address: FF92H At Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CKSEL2	0	TPS22[Note]	TPS21[Note]	TPS20[Note]	TPW23	TPW22	TPW21	TPW20

TPS22	TPS21	TPS20	Source Clock of 8-bit Counter	n
0	0	0	ASCK2/$\overline{\text{SCK3}}$/P36	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

Note To rewrite TPS0 through TPS2, clear bit 7 (POWER2) of the asynchronous serial interface mode register 2 (ASIM2) to 0.

Caution If data is written to CKSEL2 during a communication operation, the output of the baud rate generator is disturbed and the communication cannot be performed correctly. Therefore, do not rewrite CKSEL2 during communication.

(f) Transfer mode specification register 2 (TRMC2)

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

$\overline{\text{RESET}}$ input sets the value of TRMC2 to 02H.

Address: FF91H At Reset: 02H R/W

Symbol	7	6	5	4	3	2	$\boxed{1}$	$\boxed{0}$
TRMC2	TRM12[Note 1]	TRM02[Note 1]	0	0	0	0	MPIEN2	MPS2

TRM12	TRM02	Transfer Mode
0	0	UART transfer mode[Note 2]
0	1	Multi-processor transfer mode[Note 2]
1	0	Infrared data transfer (IrDA) mode[Note 2]
1	1	

Notes 1. Before rewriting TRM12 and TRM02, clear bits 6 (TXE2) and 5 (RXE2) of asynchronous serial interface mode register 2 (ASIM2) to 0.

2. The setting of bits 0 through 4 (ISEM2, SL2, CL2, PS20, and PS21) of ASIM2 is valid in all the transfer modes.

The transmit/receive clock that is used to generate the baud rate is obtained by dividing the main system clock.

- Transmit/receive clock generation for baud rate by using main system clock
  The main system clock is divided to generate the transmit/receive clock. The baud rate generated from the main system clock is determined according to the following formula.

  $$[\text{Baud rate}] = \frac{f_X}{2^{n+1} \times k} \ [\text{Hz}]$$

  f_X : Oscillation frequency of main system clock
  When ASCK2 is selected as the source clock of the 8-bit counter, substitute the input clock frequency to ASCK2 pin for f_X in the above expression.
  n : Value set via TPS20 to TPS22 ($0 \le n \le 7$)
  For details, see **Table 16-2**.
  k : Value set via MDL27 to MDL20 ($8 \le k \le 255$)

Table 15-2 shows the relationship between the 8-bit counter's source clock assigned to bits 4 to 6 (TPS20 to TPS22) of CKSEL2 and the "n" value in the above formula.

Table 15-2. Relationship between 8-Bit Counter's Source Clock and "n" Value

TPS22	TPS21	TPS20	8-Bit Counter's Source Clock Selected	n
0	0	0	ASCK2/$\overline{\text{SCK3}}$/P36	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

Remark f_X: Oscillation frequency of main system clock

Table 15-3 shows the relationship between the input clock of the baud rate generator of the counter assigned to bits 0 through 7 (MDL20 through MDL27) of BRGC2 and “k”.

Table 15-3. Relationship between Input Clock of Baud Rate Generator and “k” Value

MDL27	MDL26	MDL25	MDL24	MDL23	MDL22	MDL21	MDL20	Input Clock Selection of Baud Rate Generator	k
0	0	0	0	0	×	×	×	Setting prohibited	—
0	0	0	0	1	0	0	0	f_{SCK}/8	8
0	0	0	0	1	0	0	1	f_{SCK}/9	9
0	0	0	0	1	0	1	0	f_{SCK}/10	10
0	0	0	0	1	0	1	1	f_{SCK}/11	11
0	0	0	0	1	1	0	0	f_{SCK}/12	12
0	0	0	0	1	1	0	1	f_{SCK}/13	13
0	0	0	0	1	1	1	0	f_{SCK}/14	14
0	0	0	0	1	1	1	1	f_{SCK}/15	15
0	0	0	1	0	0	0	0	f_{SCK}/16	16
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	1	1	1	1	1	1	1	f_{SCK}/255	255

- Baud rate error

  The baud rate error can be calculated by the following expression:

$$[\text{Baud rate error}] = \frac{\text{Baud rate [bps]}}{\text{Targeted baud rate [bps]}} \times 100 - 100\ [\%]$$

Table 15-4 shows an example of the relationship between the main system clock and a baud rate.

Table 15-4. Relationship between Main System Clock and Baud Rate

Baud Rate [bps]	f_X = 7.37 MHz			f_X = 5.0 MHz			f_X = 4.19 MHz		
	n	k	Error (%)	n	k	Error (%)	n	k	Error (%)
300	7	96	0	7	65	0.16	6	109	0.18
600	7	48	0	6	65	0.16	5	109	0.18
1200	7	24	0	5	65	0.16	4	109	0.18
2400	6	24	0	4	65	0.16	3	109	0.18
4800	5	24	0	3	65	0.16	2	109	0.18
9600	4	24	0	2	65	0.16	1	109	0.18
19200	3	24	0	1	65	0.16	–	–	–
31250	1	59	–0.03	1	40	0	–	–	–
38400	2	24	0	–	–	–	–	–	–
76800	1	24	0	–	–	–	–	–	–

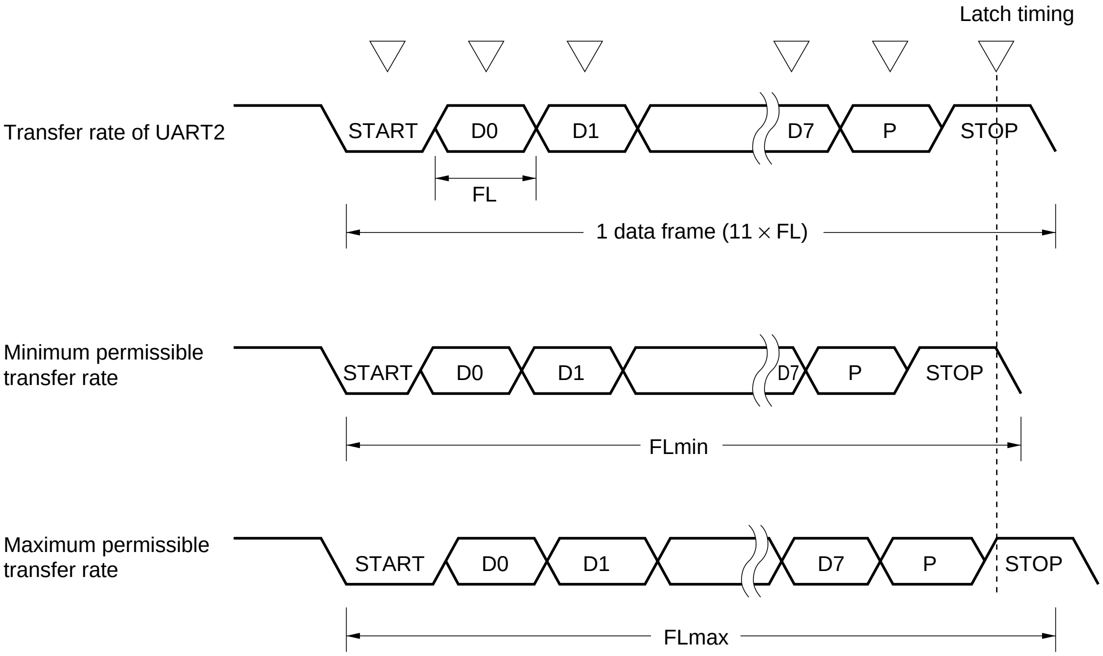
Remarks f_X : Main system clock oscillation frequency

n : Value set by TPS20 through TPS22 ($0 \le n \le 7$)

k : Value set by MDL27 through MDL20 ($8 \le k \le 255$)

- Permissible baud rate range for reception

Figure 15-9. Minimum Permissible Transfer Rate and Maximum Permissible Transfer Rate



As shown in the timing chart in Figure 15-9, the latch timing of the receive data is determined by the counter that has been set by using baud rate generator control register 2 (BRGC2) after the stop bit has been detected.

If the last data (stop bit) is received within this latch timing, the data can be correctly received.

This latch timing has a margin of two clocks.

Take reception of 11-bit data as an example,

1 bit data length of UART2: FL = $(\text{Brate})^{-1}$

Minimum permissible transfer rate: $\text{FLmin} = 11 \times \text{FL} - \frac{k-2}{2k} \times \text{FL}$

$$= \frac{21k+2}{2k} \times \text{FL}$$

Therefore, the maximum receivable baud rate of the transmission destination is as follows:

$$\text{BRmax} = (\text{Flmin}/11)^{-1} = \frac{22k}{21k+2}\ \text{Brate}$$

Similarly, the maximum permissible transfer rate is as follows:

$$\frac{10}{11} \text{ FLmax} = 11 \times \text{FL} - \frac{k+2}{2k} \times \text{FL}$$

$$= \frac{21k-2}{2k} \times \text{FL}$$

$$\text{FLmax} = \frac{21k-2}{20k} \times \text{FL} \times 11$$

Therefore, the maximum receivable baud rate of the transmission destination is as follows:

$$\text{BRmin} = (\text{FLmax}/11)^{-1} = \frac{20k}{21k-2} \text{ Brate}$$

Remark Brate: Baud rate of UART2
k : Value set by MDL27 through MDL20 ($8 \le k \le 255$)
FL : 1 bit data length

From the above expressions for the maximum and minimum baud rates, the permissible error of the baud rate between UART2 and the transmission destination can be calculated as follows:

Table 15-5. Maximum Permissible Baud Rate Error and Minimum Permissible Baud Rate Error

k	Maximum Permissible Baud Rate Error (%)	Minimum Permissible Baud Rate Error (%)
8	+3.53	–3.61
20	+4.26	–4.31
50	+4.56	–4.58
100	+4.66	–4.67
255	+4.72	–4.73

Remark k: Value set by MDL27 through MDL20 ($8 \le k \le 255$)

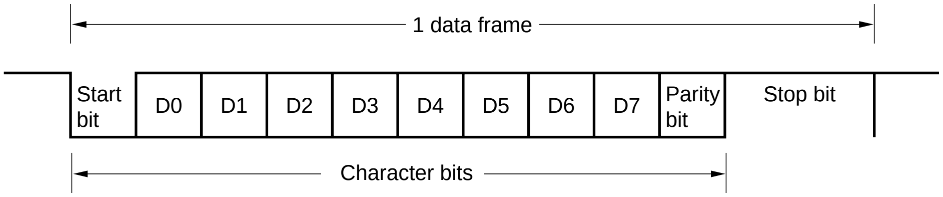
The accuracy of reception is dependent upon the number of bits in one frame, input clock frequency, and division ratio k (the higher the input clock frequency and the higher the division ratio k, the higher the accuracy).

(2) Communication operations

(a) Data format

Figure 15-10 shows the format of the transmit/receive data.

Figure 15-10. Format of Transmit/Receive Data in Asynchronous Serial Interface



1 data frame consists of the following bits.

- Start bit .... 1 bit
- Character bits ... 7 bits or 8 bits
- Parity bit .... Even parity, odd parity, zero parity, or no parity
- Stop bit(s) .... 1 bit or 2 bits

The asynchronous serial interface mode register 2 (ASIM2) 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 low-order 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”.

The ASIM2, the baud rate generator control register 2 (BRGC2), and the clock select register 2 (CKSEL2) are used to set the serial transfer rate.

If a receive error occurs, information about the receive error can be recognized by reading the asynchronous serial interface status register 2 (ASIS2).

(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.

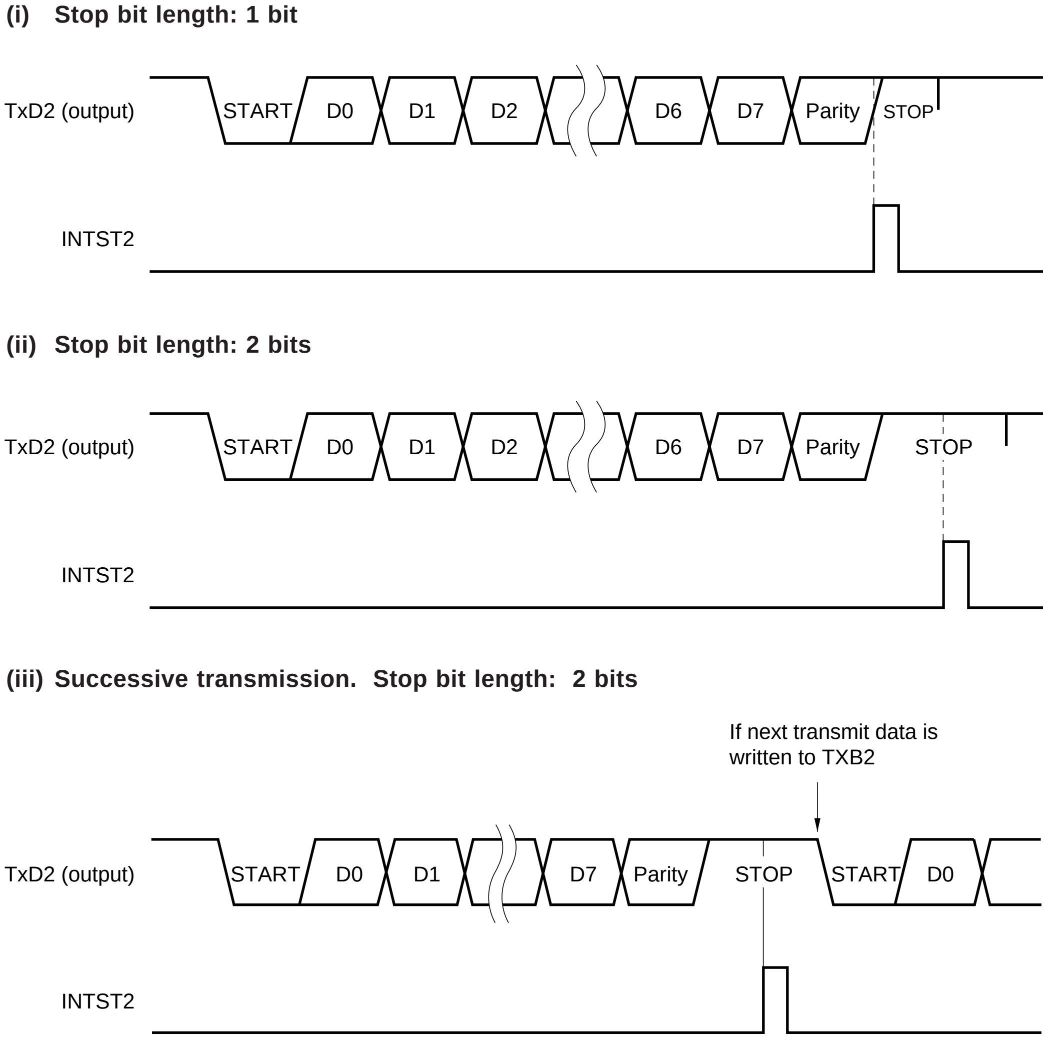
(c) Transmission

If the UART transfer mode is selected by using transfer mode specification register 2 (TRMC2) and bit 7 (POWER2) of asynchronous serial interface mode register 2 (ASIM2) is set to 1, the TxD2 pin outputs a high level. If bit 6 (TxE2) of ASIM2 is set to 1 next, transmission is enabled. Transmission can be started by writing transmit data to transmit buffer register 2 (TXB2). The start bit, parity bit, and stop bit are automatically appended to the transmit data.

When transmission has been started, the data in TXB2 is transferred to transmit shift register 2 (TXS2) and is sequentially output to the TxD2 pin, starting from the LSB. If the data to be transmitted next has been written to TXB2 when transmission has been completed, transmitting the next data is started. If no more data has been written to TXB2, transmission is stopped, until the next data is written.

Figure 15-11 illustrates the timing of the transmission interrupt.

Figure 15-11. Timing of Asynchronous Serial Interface Transmit Completion Interrupt Request



(i) Stop bit length: 1 bit

(ii) Stop bit length: 2 bits

(iii) Successive transmission. Stop bit length: 2 bits

Caution Do not rewrite to the asynchronous serial interface mode register 2 (ASIM2) during a transmit operation. Rewriting ASIM2 register during a transmit operation may disable further transmit operations (in such cases, enter a $\overline{\text{RESET}}$ to restore normal operation). Whether or not a transmit operation is in progress can be determined via software using the transmit completion interrupt request (INTST2) or the interrupt request flag (STIF2) that is set by INTST2.

(d) Successive transmission

The next transmit data can be written to transmit buffer register 2 (TXB2) as soon as transmit shift register 2 (TXB2) has started its shift operation. Consequently, even while an interrupt is being serviced after one data frame has been transmitted, data can be successively transmitted.

To successively transmit data, be sure to check, by using asynchronous serial interface transmit status register 2 (ASIF2), the transmission status and whether writing to TXB2 is enabled or disabled, and then write the data to TXB2.

The following table shows the relationship between the transmission status and writing to TXB2.

Table 15-6. Transmission Status and Writing to TXB2

TXBF	TXSF	Transmission Status	Writing to TXB2
0	0	Initial status or completion of transmission	Possible
1	0	Transmission wait (data exists in TXB2)	Impossible
0	1	During transmission (no data in TXB2)	Possible
1	1	During transmission (data exists in TXB2)	Impossible

Remark TXBF: Bit 1 of ASIF2
TXSF: Bit 0 of ASIF2

Caution Write transmit data to TXB2 when TXBF is 0. Initialize the transmit unit (TXS2) when TXSF is 0.

The following figures and tables show the timing of starting and completing successive transmission.

Figure 15-12. Timing of Starting Successive Transmission

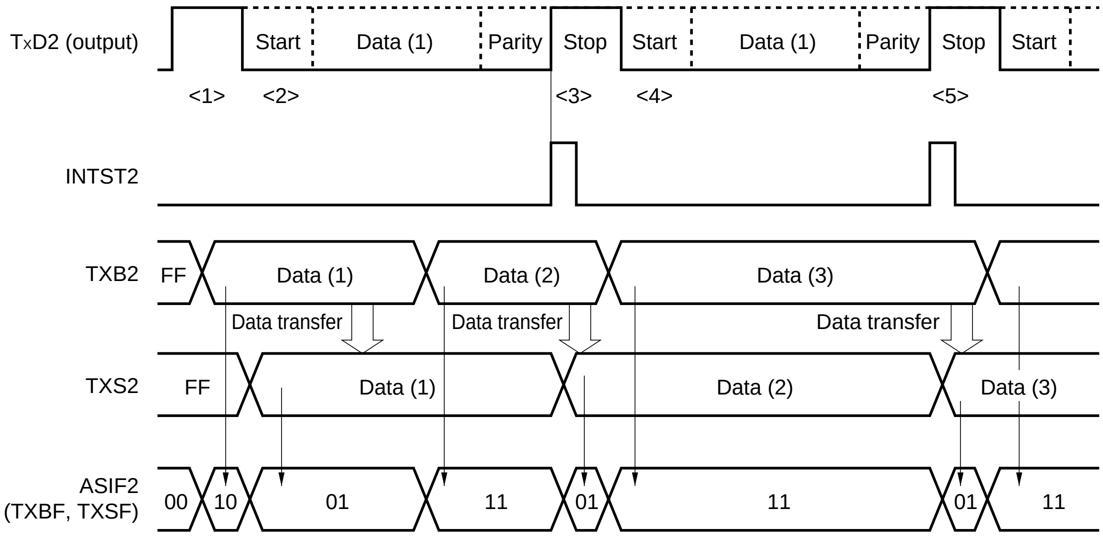


Table 15-7. Timing of Starting Successive Transmission

Transmission Procedure	Internal Operation	TXBF	TXSF
Sets transmission mode.	<1> Starts transmission unit.	0	0
Writes data (1).		1	0
	<2> Generates start bit and starts transmitting data (1).	0	1
Reads ASIF2 (to confirm TXBF = 0) and writes data (2).		1	1
	(during transmission)		
	<3> Interrupt (INTST2) occurs.	0	1
Reads ASIF2 (to confirm TXBF = 0) and writes data (3).		1	1
	<4> Generates start bit and starts transmitting data (2).		
	(during transmission)		
	<5> Interrupt (INTST2) occurs.	0	1
Reads ASIF2 (to confirm TXBF = 0) and writes data (3).		1	1

Remarks 1. <1> through <5> in this table correspond to <1> through <5> in Figure 15-12.

2. TXBF: Bit 1 (transmit buffer data flag) of asynchronous serial interface transmit status register 2 (ASIF2)

TXSF: Bit 0 of ASIF2 (transmit shift register data flag)

Figure 15-13. Timing of Completing Successive Transmission

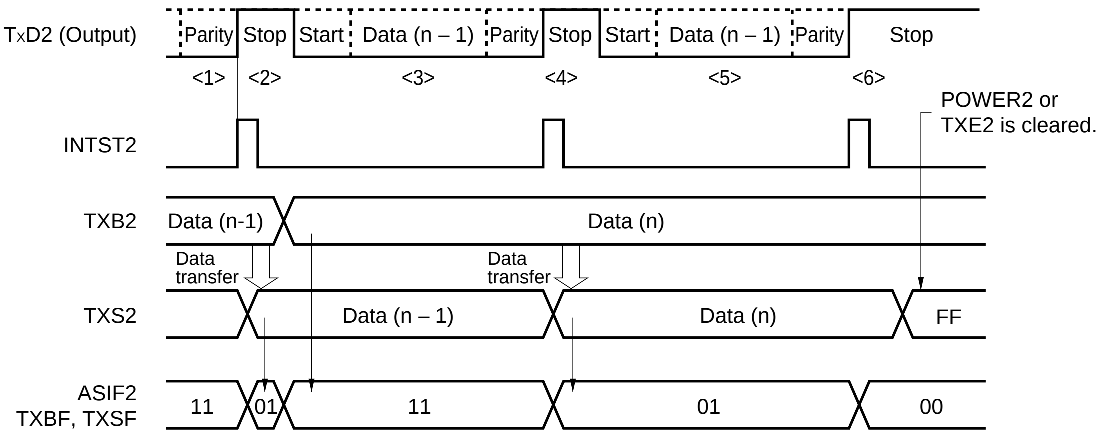


Table 15-8. Timing of Completing Successive Transmission

Transmission Procedure	Internal Operation	TXBF	TXSF
	<1> Data (n–2) is transmitted.	1	1
	<2> Interrupt (INTST2) occurs.	0	1
Reads ASIF2 (to confirm TXBF = 0) and writes data (n).		1	1
	<3> Generates start bit and starts transmitting data (n–1).		
	(during transmission)		
	<4> Interrupt (INTST2) occurs.	0	1
Reads ASIF2 (to confirm TXBF = 0). No data to be written.		1	1
	<5> Generates start bit and starts transmitting data (n).		
	(during transmission)		
	<6> Interrupt (INTST2) occurs.	0	0
Reads ASIF2 (to confirm TXBF = 0) and clears POWER2 or TXE2.	Initializes internal circuit.		

Remarks 1. <1> through <6> in this table correspond to <1> through <6> in Figure 15-13.

2. TXBF : Bit 1 (transmit buffer data flag) of asynchronous serial interface transmission status register 2 (ASIF2)

TXSF : Bit 0 of ASIF2 (transmit shift register data flag)

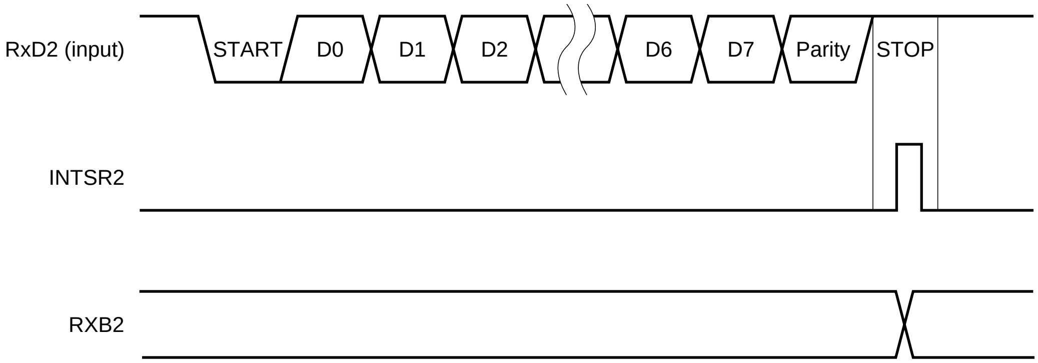
POWER2 : Bit 7 of asynchronous serial interface mode register 2 (ASIM2)

TXE2 : Bit 6 of ASIM2

(d) Reception

The interface enters the reception wait status if the UART transfer mode is specified by using transfer mode specification register 2 (TRMC2) and bit 5 (RXE2) of asynchronous serial interface mode register 2 (ASIM2) is set to 1. In this status, the RxD2 pin is monitored to detect the start bit. If the start bit is detected, reception is started, and serial data is sequentially stored in receive shift register 2 (RX2) at the specified baud rate. If the stop bit is received, a reception completion interrupt (INTSR) occurs and, at the same time, the data in RX2 is written to receive buffer register 2 (RXB2). If an overrun error (OVE2) occurs, however, the receive data is not written to RXB2 but discarded. Even if a parity error (PE2) or framing error (FE2) occurs during reception, reception continues up to the position at which the stop bit is received, and an error interrupt (INTSR2/INTSER2) occurs after completion of the reception.

Figure 15-14. Timing of Asynchronous Serial Interface Receive Completion Interrupt Request



Cautions 1. Be sure to read the contents of the receive buffer register 2 (RXB2) 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 RXB2 are read.

2. During reception, the number of stop bits is always 1. The second stop bit is discarded.

(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 the asynchronous serial interface status register 2 (ASIS2), a receive error interrupt request (INTSR2/INTSER2) will occur. Table 15-9 lists the causes behind receive errors.

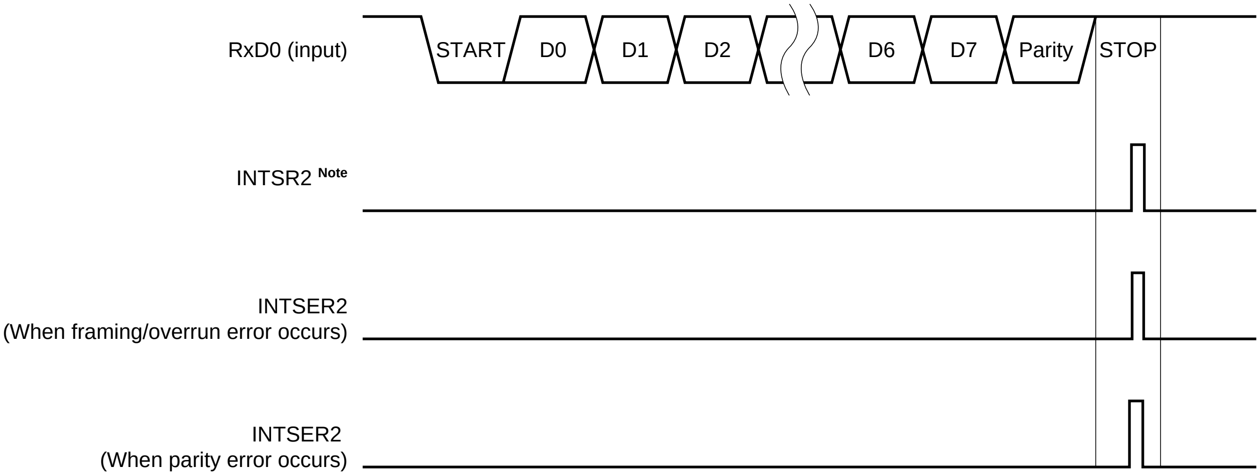
As part of receive error interrupt request (INTSR2/INTSER2) servicing, the contents of ASIS2 can be read to determine which type of error occurred during the receive operation (see **Table 15-9** and **Figure 15-15**).

The contents of ASIS2 are reset (to “0”) when the receive buffer register 2 (RXB2) is read or when the next data is received (if the next data contains an error, its error flag will be set).

Table 15-9. Causes of Receive Errors

Receive Error	Cause	ASIS2 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 the next data was completed before data was read from the receive buffer register 2 (RXB2)	01H

Figure 15-15. Receive Error Timing



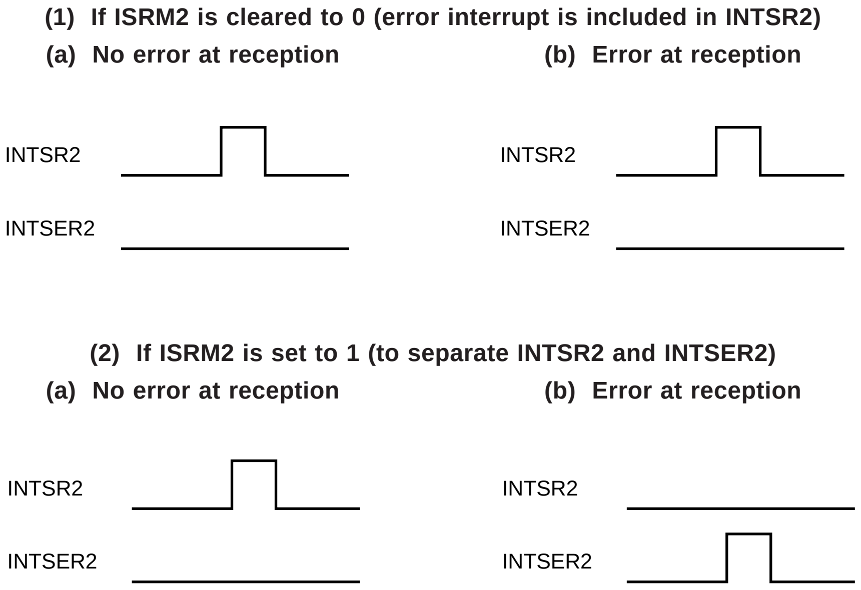
Note If a receive error occurs when ISRM2 bit has been set (1), INTSR2 does not occur.

Cautions 1. The contents of asynchronous serial interface status register 2 (ASIS2) are reset (to “0”) when the receive buffer register 2 (RXB2) is read or when the next data is received. To obtain information about the error, be sure to read the contents of ASIS2 before reading RXB2.

2. Be sure to read the contents of the receive buffer register 2 (RXB2) 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 RXB2 are read.

The error interrupts can be divided into INTSR2 and INTSER2 by setting bit 0 (ISRM2) of asynchronous serial interface mode register 2 (ASIM2) to 1.

Figure 15-16. Reception Error Interrupts (INTSR2/INTSER2)



(1) If ISRM2 is cleared to 0 (error interrupt is included in INTSR2)

(a) No error at reception

(b) Error at reception

(2) If ISRM2 is set to 1 (to separate INTSR2 and INTSER2)

(a) No error at reception

(b) Error at reception

15.4.3 Multi-processor transfer mode

In this mode, data can be transferred to or received from two or more processors.

(1) Register setting

The multi-processor transfer mode is selected by using asynchronous serial interface mode register 2 (ASIM2), asynchronous serial interface status register 2 (ASIS2), baud rate generator control register 2 (BRGC2), asynchronous serial interface transmit status register 2 (ASIF2), clock select register 2 (CKSEL2), and transfer mode specification register 2 (TRMC2).

(a) Asynchronous serial interface mode register 2 (ASIM2)

ASIM2 can be set by using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the value of ASIM2 to 00H.

Caution In the multi-processor transfer mode, set the port mode register (PMXX) as follows. Set the output latch of the port set to output mode (PMXX = 0) to 0.

- **Reception**
  Setting P35 (RxD2) in the input mode (PM35 = 1)
- **Transmission**
  Setting P34 (TxD2) in the output mode (PM34 = 0)
- **Transmission/reception**
  Setting P35 in the input mode and P34 in the output mode

Address: FF90H At Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ASIM2	POWER2	TXE2	RXE2	PS21	PS20	CL2	SL2	ISRM2

POWER2	Enables/Stops Clock Operation
0	Stops clock operation. Power consumption decreases and latch in UART2 is asynchronously reset (TxD2 pin is low).
1	Enables clock operation (TxD2 pin is high).

TXE2[Note 1]	Stops/Enables Transmission
0	Stops transmission (transmission circuit is synchronously reset).
1	Enables transmission.

RXE2[Note 2]	Stops/Enables Reception
0	Stops reception (reception circuit is synchronously reset).
1	Enables reception.

Notes 1. To transmit data with UART2, first specify the clock operation (set POWER2 to 1 and then TXE2 to 1), wait for the duration of 2 clocks[Note 3], and then write the transmit data to transmit buffer register 2 (TXB2). To stop transmission by UART2, specify stopping transmission (TXE2 = 0), wait for the duration of 2 clocks[Note 3], and then stop the clock operation (POWER2 = 0).

2. To receive data with UART2, first specify the clock operation (set POWER2 to 1 and then RXE2 to 1), wait for the duration of 2 clocks[Note 3], and then start reception. To stop reception by UART2, specify stopping reception (RXT2 = 0), wait for the duration of 2 clocks[Note 3], and then stop the clock operation (POWER2 = 0).

3. The clock is the output clock of the 8-bit counter or the input clock of the baud rate generator.

PS21[Note 1]	PS20[Note 1]	Parity Bit Specification	
		Transmission	Reception
0	0	Does not output parity bit.	Reception without parity
0	1	Outputs 0 parity.	Reception as 0 parity[Note 2]
1	0	Outputs odd parity.	Identifies odd parity.
1	1	Outputs even parity.	Identifies even parity.

CL2[Note 3]	Data Character Length Specification
0	7 bits
1	8 bits

SL2[Note 4]	Specification for Number of Stop Bits for Transmission
0	1 bit
1	2 bits

ISRM2[Note 5]	Reception Error Interrupt Signal Control
0	INTSR2 is generated.
1	INTSER2 is generated.

Notes **1.** To specify a parity bit, stop transmission and reception (TXE2 = 0 and RXE2 = 0) before rewriting PS1 and PS0.

2. The parity is not identified with this setting. Therefore, bit 2 (PE2) of asynchronous serial interface status register 2 (ASIS2) is not set and the error interrupt does not occur.

3. To specify a data character length, stop transmission and reception (TXE2 = 0 and RXE2 = 0) before rewriting CL2.

4. To specify the number of stop bits, stop transmission (TXE2 = 0) before rewriting SL2. Reception is always performed on the assumption that the number of stop bits is 1.

5. To specify an interrupt that occurs in case of an error, stop transmission (TXE2 = 0) before rewriting ISEM2.

(b) Asynchronous serial interface status register 2 (ASIS2)

When a receive error occurs during UART mode, this register indicates the type of error.

ASIS2 can be read by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the value of ASIS2 to 00H.

Address: FFA1H At Reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS2	0	0	0	0	MPR2^Note 1	PE2^Note 1	FE2^Note 1	OVE2^Note 1

MPR2	ID Reception Status Flag (during reception in multi-processor transfer mode)^Note 2
0	Multi-processor appended bit “1” is not received.
1	Multi-processor appended bit “1” is received.

PE2	Parity Error Flag
0	No parity error
1	Parity error (Incorrect parity bit detected^Note 3)

FE2	Framing Error Flag
0	No framing error
1	Framing error^Note 4 (Stop bit not detected)

OVE2	Overrun Error Flag
0	No overrun error
1	Overrun error^Note 5 (Next receive operation was completed before data was read from receive buffer register 2 (RXB2))

Notes 1. These bits are reset to 0 if bit 7 (POWER2) of asynchronous serial interface mode register 2 (ASIM2) is reset to 0.

2. This flag is affected only if the multi-processor transfer mode is selected by using bits 6 and 7 (TRM02 and TRM12) of transfer mode specification register 2 (TRMC2).

3. The operation of the parity error flag is affected by the set values of bits 3 and 4 (PS20 and PS21) of ASIM2.

4. Even if a stop bit length is set to two bits by setting bit 2 (SL2) in ASIM2, stop bit detection during a receive operation only applies to a stop bit length of 1 bit.

5. Be sure to read the contents of the receive buffer register 2 (RXB2) when an overrun error has occurred.

Until the contents of RXB2 are read, further overrun errors will occur when receiving data. The next receive data is not written to the receive buffer register 2 (RXB2) and is discarded.

(c) Baud rate generator control register 2 (BRGC2)

This register sets the serial clock for serial interface.

BRGC2 is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the value of BRGC2 to 00H.

Address: FF93H At Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
BRGC2	MDL27	MDL26	MDL25	MDL24	MDL23	MDL22	MDL21	MDL20

MDL27	MDL26	MDL25	MDL24	MDL23	MDL22	MDL21	MDL20	Input Clock Selection for Baud Rate Generator	k
0	0	0	0	0	×	×	×	Setting prohibited	—
0	0	0	0	1	0	0	0	$f_{SCK}/8$	8
0	0	0	0	1	0	0	1	$f_{SCK}/9$	9
0	0	0	0	1	0	1	0	$f_{SCK}/10$	10
0	0	0	0	1	0	1	1	$f_{SCK}/11$	11
0	0	0	0	1	1	0	0	$f_{SCK}/12$	12
0	0	0	0	1	1	0	1	$f_{SCK}/13$	13
0	0	0	0	1	1	1	0	$f_{SCK}/14$	14
0	0	0	0	1	1	1	1	$f_{SCK}/15$	15
0	0	0	1	0	0	0	0	$f_{SCK}/16$	16
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	1	1	1	1	1	1	1	$f_{SCK}/255$	255

Caution 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.

Before rewriting BRGC2, clear bits 5 and 6 (RXE2 and TXE2) of asynchronous serial interface mode register 2 (ASIM2) to 0.

Remarks 1. f_{SCK} : Source clock for 8-bit counter
Set by bits 4 through 6 (TPS20 through TPS22) of clock select register 2 (CKSEL2)

2. k : Value set via MDL27 to MDL20 ($8 \leq k \leq 255$)

(d) Asynchronous serial interface transmit status register 2 (ASIF2)

This register indicates the status of transmission.

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

$\overline{\text{RESET}}$ input sets the value of ASIF2 to 00H.

Address: FF95H At Reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIF2	0	0	0	0	0	0	TXBF	TXSF

TXBF	Transmit Buffer Data Flag
0	• If bit 7 (POWER2) or bit 6 (TXE2) of asynchronous serial interface mode register 2 (ASIM2) is cleared to 0 • If data is transferred to transmit shift register 2 (TXS2)
1	If data is written to transmit buffer register 2 (TXB2) (if data exists in TXB2)

TXSF	Transmit Shift Register Data Flag
0	• If bit 7 (POWER2) or bit 6 (TXE2) of asynchronous serial interface mode register 2 (ASIM2) is cleared to 0 • f no more data is transferred from transmit buffer register 2 (TXB2) after completion of transfer.
1	If data is transferred from transmit buffer register 2 (TXB2) (during transmission)

Caution To transmit data successively, be sure to check the value of TXBF and write the data to the transmit buffer register 2 (TXB2). Data may or may not be written to TXB2 depending on the transmission status, as indicated in the table below.

TXBF	TXSF	Transmission Status	Writing to TXB2
0	0	Initial status or completion of transmission	Possible
1	0	Transmission wait (data exists in TXB2)	Impossible
0	1	During transmission (no data in TXB2)	Possible
1	1	During transmission (data exists in TXB2)	Impossible

(e) Clock select register 2 (CKSEL2)

This register is set by using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the value of CKSEL2 to 00H.

Address: FF92H At Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CKSEL2	0	TPS22[Note]	TPS21[Note]	TPS20[Note]	TPW23	TPW22	TPW21	TPW20

TPS22	TPS21	TPS20	Source Clock of 8-bit Counter	n
0	0	0	ASCK2/$\overline{\text{SCK3}}$/P36	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

Note To rewrite TPS0 through TPS2, clear bit 7 (POWER2) of the asynchronous serial interface mode register 2 (ASIM2) to 0.

Caution If data is written to CKSEL2 during a communication operation, the output of the baud rate generator is disturbed and the communication cannot be performed correctly. Therefore, do not rewrite CKSEL2 during communication.

(f) Transfer mode specification register 2 (TRMC2)

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

RESET input sets the value of TRMC2 to 02H.

Address: FF91H At Reset: 02H R/W

Symbol	7	6	5	4	3	2	1	0
TRMC2	TRM12[Note 1]	TRM02[Note 1]	0	0	0	0	MPIEN2	MPS2[Note 2]

TRM12	TRM02	Transfer Mode
0	0	UART transfer mode[Note 3]
0	1	Multi-processor transfer mode[Note 3]
1	0	Infrared data transfer (IrDA) mode[Note 3]
1	1	

MPIEN2	Enables/disables reception completion interrupt in multi-processor transfer mode[Note 4]	
	Condition	Enables/disables INTSR2[Note 5]
0[Note 6]	If “0” is written to this bit	Disabled
1	• If it 7 (POWER2) or bit 6 (TXE2) of asynchronous serial interface mode register 2 (ASIM2) is cleared to 0 • If multi-processor appended data has received data of “1”	Enabled

MPS2	Setting of multi-processor transmission appended bit[Note 4]
0	Appends “0” to and transmits multi-processor appended bit (during data transmission).
1	Appends “1” to and transmits multi-processor appended bit (during ID transmission).

Notes 1. Before rewriting TRM12 and TRM02, clear bits 6 (TXE2) and 5 (RXE2) of asynchronous serial interface mode register 2 (ASIM2) to 0.

2. Before setting a value to MPS2, confirm that bit 1 (TXBF) of asynchronous serial interface transmit status register 2 (ASIF2) is cleared to 0. Before writing transmit data to the transmit buffer register 2 (TXB2), specify whether “0” or “1” is appended to the multi-processor appended bit.

3. The setting of bits 0 through 4 (ISEM2, SL2, CL2, PS20, and PS21) of ASIM2 is valid in all the transfer modes.

4. The specification by MPIEN2 and MPS2 is valid only when bit 7 (TRM12) is cleared to 0 and bit 6 (TRM02) is set to 1 (i.e., when the multi-processor transfer mode is set).

5. Enabling or disabling the occurrence of the reception completion interrupt (INTSR2) in case of an error is affected by the setting of bit 0 (ISRM2) of ASIM2.

6. Even if MPIEN is cleared to 0, reception is started if the start bit is detected, in order to detect address (ID) reception. At this time, error in the receive data is not detected if the multi-processor appended bit is “0”. If data “1” is received by mistake, due to bit slip or other cause, when the multi-processor appended bit is detected, however, ID reception is detected. Consequently, the error in the receive data is identified, and the error interrupt signal may be generated and the error flag may be set.

Remark To receive data in the multi-processor transfer mode, the reception completion interrupt (INTSR2) occurs, regardless of the value of MPIEN2, if data with the multi-processor appended bit set to “1” is received. Usually, this receive data is an address (ID) that indicates the other party of communication. The subsequent receive data can be ignored and the occurrence of an unnecessary reception completion interrupt (INTSR2) can be disabled by comparing this received ID with the ID of the microcontroller (for which software processing is necessary) and clearing MPIEN2 if the two IDs do not match.

For an explanation how to generate the transmit/receive clock for baud rate and details of the permissible error range of the baud rate, refer to **15.4.2 Asynchronous serial interface (UART) mode**.

(2) Communication

(a) Data format

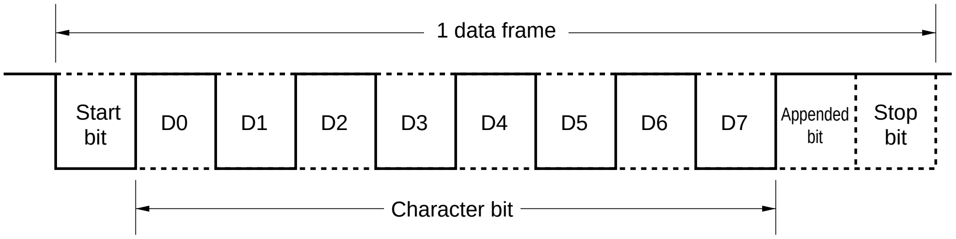
Figure 15-17 shows the format of the transmit/receive data.

Figure 15-17. Format of Transmit/Receive Data in Multi-Processor Transfer Mode

(1) ID transfer (multi-processor appended bit = 1) format

Character bit: 8 bits, No parity, Stop bit: 1 bit

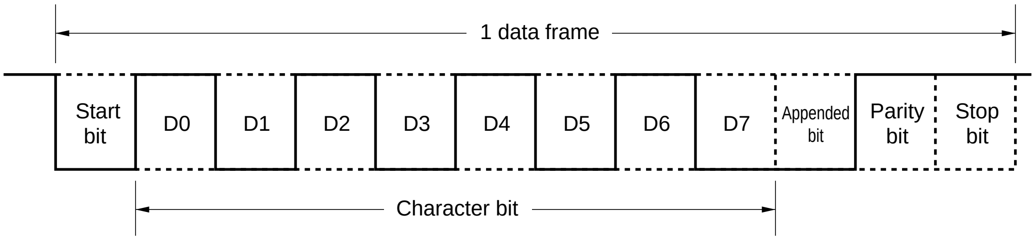
Transfer data: 55H



(2) Data transfer (multi-processor appended bit = 0) format

Character bit: 8 bits, Parity bit: odd bit

Stop bit: 1 bit, Transfer data: 55H



Caution **If parity is specified, the parity bit is output after the multi-processor appended bit. In this case, the multi-processor appended bit is subject to parity calculation during both transmission and reception.**

One data frame consists of the following bits:

• Start bit	1 bit
• Character bit	7/8 bits (LSB first)
• Multi-processor appended bit	1 bit (set to 1 or 0)
• Parity bit	Even/odd/0/none
• Stop bit	1/2 bits

The character bit length, parity, and stop bit length in one data frame are selected by asynchronous interface mode register 2 (ASIM2). Data is transferred starting from the LSB.

The multi-processor appended bit of transmit data is specified by transfer mode specification register 2 (TRMC2).

The serial transfer rate is selected by ASIM2, baud rate generator control register 2 (BRGC2), and clock select register 2 (CKSEL2).

If an error occurs when receiving serial data, the error can be identified by reading the status of asynchronous serial interface status register 2 (ASIS2).

(b) Transmission

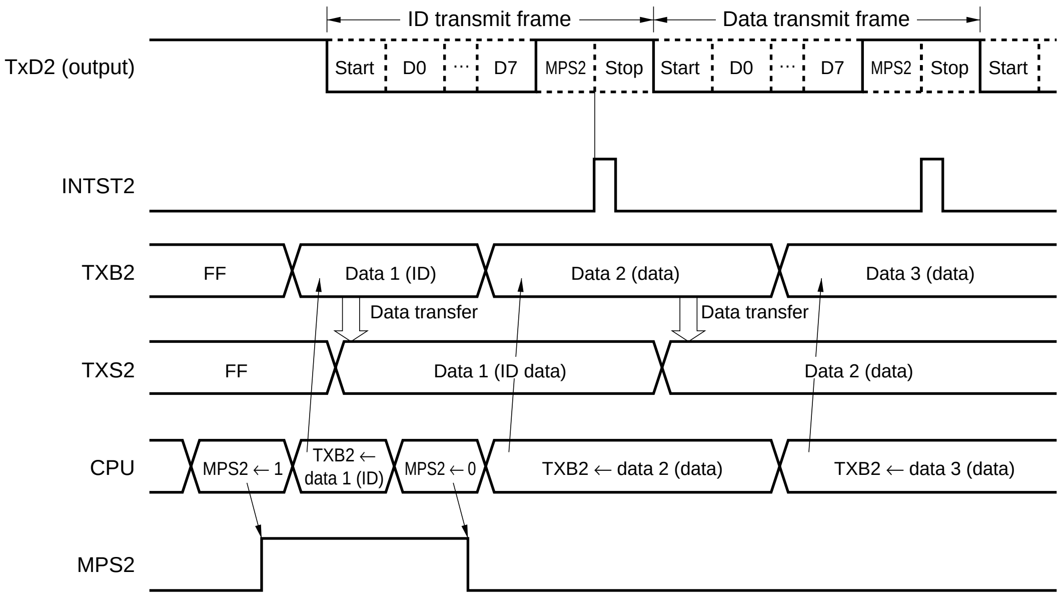
If the multi-processor transfer mode is set by using transfer mode specification register 2 (TRMC2) and bit 7 (POWER2) of asynchronous serial interface mode register 2 (ASIM2) is set to 1, the TxD2 pin outputs a high level. If bit 6 (TXE2) of ASIM2 is set to 1 next, transmission is enabled. Transmission (ID transmission) can be started by setting bit 0 (MPS2) of TRMC2 to 1 and writing transmit data to the transmit buffer register 2 (TXB2).

Next, confirm that bit 1 (TXBF) of asynchronous serial interface transmit status register 2 (ASIF2) is 0. Then clear MPS and write transmit data to TXB2 (data transmission). The start bit, multi-processor transfer appended bit, parity bit, and stop bit are automatically appended to the data.

When transmission is started, the data in TXB2 is transferred to the transmit shift register 2 (TXS2) and sequentially output to the TxD2 pin, starting from the LSB. If the data to be transmitted next has been written to TXB2 when transmission has been completed, transmitting the next data is started. If no more data has been written to TXB2, transmission is stopped, until new transmit data is written.

Figure 15-18 shows the timing of a transmission interrupt.

Figure 15-18. Timing of Transmission Completion Interrupt in Multi-Processor Transfer Mode



Caution Before writing transmit data to TXB2, confirm that TXBF = 0 and set or clear the MPS bit. If the MPS bit is set or cleared with TXBF = 1, the set data of the MPS bit may be appended to the transmit data currently in TXB2 and transferred.

(c) Reception

The interface enters the reception wait status if the multi-processor transfer mode is specified by using transfer mode specification register 2 (TRMC2) and bit 7 (POWER2) and bit 5 (RXE2) of asynchronous serial interface mode register 2 (ASIM2) are set to 1. In this status, the RxD2 pin is monitored to detect the start bit. If the start bit is detected, reception is started, and serial data is sequentially stored to the receive shift register 2 (RX2) at the specified baud rate.

If data with the multi-processor appended bit set to "1" is received (ID reception), a reception completion interrupt (INTSR) occurs after the stop bit has been detected and, at the same time, the data in RX2 is written to the receive buffer register 2 (RXB2). At this time, bit 3 (MPR2) of asynchronous serial interface register 2 (ASIS2) is set to 1. After it has been confirmed that MPR is 1, the ID of the receive data and the ID of the microprocessor are compared (for which software processing is necessary). If the two IDs match, the interface prepares for the next reception and waits for the next reception completion interrupt (INTSR2). If the IDs do not match, clear bit 1 (MPIEN2) of transfer mode specification register 2 (TRMC2) to 0. This makes receive data other than ID invalid and prevents occurrence of an unwanted reception completion interrupt (INTSR2).

Figure 15-19. Timing of Reception Completion Interrupt Request in Multi-Processor Transfer Mode (1/2)

(1) If receive data matches ID

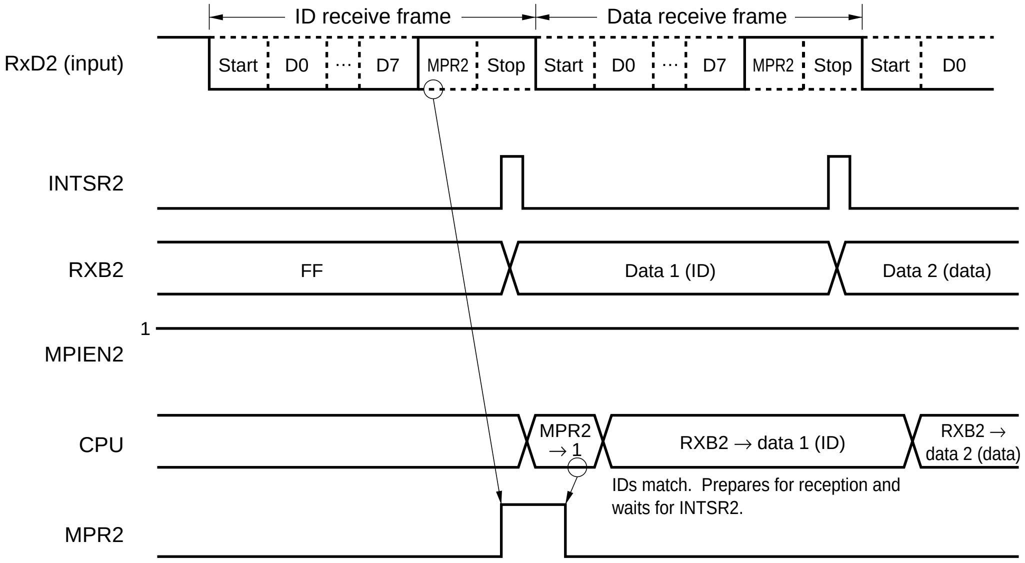
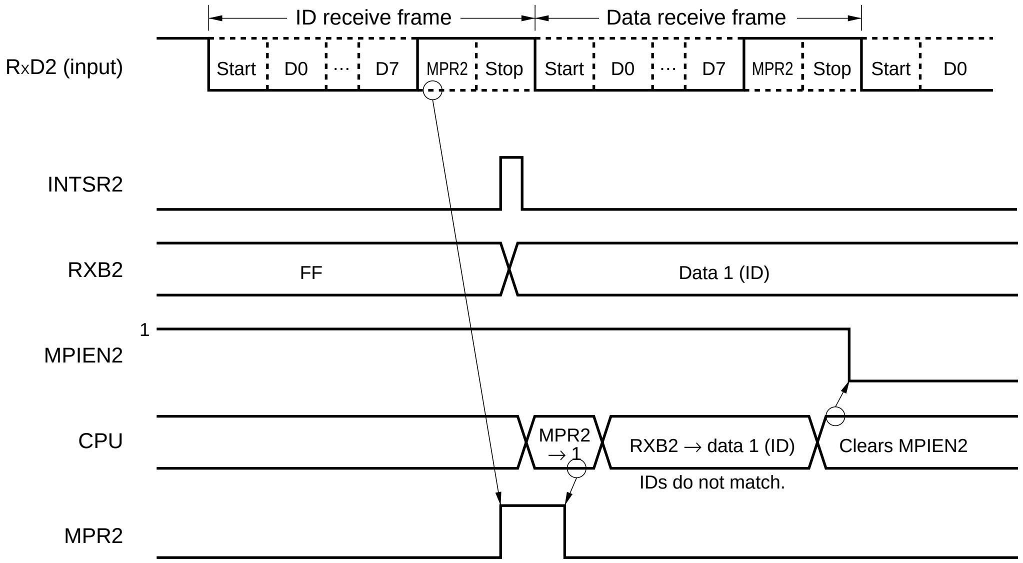


Figure 15-19. Timing of Reception Completion Interrupt Request in Multi-Processor Transfer Mode (2/2)

(2) If receive data does not match ID



15.4.4 Infrared data transfer (IrDA) mode

In this mode, pulses can be output, transmitted, or received in the data format of the IrDA specifications. This mode can be used to transmit or receive data to or from a digital device such as a personal computer.

(1) Register setting

The infrared data transfer (IrDA) mode is set by using asynchronous serial interface mode register 2 (ASIM2), clock select register 2 (CKSEL2), and transfer mode specification register 2 (TRMC2).

(a) Asynchronous serial interface mode register 2 (ASIM2)

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

$\overline{\text{RESET}}$ input sets the value of ASIM2 to 00H.

Caution In the IrDA transfer mode, set the port mode register (PMXX) as follows. Set the output latch of the port set to output mode (PMXX = 0).

- **Reception**
  Setting P35 (RxD2) in the input mode (PM35 = 1)
- **Transmission**
  Setting P34 (TxD2) in the output mode (PM34 = 0)
- **Transmission/reception**
  Setting P35 in the input mode and P34 in the output mode

Address: FF90H At Reset: 00H R/W

Symbol	$\boxed{7}$	$\boxed{6}$	$\boxed{5}$	4	3	2	1	0
ASIM2	POWER2	TXE2	RXE2	PS21	PS20	CL2	SL2	ISRM2

POWER2	Enables/Stops Clock Operation
0	Stops clock operation. Power consumption decreases and latch in UART2 is asynchronously reset (TxD2 pin is low).
1	Enables clock operation (TxD2 pin is high).

TXE2[Note 1]	Stops/Enables Transmission
0	Stops transmission (transmission circuit is synchronously reset).
1	Enables transmission.

RXE2[Note 2]	Stops/Enables Reception
0	Stops reception (reception circuit is synchronously reset).
1	Enables reception.

Notes 1. To transmit data with UART2, first specify the clock operation (set POWER2 to 1 and then TXE2 to 1), wait for the duration of 2 clocks[Note 3], and then write the transmit data to transmit buffer register 2 (TXB2). To stop transmission by UART2, specify stopping transmission (TXE2 = 0), wait for the duration of 2 clocks[Note 3], and then stop the clock operation (POWER2 = 0).

2. To receive data with UART2, first specify the clock operation (set POWER2 to 1 and then RXE2 to 1), wait for the duration of 2 clocks[Note 3], and then start reception. To stop reception by UART2, specify stopping reception (RXT2 = 0), wait for the duration of 2 clocks[Note 3], and then stop the clock operation (POWER2 = 0).

3. The clock is the output clock of the 8-bit counter or the input clock of the baud rate generator.

PS21[Note 1]	PS20[Note 1]	Parity Bit Specification	
		Transmission	Reception
0	0	Does not output parity bit.	Reception without parity
0	1	Outputs 0 parity.	Reception as 0 parity[Note 2]
1	0	Outputs odd parity.	Identifies odd parity.
1	1	Outputs even parity.	Identifies even parity.

CL2[Note 3]	Data Character Length Specification
0	7 bits
1	8 bits

SL2[Note 4]	Specification for Number of Stop Bits for Transmission
0	1 bit
1	2 bits

ISRM2[Note 5]	Reception Error Interrupt Signal Control
0	INTSR2 is generated.
1	INTSER2 is generated.

Notes 1. To specify a parity bit, stop transmission and reception (TXE2 = 0 and RXE2 = 0) before rewriting PS1 and PS0.

2. The parity is not identified with this setting. Therefore, bit 2 (PE2) of asynchronous serial interface status register 2 (ASIS2) is not set and the error interrupt does not occur.

3. To specify a data character length, stop transmission and reception (TXE2 = 0 and RXE2 = 0) before rewriting CL2.

4. To specify the number of stop bits, stop transmission (TXE2 = 0) before rewriting SL2. Reception is always performed on the assumption that the number of stop bits is 1.

5. To specify an interrupt that occurs in case of an error, stop transmission (TXE2 = 0) before rewriting ISEM2.

(b) Clock selct register 2 (CKSEL2)

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

$\overline{\text{RESET}}$ input sets the value of CKSEL2 to 00H.

Address: FF92H At Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CKSEL2	0	TPS22$^{\text{Note}}$	TPS21$^{\text{Note}}$	TPS20$^{\text{Note}}$	TPW23	TPW22	TPW21	TPW20

TPS22	TPS21	TPS20	Source Clock of 8-bit Counter	n
0	0	0	ASCK2/$\overline{\text{SCK3}}$/P36	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

TPW23	TPW22	TPW21	TPW20	Selects IrDA Transmit Pulse Width of 1-bit Data
0	0	1	0	Width of two f_{SCK} clocks
0	0	1	1	Width of three f_{SCK} clocks
0	1	0	0	Width of four f_{SCK} clocks
0	1	0	1	Width of five f_{SCK} clocks
0	1	1	0	Width of six f_{SCK} clocks
0	1	1	1	Width of seven f_{SCK} clocks
1	0	0	0	Width of eight f_{SCK} clocks
1	0	0	1	Width of nine f_{SCK} clocks
1	0	1	0	Width of ten f_{SCK} clocks
1	0	1	1	Width of 11 f_{SCK} clocks
1	1	0	0	Width of 12 f_{SCK} clocks
1	1	0	1	Width of 13 f_{SCK} clocks
1	1	1	0	Width of 14 f_{SCK} clocks
1	1	1	1	Width of 15 f_{SCK} clocks
0	0	0	0	Width of 16 f_{SCK} clocks
Others				Setting prohibited

Note To rewrite TPS0 through TPS2, clear bit 7 (POWER2) of the asynchronous serial interface mode register 2 (ASIM2) to 0.

Cautions 1. If data is written to CKSEL2 during a communication operation, the output of the baud rate generator is disturbed and the communication cannot be performed correctly. Therefore, do not rewrite CKSEL2 during communication.

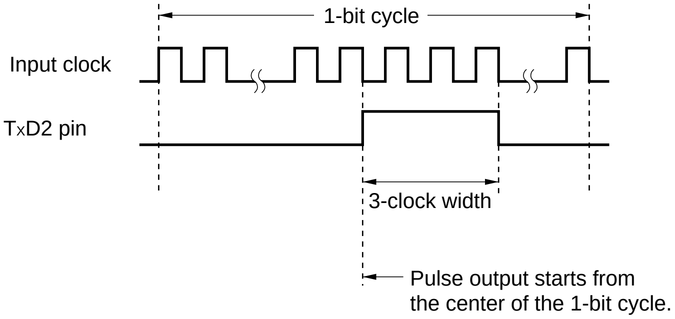
2. To transfer data in the infrared data transfer (IrDA) mode, the following conditions must be satisfied when a transmit pulse width is specified:

(Condition)

1.41 μs ≤ Transmit pulse width < Transfer rate

(Set values of bits 0 through 3 (TPW0 through TPW3) of CKSEL2) (Set values of bits 0 through 7 (MDL20 through MDL27) of BRGC2)

Example If the transmit pulse width is set to the width of three f_{SCK} clocks (TPW3 through TPW0 = 0, 0, 1, 1)



Remarks f_X : Main system clock oscillation frequency

f_{SCK} : Source clock of 8-bit counter

(c) Transfer mode specification register 2 (TRMC2)

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

$\overline{\text{RESET}}$ input sets the value of TRMC2 to 02H.

Address: FF91H At Reset: 02H R/W

Symbol	7	6	5	4	3	2	$\boxed{1}$	$\boxed{0}$
TRMC2	TRM12[Note 1]	TRM02[Note 1]	0	0	0	0	MPIEN2	MPS2

TRM12	TRM02	Transfer Mode
0	0	UART transfer mode[Note 2]
0	1	Multi-processor transfer mode[Note 2]
1	0	Infrared data transfer (IrDA) mode[Note 2]
1	1	

Notes 1. Before rewriting TRM12 and TRM02, clear bits 6 (TXE2) and 5 (RXE2) of asynchronous serial interface mode register 2 (ASIM2) to 0.

2. The setting of bits 0 through 4 (ISEM2, SL2, CL2, PS20, and PS21) of ASIM2 is valid in all the transfer modes.

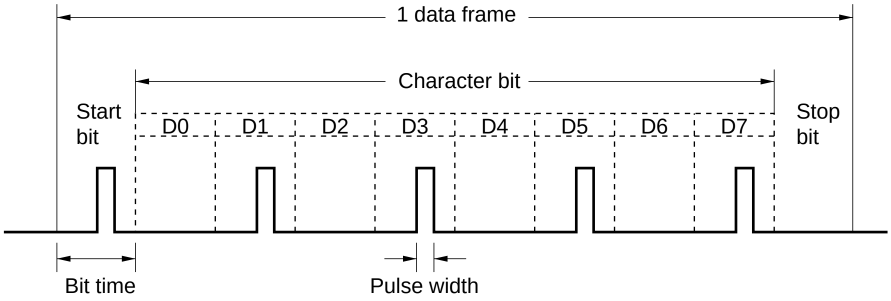
(2) Communication operation

(a) Data format

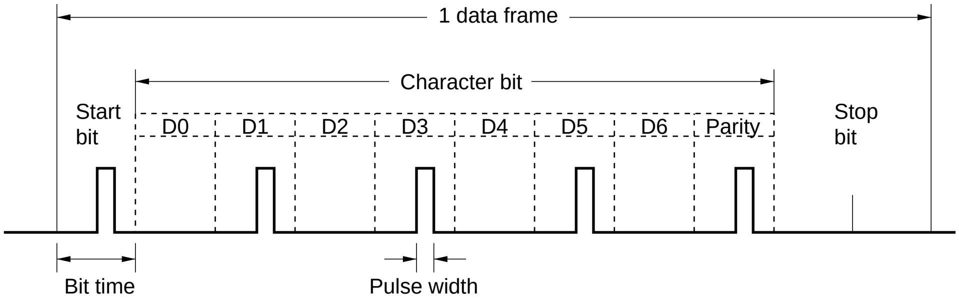
Figure 15-20 shows the format of transmit/receive data.

Figure 15-20. Data Format of Transmission/Reception in Infrared Data Transfer (IrDA) Mode

(1) IrDA standard format
(Character bit: 8 bits, Parity bit: None, Stop bit: 1 bit, Transfer data: 55H)



(2) Other format
(Character bit: 7 bits, Parity bit: Even parity, Stop bit: 2 bits, Transfer data: 55H)



One data frame consists of the following bits:

- Start bit .... 1 bit
- Character bit ...... 7/8 bits (LSB first)
- Parity bit............ Odd/even/0/None
- Stop bit .... 1/2 bits

The character bit length, parity, and stop bit length in one data frame are specified by using asynchronous serial interface mode register 2 (ASIM2). Data is transferred starting from the LSB.

The length of the electric pulse transmitted or received in one data frame can be specified by using bits 0 through 3 (TPW20 through TPW23) of clock select register 2 (CKSEL2). Usually, the pulse length is 1.41 μs (rated minimum pulse width) to lower the power consumption. The pulse bit rises at the center of a bit cycle.

(b) Transmission

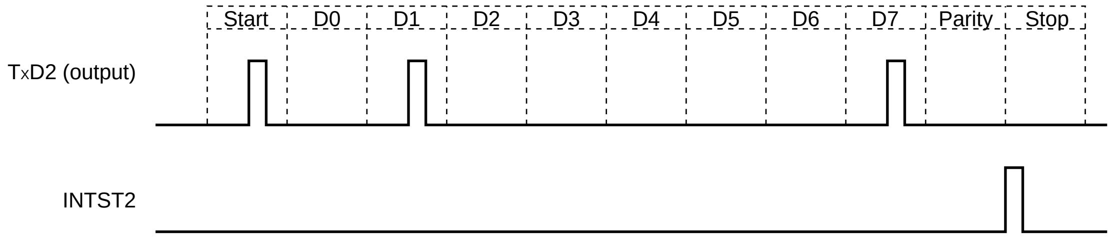
If the infrared data transfer (IrDA) mode is set by using transfer mode specification register 2 (TRMC2) and bit 7 (POWER2) of asynchronous serial interface mode register 2 (ASIM2) is set to 1, the TxD2 pin outputs a low level. If bit 6 (TXE2) of ASIM2 is set to 1 next, transmission is enabled. Transmission can be started by writing transmit data to the transmit buffer register 2 (TXB2). The start bit, parity bit, and stop bit are automatically appended to the data.

When transmission is started, the data in TXB2 is transferred to the transmit shift register 2 (TXS2) and sequentially output to the TxD2 pin, starting from the LSB. If the data to be transmitted next has been written to TXB2 when transmission has been completed, transmitting the next data is started. If no more data has been written to TXB2, transmission is stopped, until new transmit data is written.

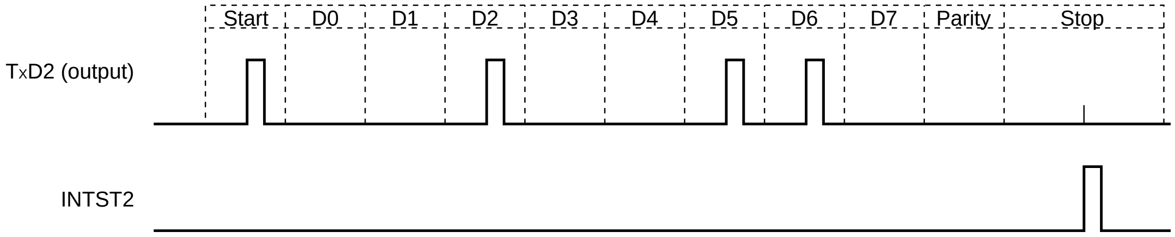
Figure 15-21 shows the timing of a transmission interrupt.

Figure 15-21. Timing of Transmission Completion Interrupt in Infrared Data (IrDA) Transfer Mode

(1) Character bit: 8 bits, Parity bit: Odd, Stop bit: 1 bit, Transfer data: 7DH



(2) Character bit: 8 bits, Parity bit: Even, Stop bit: 2 bits, Transfer data: 9BH

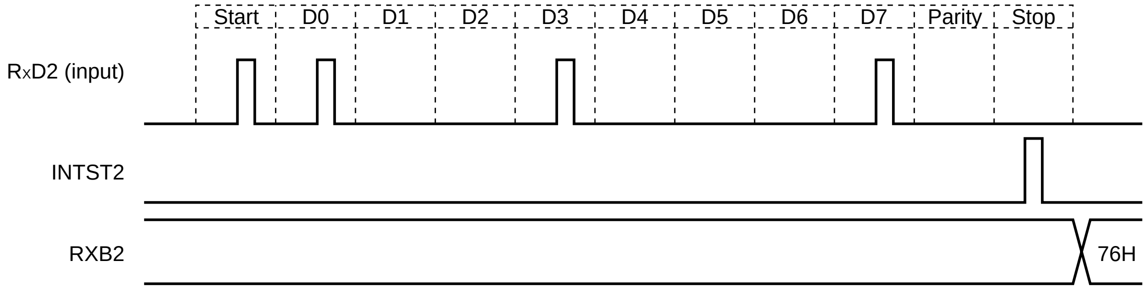


(c) Reception

The interface enters the reception wait status if the infrared data (IrDA) transfer mode is specified by using transfer mode specification register 2 (TRMC2) and bit 7 (POWER2) and bit 5 (RXE2) of asynchronous serial interface mode register 2 (ASIM2) are set to 1. In this status, the RxD2 pin is monitored to detect the start bit. If the start bit is detected, reception is started, and serial data is sequentially stored in the receive shift register 2 (RX2) at the specified baud rate.

When the stop bit is received, the data in RX2 is written to the receive buffer register 2 (RXB2). If an overrun error (OVE2) occurs, however, the receive data is not written to RXB2 and discarded. Even if a parity error (PE2) or framing error (FE2) occurs during reception, reception continues up to the position at which the stop bit is received, and an error interrupt (INTSR2/INTSER2) occurs after completion of reception.

Figure 15-22. Timing of Reception Completion Interrupt Request in Infrared Data (IrDA) Transfer Mode



Cautions 1. Be sure to read receive buffer register 2 (RXB2) even if a reception error has occurred. Otherwise, an overrun error will occur when the next data is received, and the reception error status will persist.

2. The number of stop bits is always 1 during reception. The second stop bit is discarded.

(d) Bit rate and pulse width

Table 15-10 shows the bit rate and pulse width in the infrared data transfer (IrDA) mode. The rated minimum pulse width is 1.41 μs, and the maximum pulse width is the sum of 3/16 of the bit rate and 2.5% of the bit cycle or 1.08 μs whichever greater.

Table 15-10. Bit Rate and Pulse Width

Bit Rate (bps)	Allowable Bit Rate Error (% of bit rate)	Minimum Pulse Width (μs)	Nominal Value of 3/16 of Pulse Width (μs)	Maximum Pulse Width (μs)
2400	+/–0.87	1.41	78.13	88.55
9600			19.53	22.13
19200			9.77	11.07
38400			4.88	5.96
57600			3.26	4.34
115200			1.63	2.71

The serial interface (UART2/SIO3) can be used in the asynchronous serial interface (UART) mode or 3-wire serial I/O mode.

Caution Do not enable UART2 and SIO3 at the same time.

16.1 Serial Interface (SIO3) Functions

The 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 **16.4.1 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 ($\overline{\text{SCK3}}$), serial output line (SO3), and serial input line (SI3).

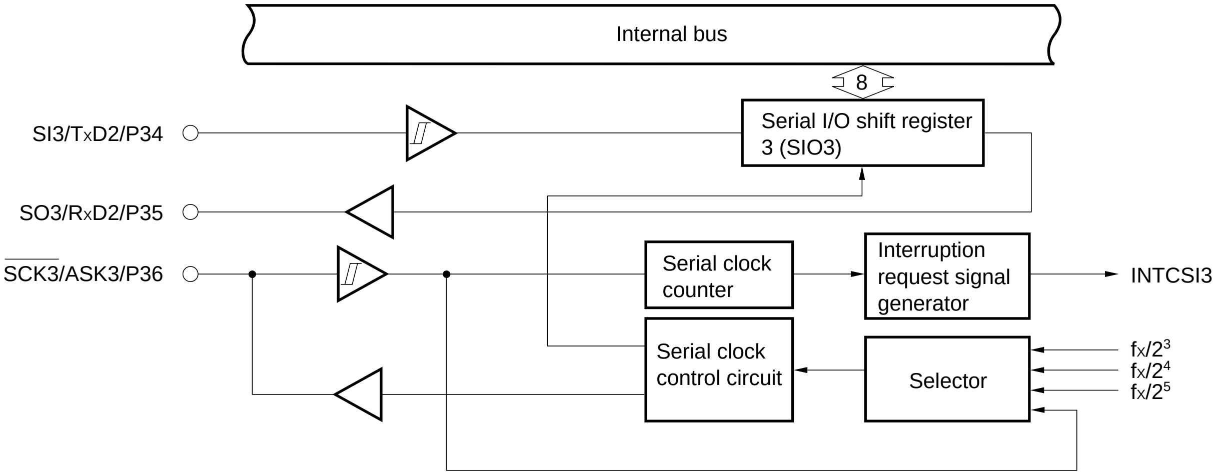
Since simultaneous transmit and receive operations are enabled in 3-wire serial I/O mode, the processing time for data transfers is reduced.

The first bit of the serial transferred 8-bit data is fixed as the MSB.

3-wire serial I/O mode is useful for connection to a peripheral I/O incorporating a clock-synchronous serial interface, or a display controller, etc. For details see **16.4.2 3-wire serial I/O mode**.

Figure 16-1 shows a block diagram of the serial interface (SIO3).

Figure 16-1. Serial Interface (SIO3) Block Diagram



16.2 Serial Interface (SIO3) Configuration

The serial interface (SIO3) includes the following hardware.

Table 16-1. Serial Interface (SIO3) Configuration

Item	Configuration
Register	Serial I/O shift register 3 (SIO3)
Control register	Serial operation mode register 3 (CSIM3)

(1) Serial I/O shift register 3 (SIO3)

This is an 8-bit register that performs parallel-serial conversion and serial transmit/receive (shift operations) synchronized with the serial clock.

SIO3 is set by an 8-bit memory manipulation instruction.

When “1” is set to bit 7 (CSIE3) of the serial operation mode register 3 (CSIM3), a serial operation can be started by writing data to or reading data from SIO3.

When transmitting, data written to SIO3 is output to the serial output (SO3).

When receiving, data is read from the serial input (SI3) and written to SIO3.

$\overline{\text{RESET}}$ input makes SIO3 undefined.

Caution Do not access SIO3 during a transmit operation unless the access is triggered by a transfer start. (Read operation is disabled when MODE3 = 0 and write operation is disabled when MODE3 = 1.)

16.3 Register to Control Serial Interface (SIO3)

The serial interface (SIO3) is controlled by serial operation mode register 3 (CSIM3).

(1) Serial operation mode register 3 (CSIM3)

This register is used to enable or disable SIO3's serial clock, operation modes, and specific operations.
CSIM3 is set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets the value of CSIM3 to 00H.

Caution **In 3-wire serial I/O mode, set the port mode register (PMXX) as follows. Set the output latch of the port set to output mode (PMXX = 0) to 0.**

<When SIO3 is used>

During serial clock output (master transmission or master reception)	PM36 = 0: Sets P36 ($\overline{\text{SCK3}}$) to output mode P36 = 0: Sets output latch of P36 to 0
During serial clock input (slave transmission or slave reception)	PM36 = 1: Sets P36 ($\overline{\text{SCK3}}$) to input mode
Transmit/receive mode	PM35 = 0: Sets P35 (SO3) to output mode P35 = 0: Sets output latch of P35 to 0
Receive mode	PM34 = 1: Sets P34 (SI3) to input mode

Figure 16-2. Serial Operation Mode Register 3 (CSIM3) Format

Address: FFB8H At Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIM3	CSIE3	0	0	0	0	MODE3	SCL31	SCL30

CSIE3	Enable/Disable Specification for SIO3		
	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]

MODE3	Transfer Operation Modes and Flags		
	Operation mode	Transfer start trigger	SO3 output
0	Transmit/transmit and receive mode	Write to SIO3	Normal output
1	Receive-only mode	Read from SIO3	Fixed at low level

SCL31	SCL30	Clock Selection
0	0	External clock input to $\overline{\text{SCK3}}$
0	1	$f_X/2^3$ (1.05 MHz)
1	0	$f_X/2^4$ (524 kHz)
1	1	$f_X/2^5$ (262 kHz)

Notes 1. When CSIE3 = 0 (SIO3 operation stop status), the pins SI3, SO3, and $\overline{\text{SCK3}}$ can be used for port functions.

2. When CSIE3 = 1 (SIO3 operation enabled state), the SI3 pin can be used as a port pin if only the send function is used, and the SO3 pin can be used as a port pin if only the receive-only mode is used.

Remarks 1. f_X: main system clock oscillation frequency

2. Figures in parentheses are for operation with f_X = 8.38 MHz.

16.4 Serial Interface (SIO3) Operations

This section explains on two modes of serial interface (SIO3).

16.4.1 Operation stop mode

Because the serial transfer is not performed during this mode, the power consumption can be reduced.
In addition, pins can be used as normal I/O ports.

(1) Register settings

Operation stop mode are set by the serial operation mode register 3 (CSIM3).
CSIM3 is set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets the value of CSIM3 to 00H.

Address: FFB8H At Reset: 00H R/W

Symbol	$\boxed{7}$	6	5	4	3	2	1	0
CSIM3	CSIE3	0	0	0	0	MODE3	SCL31	SCL30

CSIE3	SIO3 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 CSIE3 = 0 (SIO3 operation stop status), the pins SI3, SO3, and $\overline{\text{SCK3}}$ can be used for port functions.

2. When CSIE3 = 1 (SIO3 operation enabled state), the SI3 pin can be used as a port pin if only the send function is used, and the SO3 pin can be used as a port pin if only the receive-only mode is used.

16.4.2 3-wire serial I/O mode

The 3-wire serial I/O mode is useful for connection to a peripheral I/O incorporating a clock-synchronous serial interface, a display controller, etc.

This mode executes data transfers via three lines: a serial clock line ($\overline{\text{SCK3}}$), serial output line (SO3), and serial input line (SI3).

(1) Register settings

3-wire serial I/O mode is set by the serial operation mode register 3 (CSIM3).

CSIM3 is set by a 1-bit or 8-bit memory manipulation instructions.

$\overline{\text{RESET}}$ input sets the value of CSIM32 to 00H .

Caution In 3-wire serial I/O mode, set the port mode register (PMXX) as follows. Set the output latch of the port set to output mode (PMXX = 0) to 0.

During serial clock output (master transmission or master reception)	PM36 = 0: Sets P36 ($\overline{\text{SCK3}}$) to output mode P36 = 0: Sets output latch of P36 to 0
During serial clock input (slave transmission or slave reception)	PM36 = 1: Sets P36 ($\overline{\text{SCK3}}$) to input mode
Transmit/receive mode	PM35 = 0: Sets P35 (SO3) to output mode P35 = 0: Sets output latch of P35 to 0
Receive mode	PM34 = 1: Sets P34 (SI3) to input mode

Address: FFB8H At Reset: 00H R/W

Symbol	$\boxed{7}$	6	5	4	3	2	1	0
CSIM3	CSIE3	0	0	0	0	MODE3	SCL31	SCL30

CSIE3	Enable/Disable Specification for SIO3		
	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]

MODE3	Transfer Operation Modes and Flags		
	Operation mode	Transfer start trigger	SO3 output
0	Transmit/transmit and receive mode	Write to SIO3	Normal output
1	Receive-only mode	Read from SIO3	Fixed at low level

SCL31	SCL30	Clock Selection
0	0	External clock input to $\overline{SCK3}$
0	1	$f_X/2^3$ (1.05 MHz)
1	0	$f_X/2^4$ (524 kHz)
1	1	$f_X/2^5$ (262 kHz)

Notes 1. When CSIE3 = 0 (SIO3 operation stop status), the pins SI3, SO3, and $\overline{SCK3}$ can be used for port functions.

2. When CSIE3 = 1 (SIO3 operation enabled state), the SI3 pin can be used as a port pin if only the send function is used, and the SO3 pin can be used as a port pin if only the receive-only mode is used.

Remarks 1. f_X: Main system clock oscillation frequency

2. Figures in parentheses are for operation with f_X = 8.38 MHz.

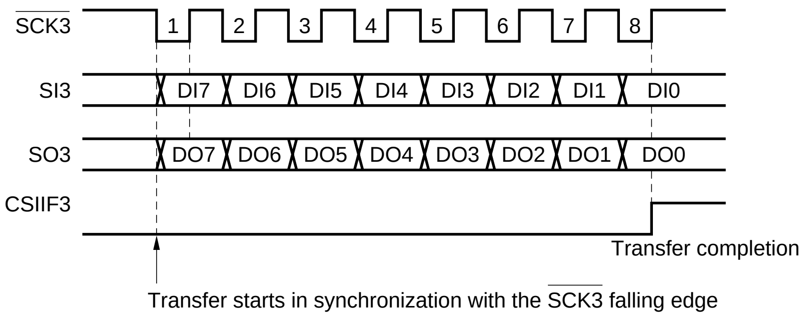
(2) Communication Operations

In the three-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.

The serial I/O shift register 3 (SIO3) is shifted in synchronization with the falling edge of the serial clock. Transmission data is held in the SO3 latch and is output from the SO3 pin. Data that is received via the SI3 pin in synchronization with the rising edge of the serial clock is latched to SIO3.

Completion of an 8-bit transfer automatically stops operation of SIO3 and sets interrupt request flag (CSIIF3).

Figure 16-3. Timing of 3-Wire Serial I/O Mode



(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 3 (SIO3).

- The SIO3 operation control bit (CSIE3) = 1
- After an 8-bit serial transfer, either the internal serial clock is stopped or $\overline{\text{SCK3}}$ is set to high level.
- Transmit/transmit and receive mode
  When CSIE3 = 1 and MODE3 = 0, transfer starts when writing to SIO3.
- Receive-only mode
  When CSIE3 = 1 and MODE3 = 1, transfer starts when reading from SIO3.

Caution After data has been written to SIO3, transfer will not start even if the CSIE3 bit value is set to “1”.

Completion of an 8-bit transfer automatically stops the serial transfer operation and interrupt request flag (CSIIF3) is set.

CHAPTER 17 SERIAL INTERFACE (CSI1)

17.1 Serial Interface (CSI1) Functions

The serial interface (CSI1) has the following two modes:

- Operation stop mode
- 3-wire serial I/O mode

(1) Operation stop mode

This mode is used when serial transfer is not performed. In this mode, the power consumption can be reduced.

(2) 3-wire serial I/O mode (MSB/LSB first selectable)

This mode is used to transfer 8-bit data by using three lines: a serial clock line (SCK1) and two serial data lines (SI1 and SO1).

The processing time of data transfer can be shortened in the 3-wire serial I/O mode because transmission and reception can be simultaneously executed in this mode. In addition, whether 8-bit data is transferred with the MSB or LSB first can be specified, so this interface can be connected to any device.

The 3-wire serial I/O mode is useful for connecting peripheral I/Os and display controllers having a conventional clocked serial interface, such as the 75XL series, 78K series, and 17K series.

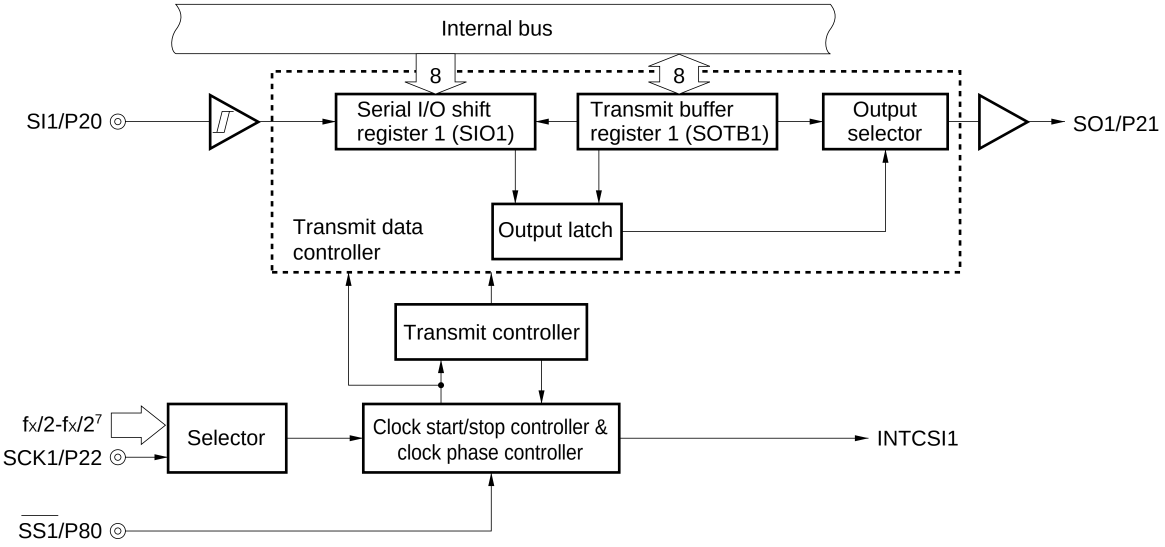
17.2 Serial Interface (CSI1) Configuration

The serial interface (CSI1) is organized with the following hardware:

Table 17-1. Serial Interface (CSI1) Configuration

Item	Organization
Registers	Transmit buffer register 1 (SOTB1) Serial I/O shift register 1 (SIO1)
Control registers	Serial operation mode register 1 (CSIM1) Serial clock select register 1 (CSIC1)

Figure 17-1. Serial Interface (CSI1) Block Diagram



(1) Transmit buffer register 1 (SOTB1)

This register sets transmit data.

Transmission is started by writing data to SOTB1 when bit 6 (TRMD1) of serial operation mode register 1 (CSIM1) is 1.

The data written to SOTB1 is converted from parallel data into serial data by serial I/O shift register 1, and output to the serial output (SO1) pin.

SOTB1 can be written or read by using an 8-bit memory manipulation instruction.

The value of this register is undefined when $\overline{\text{RESET}}$ is input.

Caution Do not access SOTB1 when CSOT1 = 1 (during serial communication).

Remark The operation is as follows if bit 5 (SSE1) of serial operation mode register 1 (CSIM1) is 1 in the slave mode:

(1) If $\overline{\text{SS1}}$ is low
··· This chip is selected and transmission is started by writing data to SOTB1.

(2) If $\overline{\text{SS1}}$ is high
··· Transmission is not started even if data is written to SOTB1 because this chip is not selected (transmission pending).

(3) If data is written to SOTB1 when transmission is kept pending because $\overline{\text{SS1}}$ is high and then $\overline{\text{SS1}}$ goes low
··· Transmission is started.

(4) If $\overline{\text{SS1}}$ goes high after transmission has been started by writing data to SOTB1 when $\overline{\text{SS1}}$ is low
... Transmission is aborted.

(2) Serial I/O shift register 1 (SIO1)

This is an 8-bit register that converts data from parallel into serial or vice versa.

This register can be read by using an 8-bit memory manipulation instruction.

Reception is started by reading data from SIO1 if bit 6 (TRMD1) of serial operation mode register 1 (CSIM1) is 0 or 1.

During reception, the data is read from the serial input pin (SI1) to SIO1.

The value of this register is undefined when $\overline{\text{RESET}}$ is input.

Caution Do not access SIO1 when CSOT1 = 1 (during serial communication).

Remark The operation is as follows if bit 5 (SSE1) of serial operation mode register 1 (CSIM1) is 1 in the slave mode:

(1) If $\overline{\text{SS1}}$ is low
··· This chip is selected and reception is started by reading data from SIO1.

(2) If $\overline{\text{SS1}}$ is high
··· Reception is not started even if data is read from SIO1 because this chip is not selected (reception pending).

(3) If data is read from SIO1 when reception is kept pending because $\overline{\text{SS1}}$ is high and then $\overline{\text{SS1}}$ goes low
··· Reception is started.

(4) If $\overline{\text{SS1}}$ goes high after reception has been started by reading data from SIO1 when $\overline{\text{SS1}}$ is low
··· Reception is aborted.

17.3 Registers to Control Serial Interface (CSI1)

The serial interface (CSI1) is controlled by the following two registers:

- Serial operation mode register 1 (CSIM1)
- Serial clock select register 1 (CSIC1)

(1) Serial operation mode register 1 (CSIM1)

This register is used to select an operation mode and enable or disable the operation.

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

$\overline{\text{RESET}}$ input sets the value of CSIM1 to 00H.

Figure 17-2. Serial Operation Mode Register 1 (CSIM1) Format

Address: FFB0H At Reset: 00H R/W[Note 1]

Symbol	$\boxed{7}$	6	5	4	3	2	1	0
CSIM1	CSIE1	TRMD1	SSE1	DIR1	0	0	0	CSOT1

CSIE1	Controls Operation in 3-wire Serial I/O Mode.
0	Stops operation (SI1/P20, SO1/P21, and SCK1/P22 pins can be used as general-purpose port pins).
1	Enables operation (SI1/P20, SO1/P21, and SCK1/P22 pins are at active level).

TRMD1[Note 2]	Selects Transmit/Receive Mode.
0[Note 3]	Reception mode (transmission disabled).
1	Transmit/receive mode

SSE1[Notes 4, 5]	Specifies Whether $\overline{\text{SS1}}$ Pin Is Used.
0	Does not use $\overline{\text{SS1}}$ pin.
1	Uses $\overline{\text{SS1}}$ pin.

DIR1[Note 4]	Specifies First Bit.
0	MSB
1	LSB

CSOT1[Note 6]	Operation Mode Flag
0	Communication is stopped.
1	Communication is in progress.

Notes

1. Bit 0 is a read-only bit.
2. Do not rewrite TRMD1 when CSOT1 = 1 (during serial communication).
3. The SO1 pin is fixed to the low level when TRMD1 is 0. Reception is started when data is read from SIO1.
4. Do not overwrite these bits when CSOT1 = 1 (during serial communication).
5. Before setting this bit to 1, fix the input level of the $\overline{\text{SS1}}$ pin to 0 or 1.
6. CSOT1 is cleared if CSIE1 is cleared to 0 (operation stops).

(2) Serial clock select register 1 (CSIC1)

This register is used to select the phase of the data clock and a count clock.

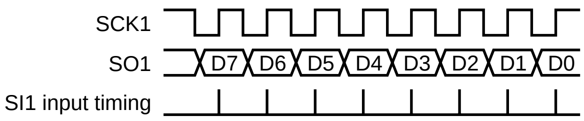
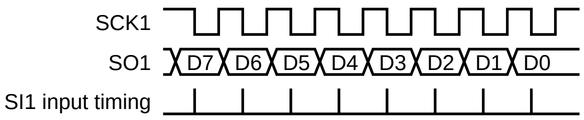
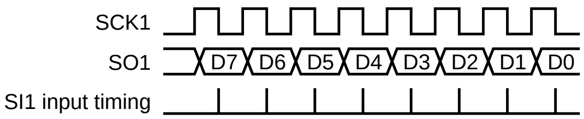
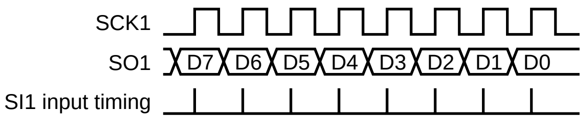
This register is set by using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the value of CSIC1 to 10H.

★ **Figure 17-3. Serial Clock Select Register 1 (CSIC1) Format**

Address: FFB1H At Reset: 10H R/W

Symbol	7	6	5	4	3	2	1	0
CSIC1	0	0	0	CKP1	DAP1	CKS12	CKS11	CKS10

CKP1	DAP1	Data Clock Phase Selection	Type
0	0	 SCK1 SO1 D7 D6 D5 D4 D3 D2 D1 D0 SI1 input timing	1
0	1	 SCK1 SO1 D7 D6 D5 D4 D3 D2 D1 D0 SI1 input timing	2
1	0	 SCK1 SO1 D7 D6 D5 D4 D3 D2 D1 D0 SI1 input timing	3
1	1	 SCK1 SO1 D7 D6 D5 D4 D3 D2 D1 D0 SI1 input timing	4

CKS12	CKS11	CKS10	Count Clock CSI1 Selection
0	0	0	$f_X/2$ (4.19 MHz)
0	0	1	$f_X/2^2$ (2.10 MHz)
0	1	0	$f_X/2^3$ (1.05 MHz)
0	1	1	$f_X/2^4$ (523.75 kHz)
1	0	0	$f_X/2^5$ (261.88 kHz)
1	0	1	$f_X/2^6$ (130.94 kHz)
1	1	0	$f_X/2^7$ (65.47 kHz)
1	1	1	External clock

Cautions 1. Do not write CSIC1 when CSIE1 = 0 (operation stops).

2. The phase type of the data clock is type 3 after reset.

Remark ( ): fx = 8.38 MHz

17.4 Serial Interface (CSI1) Operations

The serial interface 1 can be used in the following two modes:

- Operation stop mode
- 3-wire serial I/O mode

17.4.1 Operation stop mode

Serial transfer is not executed in this mode. Therefore, the power consumption can be reduced. In addition, the P20/SI1, P21/SO1, P22/SCK1, and P80/$\overline{\text{SS1}}$ pins can be used as ordinary I/O port pins in this mode.

(1) Register setting

The operation stop mode is set by the serial operation mode register 1 (CSIM1).

(a) Serial operation mode register 1 (CSIM1)

This register is set by using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the value of CSIM1 to 00H.

Address: FFB0H At Reset: 00H R/W

Symbol	$\boxed{7}$	6	5	4	3	2	1	0
CSIM1	CSIE1	TRMD1	SSE1	DIR1	0	0	0	CSOT1

CSIE1	Controls Operation in 3-wire Serial I/O Mode
0	Stops operation (SI1/P20, SO1/P21, and SCK1/P22 pins can be used as general-purpose port pins).
1	Enables operation (SI1/P20, SO1/P21, and SCK1/P22 pins are at active level).

17.4.2 3-wire serial I/O mode

The 3-wire serial I/O mode is useful for connecting peripheral I/Os ad display controllers having a conventional clocked serial interface, such as the 75XL series, 78K series, and 17K series.

In this mode, communication is executed by using three lines: serial clock (SCK1), serial output (SO1), and serial input (SI1) lines.

(1) Register setting

The 3-wire serial I/O mode is set by using serial operation 1 mode register 1 (CSIM1) and serial clock select register (CSIC1).

(a) Serial operation mode register 1 (CSIM1)

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

$\overline{\text{RESET}}$ input sets the value of CSIM1 to 00H.

Address: FFB0H At Reset: 00H R/W[Note 1]

Symbol	$\boxed{7}$	6	5	4	3	2	1	0
CSIM1	CSIE1	TRMD1	SSE1	DIR1	0	0	0	CSOT1

CSIE1	Controls Operation in 3-wire Serial I/O Mode
0	Stops operation (SI1/P20, SO1/P21, and SCK1/P22 pins can be used as general-purpose port pins).
1	Enables operation (SI1/P20, SO1/P21, and SCK1/P22 pins are at active level).

TRMD1[Note 2]	Selects Transmit/Receive Mode.
0[Note 3]	Reception mode (transmission disabled).
1	Transmit/receive mode

SSE1[Notes 4, 5]	Specifies Whether $\overline{\text{SS1}}$ Pin Is Used.
0	Does not use $\overline{\text{SS1}}$ pin.
1	Uses $\overline{\text{SS1}}$ pin.

DIR1[Note 4]	Specifies First Bit.
0	MSB
1	LSB

CSOT1[Note 6]	Operation Mode Flag
0	Communication is stopped.
1	Communication is in progress.

Notes

1. Bit 0 is a read-only bit.

2. Do not rewrite TRMD1 when CSOT1 = 1 (during serial communication).

3. The SO1 pin is fixed to the low level when TRMD1 is 0. Reception is started when data is read from SIO1.

4. Do not overwrite these bits when CSOT1 = 1 (during serial communication).

5. Before setting this bit to 1, fix the input level of the $\overline{\text{SS1}}$ pin to 0 or 1.

6. CSOT1 is cleared if CSIE1 is cleared to 0 (operation stops).

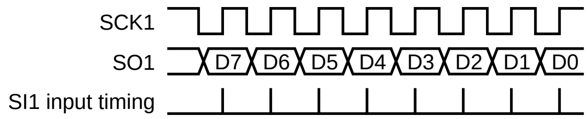
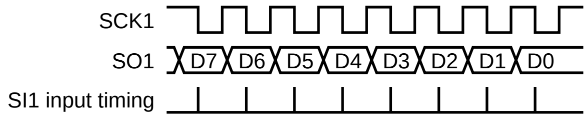
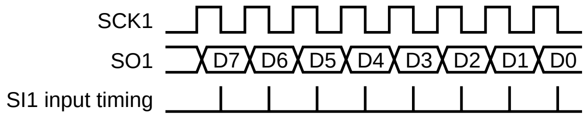
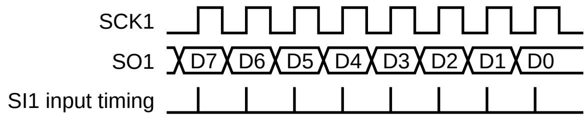
(b) Serial clock select register 1 (CSIC1)

This register is set by using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the value of CSIC1 to 10H.

Address: FFB1H At Reset: 10H R/W

Symbol	7	6	5	4	3	2	1	0
CSIC1	0	0	0	CKP1	DAP1	CKS12	CKS11	CKS10

CKP1	DAP1	Data Clock Phase Selection	Type
0	0	 SCK1 SO1 D7 D6 D5 D4 D3 D2 D1 D0 SI1 input timing	1
0	1	 SCK1 SO1 D7 D6 D5 D4 D3 D2 D1 D0 SI1 input timing	2
1	0	 SCK1 SO1 D7 D6 D5 D4 D3 D2 D1 D0 SI1 input timing	3
1	1	 SCK1 SO1 D7 D6 D5 D4 D3 D2 D1 D0 SI1 input timing	4

CKS12	CKS11	CKS10	Count Clock CSI1 Selection
0	0	0	$f_X/2$ (4.19 MHz)
0	0	1	$f_X/2^2$ (2.10 MHz)
0	1	0	$f_X/2^3$ (1.05 MHz)
0	1	1	$f_X/2^4$ (523.75 kHz)
1	0	0	$f_X/2^5$ (261.88 kHz)
1	0	1	$f_X/2^6$ (130.94 kHz)
1	1	0	$f_X/2^7$ (65.47 kHz)
1	1	1	External clock

Cautions 1. Do not write CSIC1 when CSIE1 = 0 (operation stops).

2. The phase type of the data clock is type 3 after reset.

Remark ( ): $f_X = 8.38$ MHz

(2) Setting of port

<1> Transmit/receive mode

(a) To use externally input clock as system clock (SCK1)

Bit 0 (PM20) of port mode register 2: Set to 1

Bit 1 (PM21) of port mode register 2: Cleared to 0

Bit 2 (PM22) of port mode register 2: Set to 1

Bit 1 (P21) of port 2 : Cleared to 0

(b) To use internal clock as system clock (SCK1)

Bit 0 (PM20) of port mode register 2: Set to 1

Bit 1 (PM21) of port mode register 2: Cleared to 0

Bit 2 (PM22) of port mode register 2: Cleared to 0

Bit 1 (P21) of port 2 : Cleared to 0

Bit 2 (P22) of port 2 : Cleared to 0

<2> Reception mode (with transmission disabled)

(a) To use externally input clock as system clock (SCK1)

Bit 0 (PM20) of port mode register 2: Set to 1

Bit 2 (PM22) of port mode register 2: Set to 1

(b) To use internal clock as system clock (SCK1)

Bit 0 (PM20) of port mode register 2: Set to 1

Bit 2 (PM22) of port mode register 2: Cleared to 0

Bit 5 (P25) of port 2 : Cleared to 0

Remark The transmit/receive mode or reception mode is selected by using bit (TRMD1) of serial operation mode register 1 (CSIM1).

(3) Communication operation

In the 3-wire serial I/O mode, data is transmitted or received in 8-bit units. Each bit of the data is transmitted or received in synchronization with the serial clock.

Data can be transmitted or received if bit 6 (TRMD1) of the serial operation mode register 1 (CSIM1) is 1. Transmission is started when a value is written to the transmit buffer register 1 (SOTB1). Reception is started when data is read from the serial I/O shift register 1 (SIO1).

However, the communication operation is not started if bit 5 (SSE1) of CSIM1 is 1 in the slave mode, and the $\overline{\text{SS1}}$ pin is at the high level.

After communication has been started, bit 0 (CSOT1) of CSIM1 is set to 1. When communication of 8-bit data has been completed, a communication completion interrupt request flag (CSIIF1) is set, and CSOT1 is cleared to 0. Then the next communication is enabled.

Cautions 1. Do not access the control register and data register when CSOT1 = 1 (during serial communication).

2. To change the level of the $\overline{\text{SS1}}$ pin in the slave mode, wait for the duration of at least one clock before the clock operation is started; otherwise, malfunctioning may occur.

Figure 17-4. Timing of 3-Wire Serial I/O Mode (1/2)

(1) Transmission timing (Type 1; TRMD1 = 1, DIR1 = 0, CKP1 = 0, DAP1 = 0)

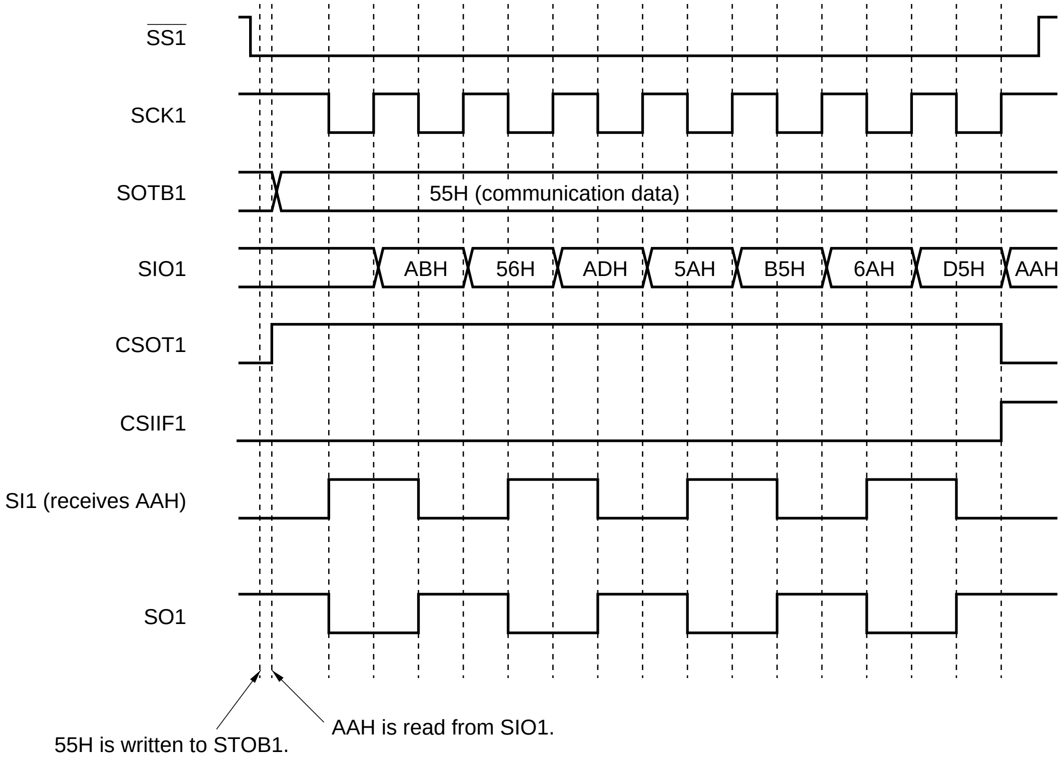


Figure 17-4. Timing of 3-Wire Serial I/O Mode (2/2)

(2) Transmission timing (Type 2; TRMD1 = 1, DIR1 = 0, CKP1 = 0, DAP1 = 1)

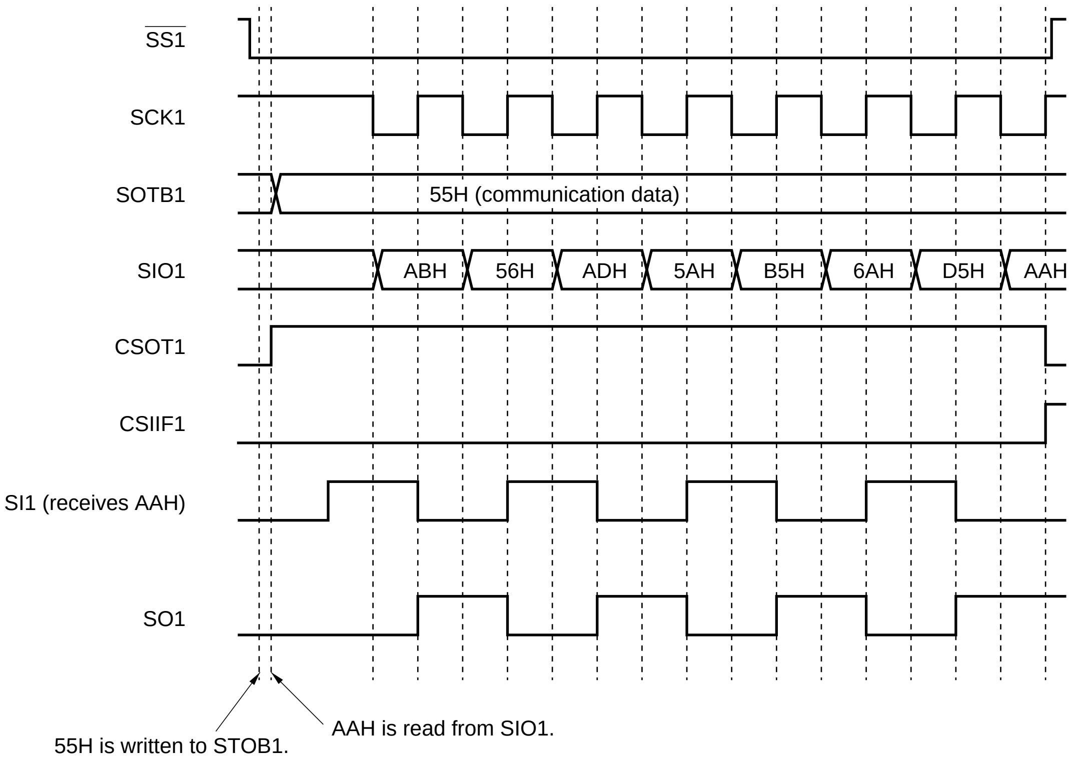
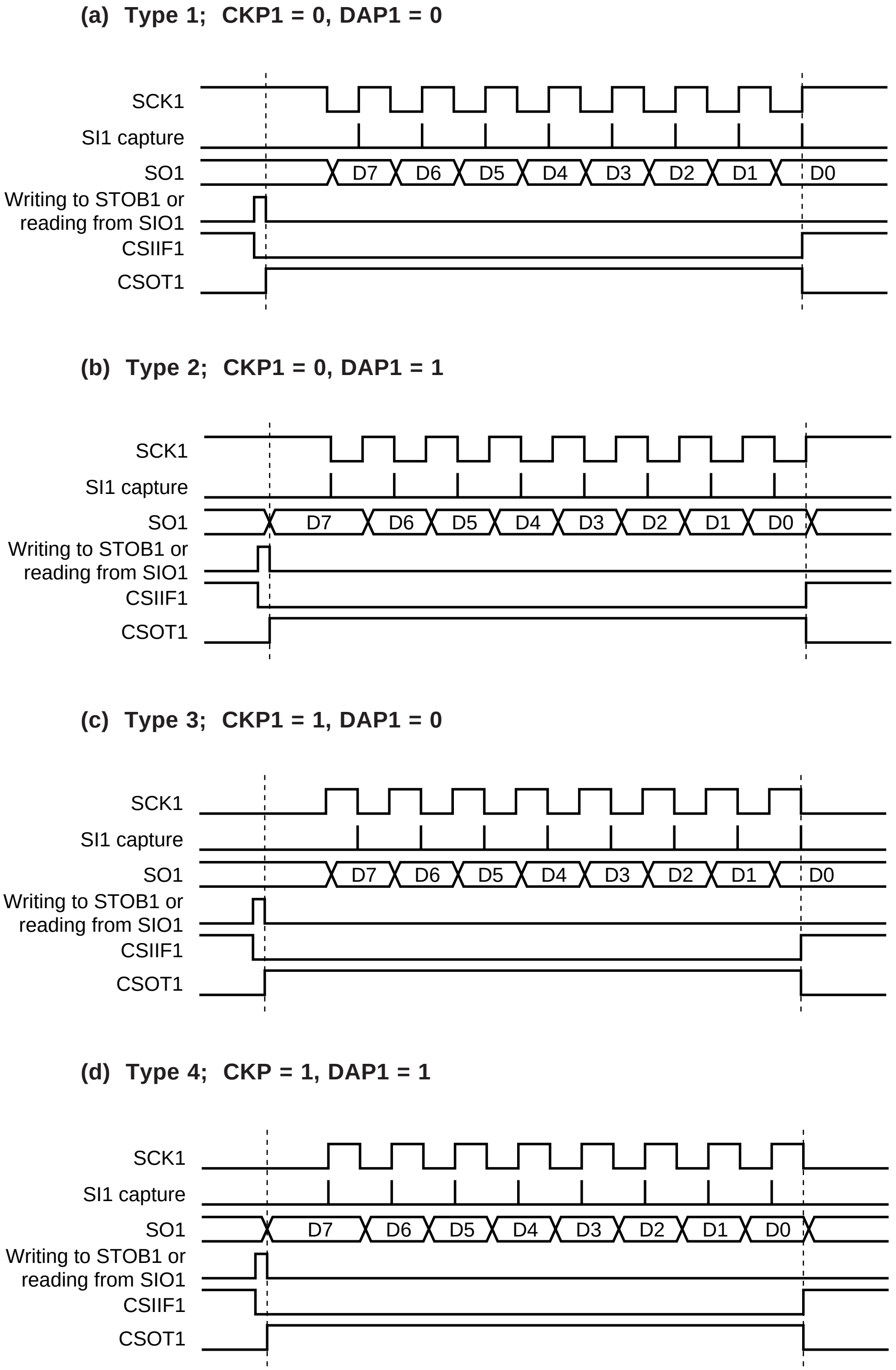


Figure 17-5. Timing of Clock/Data Phase



(a) Type 1; CKP1 = 0, DAP1 = 0

(b) Type 2; CKP1 = 0, DAP1 = 1

(c) Type 3; CKP1 = 1, DAP1 = 0

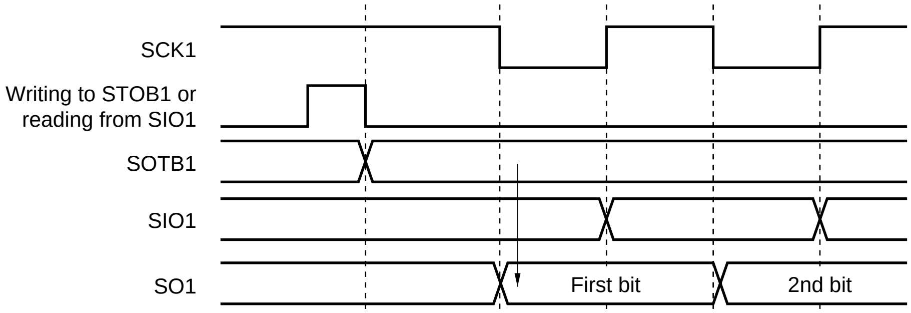
(d) Type 4; CKP = 1, DAP1 = 1

(4) Timing of output to SO1 pin (first bit)

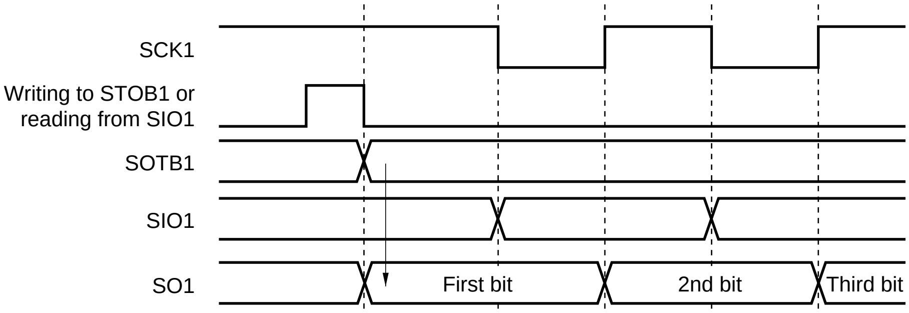
When communication is started, the value of transmit buffer register 1 (SOTB1) is output to the SO1 pin. At this time, the output timing of the first bit must be determined.

Figure 17-6. Timing of Output to SO1 Pin (first bit) (1/2)

(1) Type 1; when CKP1 = 0 and DAP1 = 0



(2) Type 2; when CKP1 = 0 and DAP1 = 1



(3) Type 3; when CKP1 = 1 and DAP1 = 0

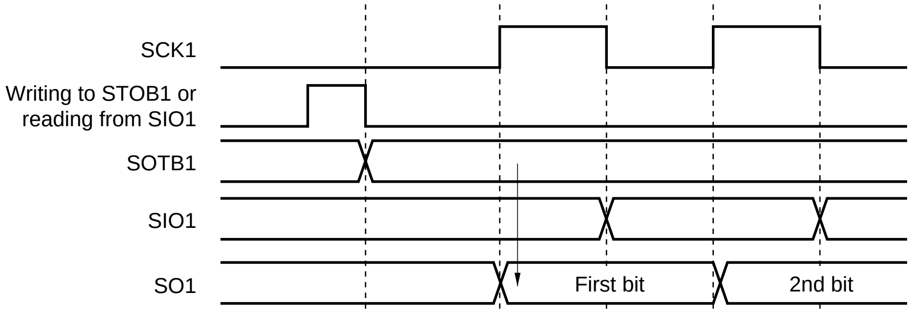
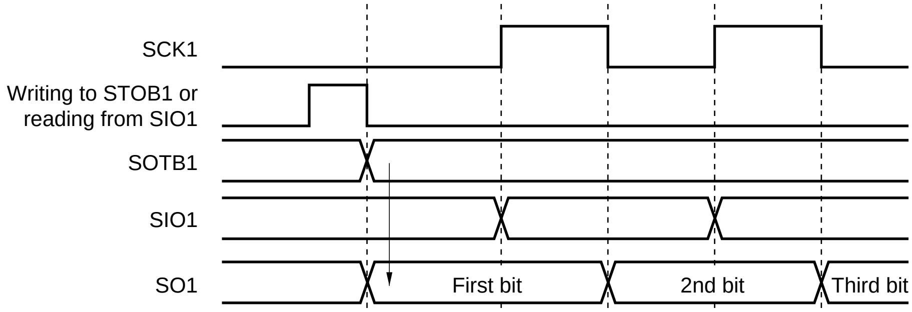


Figure 17-6. Timing of Output to SO1 Pin (first bit) (2/2)

(4) Type 4; when CKP1 = 1 and DAP1 = 1

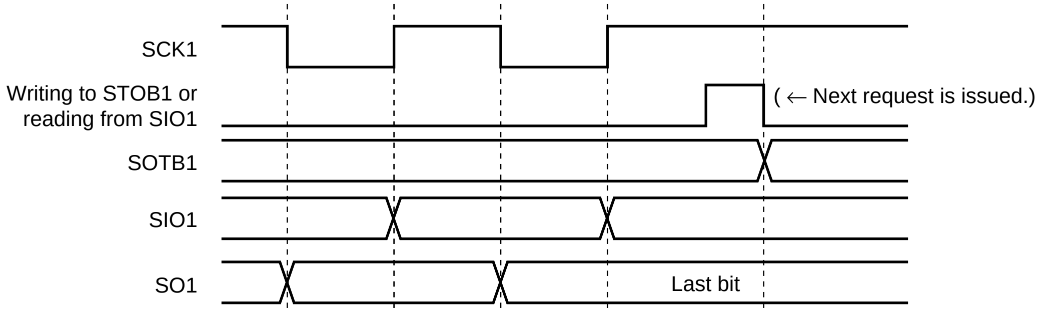


(5) Output value of SO1 pin (last bit)

After communication has been completed, the SO1 pin holds the output value of the last bit.

Figure 17-7. Output Value of SO1 Pin (last bit) (1/2)

(1) Type 1; when CKP1 = 0 and DAP1 = 0


SCK1
Writing to STOB1 or reading from SIO1
(← Next request is issued.)
SOTB1
SIO1
SO1
Last bit

(2) Type 2; when CKP1 = 0 and DAP1 = 1

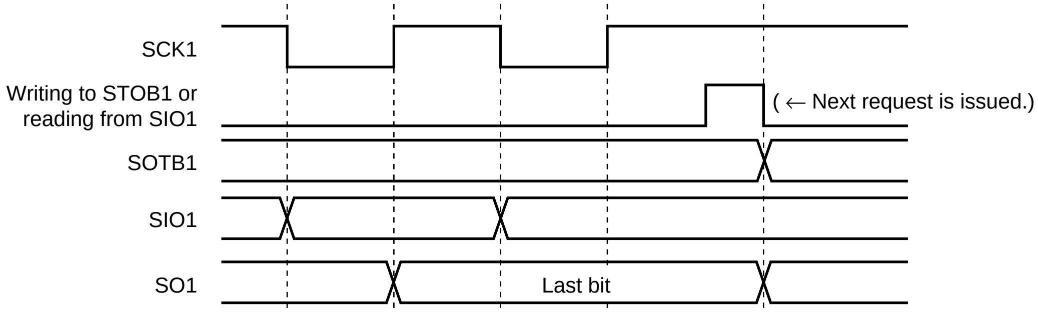
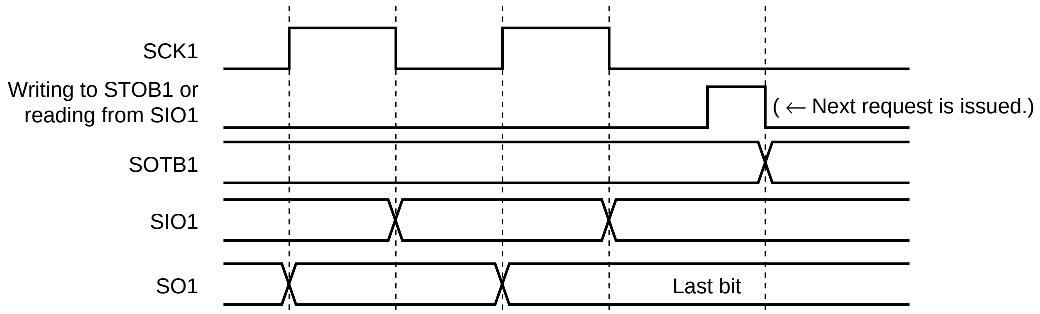
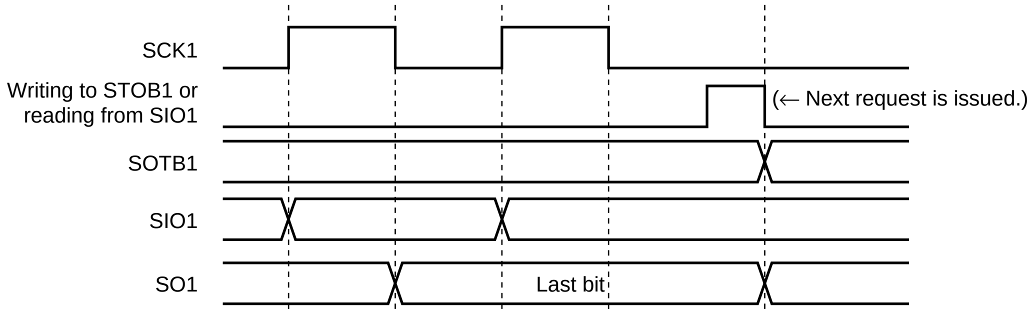

SCK1
Writing to STOB1 or reading from SIO1
(← Next request is issued.)
SOTB1
SIO1
SO1
Last bit

Figure 17-7. Output Value of SO1 Pin (last bit) (2/2)

(3) Type 3; when CKP1 = 1 and DAP1 = 0


SCK1
Writing to STOB1 or reading from SIO1
(← Next request is issued.)
SOTB1
SIO1
SO1
Last bit

(4) Type 4; when CKP1 = 1 and DAP1 = 1


SCK1
Writing to STOB1 or reading from SIO1
(← Next request is issued.)
SOTB1
SIO1
SO1
Last bit

(6) SCK1 pin

The status of the SCK1 pin is as follows if bit 7 (CSIE1) of serial operation mode register 1 (CSIM1) is cleared to 0.

Table 17-2. Status of SCK1 Pin

CKP1	CKS12-10	SCK1 Pin
CKP1 = 0	CKS12, 11, 10 ≠ 1, 1, 1	Outputs high level.
	CKS12, 11, 10 = 1, 1, 1	Outputs high level.
CKP1 = 1[Note]	CKS12, 11, 10 ≠ 1, 1, 1[Note]	Outputs low level[Note].
	CKS12, 11, 10 = 1, 1, 1	Outputs high level.

Note Status at reset

(7) SO1 pin

The status of the SO1 pin is as follows if bit 7 (CSIE1) of serial operation mode register 1 (CSIM1) is cleared to 0.

Table 17-3. Status of SO1 Pin

TRMD1	DAP1	DIR1	SO1 Pin
TRMD1 = 0[Note]	–	–	Outputs low level[Note].
TRMD1 = 1	DAP1 = 0	–	Value of SO1 latch (low-level output).
	DAP1 = 1	DIR1 = 0	Value of bit 7 of SOTB1
		DIR1 = 1	Value of bit 0 of SOTB1

Note Status at reset

Caution **If a value is written to TRMD1, DAP1, and DIR1, the output value of the SO1 pin changes.**

[MEMO]

CHAPTER 18 SERIAL INTERFACE (IIC0) (μPD780078Y SUBSERIES ONLY)

18.1 Serial Interface (IIC0) Functions

The 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, the IIC0 requires pull-up resistors for the serial clock line (SCL0) and the serial data bus line (SDA0).

Figure 18-1 shows a block diagram of serial interface (IIC0).

Figure 18-1. Serial Interface (IIC0) Block Diagram

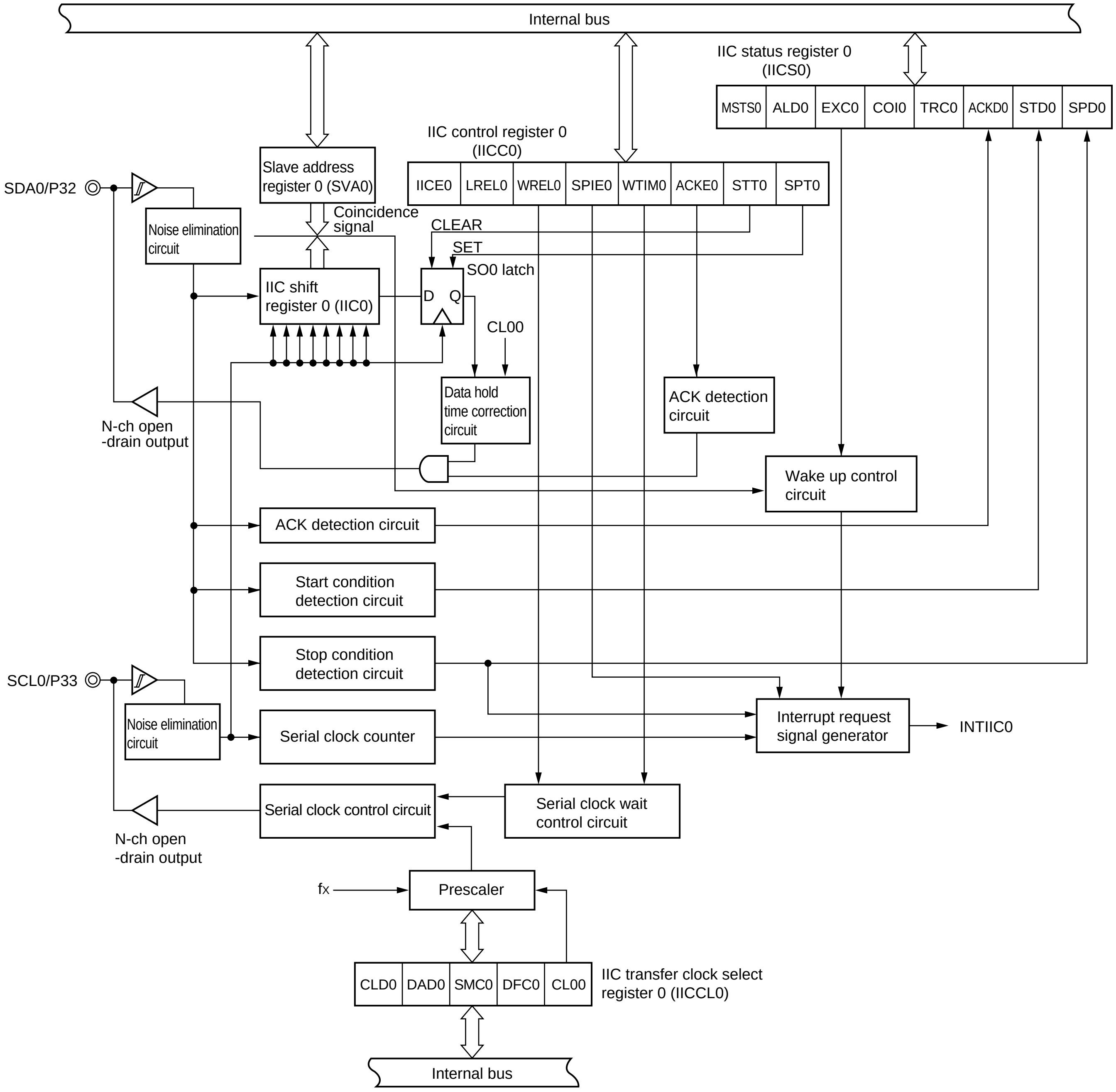
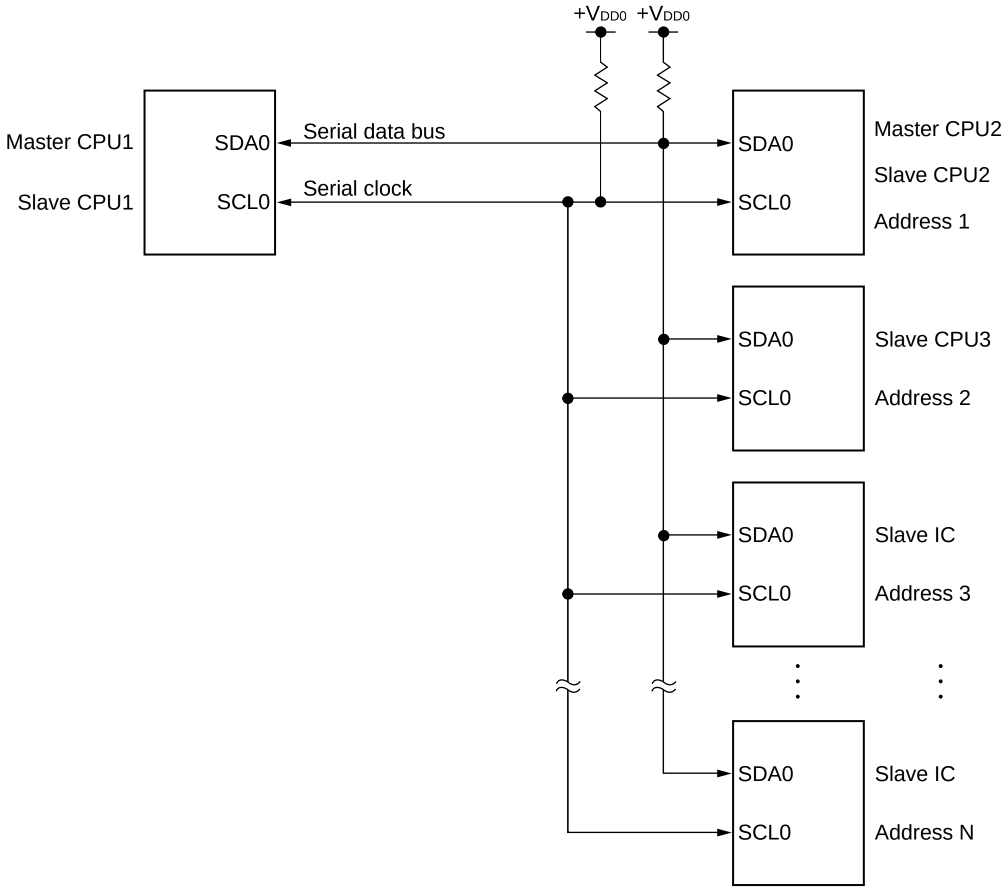


Figure 18-2 shows a serial bus configuration example.

Figure 18-2. Serial Bus Configuration Example Using I²C Bus



18.2 Serial Interface (IIC0) Configuration

The serial interface (IIC0) includes the following hardware.

Table 18-1. Serial Interface (IIC0) Configuration

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 select register 0 (IICCL0)

(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 by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the value of IIC0 to 00H.

(2) Slave address register 0 (SVA0)

This register sets local addresses when in slave mode.

SVA0 is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the value of SVA0 to 00H.

(3) SO0 latch

The SO0 latch is used to retain the SDA0 pin's output level.

(4) Wake-up control circuit

This circuit generates an interrupt request when the address received by this register matches the address value set to the slave address register 0 (SVA0) or when an extension code is received.

(5) Clock selector

This selects the sampling clock to be used.

(6) Serial clock counter

This counter counts the serial clocks that are output or input during transmit/receive operations and is used to verify that 8-bit data was 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 : Bit 3 of the IIC control register 0 (IICC0)
SPIE0 bit : Bit 4 of the IIC control register 0 (IICC0)

(8) Serial clock control circuit

During master mode, this circuit generates the clock output via the SCL0 pin from a sampling clock.

(9) Serial clock wait control circuit

This circuit controls the wait timing.

(10) ACK output circuit, stop condition detection circuit, start condition detection circuit, and ACK detection circuit

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.

18.3 Registers to Control Serial Interface (IIC0)

Serial interface (IIC0) is controlled via three types of registers.

- IIC control register 0 (IICC0)
- IIC status register 0 (IICS0)
- IIC transfer clock select register 0 (IICCL0)

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^2C operations, set wait timing, and set other I^2C operations.
IICC0 is set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets the value of IICC0 to 00H.

Caution In I^2C bus mode, set the port mode register (PMXX) as follows. Set the output latch to 0.

- **Set P32 (SDA0) to output mode (PM32 = 0)**
- **Set P33 (SCL0) to output mode (PM33 = 0)**

Figure 18-3. IIC Control Register 0 (IICC0) Format (1/3)

Address: FFA8H At 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	Stops operation. Presets IIC status register 0 (IICS0). Stops internal operation.
1	Enables operation.
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	This exits from the current communications operation and sets standby mode. 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 go 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	Does not cancel wait
1	Cancels wait. This setting is automatically cleared after wait is canceled.
When WREL0 is set (wait released) during the wait period at the ninth clock pulse in transmission status (TRC0 = 1), the SDA0 line goes into the high impedance state (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	Disable
1	Enable
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 18-3. IIC Control Register 0 (IICC0) Format (2/3)

<table>
<tr><th>WTIM0</th><th colspan="2">Control of Wait and Interrupt Request Generation</th></tr>
<tr><td>0</td><td colspan="2">Interrupt request is generated at the eighth clock's falling edge.
Master mode : After output of eight clocks, clock output is set to low level and wait is set.
Slave mode : After input of eight clocks, the clock is set to low level and wait is set for master device.</td></tr>
<tr><td>1</td><td colspan="2">Interrupt request is generated at the ninth clock's falling edge.
Master mode : After output of nine clocks, clock output is set to low level and wait is set.
Slave mode : After input of nine clocks, the clock is set to low level and wait is set for master device.</td></tr>
<tr><td colspan="3">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.</td></tr>
<tr><td colspan="2">Condition for clearing (WTIM0 = 0) Note</td><td>Condition for setting (WTIM0 = 1)</td></tr>
<tr><td colspan="2">• Cleared by instruction
• When $\overline{\text{RESET}}$ is input</td><td>• Set by instruction</td></tr>
</table>

<table>
<tr><th>ACKE0</th><th colspan="2">Acknowledge Control</th></tr>
<tr><td>0</td><td colspan="2">Disable acknowledge.</td></tr>
<tr><td>1</td><td colspan="2">Enable acknowledge. During the ninth clock period, the SDA0 line is set to low level. However, the ACK is invalid during address transfers and is valid when EXC0 = 1.</td></tr>
<tr><td colspan="2">Condition for clearing (ACKE0 = 0) Note</td><td>Condition for setting (ACKE0 = 1)</td></tr>
<tr><td colspan="2">• Cleared by instruction
• When $\overline{\text{RESET}}$ is input</td><td>• Set by instruction</td></tr>
</table>

<table>
<tr><th>STT0</th><th colspan="2">Start Condition Trigger</th></tr>
<tr><td>0</td><td colspan="2">Does not generate a start condition.</td></tr>
<tr><td>1</td><td colspan="2">When bus is released (during STOP mode):
Generates a start condition (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, it releases the bus and then automatically generates a start condition.
Wait status (during master mode):
Generates a restart condition after wait is released.</td></tr>
<tr><td colspan="3">Cautions concerning set timing
• For master reception : Cannot be set during transfer. Can be set only at the waiting period when ACKE0 has been set to 0 and 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 waiting period.
• Cannot be set at the same time as SPT0</td></tr>
<tr><td colspan="2">Condition for clearing (STT0 = 0) Note</td><td>Condition for setting (STT0 = 1)</td></tr>
<tr><td colspan="2">• Cleared by instruction
• Cleared by loss in arbitration
• Cleared after start condition is generated by master device
• Cleared when by setting LREL0 to 1
• When IICE0 = 0
• When $\overline{\text{RESET}}$ is input</td><td>• Set by instruction</td></tr>
</table>

Note This flag's signal is invalid when IICE0 = 0.

Figure 18-3. IIC Control Register 0 (IICC0) Format (3/3)

<table>
<tr><th>SPT0</th><th colspan="2">Stop Condition Trigger</th></tr>
<tr><td>0</td><td colspan="2">Stop condition is not generated.</td></tr>
<tr><td>1</td><td colspan="2">Stop condition is generated (termination of master device’s transfer).
After the SDA0 line goes to low level, either set the SCL0 line to high level or wait until it goes to high level. Next, after the rated amount of time has elapsed, the SDA0 line changes from low level to high level and a stop condition is generated.</td></tr>
<tr><td colspan="3">Cautions concerning set timing
• For master reception : Cannot be set during transfer. Can be set only at the waiting period when ACKE0 has been set to 0 and 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 waiting 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.</td></tr>
<tr><td colspan="2">Condition for clearing (SPT0 = 0)</td><td>Condition for setting (SPT0 = 1)</td></tr>
<tr><td colspan="2">• Cleared by instruction
• Cleared by loss in arbitration
• Automatically cleared after stop condition is detected
• Cleared when by setting LREL0 to 1
• When IICE0 = 0
• When $\overline{\text{RESET}}$ is input</td><td>• Set by instruction</td></tr>
</table>

Note Set SPT0 only during master mode. However, you must set SPT0 and generate a stop condition before the first stop condition is detected following the switch to operation enable status. For details, see **18.5.15 Other cautions**.

Caution When bit 3 (TRC0) of the 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 for 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 setting.

(2) IIC status register 0 (IICS0)

This register indicates the status of the I^2C.

IICS0 is set by a 1-bit or 8-bit memory manipulation instruction. IICS00 is a read-only register.

$\overline{\text{RESET}}$ input sets the value of IICS0 to 00H.

Figure 18-4. IIC Status Register 0 (IICS0) Format (1/3)

Address: FFA9H At 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 high-order 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 18-4. IIC Status Register 0 (IICS0) Format (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 for high impedance.
1	Transmit status. The value in the SO0 latch is enabled for output to the SDA0 line (valid starting at the rising 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 by 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 by 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 (IIS0) to 1 at the ninth clock when bit 3 (TRC0) of IIC status register 0 (IICS0) is 1, TRC0 is cleared, and the SDA0 line goes into a high-impedance state.

Figure 18-4. IIC Status Register 0 (IICS0) Format (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 was terminated and the bus was 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 select register 0 (IICCL0)

This register is used to set the transfer clock for the I^{2}C bus.

IICCL0 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the value of IICCL0 to 00H.

Figure 18-5. IIC Transfer Clock Select Register 0 (IICCL0) Format (1/2)

Address: FFAAH At Reset: 00H R/W [Note]

Symbol	7	6	$\boxed{5}$	$\boxed{4}$	3	2	1	0
IICCL0	0	0	CLD0	DAD0	SMC0	DFC0	0	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 18-5. IIC Transfer Clock Select Register 0 (IICCL0) Format (2/2)

DFC0	Control of Digital Filter Operation Note 1
0	Digital filter OFF
1	Digital filter ON

CL00	Selection of Transfer Rate	
	Standard mode	High-speed mode
0	f_X/44 Note 2	f_X/18 (465 kHz)
1	f_X/86 (97.5 kHz)	

Notes 1. The digital filter can be used when in high-speed mode. Response time is slower when the digital filter is used.

2. Transfer rate in standard mode can only be used when f_X is less than 100 kHz.

Caution Stop serial transfer once before rewriting CL00 to other than the same value.

Remarks 1. IICE0: Bit 7 of IIC control register 0 (IICC0)

2. f_X: Main system clock oscillation frequency

3. Figures in parentheses are for operation with f_X = 8.38 MHz.

4. The transfer clock does not change in the high-speed mode even if DFC0 is turned on and off.

(4) IIC shift register 0 (IIC0)

This register is used for serial transmission/reception (shift operations) that are 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 At Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IIC0								

(5) Slave address register 0 (SVA0)

This register holds the I^2C's slave addresses.

It can be read from or written to in 8-bit units, but bit 0 is fixed to "0".

Address: FFABH At Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SVA0								0

18.4 I²C Bus Mode Functions

18.4.1 Pin configuration

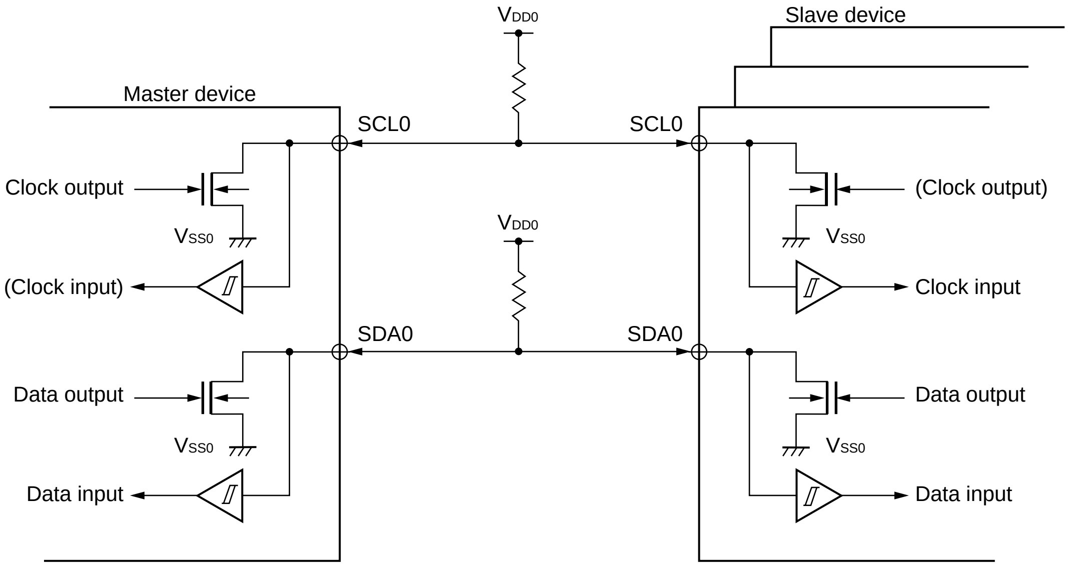
The serial clock pin (SCL0) and serial data bus pin (SDA0) are configured as follows.

(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.

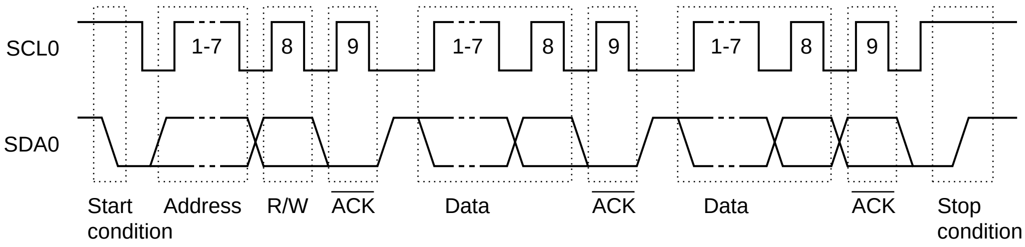
Figure 18-6. Pin Configuration Diagram



18.5 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 18-7 shows the transfer timing for the "start condition", "data", and "stop condition" output via the I^{2}C bus's serial data bus.

Figure 18-7. I^{2}C Bus Serial Data Transfer Timing



The master device outputs the start condition, slave address, and stop condition.

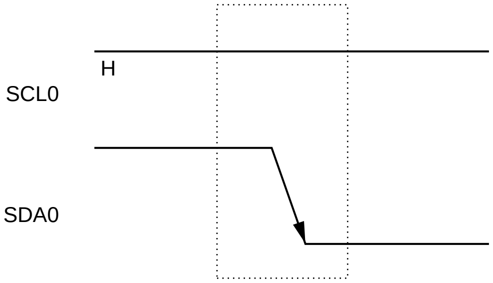
The acknowledge signal ($\overline{\text{ACK}}$) can be output by either the master or slave device (normally, it is output by the device that receives 8-bit data).

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.

18.5.1 Start conditions

A start condition is met when the SCL0 pin is at high level and the SDA0 pin changes from high level to low level. The start conditions for the SCL0 pin and SDA0 pin are signals that the master device outputs to the slave device when starting a serial transfer. The slave device includes hardware for detecting start conditions.

Figure 18-8. Start Conditions



A start condition is output when the IIC control register 0 (IICC0)'s bit 1 (STT0) is set (to "1") after a stop condition has been detected (SPD0: Bit 0 = 1 in the IIC status register 0 (IICS0)). When a start condition is detected, IICS0's bit 1 (STD0) is set (to "1").

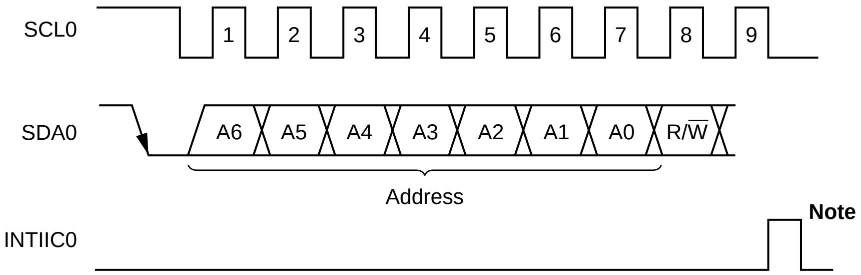
18.5.2 Addresses

The address is defined by the 7 bits of data that follow the start condition.

An address is a 7-bit data segment that is output in order to select one of the slave devices that are connected to the master device via bus lines. Therefore, each slave device connected via the bus lines must have a unique address.

The slave devices include hardware that detects the start condition and checks whether or not the 7-bit address data matches the data values stored in the slave address register 0 (SVA0). If the address data matches the SVA0 values, the slave device is selected and communicates with the master device until the master device transmits a start condition or stop condition.

Figure 18-9. Address



Note INTIIC0 is not issued if data other than a local address or extension code is received during slave device operation.

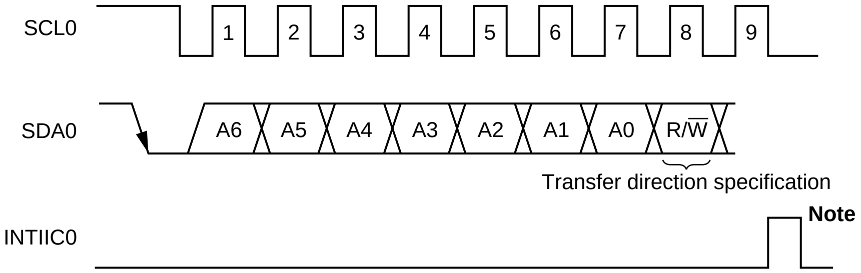
The slave address and the eighth bit, which specifies the transfer direction as described in **18.5.3 Transfer direction specification** below, are together written to the IIC shift register 0 (IIC0) and are then output. Received addresses are written to IIC0.

The slave address is assigned to the high-order 7 bits of IIC0.

18.5.3 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 18-10. Transfer Direction Specification



Note INTIIC0 is not issued if data other than a local address or extension code is received during slave device operation.

18.5.4 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).

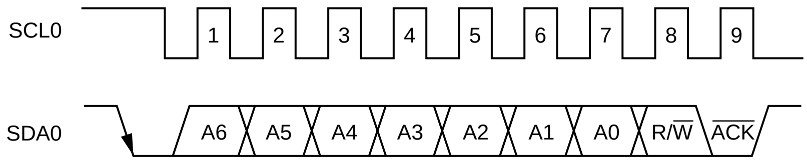
When bit 2 (ACKE0) of the IIC control register 0 (IICC0) is set to 1, automatic $\overline{\text{ACK}}$ signal generation is enabled.

Transmission of the eighth bit following the 7 address data bits causes bit 3 (TRC0) of the 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 18-11. $\overline{\text{ACK}}$ Signal



When the local address is received, an $\overline{\text{ACK}}$ signal is automatically output in sync 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.

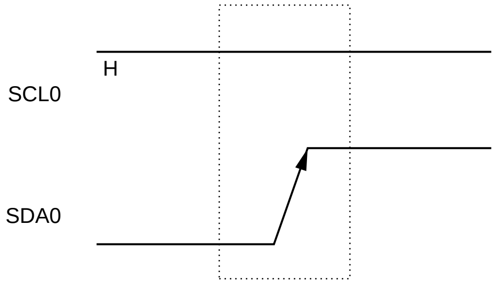
- When 8-clock wait is selected : $\overline{\text{ACK}}$ signal is output when ACKE0 is set to "1" before wait cancellation.
- When 9-clock wait is selected : $\overline{\text{ACK}}$ signal is automatically output at the falling edge of the SCL0's eighth clock if ACKE0 has already been set to "1".

18.5.5 Stop condition

When the SCL0 pin is at high level, changing the SDA0 pin from low level to high level generates a stop condition.

A stop condition is a signal that the master device outputs to the slave device when serial transfer has been completed. The slave device includes hardware that detects stop conditions.

Figure 18-12. Stop Condition



A stop condition is generated when bit 0 (SPT0) of the IIC control register 0 (IICC0) is set (to “1”). When the stop condition is detected, bit 0 (SPD0) of the IIC status register 0 (IICS0) is set (to “1”) and INTIIC0 is generated when bit 4 (SPIE0) of IICC0 is set (to “1”).

18.5.6 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 18-13. Wait Signal (1/2)

(1) When master device has a nine-clock wait and slave device has an eight-clock wait (master transmits, slave receives, and ACKE0 = 1)

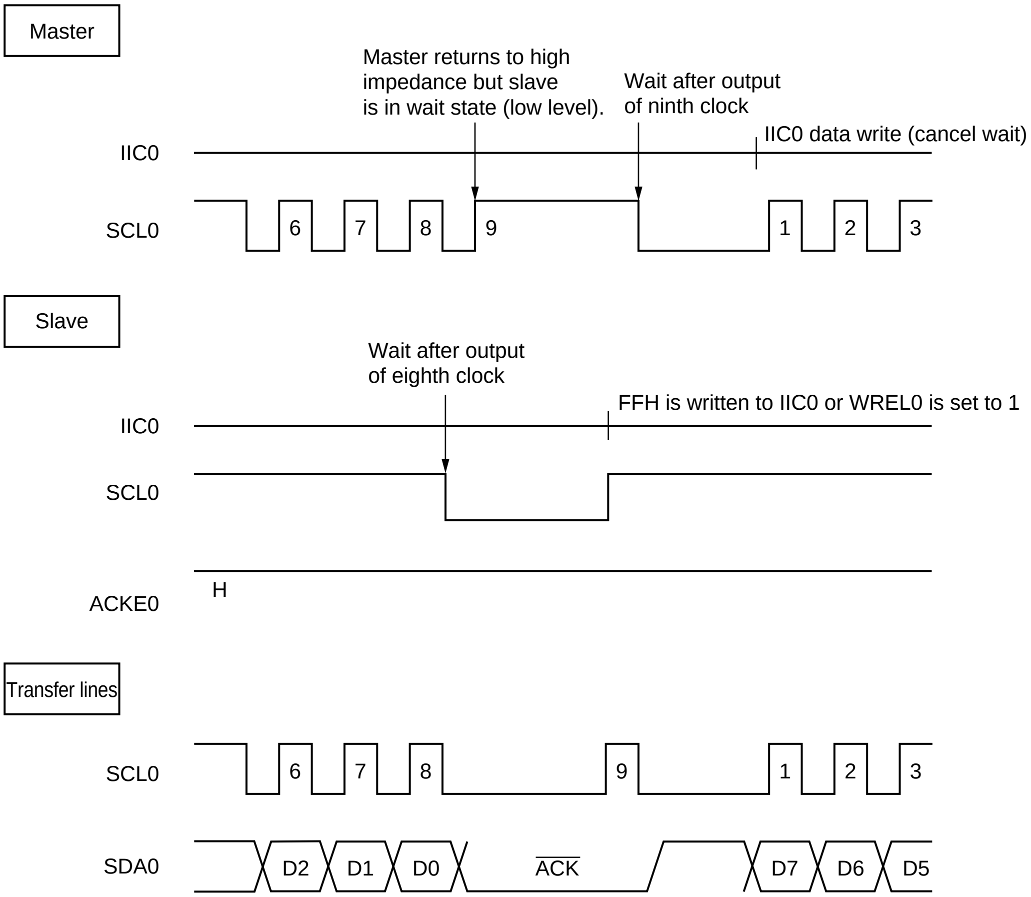
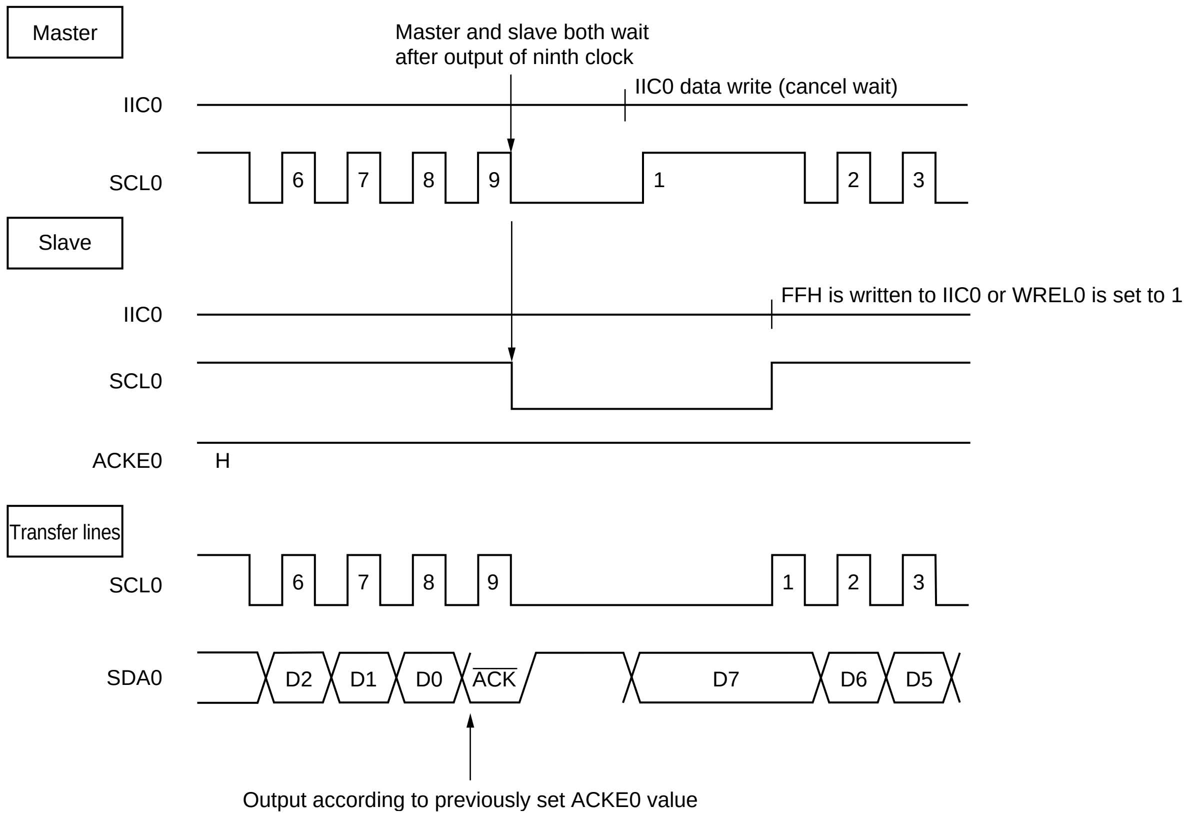

Master
Master returns to high impedance but slave is in wait state (low level).
Wait after output of ninth clock
IIC0 data write (cancel wait)
IIC0
SCL0
6
7
8
9
1
2
3
Slave
Wait after output of eighth clock
FFH is written to IIC0 or WREL0 is set to 1
IIC0
SCL0
ACKE0
H
Transfer lines
SCL0
6
7
8
9
1
2
3
SDA0
D2
D1
D0
$\overline{\text{ACK}}$
D7
D6
D5

Figure 18-13. Wait Signal (2/2)

(2) When master and slave devices both have a nine-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 (cancel wait)
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
$\overline{ACK}$
D7
D6
D5
Output according to previously set ACKE0 value

Remarks ACKE0 : Bit 2 of IIC control register (IICC0)
WREL0 : Bit 5 of IIC control register (IICC0)

A wait may be automatically generated depending on the setting for bit 3 (WTIM0) of the IIC control register 0 (IICC0).

Normally, when bit 5 (WREL0) of IICC0 is set to “1” or when FFH is written to the IIC shift register 0 (IIC0), the wait status is canceled and the transmitting side write 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”

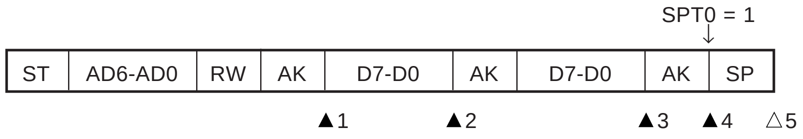
18.5.7 I^{2}C interrupt requests (INTIIC0)

The INTIIC0 interrupt request timing and the 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)

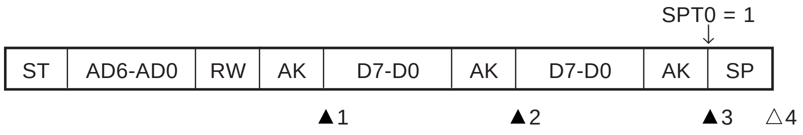
△5 : IICS0 = 00000001B

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

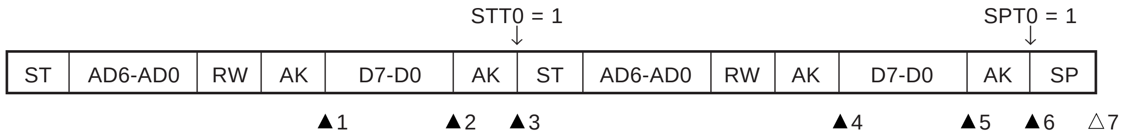
Remark ▲ : Always generated

△ : 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)

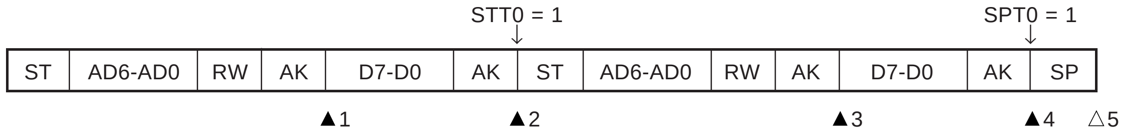
△7 : IICS0 = 00000001B

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

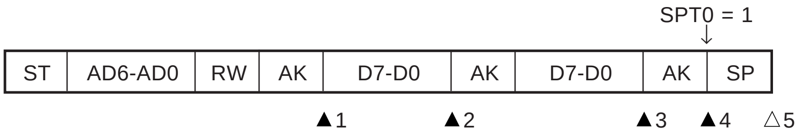
Remark ▲: Always generated

△: Generated only when SPIE0 = 1

× : Don’t care

(c) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

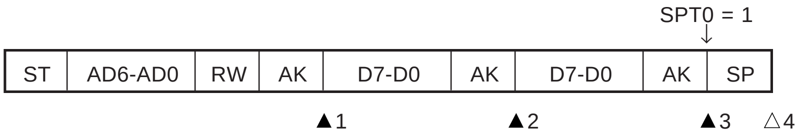
(i) When WTIM0 = 0



▲1 : IICS0 = 1010×110B
▲2 : IICS0 = 1010×000B
▲3 : IICS0 = 1010×000B (Sets WTIM0)
▲4 : IICS0 = 1010××00B
△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 time (matches with SVA0))

(a) Start ~ Address ~ Data ~ Data ~ Stop

(i) When WTIM0 = 0

ST	AD6-AD0	RW	AK	D7-D0	AK	D7-D0	AK	SP
			▲1	▲2		▲3		△4

▲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

ST	AD6-AD0	RW	AK	D7-D0	AK	D7-D0	AK	SP
			▲1		▲2		▲3	△4

▲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)

ST	AD6-AD0	RW	AK	D7-D0	AK	ST	AD6-AD0	RW	AK	D7-D0	AK	SP
			▲1	▲2					▲3	▲4		△5

▲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)

ST	AD6-AD0	RW	AK	D7-D0	AK	ST	AD6-AD0	RW	AK	D7-D0	AK	SP
			▲1		▲2				▲3		▲4	△5

▲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)

ST	AD6-AD0	RW	AK	D7-D0	AK	ST	AD6-AD0	RW	AK	D7-D0	AK	SP
			▲1	▲2				▲3		▲4		△5

▲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)

ST	AD6-AD0	RW	AK	D7-D0	AK	ST	AD6-AD0	RW	AK	D7-D0	AK	SP
			▲1		▲2			▲3	▲4		▲5	△6

▲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))

ST	AD6-AD0	RW	AK	D7-D0	AK	ST	AD6-AD0	RW	AK	D7-D0	AK	SP
			▲1	▲2					▲3			△4

▲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))

ST	AD6-AD0	RW	AK	D7-D0	AK	ST	AD6-AD0	RW	AK	D7-D0	AK	SP
			▲1		▲2				▲3			△4

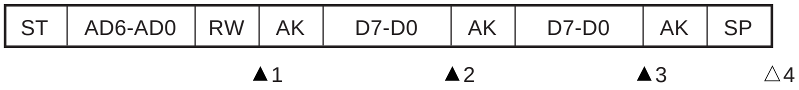
▲1 : IICS0 = 0001×110B
▲2 : IICS0 = 0001××00B
▲3 : IICS0 = 00000×10B
△4 : IICS0 = 00000001B

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

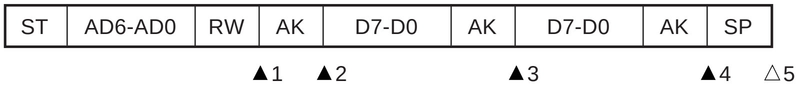
(i) When WTIM0 = 0



▲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)

ST	AD6-AD0	RW	AK	D7-D0	AK	ST	AD6-AD0	RW	AK	D7-D0	AK	SP
			▲1		▲2					▲3	▲4	△5

▲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)

ST	AD6-AD0	RW	AK	D7-D0	AK	ST	AD6-AD0	RW	AK	D7-D0	AK	SP
			▲1	▲2		▲3				▲4		▲5 △6

▲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)

ST	AD6-AD0	RW	AK	D7-D0	AK	ST	AD6-AD0	RW	AK	D7-D0	AK	SP	
			▲1		▲2				▲3		▲4		△5

▲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)

ST	AD6-AD0	RW	AK	D7-D0	AK	ST	AD6-AD0	RW	AK	D7-D0	AK	SP	
			▲1	▲2		▲3			▲4	▲5		▲6	△7

▲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))

ST	AD6-AD0	RW	AK	D7-D0	AK	ST	AD6-AD0	RW	AK	D7-D0	AK	SP
		▲1		▲2					▲3			△4

▲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))

ST	AD6-AD0	RW	AK	D7-D0	AK	ST	AD6-AD0	RW	AK	D7-D0	AK	SP
		▲1	▲2		▲3				▲4			△5

▲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

ST	AD6-AD0	RW	AK	D7-D0	AK	D7-D0	AK	SP
								△1

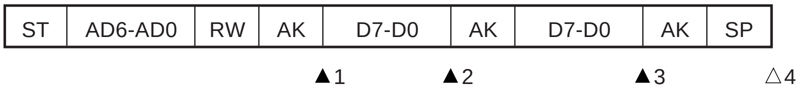
△1 : IICS0 = 00000001B

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

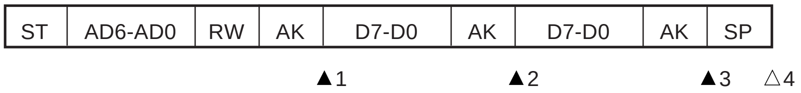
(i) When WTIM0 = 0



▲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

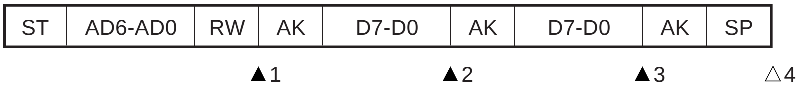


▲1 : IICS0 = 0101×110B (**Example** When ALD0 is read during interrupt servicing)
▲2 : IICS0 = 0001×100B
▲3 : IICS0 = 0001××00B
△4 : IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
× : Don’t care

(b) When arbitration loss occurs during transmission of extension code

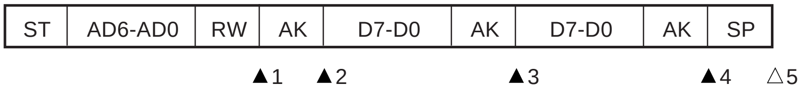
(i) When WTIM0 = 0



▲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

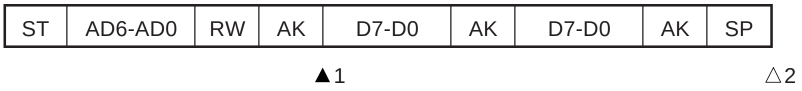


▲1 : IICS0 = 0110×010B (**Example** When ALD0 is read during interrupt servicing)
▲2 : IICS0 = 0010×110B
▲3 : IICS0 = 0010×100B
▲4 : IICS0 = 0010××00B
△5 : IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(6) Operation when arbitration loss occurs (no communication after arbitration loss)

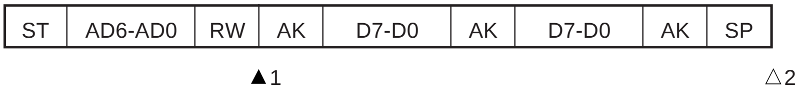
(a) When arbitration loss occurs during transmission of slave address data (when WTIM0 = 1)



▲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 data

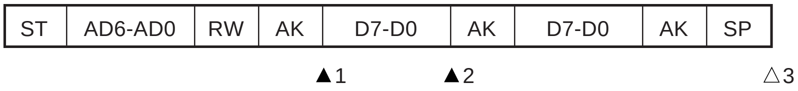


▲1 : IICS0 = 0110×010B (**Example** When ALD0 is read during interrupt servicing)
LREL0 is set to “1” by software
△2 : IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(c) When arbitration loss occurs during transmission of data

(i) When WTIM0 = 0



▲1 : IICS0 = 10001110B

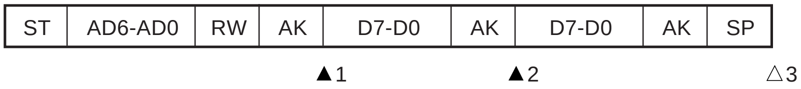
▲2 : IICS0 = 01000000B (**Example** When ALD0 is read during interrupt servicing)

△3 : IICS0 = 00000001B

Remark ▲: Always generated

△: Generated only when SPIE0 = 1

(ii) When WTIM0 = 1



▲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: unmatches with SVA0, WTIM0 = 1)

ST	AD6-AD0	RW	AK	D7-Dn	ST	AD6-AD0	RW	AK	D7-D0	AK	SP	
			▲1					▲2				△3

▲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 - 0

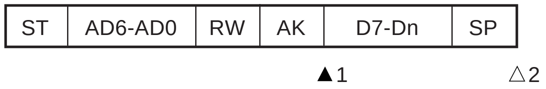
(ii) Extension code

ST	AD6-AD0	RW	AK	D7-Dn	ST	AD6-AD0	RW	AK	D7-D0	AK	SP	
			▲1				▲2					△3

▲1 : IICS0 = 1000×110B
▲2 : IICS0 = 0110×010B (**Example** When ALD0 is read during interrupt servicing)
Sets LREL0 = 1 by software
△3 : IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care
n = 6 - 0

(e) When loss occurs due to stop condition during data transfer

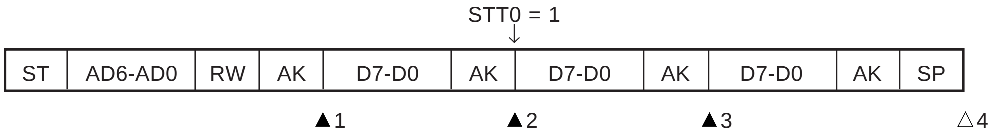


▲1 : IICS0 = 1000×110B
△2 : IICS0 = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
× : Don't care
n = 6 - 0

(f) When arbitration loss occurs due to low-level data when attempting to generate a restart condition

(i) When WTIM0 = 1

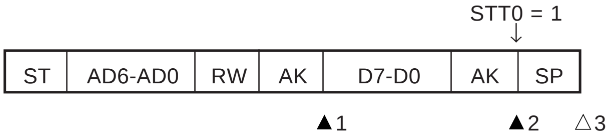


▲1 : IICS0 = 1000×110B
▲2 : IICS0 = 1000×100B (Sets STT0)
▲3 : IICS0 = 01000100B (**Example** When ALD0 is read during interrupt servicing)
△4 : IICS0 = 00000001B

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

(i) When WTIM0 = 1

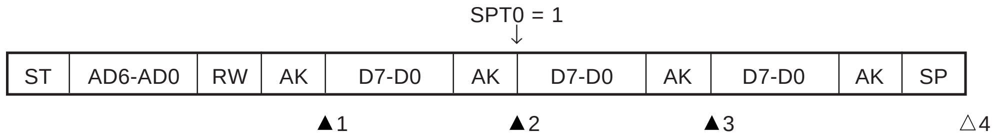


▲1 : IICS0 = 1000×110B
▲2 : IICS0 = 1000××00B (Sets STT0)
△3 : IICS0 = 01000001B

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

(i) When WTIM0 = 1



▲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

18.5.8 Interrupt request (INTIIC0) generation timing and wait control

The setting of bit 3 (WTIM0) in the IIC control register 0 (IICC0) determines the timing by which INTIIC0 is generated and the corresponding wait control, as shown in Table 18-2.

Table 18-2. 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 the slave address register 0 (SVA0).
At this point, $\overline{\text{ACK}}$ is output regardless of the value set to IICC0's bit 2 (ACKE0). For a slave device that has received an extension code, INTIIC0 occurs at the falling edge of the eighth clock.

2. If the received address does not match the contents of the slave address register 0 (SVA0), neither INTIIC0 nor a wait occurs.

Remark The numbers in the table indicate the number of the serial clock's clock signals. Interrupt requests and wait control are both synchronized with the falling edge of these clock signals.

(1) During address transmission/reception

- Slave device operation : Interrupt and wait timing are determined regardless of the WTIM0 bit.
- Master device operation: Interrupt and wait timing occur at the falling edge of the ninth clock regardless of the WTIM0 bit.

(2) During data reception

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM0 bit.

(3) During data transmission

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM0 bit.

(4) Wait cancellation method

The four wait cancellation methods are as follows.

- By setting bit 5 (WREL0) of IIC control register 0 (IICC0) to "1"
- By writing to the IIC shift register 0 (IIC0)
- By setting a start condition (setting bit 1 (STT0) of IICC0 to "1")
- By setting a stop condition (setting IICC0's bit 0 (SPT0) to "1")

When 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.

18.5.9 Address match detection method

When in I^2C bus mode, the master device can select a particular slave device by transmitting the corresponding slave address.

Address match can be detected automatically by hardware. An interrupt frequency (INTIIC0) occurs when a local address has been set to the slave address register 0 (SVA0) and when the address set to SVA0 matches the slave address sent by the master device, or when an extension code has been received.

18.5.10 Error detection

During I^2C bus mode, the status of the serial data bus (SDA0) during data transmission is captured by the 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.

18.5.11 Extension code

(1) When the high-order 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 the slave address register 0 (SVA0) is not affected.

(2) If “111110××” is set to SVA0 by a 10-bit address transfer and “111110××” is transferred from the master device, the results are as follows. Note that INTIIC0 occurs at the falling edge of the eighth clock.

- High-order four 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 the IIC control register 0 (IICC0) to “1” to set the standby mode for the next communication operation.

Table 18-3. 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 that is reserved for different bus format
1111 0××	×	10-bit slave address specification

18.5.12 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 lo™Ωs in arbitration, an arbitration loss flag (ALD0) in the IIC status register 0 (IICS0) is set via the timing by which the arbitration loss occurred, and the SCL0 and SDA0 lines are both set for 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 **18.5.7 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 18-14. Arbitration Timing Example

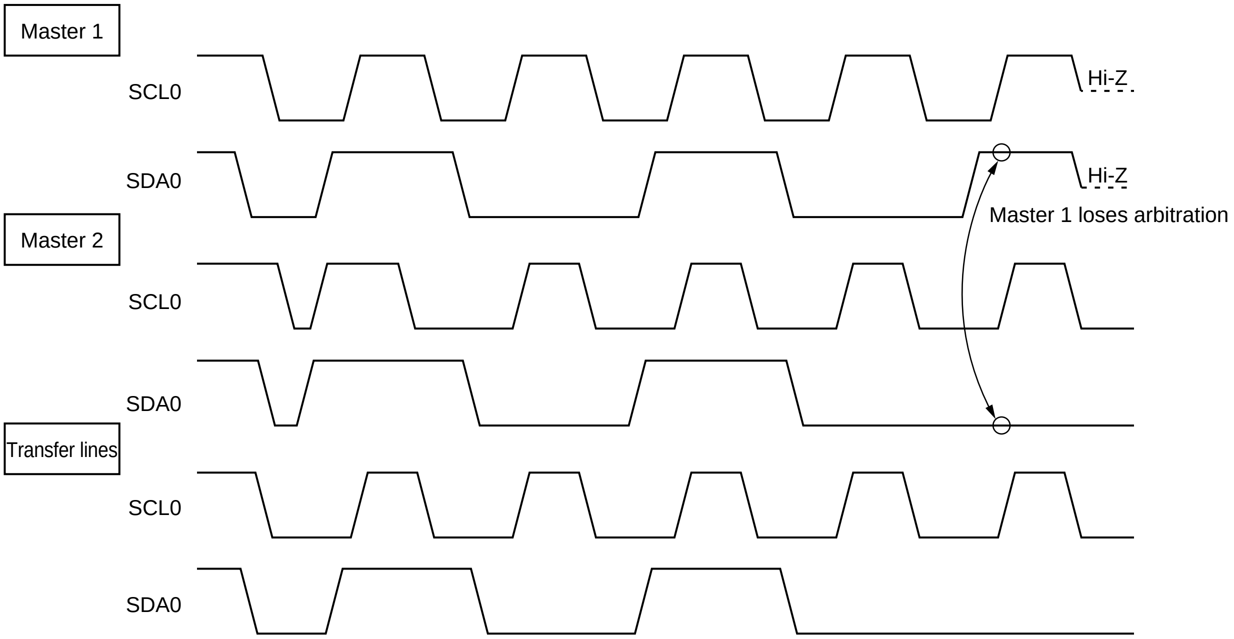


Table 18-4. Status during Arbitration and Interrupt Request Generation Timing

Status during Arbitration	Interrupt Request Generation Timing
During address transmission	At falling edge of eighth or ninth clock following byte transfer $^{\text{Note 1}}$
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) $^{\text{Note 2}}$
When data is at low level while attempting to output a restart condition	At falling edge of eighth or ninth clock following byte transfer $^{\text{Note 1}}$
When stop condition is detected while attempting to output a restart condition	When stop condition is output (when SPIE0 = 1) $^{\text{Note 2}}$
When data is at low level while attempting to output a stop condition	At falling edge of eighth or ninth clock following byte transfer $^{\text{Note 1}}$
When SCL0 is at low level while attempting to output a restart condition	

Notes 1. When WTIM0 (bit 3 of the 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 the IIC control register 0 (IICC0)

18.5.13 Wake up function

The I^{2}C 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 the IIC control register 0 (IICC0) is set regardless of the wake up function, and this determines whether interrupt requests are enabled or prohibited.

18.5.14 Communication reservation

To start master device communications when not currently using a bus, a communication reservation can be made to enable transmission of a start condition when the bus is released. There are two modes under which the bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled ($\overline{\text{ACK}}$ is not returned and the bus was released when bit 6 (LREL0) of the IIC control register 0 (IICC0) was set to “1”).

If bit 1 (STT0) of IICC0 is set while the bus is not used (after a stop condition is detected), a start condition is automatically generated and wait status is set.

When the bus release is detected (when a stop condition is detected), writing to the IIC shift register 0 (IIC0) causes the master’s address transfer to start. At this point, IICC0’s bit 4 (SPIE0) 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 with MSTS0 (bit 7 of the IIC status register 0 (IICS0)) after SST0 is set and a wait time elapses.

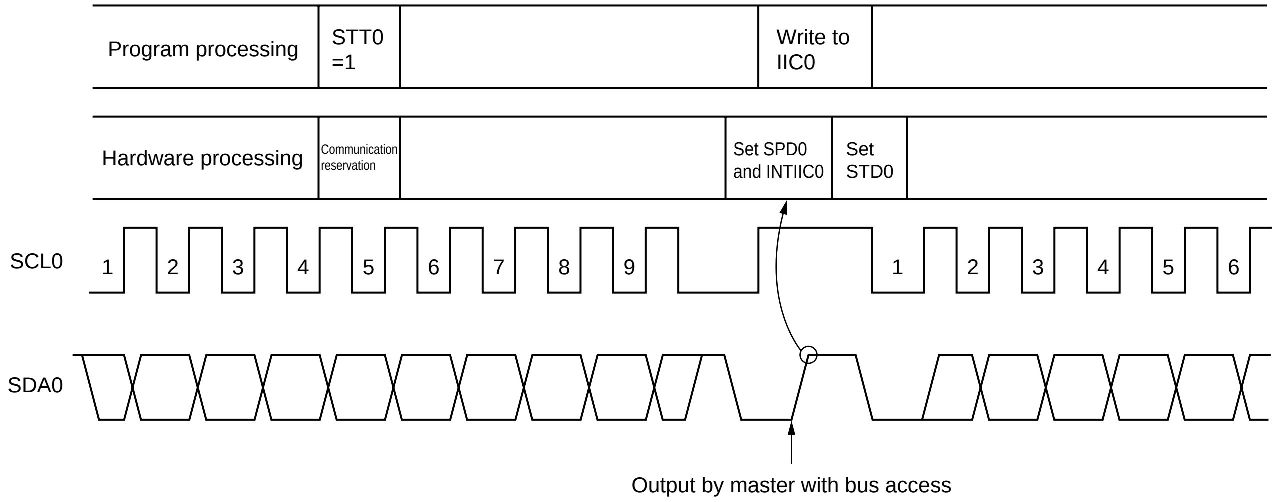
Wait periods, which should be set via software, are listed in Table 18-5. These wait periods can be set via the settings for bits 3 and 0 (SMC0 and CL00) in the IIC transfer clock select register 0 (IICCL0).

Table 18-5. Wait Periods

SMC0	CL00	Wait Period
0	0	26 clocks
0	1	46 clocks
1	0	16 clocks
1	1	

Figure 18-15 shows communication reservation timing.

Figure 18-15. Communication Reservation Timing



Remark IIC0 : IIC shift register 0

STT0 : Bit 1 of IIC control register 0 (IICC0)

STD0 : Bit 1 of IIC status register 0 (IICS0)

SPD0 : Bit 0 of IIC status register 0 (IICS0)

Communication reservations are accepted via the following timing. After bit 1 (STD0) of the IIC status register 0 (IICS0) is set to “1”, a communication reservation can be made by setting bit 1 (STT0) of the IIC control register 0 (IICC0) to “1” before a stop condition is detected.

Figure 18-16. Timing for Accepting Communication Reservations

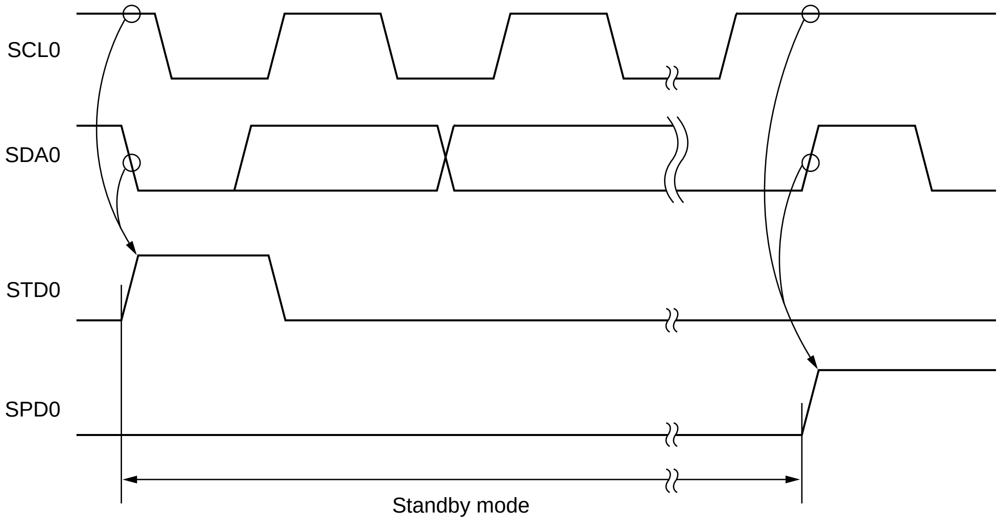
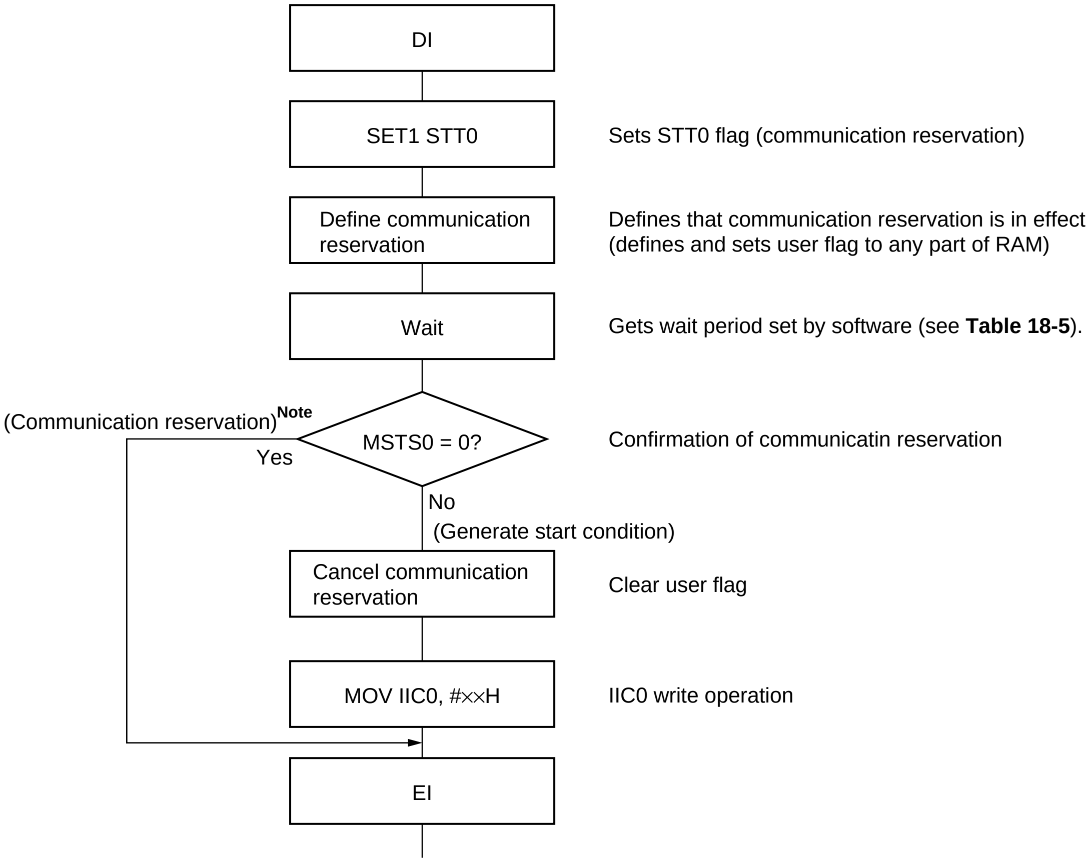


Figure 18-17 shows the communication reservation protocol.

Figure 18-17. 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
Gets wait period set by software (see **Table 18-5**).
(Communication reservation)Note
MSTS0 = 0?
Yes
Confirmation of communicatin reservation
No
(Generate start condition)
Cancel communication reservation
Clear user flag
MOV IIC0, #××H
IIC0 write operation
EI

Note The communication reservation operation executes a write to the 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

18.5.15 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.

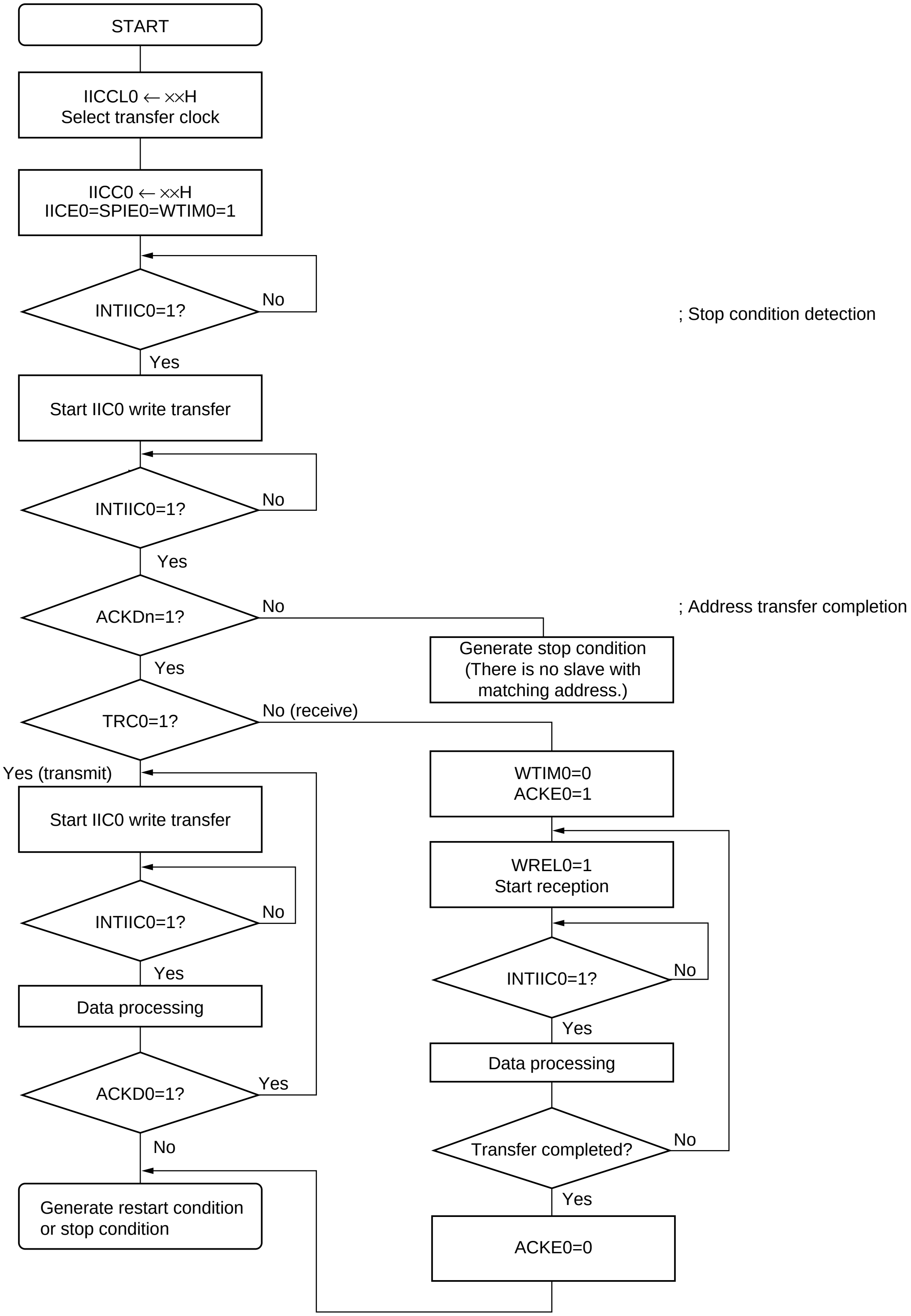
(a) Set IIC transfer clock select register 0 (IICCL0).
(b) Set bit 7 (IICE0) of the IIC control register 0 (IICC0).
(c) Set bit 0 (SPT0) of IICC0.

18.5.16 Communication operations

(1) Master operations

The following is a flow chart of the master operations.

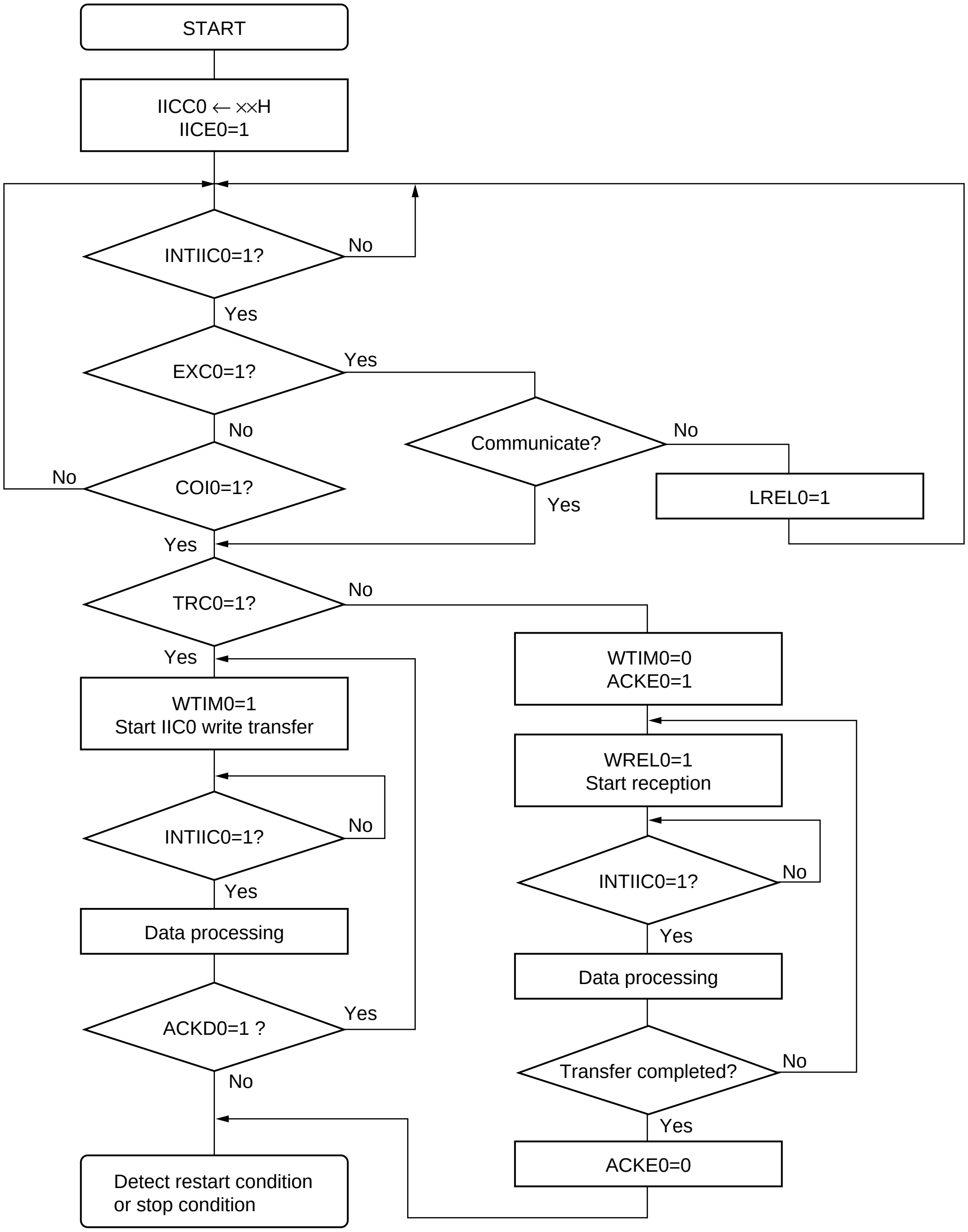
Figure 18-18. Master Operation Flow Chart



(2) Slave operation

An example of slave operation is shown below.

Figure 18-19. Slave Operation Flow Chart



18.6 Timing Charts

When using the I^2C bus mode, the master device outputs an address via the serial bus to select one of several slave devices as its communication partner.

After outputting the slave address, the master device transmits the TRC0 bit (bit 3 of the IIC status register 0 (IICS0)) that specifies the data transfer direction and then starts serial communication with the slave device.

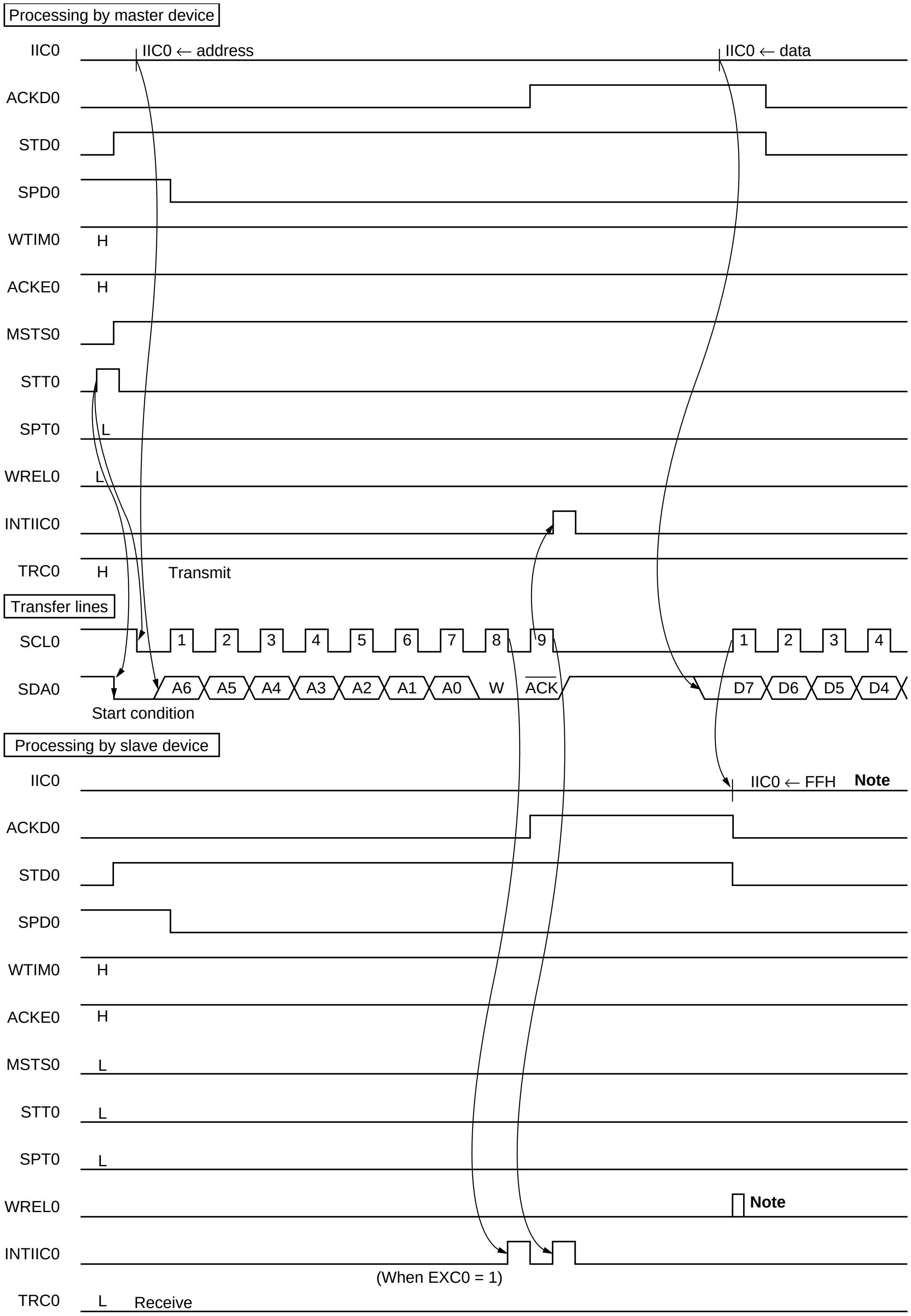
Figures 18-20 and 18-21 show timing charts of the data communication.

The 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 18-20. Example of Master to Slave Communication (When 9-clock Wait Is Selected for Both Master and Slave) (1/3)

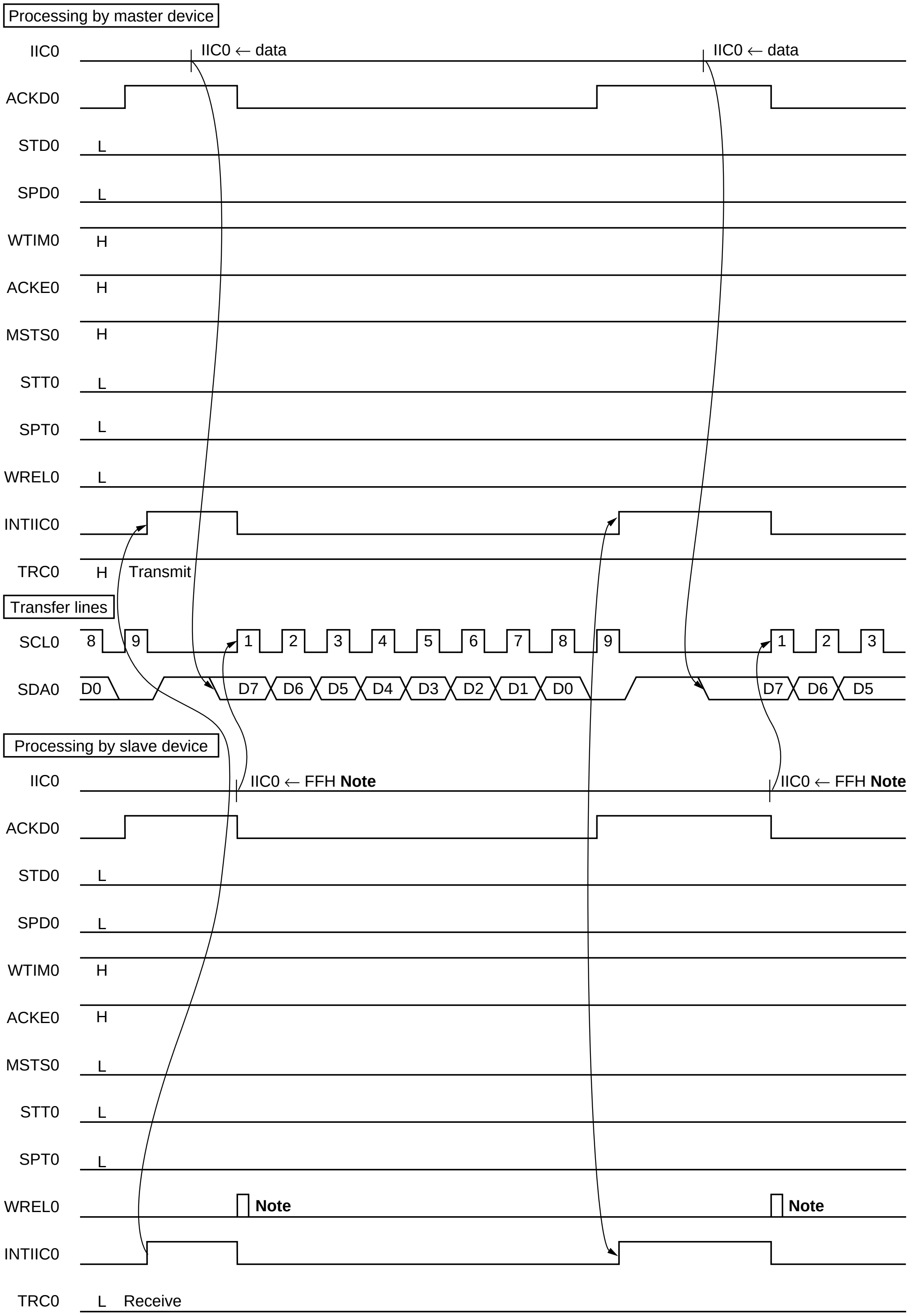
(1) Start condition ~ address



Note To cancel slave wait, write “FFH” to IIC0 or set WREL0.

Figure 18-20. Example of Master to Slave Communication
(When 9-clock Wait Is Selected for Both Master and Slave) (2/3)

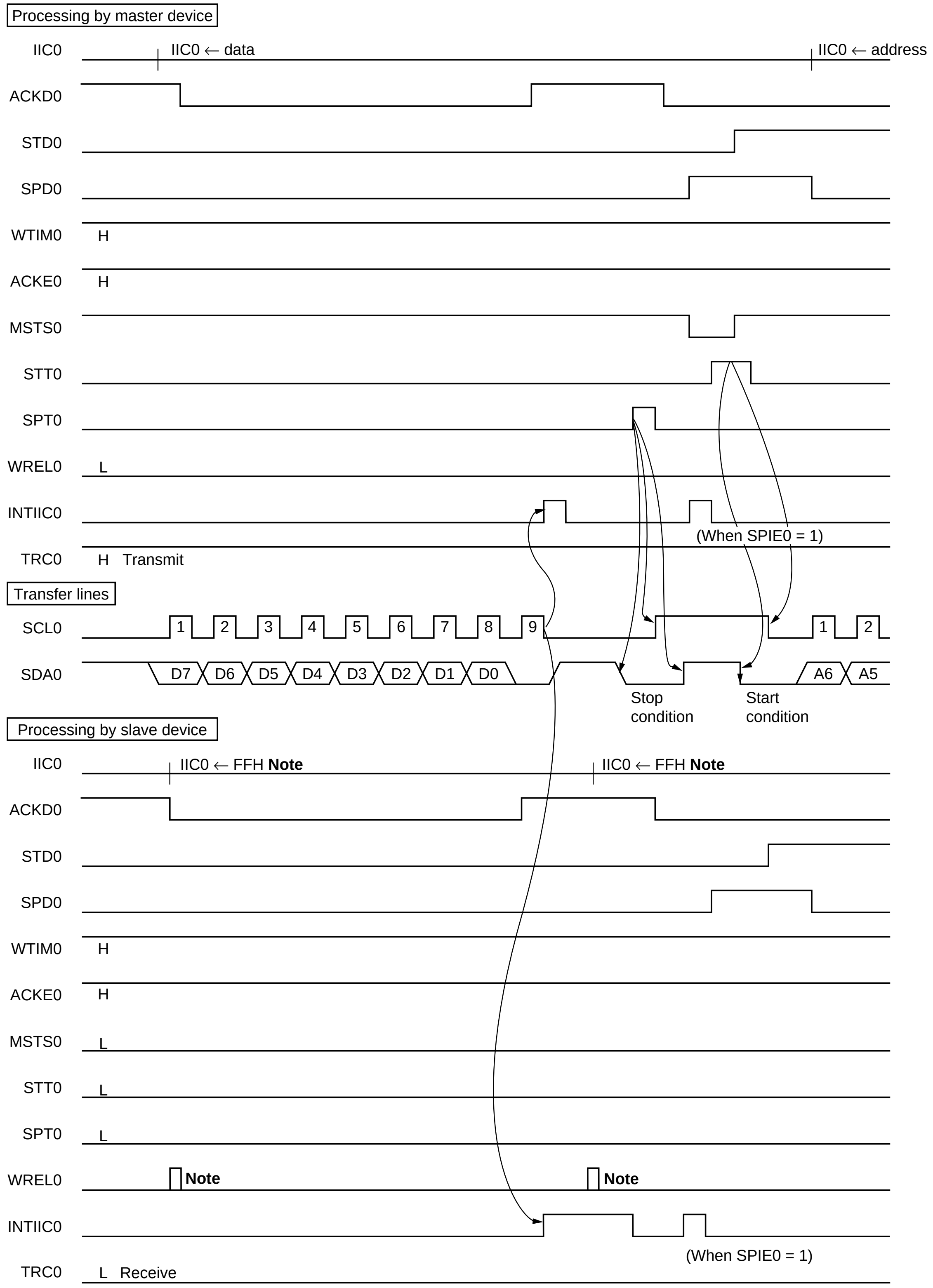
(2) Data



Note To cancel slave wait, write “FFH” to IIC0 or set WREL0.

Figure 18-20. Example of Master to Slave Communication
(When 9-clock Wait Is Selected for Both Master and Slave) (3/3)

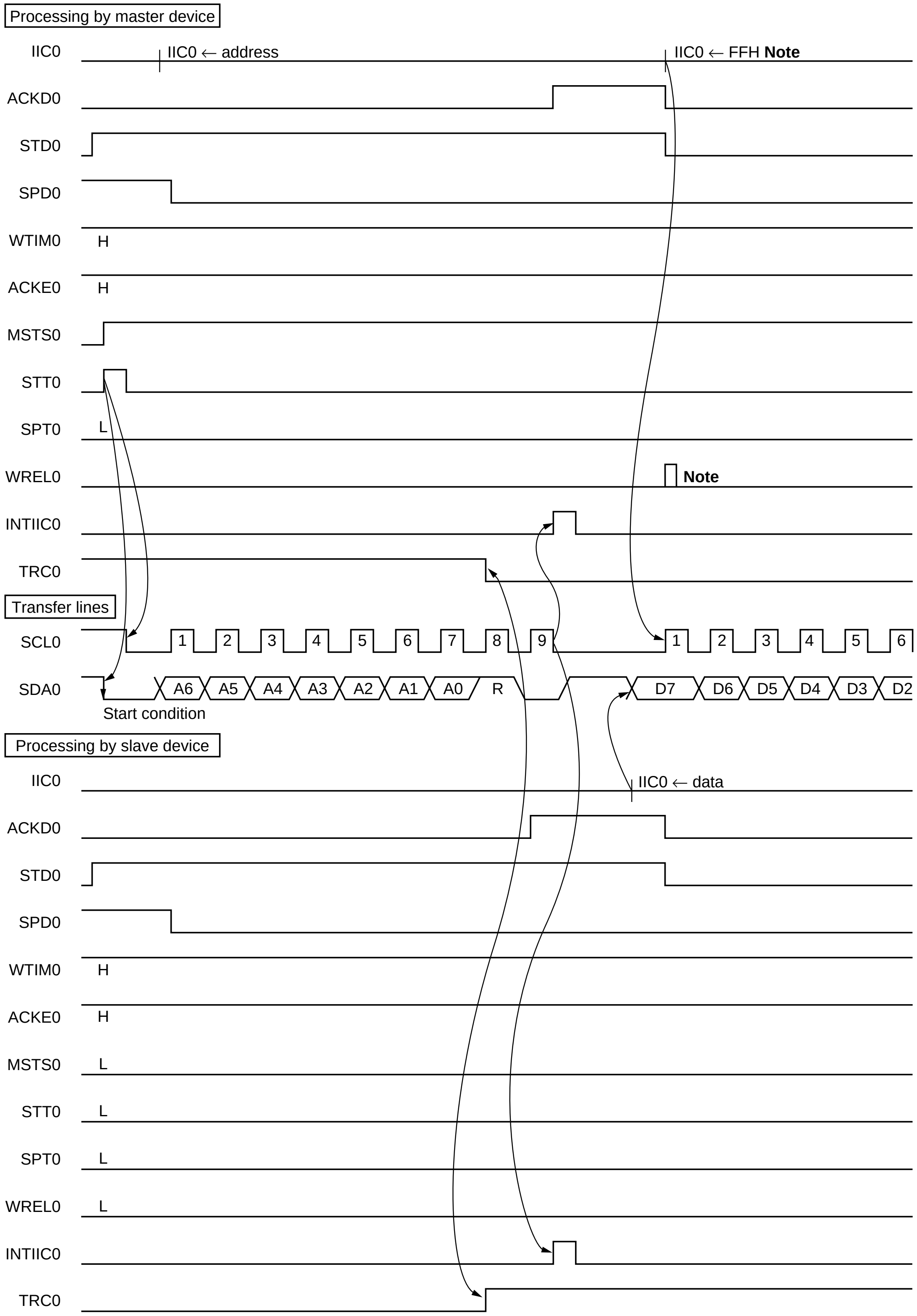
(3) Stop condition



Note To cancel slave wait, write “FFH” to IIC0 or set WREL0.

Figure 18-21. Example of Slave to Master Communication (When 9-clock Wait Is Selected for Both Master and Slave) (1/3)

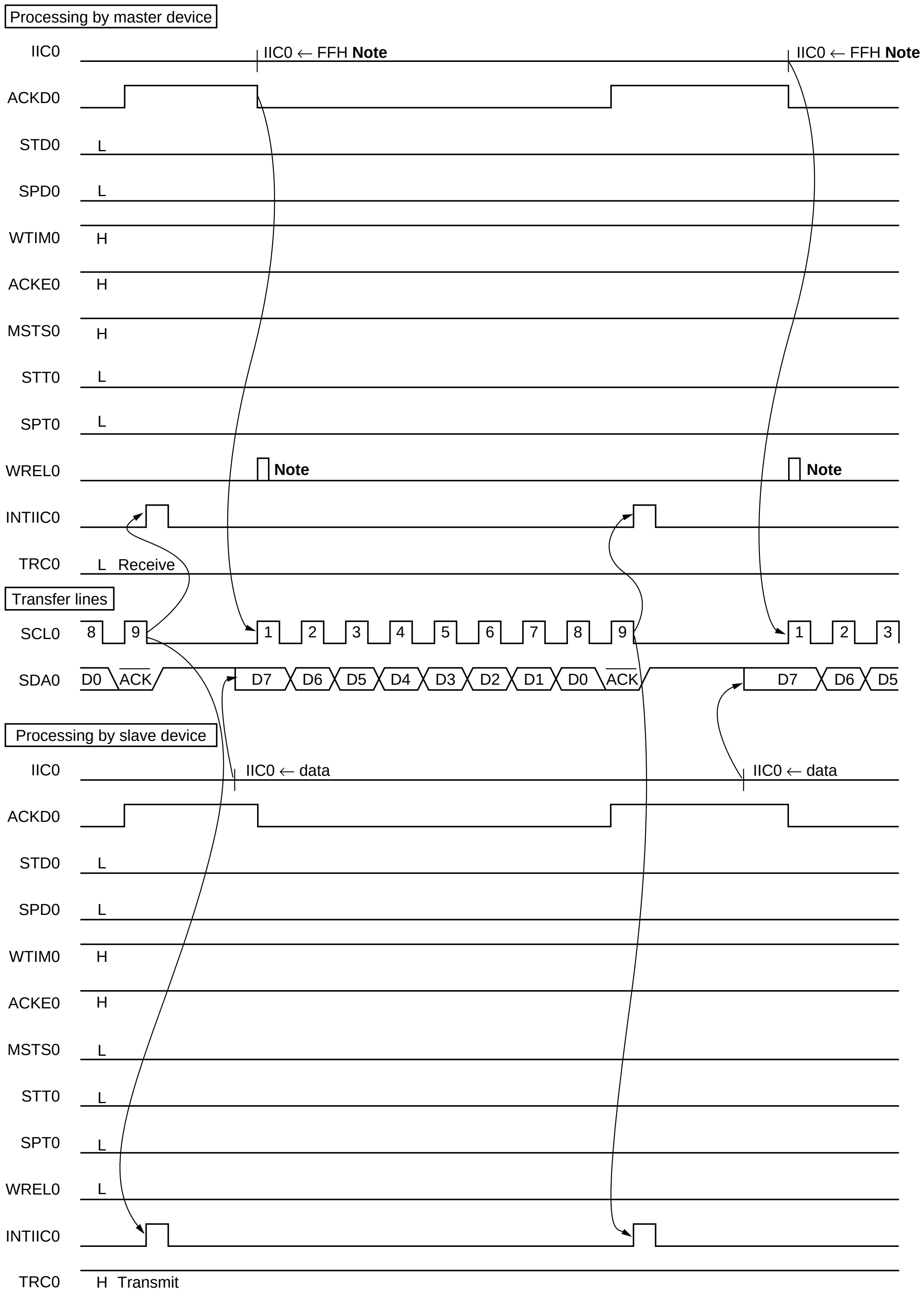
(1) Start condition ~ address



Note To cancel slave wait, write “FFH” to IIC0 or set WREL0.

Figure 18-21. Example of Slave to Master Communication (When 9-clock Wait Is Selected for Both Master and Slave) (2/3)

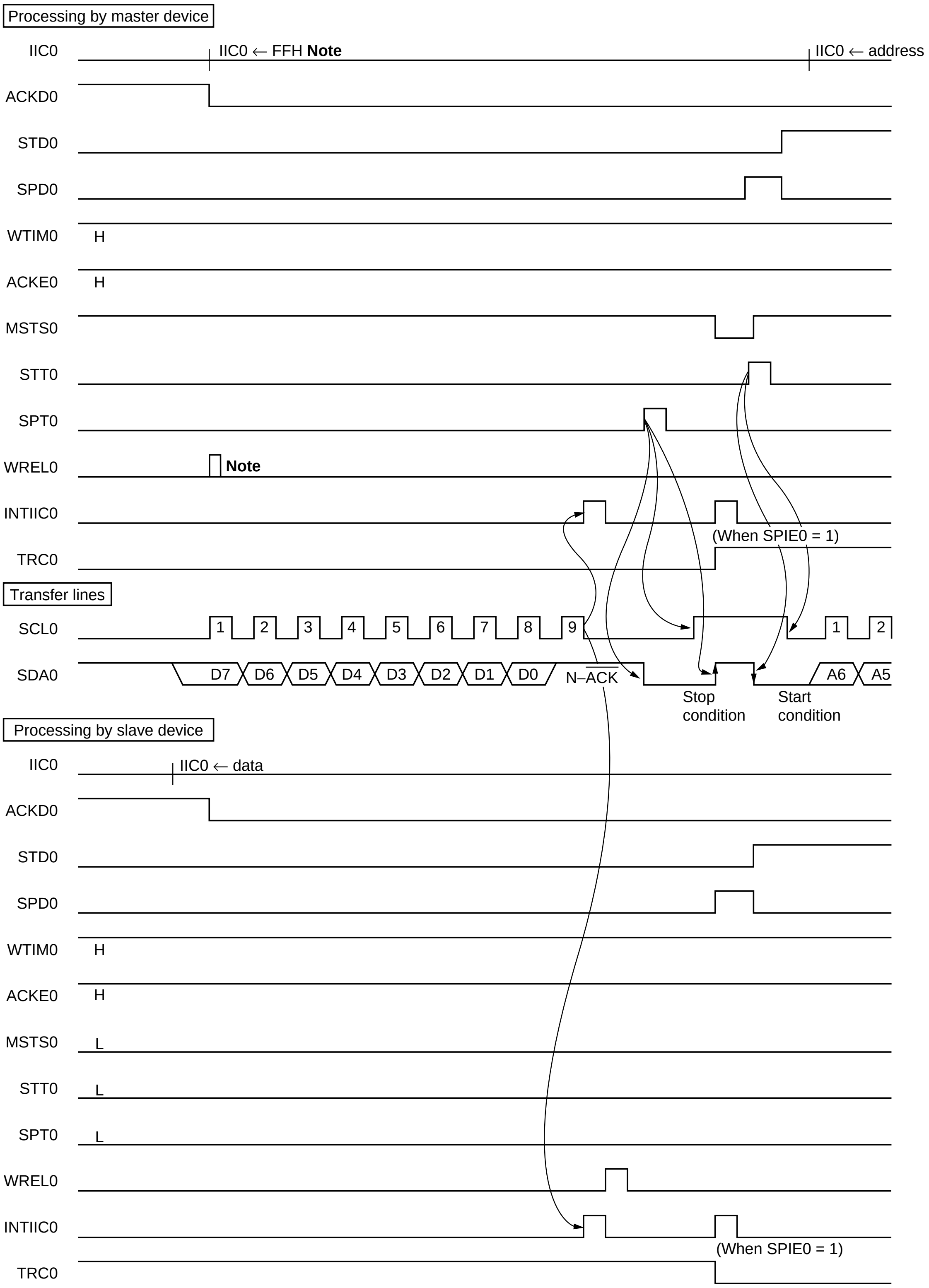
(2) Data



Note To cancel slave wait, write “FFH” to IIC0 or set WREL0.

Figure 18-21. 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 19 INTERRUPT FUNCTIONS

19.1 Interrupt Function Types

The following three types of interrupt functions 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.

One interrupt from the watchdog timer is incorporated as a non-maskable interrupt.

(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).

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 19-1**).

A standby release signal is generated.

Five external interrupt requests and 18 internal interrupt requests (19 internal interrupt requests for μPD780078Y subseries) are incorporated as maskable interrupts.

(3) Software interrupt

This is a vectored interrupt to be generated by executing the BRK instruction. It is acknowledged even in a disabled stated. The software interrupt does not undergo interrupt priority control.

19.2 Interrupt Sources and Configuration

A total of 25 interrupt sources (26 interrupt sources for μPD780078Y) exist among non-maskable, maskable, and software interrupts (see **Table 19-1**).

Remark As the watchdog timer interrupt source (INTWDT), a non-maskable interrupt or maskable interrupt (internal) can be selected.

Table 19-1. Interrupt Source List (1/2)

Type of	Default	Interrupt Source		Internal/	Vector Table	Basic Configuration
Interrupt	Priority[Note 1]	Name	Trigger	External	Address	Type[Note 2]
Non-maskable	—	INTWDT	Watchdog timer overflow (non-maskable interrupt selected)	Internal	0004H	(A)
Maskable	0	INTWDT	Watchdog timer overflow (interval timer mode selected)			(B)
	1	INTP0	Pin input edge detection	External	0006H	(C)
	2	INTP1			0008H	
	3	INTP2			000AH	
	4	INTP3			000CH	
	5	INTSER0	Generation of UART0 reception error	Internal	000EH	(B)
	6	INTSR0	End of UART0 reception		0010H	
	7	INTST0	End of UART0 transmission		0012H	
	8	INTCSI1	End of CSI1 transfer		0014H	
	9	INTCSI3	End of SIO3 transfer		0016H	
	10	INTIIC0[Note 3]	End of serial interface IIC0 transfer		0018H	
	11	INTWTI	Reference time interval signal from watch timer		001AH	
	12	INTTM000	Coincidence of TM00 and CR000 (when compare register is specified) Detection of valid edge of TI010 (when capture register is specified)		001CH	
	13	INTTM010	Coincidence of TM00 and CR010 (when compare register is specified) Detection of valid edge of TI000 (when capture register is specified)		001EH	
	14	INTTM50	Coincidence of TM50 and CR50		0020H	
	15	INTTM51	Coincidence of TM51 and CR51		0022H	
	16	INTAD0	End of conversion by A/D converter		0024H	
	17	INTWT	Watch timer overflow		0026H	
	18	INTKR	Falling edge detection of port 4	External	0028H	(D)

Notes 1. The default priority is a priority order when two or more maskable interrupt requests are generated simultaneously. 0 is the highest order and 23, the lowest.

2. Basic configuration types (A) to (E) correspond to (A) to (E) in Figure 19-1, respectively.

3. μPD780078Y subseries only.

Table 19-1. Interrupt Source List (2/2)

Type of Interrupt	Default Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Maskable	19	INTSER2	Generation of UART2 reception error	Internal	002AH	(B)
	20	INTSR2	End of UART2 reception		002CH	
	21	INTST2	End of UART2 transmission/data transfer[Note 3]		002EH	
	22	INTTM001	Coincidence of TM01 and CR001 (when compare register specified) Detection of TI011 valid edge (when capture register specified)		0030H	
	23	INTTM011	Coincidence of TM01 and CR011 (when compare register specified) Detection of TI001 valid edge (when capture register specified)		0032H	
Software	—	BRK	BRK instruction execution	—	003EH	(E)

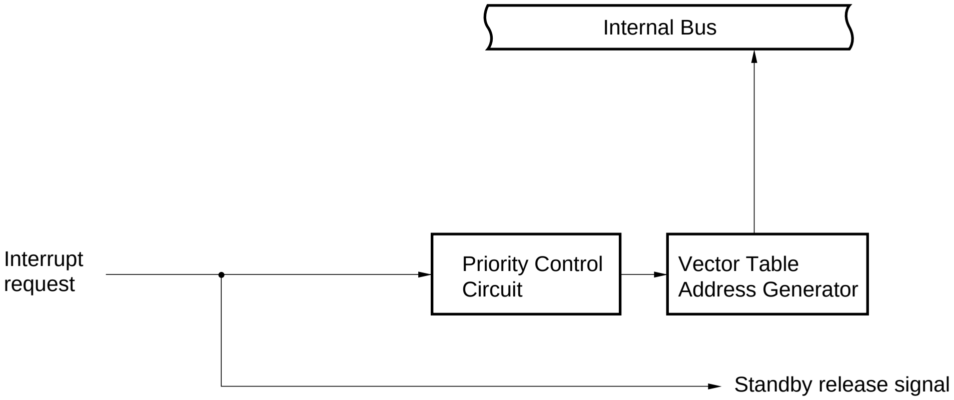
Notes 1. The default priority is a priority order when two or more maskable interrupt requests are generated simultaneously. 0 is the highest order and 23, the lowest.

2. Basic configuration types (A) to (E) correspond to (A) to (E) in Figure 19-1, respectively.

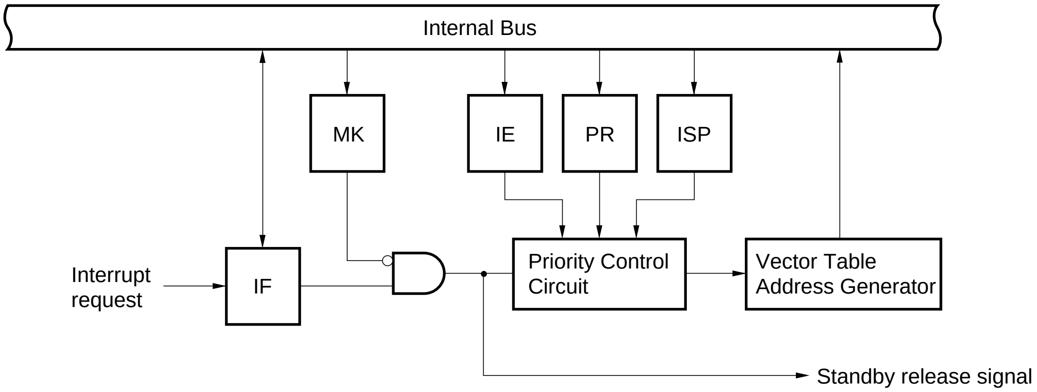
3. This source generates an interrupt request signal during data transfer from the transmit buffer register 2 (TXB2) to the transmit shift register. Interrupt sources can be selected by the transmit interrupt signal select flag (ISMD).

Figure 19-1. Basic Configuration of Interrupt Function (1/2)

(A) Internal non-maskable interrupt



(B) Internal maskable interrupt



(C) External maskable interrupt (INTP0 to INTP3)

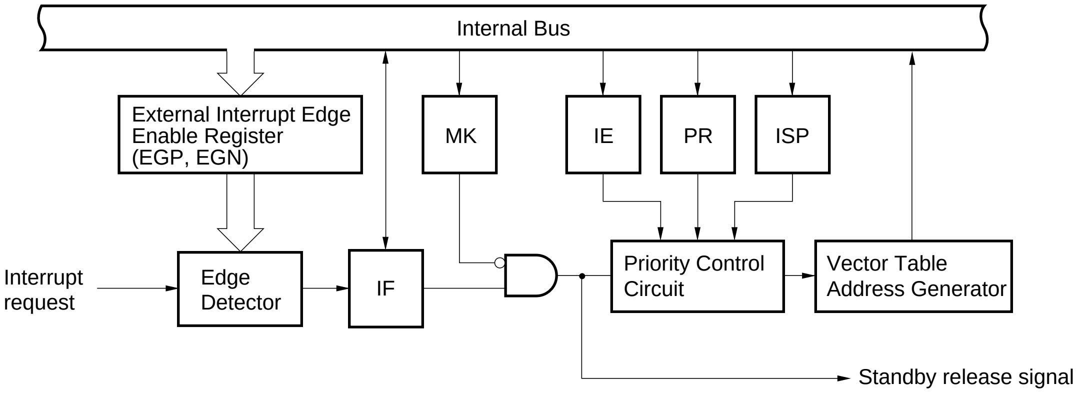
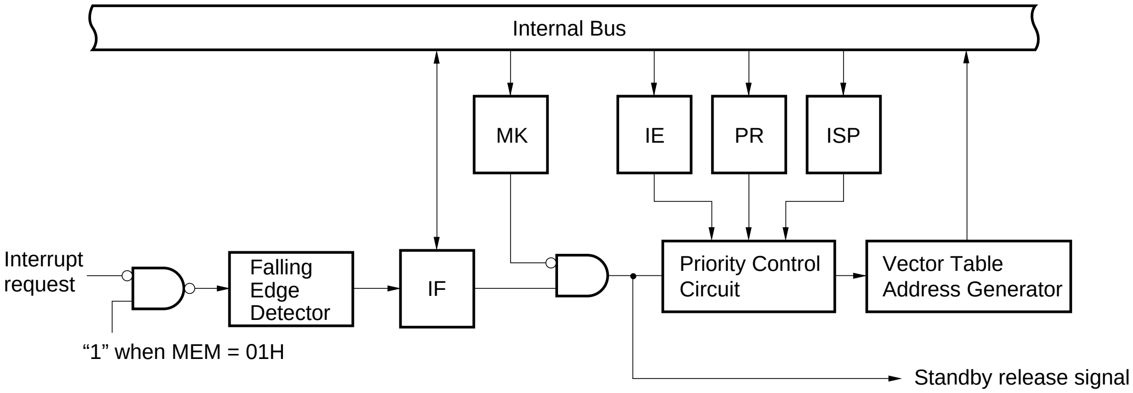
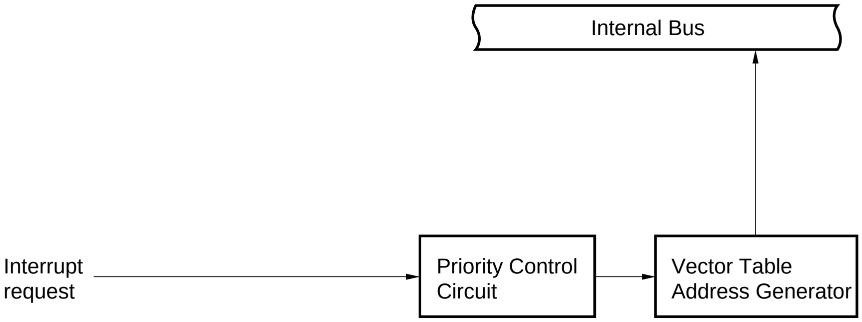


Figure 19-1. Basic Configuration of Interrupt Function (2/2)

(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
MEM	:	Memory expansion mode register

19.3 Interrupt Function Control Registers

The following 6 types of registers are used to control the interrupt functions.

- Interrupt request flag register (IF0L, IF0H, IF1L)
- Interrupt mask flag register (MK0L, MK0H, MK1L)
- Priority specification flag register (PR0L, PR0H, PR1L)
- External interrupt rising edge enable flag (EGP)
- External interrupt falling edge enable flag (EGN)
- Program status word (PSW)

Table 19-2 gives a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 19-2. Flags Corresponding to Interrupt Request Sources

Interrupt Request	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
		Register		Register		Register
INTWDT INTP0 INTP1 INTP2 INTP3 INTSER0 INTSR0 INTST0	WDTIF[Note 1] PIF0 PIF1 PIF2 PIF3 SERIF0 SRIF0 STIF0	IF0L	WDTMK[Note 1] PMK0 PMK1 PMK2 PMK3 SERMK0 SRMK0 STMK0	MK0L	WDTPR[Note 1] PPR0 PPR1 PPR2 PPR3 SERPR0 SRPR0 STPR0	PR0L
INTCSI1 INTCSI3 INTIIC0[Note 2] INTWTI INTTM000 INTTM010 INTTM50 INTTM51	CSIIF1 CSIIF3 IICIF0[Note 2] WTIIF0 TMIF000 TMIF010 TMIF50 TMIF51	IF0H	CSIMK1 CSIMK3 IICMK0[Note 2] WTIMK0 TMMK000 TMMK010 TMMK50 TMMK51	MK0H	CSIPR1 CSIPR3 IICPR0[Note 2] WTIPR0 TMPR000 TMPR010 TMPR50 TMPR51	PR0H
INTAD0 INTWT INTKR INTSER2 INTSR2 INTST2 INTTM001 INTTM011	ADIF0 WTIF KRIF SERIF2 SRIF2 STIF2 TMIF001 TMIF011	IF1L	ADMK0 WTMK KRMK SERMK2 SRMK2 STMK2 TMMK001 TMMK011	MK1L	ADPR0 WTPR KRPR SERPR2 SRPR2 STPR2 TMPR001 TMPR011	PR1L

Notes 1. Interrupt control flag when watchdog timer is used as interval timer

2. μPD780078Y Subseries only

(1) Interrupt request flag registers (IF0L, IF0H, IF1L)

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, and IF1L are set by a 1-bit or 8-bit memory manipulation instruction. When IF0L and IF0H are combined to form 16-bit register IF0, they are read with a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the values of IF0L, IF0H, and IF1L to 00H.

Figure 19-2. Interrupt Request Flag Register (IF0L, IF0H, IF1L) Format

Address: FFE0H At Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IF0L	STIF0	SRIF0	SERIF0	PIF3	PIF2	PIF1	PIF0	WDTIF

Address: FFE1H At Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IF0H	TMIF51	TMIF50	TMIF010	TMIF000	WTIIF0	IICIF0[Note]	CSIIF3	CSIIF1

Address: FFE2H At Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IF1L	TMIF011	TMIF001	STIF2	SRIF2	SERIF2	KRIF	WTIF	ADIF0

XXIFX	Interrupt Request Flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Note Incorporated only in the μPD780078Y Subseries. Be sure to set 0 for the μPD780078 Subseries.

Cautions 1. The WDTIF flag is R/W enabled only when the watchdog timer is used as the interval timer. If watchdog timer mode 1 is used, set the WDTIF flag to 0.

2. When operating a timer, serial interface, or A/D converter after stand-by release, run it once after clearing an interrupt request flag. An interrupt request flag may be set by noise.

(2) Interrupt mask flag registers (MK0L, MK0H, MK1L)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt service and to set standby clear enable/disable.

MK0L, MK0H, and MK1L are set by a 1-bit or 8-bit memory manipulation instruction. When MK0L and MK0H are combined to form a 16-bit register, they are set with a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the values of MK0L, MK0H, and MK1L to FFH.

Figure 19-3. Interrupt Mask Flag Register (MK0L, MK0H, MK1L) Format

Address: FFE4H At Reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
MK0L	STMK0	SRMK0	SERMK0	PMK3	PMK2	PMK1	PMK0	WDTMK

Address: FFE5H At Reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
MK0H	TMMK51	TMMK50	TMMK010	TMMK000	WTIMK0	IICMK0[Note]	CSIMK3	CSIMK1

Address: FFE6H At Reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
MK1L	TMMK011	TMMK001	STMK2	SRMK2	SERMK2	KRMK	WTMK	ADMK0

XXMKX	Interrupt Servicing Control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Note Incorporated only in the μPD780078Y Subseries. Be sure to set 1 for the μPD780078 Subseries.

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 have an alternate function as external interrupt request input, when the output level is changed by specifying the output mode of the port function, an interrupt request flag is set. Therefore, 1 should be set in the interrupt mask flag before using the output mode.

(3) Priority specification flag registers (PR0L, PR0H, PR1L)

The priority specification flag registers are used to set the corresponding maskable interrupt priority orders.
PR0L, PR0H, and PR1L are set by a 1-bit or 8-bit memory manipulation instruction. If PR0L and PR0H are combined to form 16-bit register PR0, they are set with a 16-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets the values of PR0L, PR0H, and PR1L to FFH.

Figure 19-4. Priority Specification Flag Register (PR0L, PR0H, PR1L) Format

Address: FFE8H At Reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PR0L	STPR0	SRPR0	SERPR0	PPR3	PPR2	PPR1	PPR0	WDTPR

Address: FFE9H At Reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PR0H	TMPR51	TMPR50	TMPR010	TMPR000	WTIPR0	IICPR0^Note	CSIPR3	CSIPR1

Address: FFEAH At Reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PR1L	TMPR011	TMPR001	STPR2	SRPR2	SERPR2	KRPR	WTPR	ADPR0

XXPRX	Priority Level Selection
0	High priority level
1	Low priority level

Note Incorporated only in the μPD780078Y Subseries. Be sure to set 1 for the μPD780078 Subseries.

Caution When the watchdog timer is used in the watchdog timer 1 mode, set 1 in the WDTPR flag.

(4) External interrupt rising edge enable register (EGP), External interrupt falling edge enable register (EGN)

These registers specify the valid edge for INTP0 to INTP3.

EGP and EGN are set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the values of these registers to 00H.

Figure 19-5. External Interrupt Rising Edge Enable Register (EGP), External Interrupt Falling Edge Enable Register (EGN) Format

Address: FF48H At Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP	0	0	0	0	EGP3	EGP2	EGP1	EGP0

Address: FF49H At Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN	0	0	0	0	EGN3	EGN2	EGN1	EGN0

EGPn	EGNn	INTPn Pin Valid Edge Selection (n = 0 to 3)
0	0	Interrupt disable
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

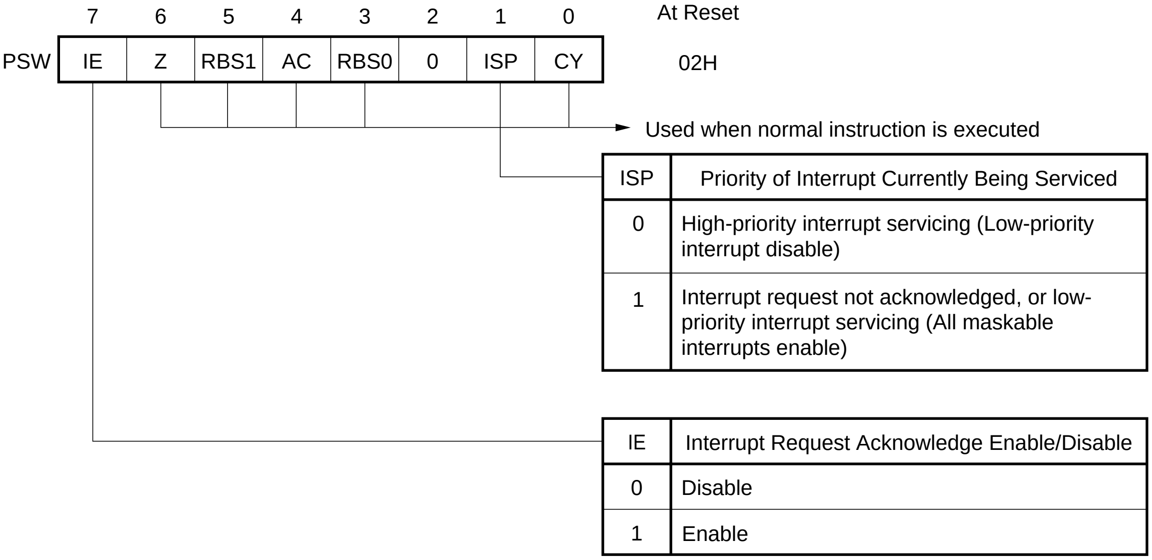
(5) Program status word (PSW)

The program status word is a register 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 nesting processing are mapped.

Besides 8-bit read/write, this register can carry out operations with a bit manipulation instruction and dedicated instructions (EI and DI). When a vectored interrupt request is acknowledged, if the BRK instruction is executed, the contents of PSW are automatically saved into a stack and the IE flag is reset to 0. If a maskable interrupt request is acknowledged, the contents of the priority specification flag of the acknowledged interrupt are transferred to the ISP flag. The PSW contents are also saved into the stack with the PUSH PSW instruction. They are reset from the stack with the RETI, RETB, and POP PSW instructions.

$\overline{\text{RESET}}$ input sets the value of PSW to 02H.

Figure 19-6. Program Status Word Format



	7	6	5	4	3	2	1	0	At Reset
PSW	IE	Z	RBS1	AC	RBS0	0	ISP	CY	02H

Used when normal instruction is executed

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	Disable
1	Enable

19.4 Interrupt Servicing Operations

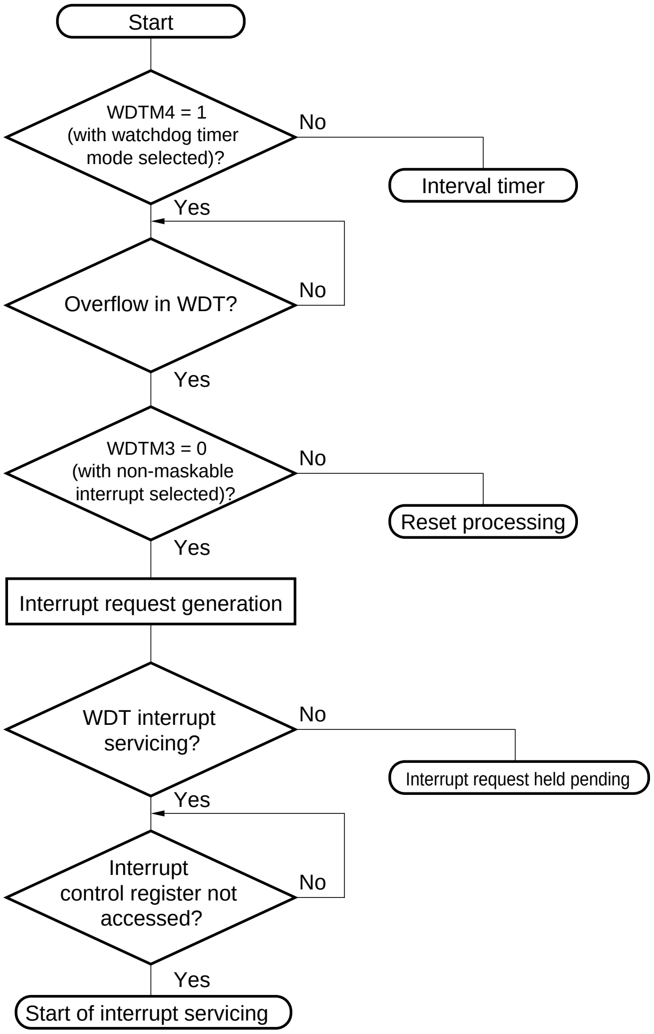
19.4.1 Non-maskable interrupt request acknowledge operation

A non-maskable interrupt request is unconditionally acknowledged even if in an interrupt acknowledge disable state. It does not undergo interrupt priority control and has highest priority over all other interrupts.

If a non-maskable interrupt request is acknowledged, the contents are saved into the stacks 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 PC and branched.

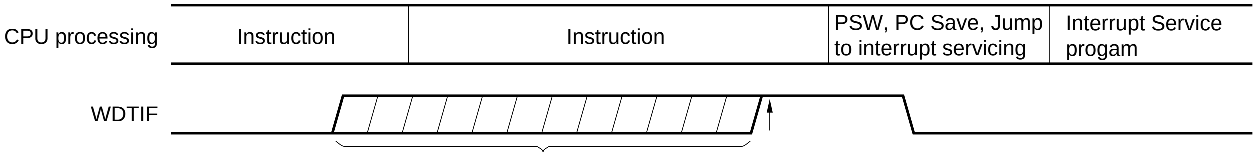
A new non-maskable interrupt request generated during execution of a non-maskable interrupt servicing program is acknowledged after the current execution of the non-maskable interrupt servicing program is terminated (following RETI instruction execution) and one main routine instruction is executed. 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 19-7, 19-8, and 19-9 show the flowchart of the non-maskable interrupt request generation through acknowledge, acknowledge timing of non-maskable interrupt request, and acknowledge operation at multiple non-maskable interrupt request generation, respectively.

Figure 19-7. Non-Maskable Interrupt Request Generation to Acknowledge Flowchart



WDTM: Watchdog timer mode register
WDT : Watchdog timer

Figure 19-8. Non-Maskable Interrupt Request Acknowledge Timing

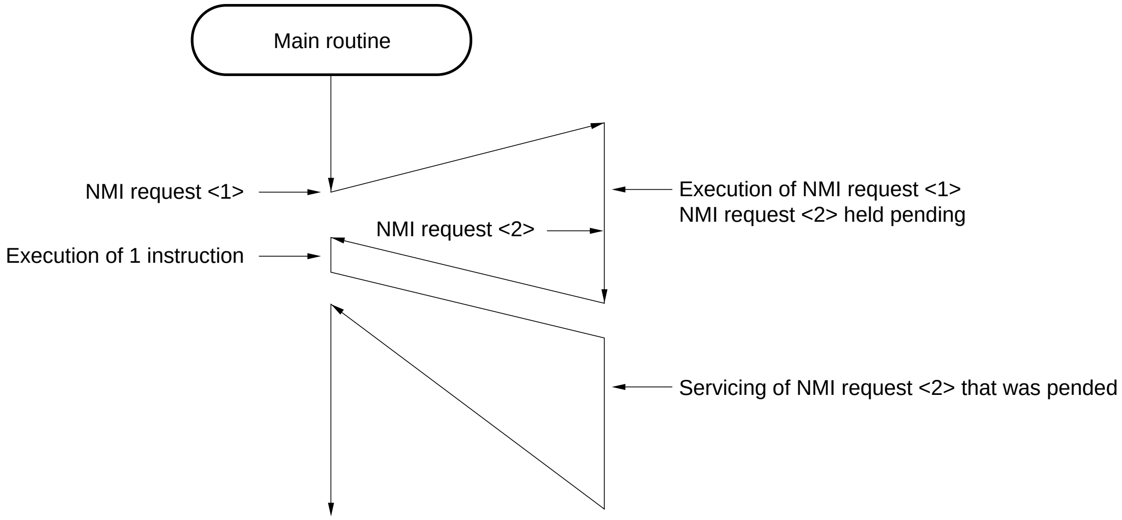


Interrupt request generated during this interval is acknowledged at ↑ .

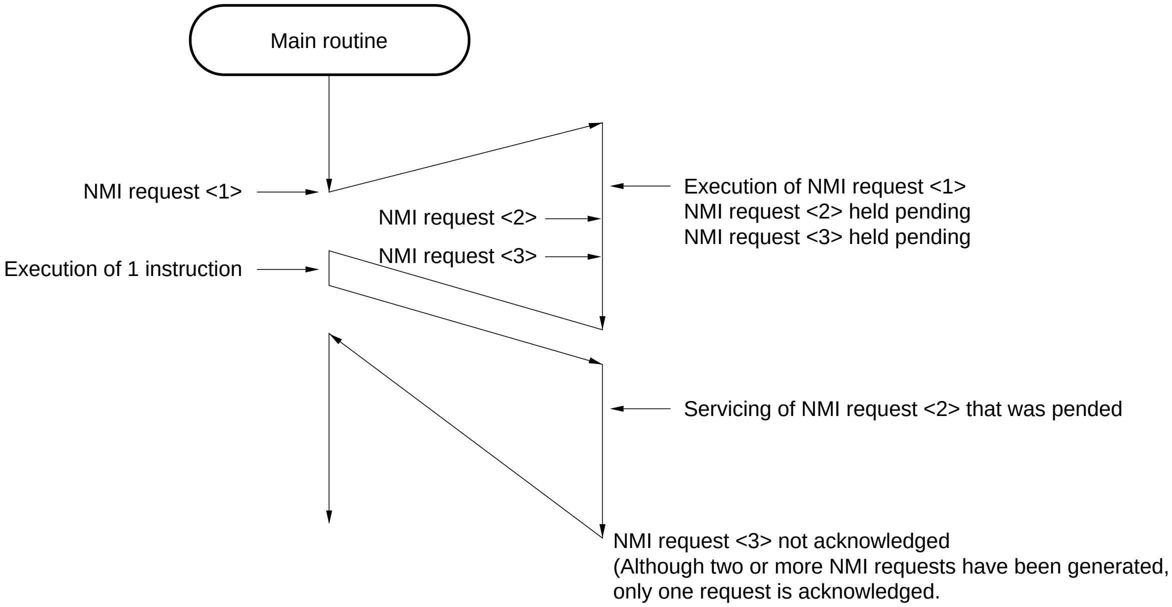
WDTIF: Watchdog timer interrupt request flag

Figure 19-9. Non-Maskable Interrupt Request Acknowledge 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 pended

(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 pended
NMI request <3> not acknowledged
(Although two or more NMI requests have been generated,
only one request is acknowledged.

19.4.2 Maskable interrupt acknowledge operation

A maskable interrupt becomes acknowledgeable when an interrupt request flag is set to 1 and the mask (MK) flag corresponding to that interrupt request is cleared to 0. A vectored interrupt request is acknowledged if in the interrupt enable state (when IE flag is set to 1). However, a low-priority interrupt request is not acknowledged during servicing of a higher priority interrupt request (when the ISP flag is reset to 0). The times from generation of a maskable interrupt request until interrupt servicing is performed are listed in Table 19-3 below.

For the interrupt request acknowledge timing, see **Figures 19-11** and **19-12**.

Table 19-3. Times 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 in the priority specification flag is acknowledged first. If two or more interrupts requests have the same priority level, the request with the highest default priority is acknowledged first.

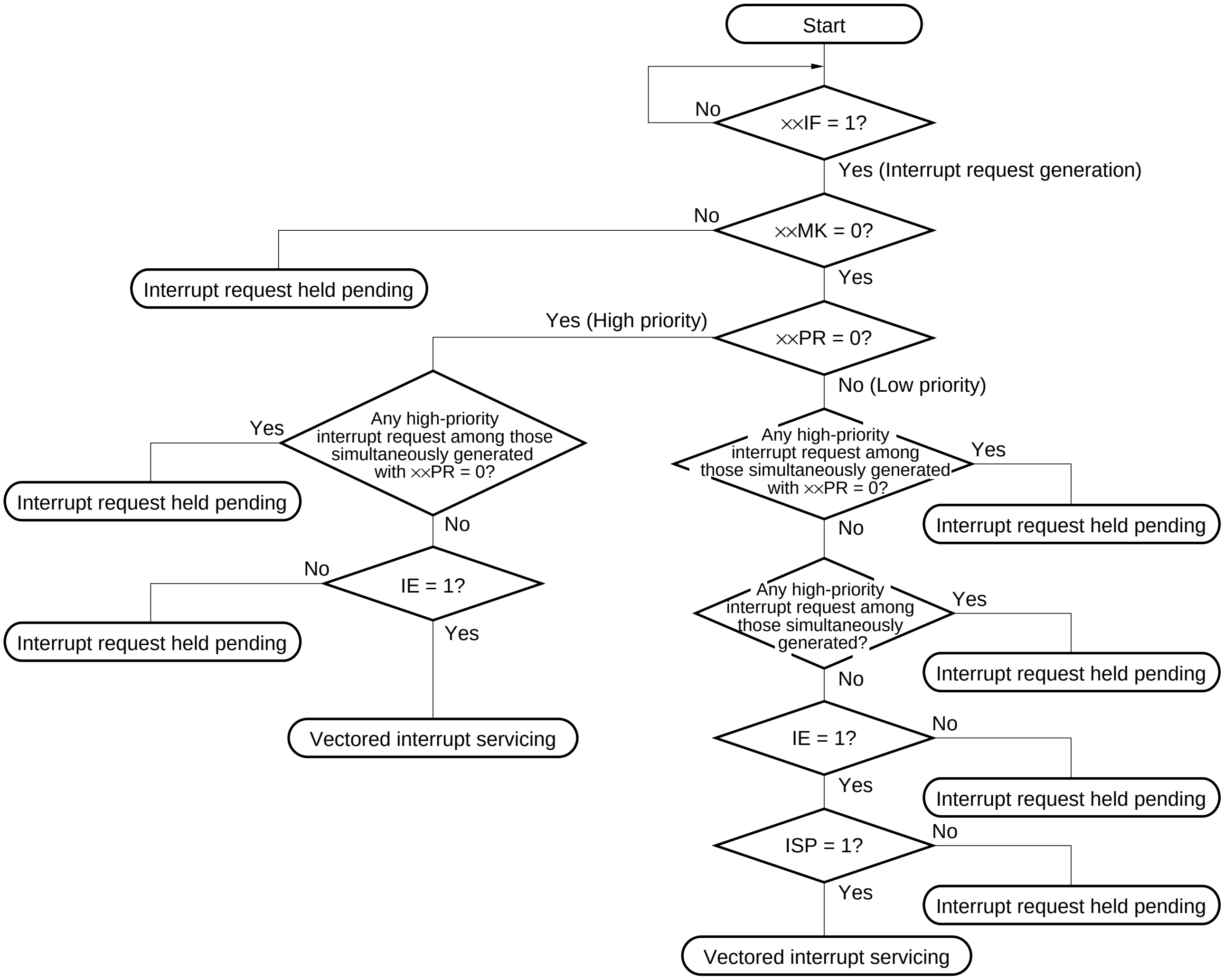
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 19-10 shows the interrupt request acknowledge algorithm.

If a maskable interrupt request is acknowledged, the contents are saved into the stacks in the order of PSW, then PC, the IE flag is reset (0), and the contents of the priority specification flag corresponding to the acknowledged interrupt are transferred to the ISP flag. Further, the vector table data determined for each interrupt request is loaded into PC and branched.

Return from an interrupt is possible with the RETI instruction.

Figure 19-10. Interrupt Request Acknowledge 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

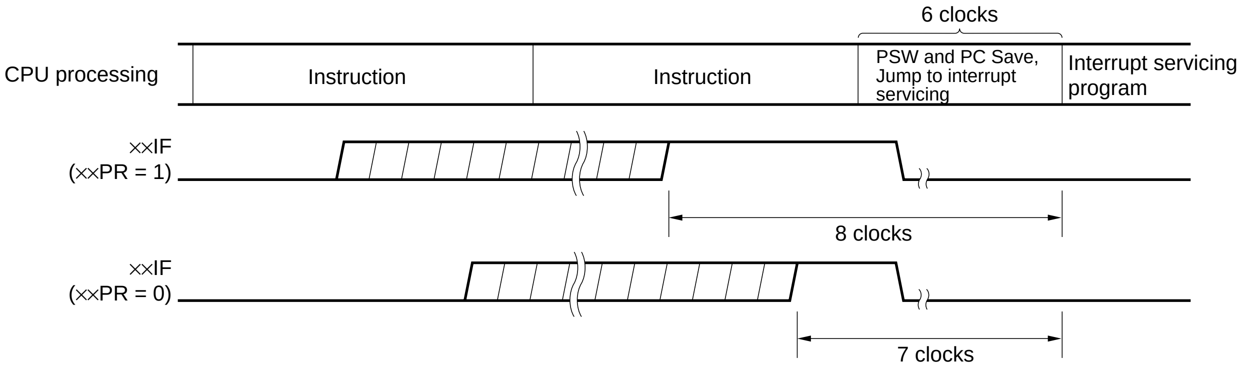
××MK : Interrupt mask flag

××PR : Priority specification flag

IE : Flag that controls acknowledge of maskable interrupt request (1 = Enable, 0 = Disable)

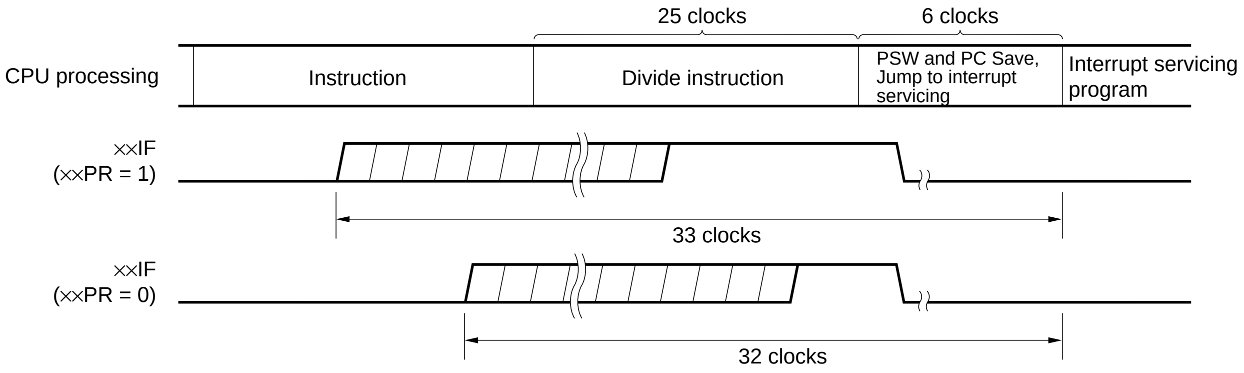
ISP : Flag that indicates the priority level of the interrupt currently being serviced (0 = high-priority interrupt servicing, 1 = No interrupt request received, or low-priority interrupt servicing)

Figure 19-11. Interrupt Request Acknowledge Timing (Minimum Time)



Remark 1 clock: 1/f_{CPU} (f_{CPU}: CPU clock)

Figure 19-12. Interrupt Request Acknowledge Timing (Maximum Time)



Remark 1 clock: 1/f_{CPU} (f_{CPU}: CPU clock)

19.4.3 Software interrupt request acknowledge operation

A software interrupt acknowledge is acknowledged by BRK instruction execution. Software interrupts cannot be disabled.

If a software interrupt request is acknowledged, the contents are saved into the stacks in the order of the program status word (PSW), then program counter (PC), the IE flag is reset (0), and the contents of the vector table (003EH, 003FH) are loaded into PC and branched.

Return from a software interrupt is possible with the RETB instruction.

Caution Do not use the RETI instruction for returning from the software interrupt.

19.4.4 Nesting interrupt servicing

Nesting occurs when another interrupt request is acknowledged during execution of an interrupt.

Nesting does 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 requests acknowledge becomes disabled (IE = 0). Therefore, to enable nesting, it is necessary to set (1) the IE flag with the EI instruction during interrupt servicing to enable interrupt acknowledge.

Moreover, even if interrupts are enabled, nesting 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 nesting.

In the interrupt enable 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 nesting. 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 nesting. Interrupt requests that are not enabled because of the interrupt disable state or they have a lower priority are held pending. When servicing of the current interrupt ends, the pended interrupt request is acknowledged following execution of one main processing instruction execution.

Nesting is not possible during non-maskable interrupt servicing.

Table 19-4 shows interrupt requests enabled for nesting and Figure 19-13 shows nesting examples.

Table 19-4. Interrupt Request Enabled for Nesting during Interrupt Servicing

Nesting 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. ○ : Nesting enable

2. × : Nesting disable

3. ISP and IE are flags contained in 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 acknowledge is disabled.

IE = 1 : Interrupt request acknowledge is enabled.

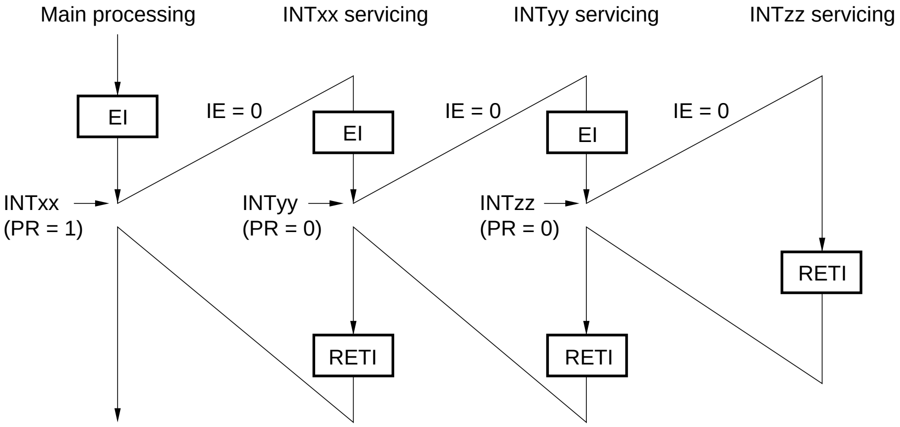
4. PR is a flag contained in PR0L, PR0H, and PR1L.

PR = 0 : Higher priority level

PR = 1 : Lower priority level

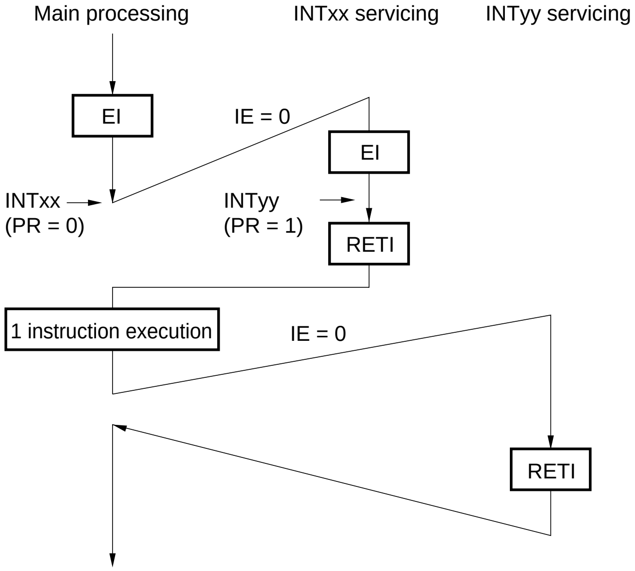
Figure 19-13. Nesting Examples (1/2)

Example 1. Nesting occurs twice



During servicing of interrupt INTxx, two interrupt requests, INTyy and INTzz, are acknowledged, and nesting takes place. Before each interrupt request is acknowledged, the EI instruction must always be issued to enable interrupt request acknowledge.

Example 2. Nesting does not occur due to priority control

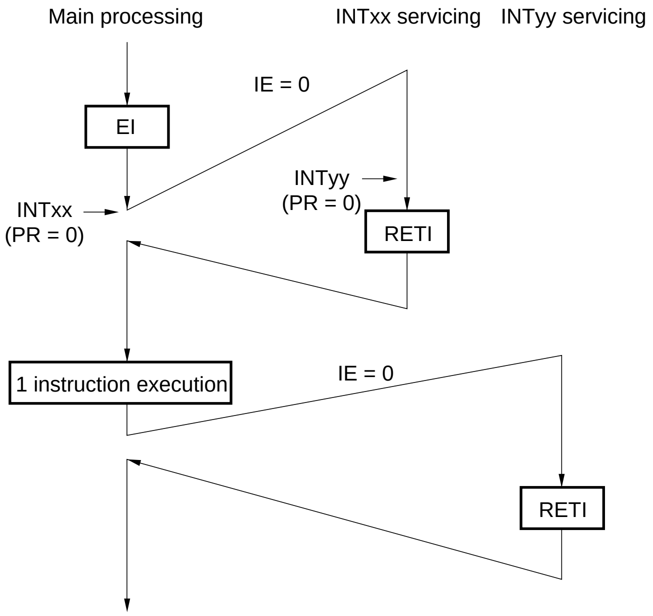


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and nesting does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 0	:	Higher priority level
PR = 1	:	Lower priority level
IE = 0	:	Interrupt request acknowledge disable

Figure 19-13. Nesting Examples (2/2)

Example 3. Nesting does not occur because interrupt is not enabled



Interrupt is not enabled during servicing of interrupt INTxx (EI instruction is not issued), therefore, interrupt request INTyy is not acknowledged and nesting 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

19.4.5 Interrupt request hold

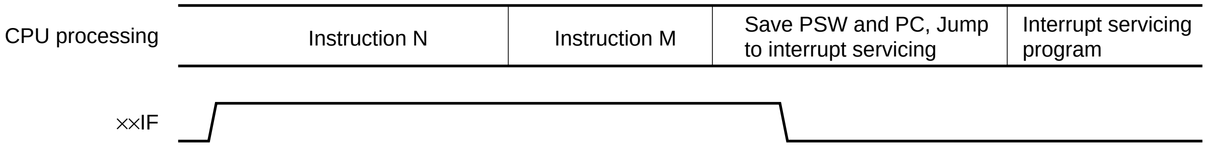
There are instructions where, even if an interrupt request is issued for them while another instruction is executed, request acknowledge is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- 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
- Manipulate instructions for the IF0L, IF0H, IF1L, MK0L, MK0H, MK1L, PR0L, PR0H, and PR1L 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 requests is generated during execution of the BRK instruction, the interrupt request is not acknowledged. However, a non-maskable interrupt request is acknowledged.

Figure 19-14 shows the timing with which interrupt requests are held pending.

Figure 19-14. 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)

[MEMO]

CHAPTER 20 EXTERNAL DEVICE EXPANSION FUNCTION

20.1 External Device Expansion Function

The external device expansion function connects external devices to areas other than the internal ROM, RAM, and SFR. Connection of external devices uses ports 4 to 6. Ports 4 to 6 control address/data, read/write strobe, wait, address strobe, etc.

Table 20-1. Pin Functions in External Memory Expansion Mode

Pin Function at External Device Connection		Alternate Function
Name	Function	
AD0 to AD7	Multiplexed address/data bus	P40 to P47
A8 to A15	Address bus	P50 to P57
$\overline{RD}$	Read strobe signal	P64
$\overline{WR}$	Write strobe signal	P65
$\overline{WAIT}$	Wait signal	P66
ASTB	Address strobe signal	P67

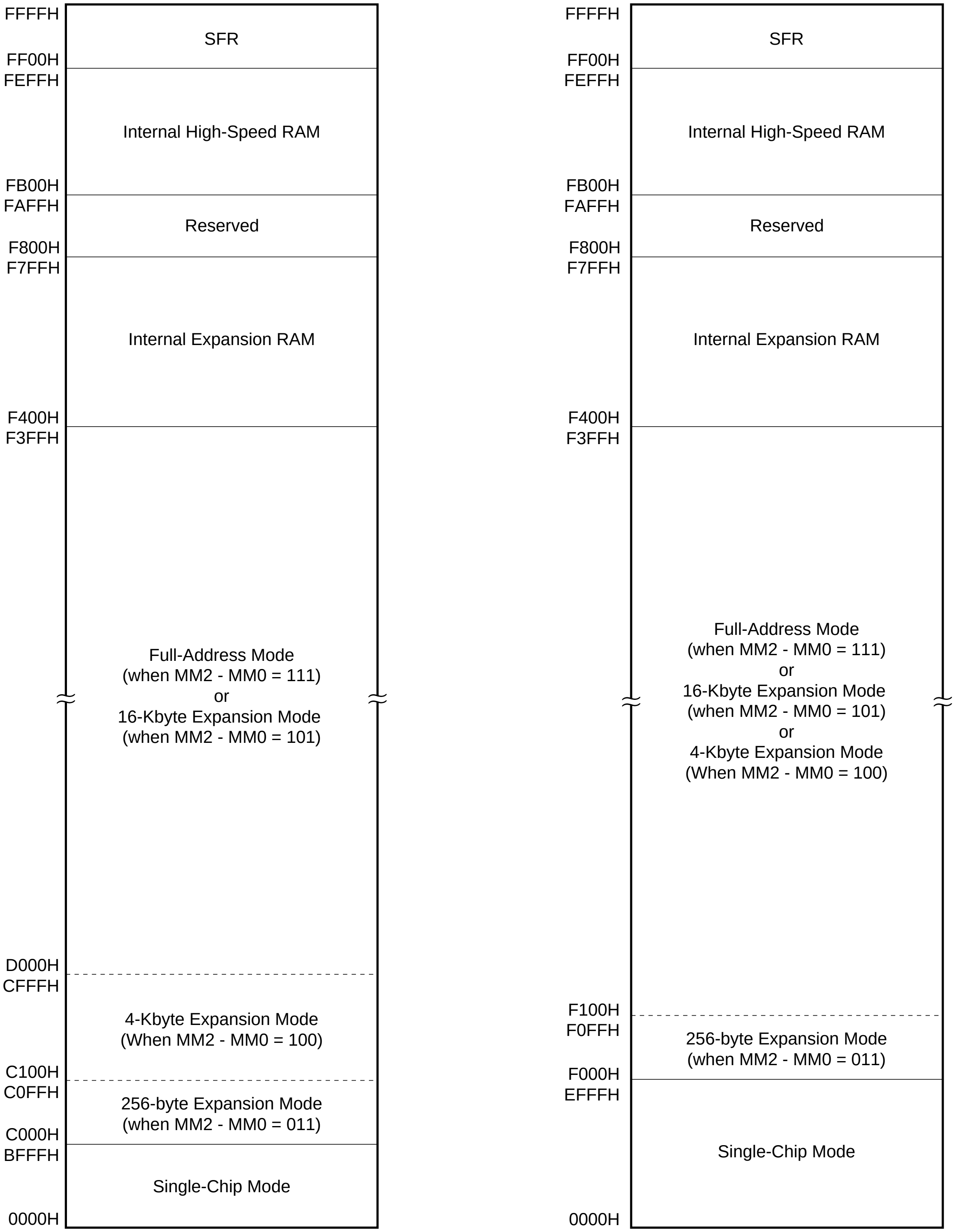
Table 20-2. State of Port 4 to 6 Pins in External Memory Expansion Mode

Port	Port 4	Port 5								Port 6							
External Expansion Mode	0 to 7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Single-chip mode	Port	Port								Port							
256-byte expansion mode	Address/data	Port								Port				$\overline{RD}$, $\overline{WR}$, $\overline{WAIT}$, ASTB			
4-Kbyte expansion mode	Address/data	Address				Port				Port				$\overline{RD}$, $\overline{WR}$, $\overline{WAIT}$, ASTB			
16-Kbyte expansion mode	Address/data	Address						Port		Port				$\overline{RD}$, $\overline{WR}$, $\overline{WAIT}$, ASTB			
Full-address mode	Address/data	Address								Port				$\overline{RD}$, $\overline{WR}$, $\overline{WAIT}$, ASTB			

Caution When the external wait function is not used, the $\overline{WAIT}$ pin can be used as a port in all modes.

The memory maps when the external device expansion function is used are as follows.

Figure 20-1. Memory Map When Using External Device Function



(a) Memory map of μPD780076, 780076Y, and of μPD78F0078, 78F0078Y when flash memory size is 48 KB

(b) Memory map of μPD780078, 780078Y and of μPD78F0078, 78F0078Y when flash memory size is 60 KB

20.2 External Device Expansion Function Control Register

The external device expansion function is controlled by the following two types of registers.

- Memory expansion mode register (MEM)
- Memory expansion wait setting register (MM)

(1) Memory expansion mode register (MEM)

MEM sets the external expansion area.

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

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

Figure 20-2. Memory Expansion Mode Register (MEM) Format

Address: FF47H At Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MEM	0	0	0	0	0	MM2	MM1	MM0

<table>
<tr><th rowspan="2">MM2</th><th rowspan="2">MM1</th><th rowspan="2">MM0</th><th rowspan="2" colspan="2">Single-Chip/Memory Expansion Mode Selection</th><th colspan="5">P40 to P47, P50 to P57, P64 to P67 Pin State</th></tr>
<tr><th>P40 to P47</th><th>P50 to P53</th><th>P54, P55</th><th>P56, P57</th><th>P64 to P67</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td colspan="2">Single-chip mode</td><td colspan="5" rowspan="2">Port mode</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td colspan="2">Port 4 falling edge detection mode</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td rowspan="4">Memory expansion mode</td><td>256-byte mode</td><td rowspan="4">AD0 to AD7</td><td colspan="3">Port mode</td><td rowspan="4">P64 = $\overline{\text{RD}}$
P65 = $\overline{\text{WR}}$
P66 =$\overline{\text{WAIT}}$
P67 = ASTB</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>4-Kbyte mode</td><td rowspan="3">A8 to A11</td><td colspan="2">Port mode</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>16-Kbyte mode</td><td rowspan="2">A12, A13</td><td>Port mode</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Full-address mode[Note]</td><td>A14, A15</td></tr>
<tr><td colspan="3">Other than above</td><td colspan="7">Setting prohibited</td></tr>
</table>

Note The full-address mode allows external expansion to the entire 64-Kbyte address space except for the internal ROM, RAM, SFR areas and the reserved areas.

Caution When using the falling edge detection function of port 4, be sure to set MEM to 01H.

(2) Memory expansion wait setting register (MM)

MM sets the number of waits.

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

$\overline{\text{RESET}}$ input sets the value of MM to 10H.

Figure 20-3. Memory Expansion Wait Setting Register (MM) Format

Address: FFF8H At Reset: 10H R/W

Symbol	7	6	5	4	3	2	1	0
MM	0	0	PW1	PW0	0	0	0	0

PW1	PW0	Wait Control
0	0	No wait
0	1	Wait (one wait state inserted)
1	0	Setting prohibited
1	1	Wait control by external wait pin

Caution To control wait with external wait pin, be sure set WAIT/P66 pin to input mode (set bit 6 (PM66) of port mode register 6 (PM6) to 1).

20.3 External Device Expansion Function Timing

Timing control signal output pins in the external memory expansion mode are as follows.

(1) $\overline{\text{RD}}$ pin (Alternate function: P64)

Read strobe output pin. The read strobe output pin is output in data accesses and instruction fetches from external memory.

During internal memory access, the read strobe signal is not output (maintains high level).

(2) $\overline{\text{WR}}$ pin (Alternate function: P65)

Write strobe signal output pin. The write strobe signal is output in data access to external memory.

During internal memory access, the write strobe signal is not output (maintains high level)

(3) $\overline{\text{WAIT}}$ pin (Alternate function: P66)

External wait signal input pin.

When the external wait is not used, the $\overline{\text{WAIT}}$ pin can be used as an input/output port.

During internal memory access, the external wait signal is ignored.

(4) ASTB pin (Alternate function: P67)

Address strobe signal output pin. The address strobe signal is output regardless of data access and instruction fetch from external memory. During internal memory access, the address strobe signal is not output.

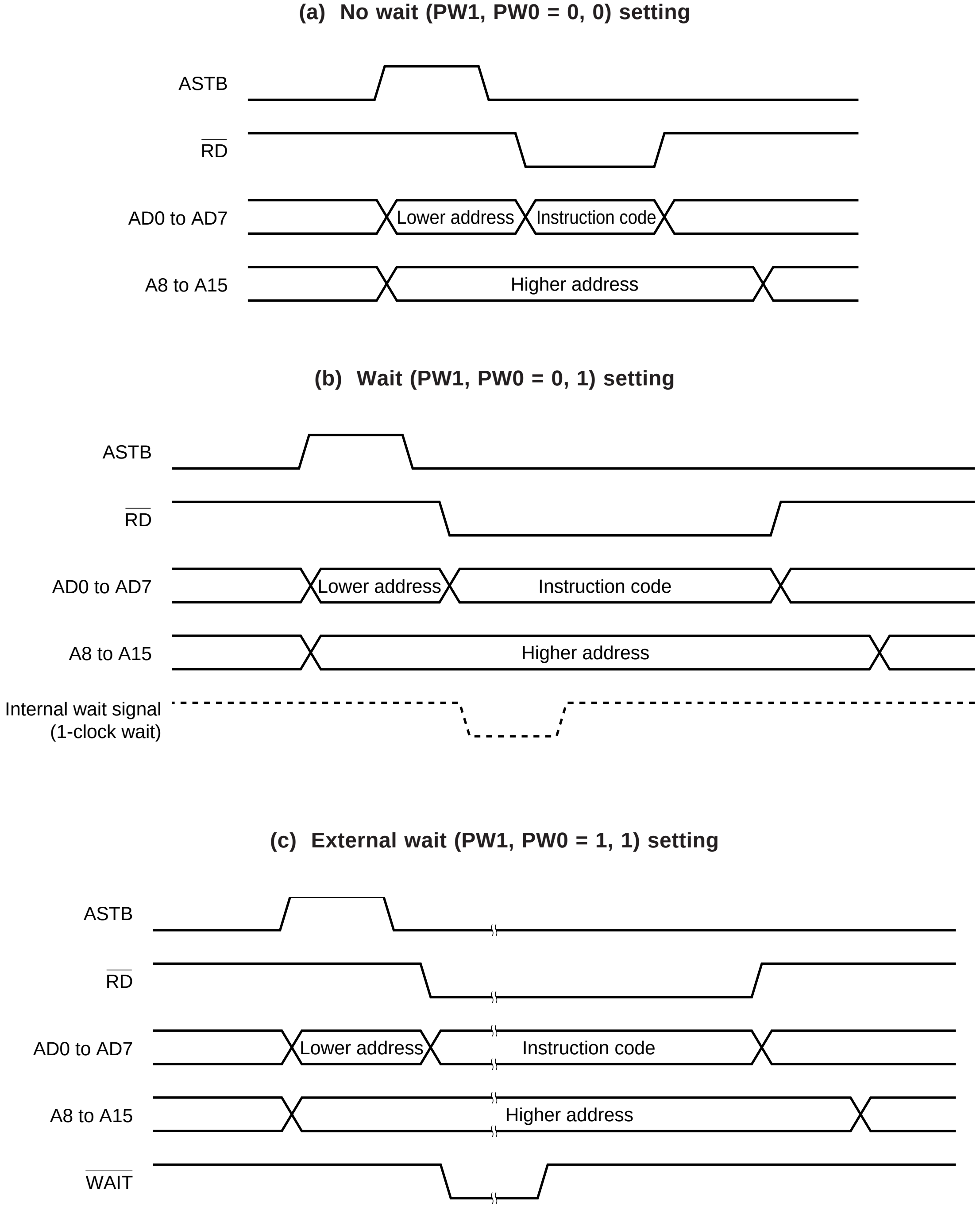
(5) AD0 to AD7, A8 to A15 pins (Alternate function: P40 to P47, P50 to P57)

Address/data signal output pins. Valid signal is output or input during data accesses and instruction fetches from external memory.

These signals change even during internal memory access (output values are undefined).

The timing charts are shown in Figures 20-4 to 20-7.

Figure 20-4. Instruction Fetch from External Memory

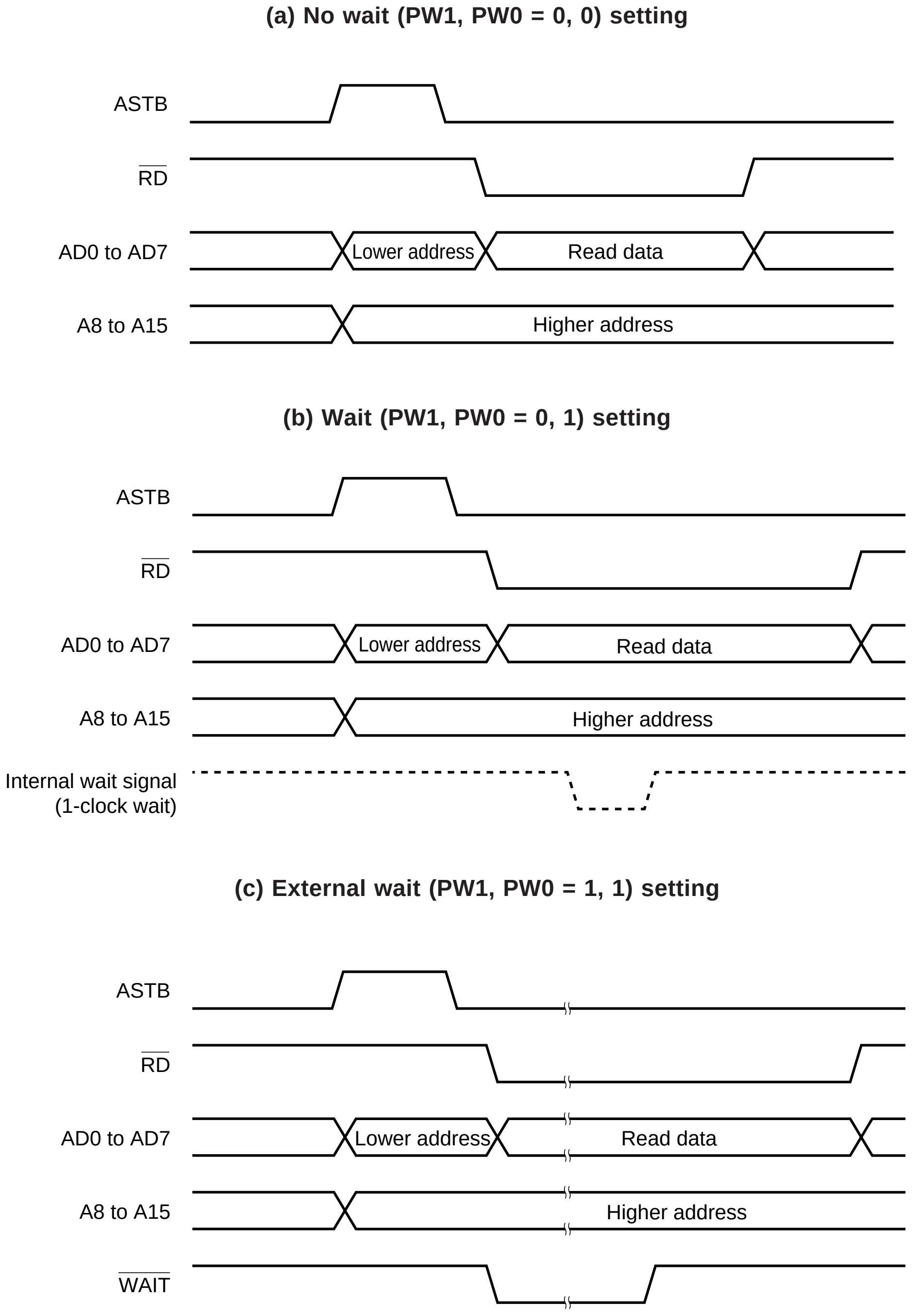


(a) No wait (PW1, PW0 = 0, 0) setting

(b) Wait (PW1, PW0 = 0, 1) setting

(c) External wait (PW1, PW0 = 1, 1) setting

Figure 20-5. External Memory Read Timing



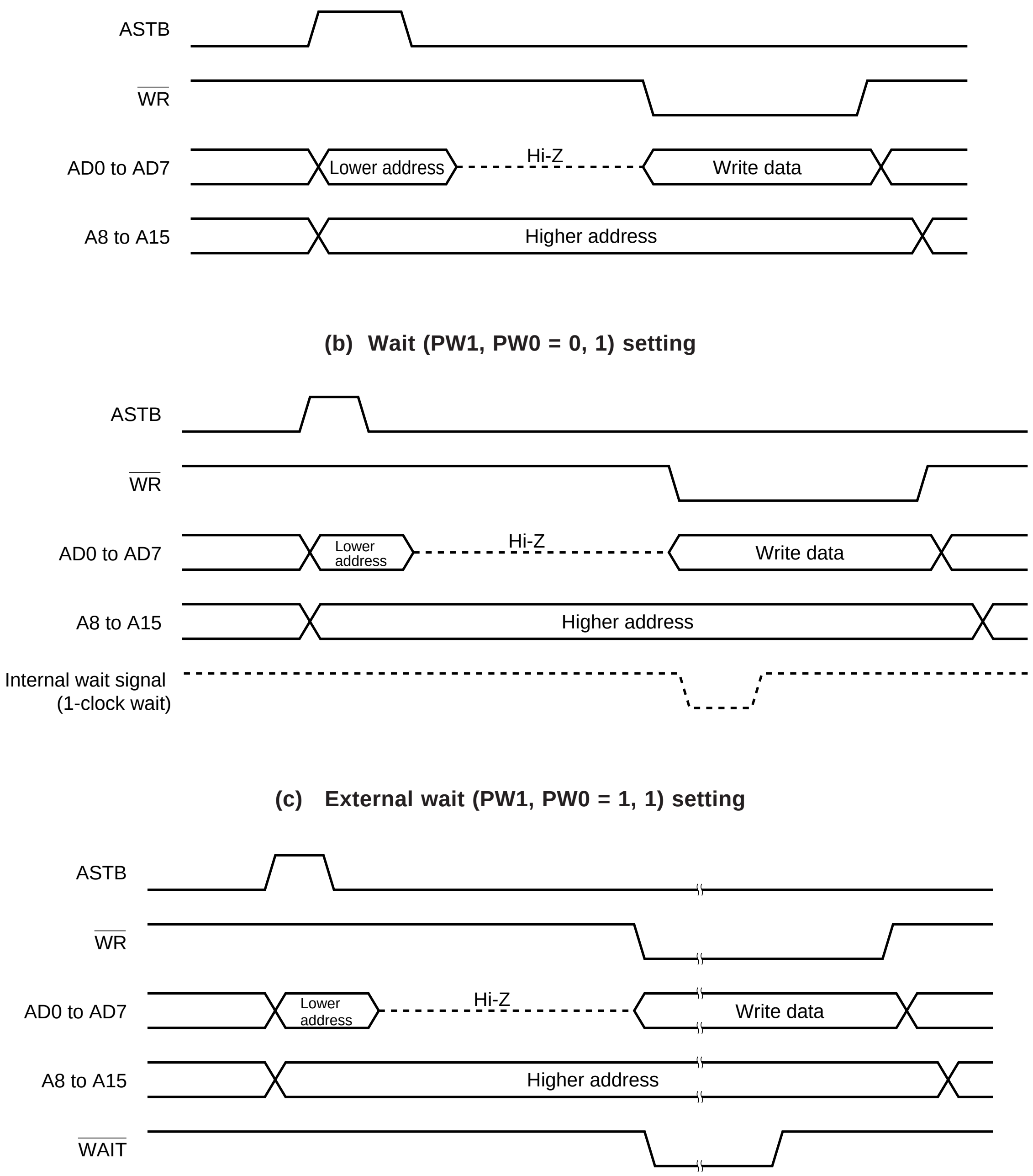
(a) No wait (PW1, PW0 = 0, 0) setting

(b) Wait (PW1, PW0 = 0, 1) setting

(c) External wait (PW1, PW0 = 1, 1) setting

Figure 20-6. External Memory Write Timing

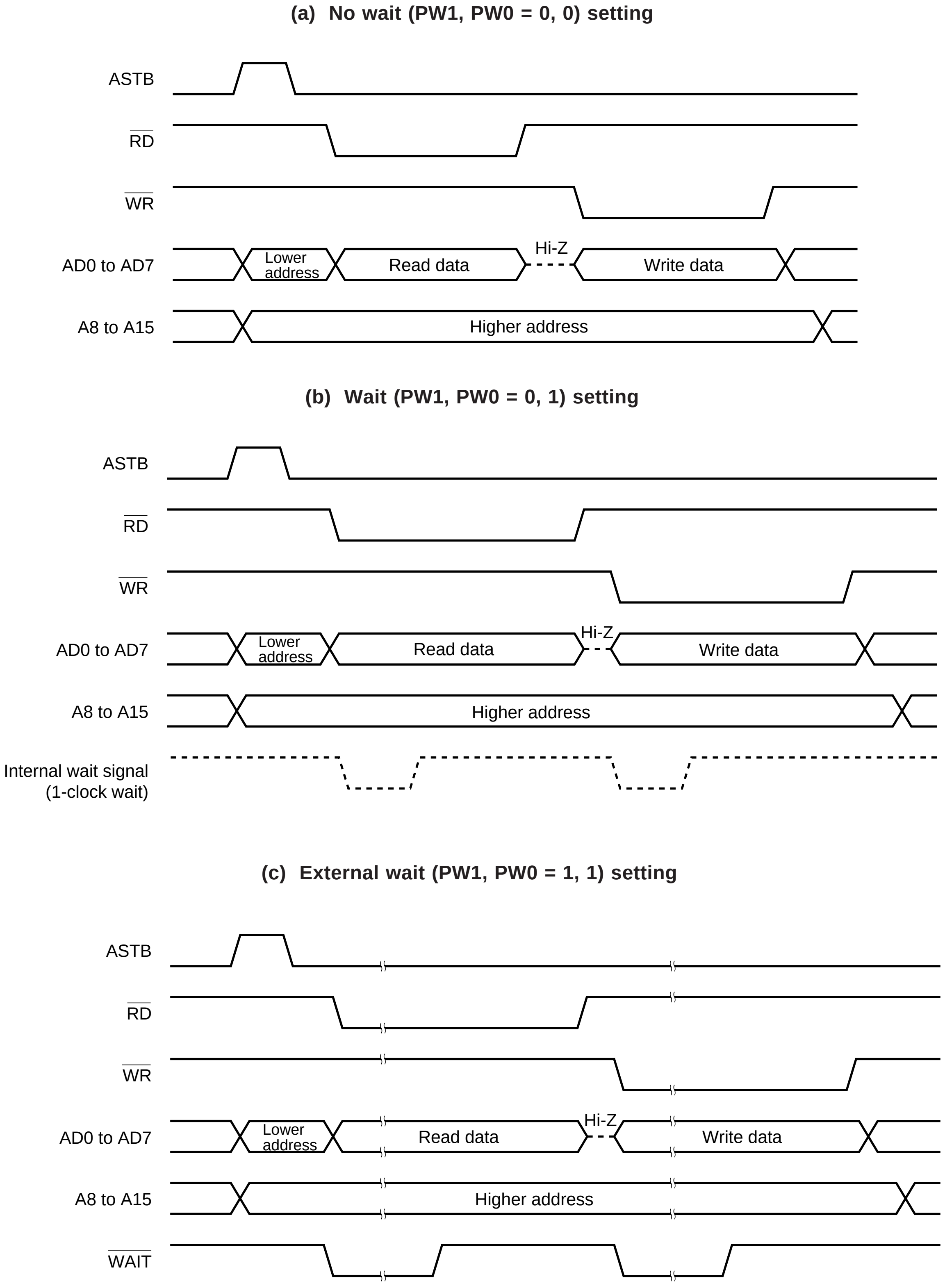
(a) No wait (PW1, PW0 = 0, 0) setting



(b) Wait (PW1, PW0 = 0, 1) setting

(c) External wait (PW1, PW0 = 1, 1) setting

Figure 20-7. External Memory Read Modify Write Timing


(a) No wait (PW1, PW0 = 0, 0) setting

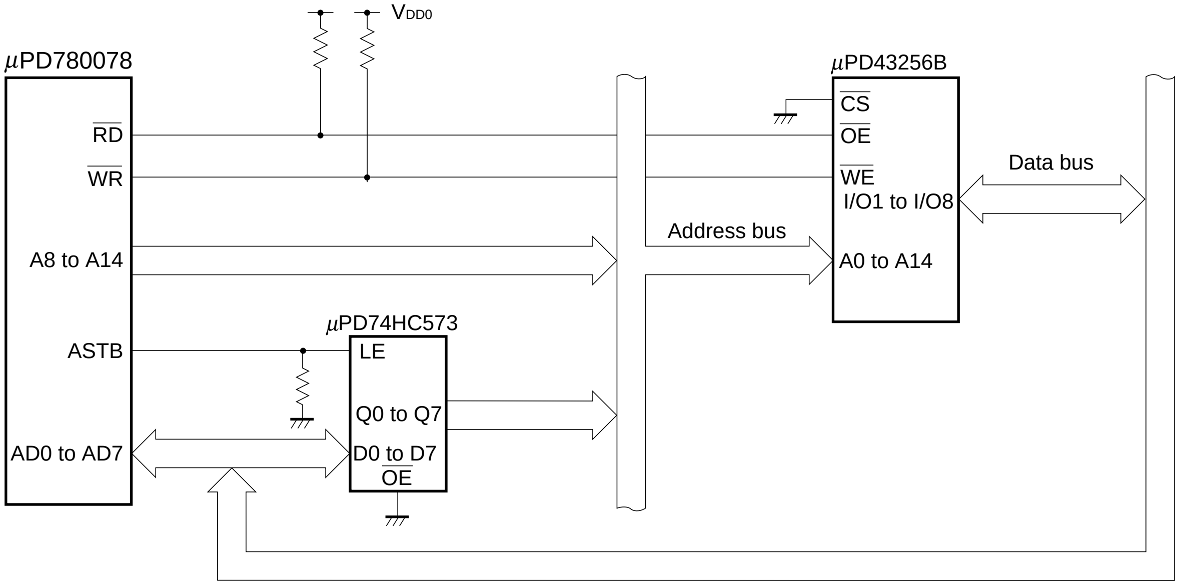
(b) Wait (PW1, PW0 = 0, 1) setting

(c) External wait (PW1, PW0 = 1, 1) setting

20.4 Example of Connection with Memory

This section provide an example of connecting the μPD780078 with external memory (in this example, SRAM) in Figure 20-8. In addition, the external device expansion function is used in the full-address mode, and the addresses from 0000H to 7FFFH (32 Kbytes) are allocated for internal ROM, and the addresses after 8000H from SRAM.

Figure 20-8. Connection Example of μPD780078 and Memory



CHAPTER 21 STANDBY FUNCTION

21.1 Standby Function and Configuration

21.1.1 Standby function

The standby function is designed to decrease power consumption of the system. The following two modes are available.

(1) HALT mode

Halt instruction execution sets the HALT mode. The HALT mode is intended to stop the CPU operation clock. The system clock oscillator continues oscillating. In this mode, current consumption is not decreased as much as in the STOP mode. However, the HALT mode is effective to restart operation immediately upon interrupt request and to carry out intermittent operations such as watch applications.

(2) STOP mode

Stop instruction execution sets the STOP mode. In the STOP mode, the main 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 to hold 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 a wait time is required to secure an oscillation stabilization time after the STOP mode is cleared, select the HALT mode if it is necessary to start processing immediately upon interrupt request.

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 input/output port output latch and output buffer statuses are also held.

Cautions 1. The STOP mode can be used only when the system operates with the main system clock (subsystem clock oscillation cannot be stopped). The HALT mode can be used with either the main system clock or the subsystem clock.

2. When operation is transferred to the STOP mode, be sure to stop the peripheral hardware operation and execute the STOP instruction.

3. The following sequence is recommended for power consumption reduction of the A/D converter when the standby function is used: First clear bit 7 (ADCS0) of the A/D converter mode register 0 (ADM0) to 0 to stop the A/D conversion operation, and then execute the HALT or STOP instruction.

21.1.2 Standby function control register

The wait time after the STOP mode is cleared upon interrupt request is controlled with the oscillation stabilization time select register (OSTS).

OSTS is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the value of OSTS to 04H.

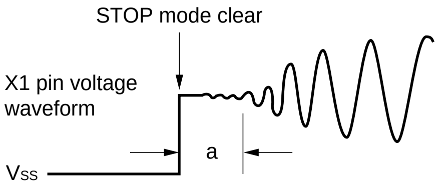
Figure 21-1. Oscillation Stabilization Time Select Register (OSTS) Format

Address: FFFAH At 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	$2^{12}/f_X$ (488 μs)
0	0	1	$2^{14}/f_X$ (1.95 ms)
0	1	0	$2^{15}/f_X$ (3.91 ms)
0	1	1	$2^{16}/f_X$ (7.81 ms)
1	0	0	$2^{17}/f_X$ (15.6 ms)
Other than above			Setting prohibited

Caution The wait time after the STOP mode is cleared does not include the time (see "a" in the illustration below) from STOP mode clear to clock oscillation start. The time is not included either by $\overline{\text{RESET}}$ input or by interrupt request generation.



Remarks 1. f_X: Main system clock oscillation frequency

2. Values in parentheses are for operation with f_X = 8.38 MHz.

21.2 Standby Function Operations

21.2.1 HALT mode

(1) HALT mode setting and operating statuses

The HALT mode is set by executing the HALT instruction. It can be set with the main system clock or the subsystem clock.

The operating statuses in the HALT mode are described below.

Table 21-1. HALT Mode Operating Statuses

HALT Mode Setting Item		During HALT Instruction Execution Using Main System Clock		During HALT Instruction Execution Using Subsystem Clock	
		Without subsystem clock[Note 1]	With subsystem clock[Note 2]	With main system clock oscillation	With main system clock oscillation stopped
Clock generator		Both main system clock and subsystem clock can be oscillated. Clock supply to CPU stops.			
CPU		Operation stops.			
Port (Output latch)		Status before HALT mode setting is held.			
★ 16-bit timer/event counter		Operable			Stop
8-bit timer/event counter		Operable			Operable when TI50, TI51 are selected as count clock.
Watch timer		Operable when $f_X/2^7$ is selected as count clock	Operable		Operable when f_{XT} is selected as count clock.
Watchdog timer		Operable		Operation stops.	
A/D converter		Stop			
Serial interface		Operable			Operable during external SCK.
External interrupt		Operable			
Bus line during external expansion	AD0 to AD7	High impedance			
	A8 to A15	Status before HALT mode setting is held.			
	ASTB	Low level			
	$\overline{\text{WR}}$, $\overline{\text{RD}}$	High level			
	$\overline{\text{WAIT}}$	High impedance			

Notes 1. Including case when external clock is not supplied.

2. Including case when external clock is supplied.

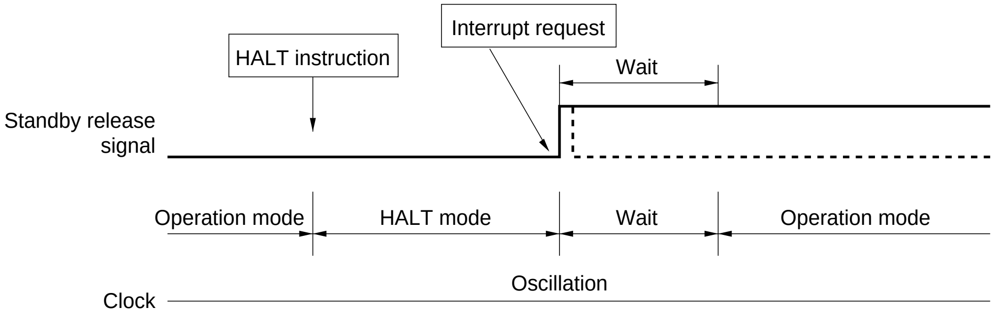
(2) HALT mode release

The HALT mode can be released with the following three types of sources.

(a) Release by unmasked interrupt request

When an unmasked interrupt request is generated, the HALT mode is released. If interrupt acknowledge is enabled, vectored interrupt service is carried out. If interrupt acknowledge is disabled, the next address instruction is executed.

Figure 21-2. HALT Mode Release by Interrupt Request Generation



Remarks 1. The broken line indicates the case when the interrupt request which has released the standby mode is acknowledged.

2. Wait times are as follows:

- When vectored interrupt service is carried out : 8 or 9 clocks
- When vectored interrupt service is not carried out : 2 or 3 clocks

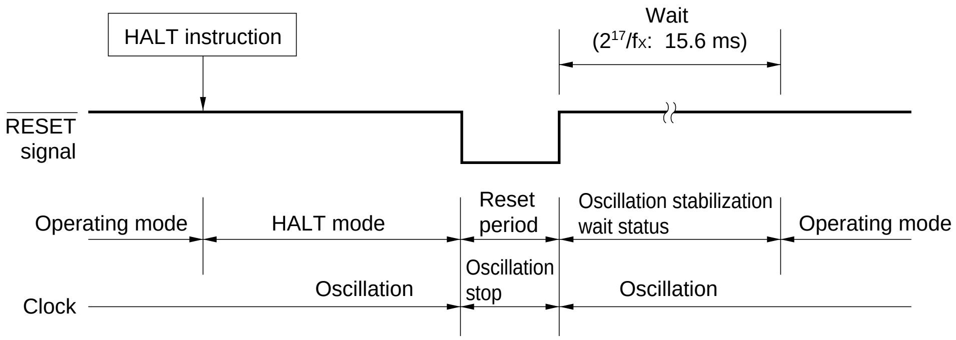
(b) Release by non-maskable interrupt request

When an non-maskable interrupt request is generated, the HALT mode is released and vectored interrupt service is carried out whether interrupt acknowledge is enabled or disabled.

(c) Release by $\overline{\text{RESET}}$ input

When $\overline{\text{RESET}}$ signal is input, HALT mode is released. And, as in the case with normal reset operation, a program is executed after branch to the reset vector address.

Figure 21-3. HALT Mode Release by $\overline{\text{RESET}}$ Input



Remarks 1. fX: Main system clock oscillation frequency

2. Values in parentheses are for operation with fX = 8.38 MHz.

Table 21-2. 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 service execution
	0	1	0	1	Next address instruction execution
	0	1	×	0	
	0	1	1	1	Interrupt service execution
	1	×	×	×	HALT mode hold
Non-maskable interrupt request	—	—	×	×	Interrupt service execution
$\overline{\text{RESET}}$ input	—	—	×	×	Reset processing

×: Don’t care

21.2.2 STOP mode

(1) STOP mode setting and operating status

The STOP mode is set by executing the STOP instruction. It can be set only with the main system clock.

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 main system clock.

2. Because the interrupt request signal is used to clear the standby mode, if there is an interrupt source with the interrupt request flag set and the interrupt mask flag reset, the standby mode is immediately cleared if set. Thus, the STOP mode is reset to the HALT mode immediately after execution of the STOP instruction. After the wait set using the oscillation stabilization time select register (OSTS), the operating mode is set.

The operating status in the STOP mode is described in Table 21-3 below.

Table 21-3. STOP Mode Operating Status

STOP Mode Setting Item		With Subsystem Clock	Without Subsystem Clock
Clock generator		Only main system clock oscillation is stopped.	
CPU		Operation stops.	
Port (Output latch)		Status before STOP mode setting is held.	
16-bit timer/event counter		Operation stops.	
8-bit timer/event counter		Operable only when TI50, TI51 are selected as count clock.	
Watch timer		Operable when f_{XT} is selected as counter clock.	Operation stops.
Watchdog timer		Operation stops.	
Clock output/buzzer output		PCL and BUZ at low level.	
A/D converter		Operation stops	
Serial interface	Other than UART0, 2	Operable only when externally supplied clock is specified as the serial clock.	
	UART0, 2	Operation stops. (transmit shift register 0, 2 (TXS0, TXS2), receive shift register 0, 2 (RX0, RX2), receive buffer register 0, 2 (RXB0, RXB2) and transmit buffer register 2 (TXB2) hold the value just before the clock stop.)	
External interrupt		Operatable	
Bus line during external expansion	AD0 to AD7	High impedance	
	A8 to A15	Status before STOP mode setting is held.	
	ASTB	Low level	
	$\overline{WR}$, $\overline{RD}$	High level	
	$\overline{WAIT}$	High impedance	

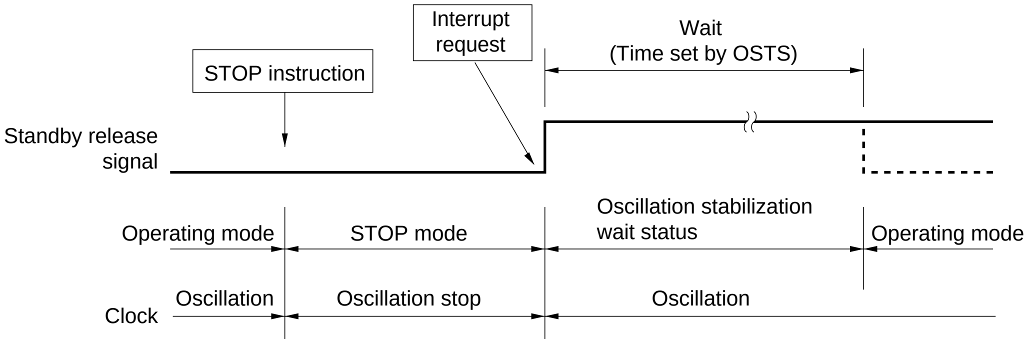
(2) STOP mode release

The STOP mode can be released by the following two types of sources.

(a) Release by unmasked interrupt request

When an unmasked interrupt request is generated, the STOP mode is released. If interrupt acknowledge is enabled after the lapse of oscillation stabilization time, vectored interrupt service is carried out. If interrupt acknowledge is disabled, the next address instruction is executed.

Figure 21-4. STOP Mode Release by Interrupt Request Generation

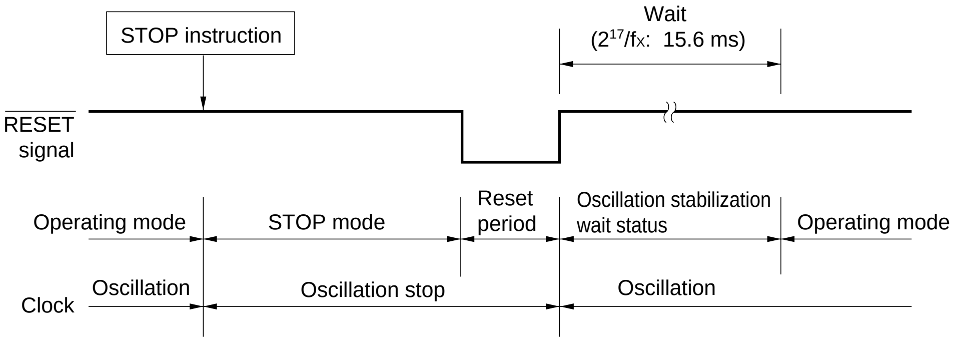


Remark The broken line indicates the case when the interrupt request which has cleared the standby status is acknowledged.

(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, reset operation is carried out.

Figure 21-5. STOP Mode Release by $\overline{\text{RESET}}$ Input



Remarks 1. fx: Main system clock oscillation frequency

2. Values in parentheses are for operation with fx = 8.38 MHz.

Table 21-4. 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 service execution
	0	1	0	1	Next address instruction execution
	0	1	×	0	
	0	1	1	1	Interrupt service execution
	1	×	×	×	STOP mode hold
$\overline{\text{RESET}}$ input	—	—	×	×	Reset processing

×: Don’t care

CHAPTER 22 RESET FUNCTION

22.1 Reset Function

The following two operations are available to generate the reset function.

(1) External reset input via $\overline{\text{RESET}}$ pin
(2) Internal reset by watchdog timer runaway time detection

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 hardware is set to the status shown in Table 22-1. Each pin has high impedance during reset input or during oscillation stabilization time just after reset clear.

When a high level is input to the $\overline{\text{RESET}}$ pin, the reset is cleared and program execution starts after the lapse of oscillation stabilization time $2^{17}/f_X$. The reset applied by watchdog timer overflow is automatically cleared after a reset and program execution starts after the lapse of oscillation stabilization time $2^{17}/f_X$ (see **Figures 22-2** to **22-4**).

Cautions 1. For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin.

2. During reset input, main system clock oscillation remains stopped but subsystem clock oscillation continues.

3. When the STOP mode is cleared by reset, the STOP mode contents are held during reset input. However, the port pin becomes high-impedance.

Figure 22-1. Reset Function Block Diagram

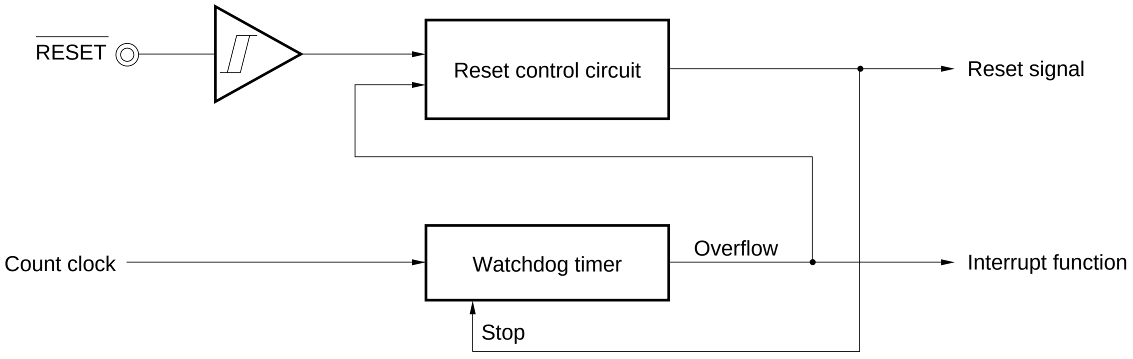


Figure 22-2. Timing of Reset by $\overline{\text{RESET}}$ Input

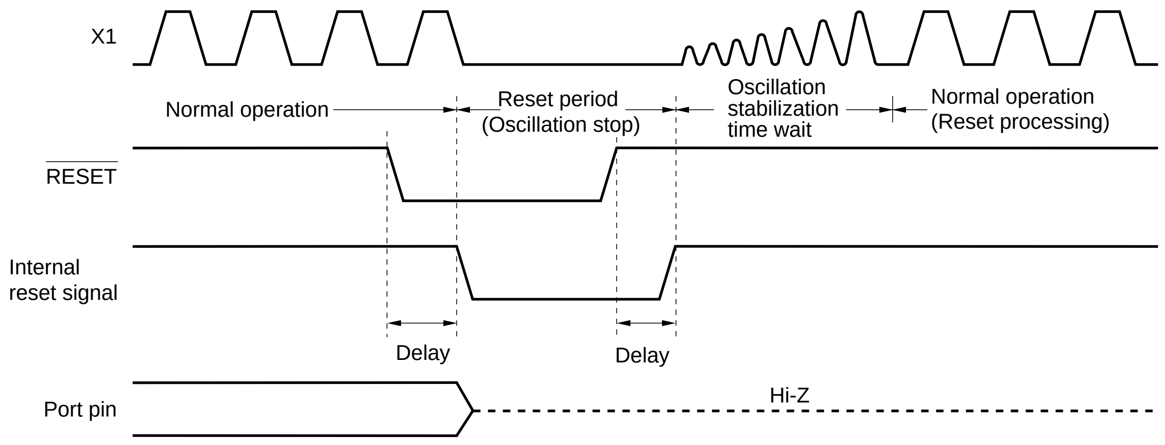


Figure 22-3. Timing of Reset Due to Watchdog Timer Overflow

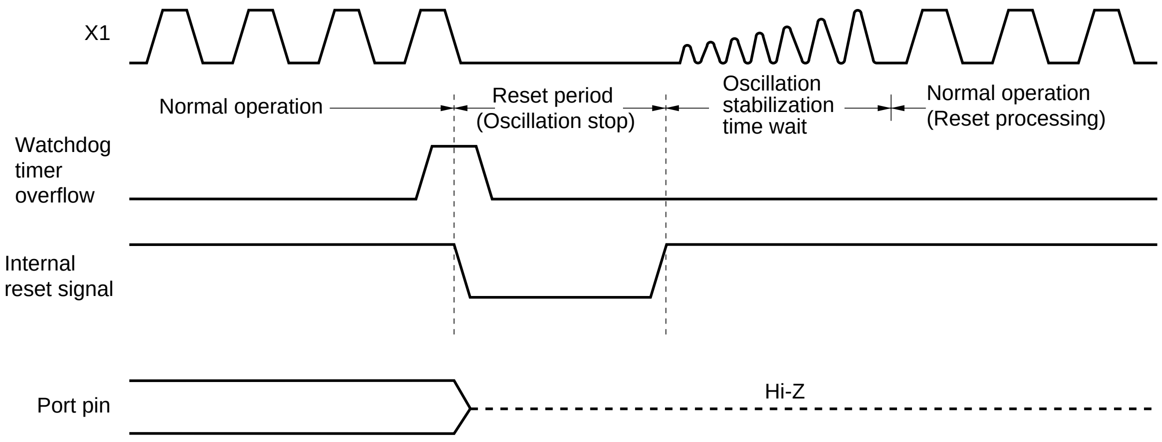


Figure 22-4. Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input

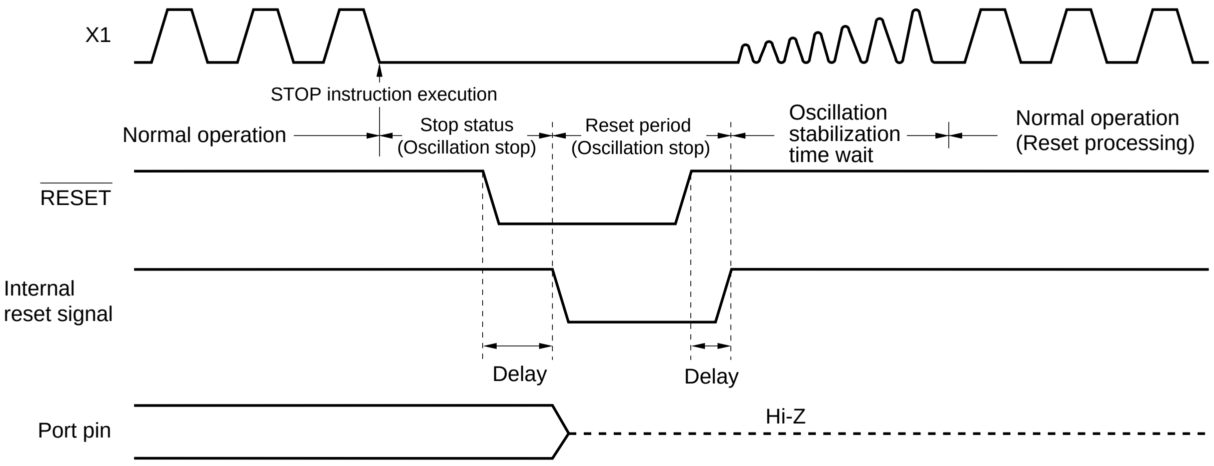


Table 22-1. Hardware Statuses after Reset (1/2)

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	Undefined[Note 2]
	General-purpose register	Undefined[Note 2]
Port (Output latch)		00H
Port mode registers (PM0, PM2 to PM8)		FFH
Pull-up resistor option registers (PU0, PU2 to PU8)		00H
Processor clock control register (PCC)		04H
Memory size switching register (IMS)		CFH[Note 3]
Internal RAM size switching register (IXS)		0CH[Note 4]
Memory expansion mode register (MEM)		00H
Memory expansion wait setting register (MM)		10H
Oscillation stabilization time select register (OSTS)		04H
16-bit timer/event counter	Timer counters 00, 01 (TM00, TM01)	0000H
	Capture/compare registers 000, 001, 010, 011 (CR000, CR001, CR010, CR011)	Undefined
	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 (TM50, TM51)	00H
	Compare registers 50, 51 (CR50, CR51)	Undefined
	Clock select registers 50, 51 (TCL50, TCL51)	00H
	Mode control register 50, 51 (TMC50, TMC51)	00H
Watch timer	Operation mode register (WTM)	00H
Watchdog timer	Clock select 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. Although the initial value is CFH, use the following value to be set for each version.

μPD780076, 780076Y : CCH
μPD780078, 780078Y : CFH
μPD78F0078, 78F0078Y : Value for mask ROM versions

4. Although the default value of this register is 0CH, initialize this register to 0AH.

Table 22-1. Hardware Statuses after Reset (2/2)

Hardware		Status after Reset
Clock output/buzzer output controller	Clock output select register (CKS)	00H
A/D converter	Conversion result registers 0 (ADCR0)	00H
	Mode register 0 (ADM0)	00H
	Analog input channel specification register 0 (ADS0)	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 (UART2)	Asynchronous serial interface mode register 2 (ASIM2)	00H
	Transfer mode specification register 2 (TRMC2)	02H
	Clock select register 2 (CKSEL2)	00H
	Baud rate generator control register 2 (BRGC2)	00H
	Asynchronous serial interface status register (ASIS2)	00H
	Asynchronous serial interface transmit status register (ASIF2)	00H
	Transmit buffer register 2 (TXB2)	FFH
	Receive buffer register 2 (RXB2)	FFH
Serial interface (SIO3)	Shift register 3 (SIO3)	Undefined
	Operation mode registers 3 (CSIM3)	00H
Serial interface (CSI1)	Transmit buffer register 1 (SOTB1)	Undefined
	Shift register 1(SIO1)	Undefined
	Operation mode register 1 (CSIM1)	00H
	Clock select register 1 (CSIC1)	10H
Serial interface (IIC0)[Note]	Transfer clock select register 0 (IICCL0)	00H
	Shift register 0 (IIC0)	00H
	Control register 0 (IICC0)	00H
	Status register 0 (IICS0)	00H
	Slave address register 0 (SVA0)	00H
Interrupt	Request flag registers (IF0L, IF0H, IF1L)	00H
	Mask flag registers (MK0L, MK0H, MK1L)	FFH
	Priority specification flag registers (PR0L, PR0H, PR1L)	FFH
	External interrupt rising edge enable register (EGP)	00H
	External interrupt falling edge enable register (EGN)	00H

Note Provided only in the μPD780078Y Subseries.

CHAPTER 23 μPD78F0078, 78F0078Y

The μPD78F0078 and 78F0078Y are provided as the flash memory versions of the μPD780078, 78F0078Y Subseries.

For purposes of simplification, throughout this chapter, the μPD78F0078 is used to refer to both the μPD78F0078 and 78F0078Y. In the same way, with regard to mask ROM versions, the μPD780076 and 780078 is used.

The μPD78F0078 replaces the internal mask ROM of the μPD780078 with flash memory to which a program can be written, deleted and overwritten while mounted on the substrate. Table 23-1 lists the differences among the μPD78F0078 and the mask ROM versions.

Table 23-1. Differences among μPD78F0078 and Mask ROM Versions

Item	μPD78F0078	μPD780076	μPD780078
Internal ROM configuration	Flash memory	Mask ROM	
Internal ROM capacity	60 Kbytes[Note 1]	48 Kbytes	60 Kbytes
Mask option to specify the on-chip pull-up resistors of pins P30 to P33[Note 2]	Not possible	Possible	
IC pin	None	Available	
V_{PP} pin	Available	None	
Electrical specifications	Refer to data sheet of each product.		

Notes 1. The same capacity as the mask ROM versions can be specified by means of the memory size switching register (IMS).

2. P30 and P31 pins are provided only on the μPD780078Y Subseries.

Caution **Flash memory versions and mask ROM versions differ in their noise immunity and noise radiation. If replacing flash memory versions with mask ROM versions when changing from test production to mass production, be sure to perform sufficient evaluation with CS versions (not ES versions) of mask ROM versions.**

23.1 Memory Size Switching Register

The μPD78F0078 allows users to select the internal memory capacity using the memory size switching register (IMS) so that the same memory map as that of the μPD780076, 780078 with a different size of internal memory capacity can be achieved.

IMS is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the value of IMS to CFH.

Caution The initial value of IMS is "setting prohibited (CFH)". Be sure to set the value of the relevant mask ROM versions at initialization.

Figure 23-1. Memory Size Switching Register (IMS) Format

Address: FFF0H At Reset: CFH R/W

Symbol	7	6	5	4	3	2	1	0
IMS	RAM2	RAM1	RAM0	0	ROM3	ROM2	ROM1	ROM0

RAM2	RAM1	RAM0	Internal High-Speed RAM Capacity Selection
1	1	0	1024 bytes
Other than above			Setting prohibited

ROM3	ROM2	ROM1	ROM0	Internal ROM Capacity Selection
0	0	1	0	8 Kbytes
0	1	0	0	16 Kbytes
0	1	1	0	24 Kbytes
1	0	0	0	32 Kbytes
1	1	0	0	48 Kbytes
1	1	1	1	60 Kbytes
Other than above				Setting prohibited

The IMS settings to obtain the same memory map as mask ROM versions are shown in Table 23-2.

Table 23-2. Memory Size Switching Register Settings

Target Mask ROM Versions	IMS Setting
μPD780076	CCH
μPD780078	CFH

Caution When using the mask ROM versions, be sure to set the value indicated in Table 23-2 to IMS.

23.2 Internal Expansion RAM Size Select Register (IXS)

This register is used to set the internal expansion RAM capacity through software.
This register is set by using an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets the value of IXS to 0CH.

Caution Setting the default value of IXS (0CH) is prohibited. Be sure to initialize the value of this register to 0AH.

Figure 23-2. Internal Expansion RAM Size Select Register (IXS) Format

Address: FFF4H At 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	Selects Internal High-speed RAM Capacity.
0	1	0	1	0	1024 bytes
Other than above					Setting prohibited

23.3 Flash Memory Programming

On-board writing of flash memory (with device mounted on target system) is supported.

On-board writing is done after connecting a dedicated flash programmer (Flashpro II (part number: FL-PR2), Flashpro III (FL-PR3, 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 II or Flashpro III.

Remark FL-PR2 and FL-PR3 are products of Naito Densei Machida Mfg. Co., Ltd.

23.3.1 Selection of transmission method

Writing to flash memory is performed using Flashpro II or Flashpro III and serial communication. Select the transmission method for writing from Table 23-3. For the selection of the transmission method, a format like the one shown in Figure 23-2 is used. The transmission methods are selected with the V_{PP} pulse numbers shown in Table 23-3.

Table 23-3. Transmission Method List

Transmission Method	Number of Channels	Pin Used	Number of V_{PP} Pulses
3-wire serial I/O (SIO3)	1	SI3/P34 SO3/P35 $\overline{\text{SCK3}}$/P36	1
		HS/P31[Note] SI3/P34 SO3/P35 $\overline{\text{SCK3}}$/P36	3
I^2C bus (IIC0) (μPD78F0078Y only)	1	SDA0/P32 SCL0/P33	4
UART (UART0)	1	RxD0/P23 TxD0/P24	8

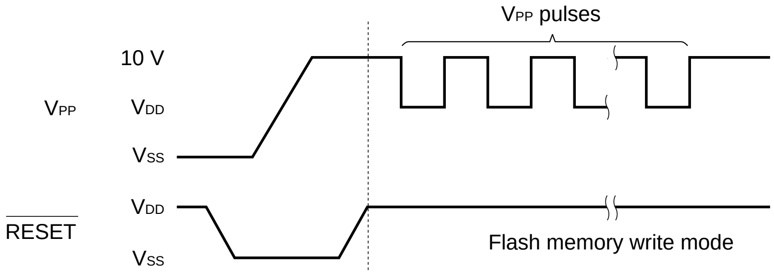
Note When using the handshake function, use the Flashpro III. Do not use the Flashpro II.

Cautions 1. Be sure to select the number of V_{PP} pulses shown in Table 23-3 for the transmission method.

2. If performing write operations to flash memory with the UART transmission method, set the main system clock oscillation frequency to 3 MHz or higher.

3. When writing data to flash memory using the I^2C bus communication method, the range of the usable main system clock oscillation frequency is 4.19 MHz $\leq f_X \leq$ 8.38 MHz.

Figure 23-3. Transmission Method Selection Format



23.3.2 Flash memory programming function

Flash memory writing is performed through command and data transmit/receive operations using the selected transmission method. The main functions are listed in Table 23-4.

Table 23-4. Main Functions of Flash Memory Programming

Function	Description
Reset	Used to detect write stop and transmission synchronization.
Batch verify	Compares entire memory contents and input data.
Batch delete	Deletes the entire memory contents.
Batch blank check	Checks the deletion status of the entire memory.
High-speed write	Performs writing to flash memory according to write start address and number of write data (bytes).
Continuous write	Performs successive write operations using the data input with high-speed write operation.
Status	Checks the current operation mode and operation end.
Oscillation frequency setting	Inputs the resonator oscillation frequency information.
Delete time setting	Inputs the memory delete time.
Baud rate setting	Sets the transmission rate when the UART method is used.
I^2C mode setting	Sets the standard/high-speed mode when the I^2C bus method is used.
Silicon signature read	Outputs the device name, memory capacity, and device block information.

23.3.3 Connection of Flashpro II and Flashpro III

Connection of the Flashpro II and the μPD78F0078 differs depending on communication method (3-wire serial I/O (SIO3), UART (UART0), and I^{2}C bus (IIC0)**Note**). Each type of connection is shown in Figures 23-4 to 23-7.

Note μPD78F0078Y only

Figure 23-4. Connection of Flashpro II and Flashpro III Using 3-Wire Serial I/O (SIO3) Method

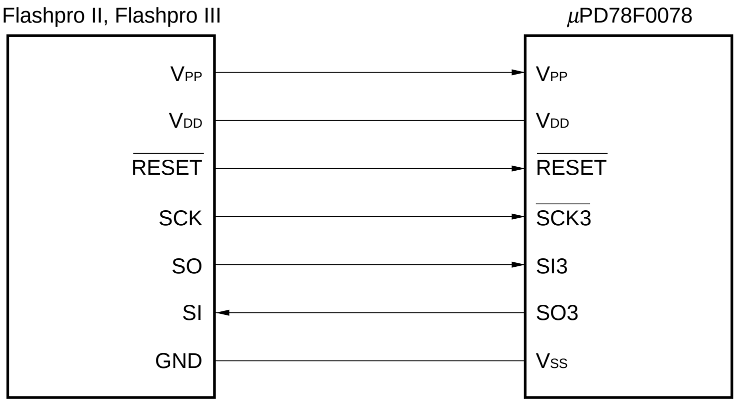


Figure 23-5. Connection of Flashpro III Using 3-wire Serial I/O (SIO3) (When Handshake Is Used)

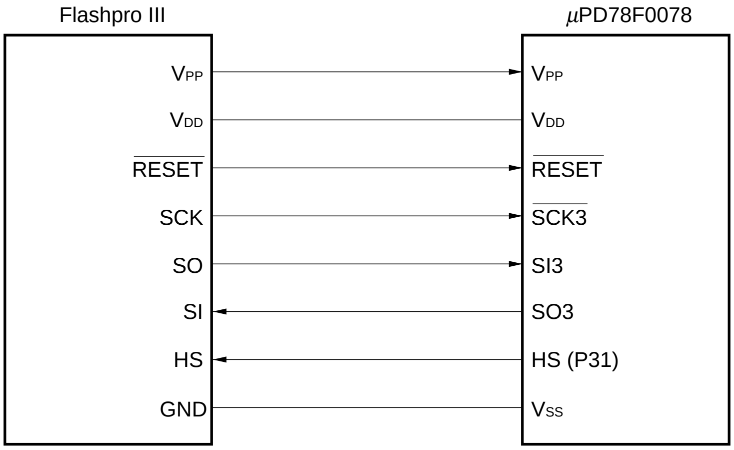


Figure 23-6. Connection of Flashpro II and Flashpro III Using I²C (IIC0) Bus Method (μPD78F0078Y Only)

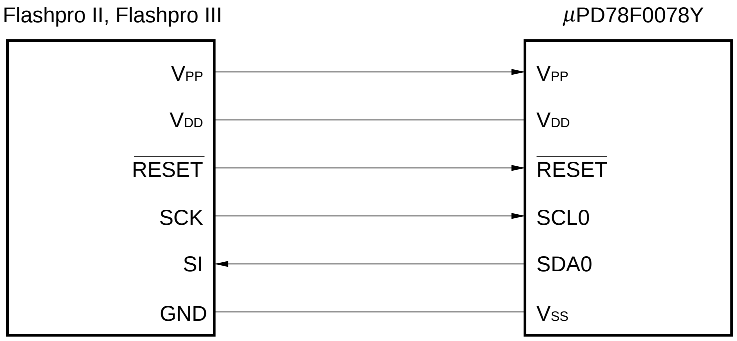
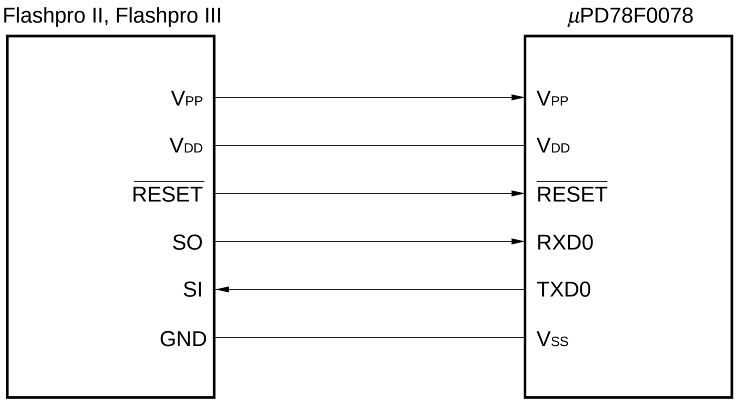


Figure 23-7. Connection of Flashpro II and Flashpro III Using UART (UART0) Method



[MEMO]

CHAPTER 24 INSTRUCTION SET

This chapter lists each instruction set of the μPD780078, 780078Y Subseries in table form. For details of its operation and operation code, refer to the separate document **78K/0 Series User's Manual - Instructions (U12326E)**.

24.1 Legends Used in Operation List

24.1.1 Operand identifiers and specification methods

Operands are written in “Operand” column of each instruction in accordance with the specification method of the instruction operand identifier (refer to the assembler specifications for detail). When there are two or more methods, select one of them. Alphabetic letters in capitals and symbols, #, !, $ and [ ] are key words and must be written as they are. Each symbol has the following meaning.

- # : Immediate data specification
- ! : Absolute address specification
- $: Relative address specification
- [ ] : Indirect address specification

In the case of immediate data, describe an appropriate numeric value or a label. When using a label, be sure to write the #, !, $, and [ ] symbols.

For operand register identifiers, r and rp, either function names (X, A, C, etc.) or absolute names (names in parentheses in the table below, R0, R1, R2, etc.) can be used for specification.

Table 24-1. Operand Identifiers and Specification Methods

Identifier	Specification Method
r rp sfr sfrp	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7), AX (RP0), BC (RP1), DE (RP2), HL (RP3) Special function register symbol**Note** Special function register symbol (16-bit manipulatable register even addresses only)**Note**
saddr saddrp	FE20H to FF1FH Immediate data or labels FE20H to FF1FH Immediate data or labels (even address only)
addr16 addr11 addr5	0000H to FFFFH Immediate data or labels (Only even addresses for 16-bit data transfer instructions) 0800H to 0FFFH Immediate data or labels 0040H to 007FH Immediate data or labels (even address only)
word byte bit	16-bit immediate data or label 8-bit immediate data or label 3-bit immediate data or label
RBn	RB0 to RB3

Note Addresses from FFD0H to FFDFH cannot be accessed with these operands.

Remark For special function register symbols, refer to **Table 5-3 Special Function Register List**.

24.1.2 Description of “operation” column

A	:	A register; 8-bit accumulator
X	:	X register
B	:	B register
C	:	C register
D	:	D register
E	:	E register
H	:	H register
L	:	L register
AX	:	AX register pair; 16-bit accumulator
BC	:	BC register pair
DE	:	DE register pair
HL	:	HL register pair
PC	:	Program counter
SP	:	Stack pointer
PSW	:	Program status word
CY	:	Carry flag
AC	:	Auxiliary carry flag
Z	:	Zero flag
RBS	:	Register bank select flag
IE	:	Interrupt request enable flag
NMIS	:	Non-maskable interrupt servicing flag
( )	:	Memory contents indicated by address or register contents in parentheses
X_H, X_L	:	High-order 8 bits and low-order 8 bits of 16-bit register
∧	:	Logical product (AND)
∨	:	Logical sum (OR)
⊻	:	Exclusive logical sum (exclusive OR)
——	:	Inverted data
addr16	:	16-bit immediate data or label
jdisp8	:	Signed 8-bit data (displacement value)

24.1.3 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

24.2 Operation List

Instruction Group	Mnemonic	Operands	Byte	Clock		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 + n	A ← (addr16)			
		!addr16, A	3	8	9 + m	(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 + n	A ← (DE)			
		[DE], A	1	4	5 + m	(DE) ← A			
		A, [HL]	1	4	5 + n	A ← (HL)			
		[HL], A	1	4	5 + m	(HL) ← A			
		A, [HL + byte]	2	8	9 + n	A ← (HL + byte)			
		[HL + byte], A	2	8	9 + m	(HL + byte) ← A			
		A, [HL + B]	1	6	7 + n	A ← (HL + B)			
		[HL + B], A	1	6	7 + m	(HL + B) ← A			
		A, [HL + C]	1	6	7 + n	A ← (HL + C)			
		[HL + C], A	1	6	7 + m	(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 + n + m	A ↔ (addr16)			
		A, [DE]	1	4	6 + n + m	A ↔ (DE)			
		A, [HL]	1	4	6 + n + m	A ↔ (HL)			
		A, [HL + byte]	2	8	10 + n + m	A ↔ (HL + byte)			
		A, [HL + B]	2	8	10 + n + m	A ↔ (HL + B)			
		A, [HL + C]	2	8	10 + n + m	A ↔ (HL + C)			

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed.

3. Except “r = A”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to internal ROM program.

3. n is the number of waits when external memory expansion area is read from.

4. m is the number of waits when external memory expansion area is written to.

Instruction Group	Mnemonic	Operands	Byte	Clock		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 + 2n	AX ← (addr16)			
		!addr16, AX	3	10	12 + 2m	(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 + n	A, CY ← A + (addr16)	×	×	×
		A, [HL]	1	4	5 + n	A, CY ← A + (HL)	×	×	×
		A, [HL + byte]	2	8	9 + n	A, CY ← A + (HL + byte)	×	×	×
		A, [HL + B]	2	8	9 + n	A, CY ← A + (HL + B)	×	×	×
		A, [HL + C]	2	8	9 + n	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 + n	A, CY ← A + (addr16) + CY	×	×	×
		A, [HL]	1	4	5 + n	A, CY ← A + (HL) + CY	×	×	×
		A, [HL + byte]	2	8	9 + n	A, CY ← A + (HL + byte) + CY	×	×	×
		A, [HL + B]	2	8	9 + n	A, CY ← A + (HL + B) + CY	×	×	×
		A, [HL + C]	2	8	9 + n	A, CY ← A + (HL + C) + CY	×	×	×

Notes **1.** When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Only when rp = BC, DE or HL

4. Except “r = A”

Remarks **1.** One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to internal ROM program.

3. n is the number of waits when external memory expansion area is read from.

4. m is the number of waits when external memory expansion area is written to.

Instruction Group	Mnemonic	Operands	Byte	Clock		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 + n	A, CY ← A – (addr16)	×	×	×
		A, [HL]	1	4	5 + n	A, CY ← A – (HL)	×	×	×
		A, [HL + byte]	2	8	9 + n	A, CY ← A – (HL + byte)	×	×	×
		A, [HL + B]	2	8	9 + n	A, CY ← A – (HL + B)	×	×	×
		A, [HL + C]	2	8	9 + n	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 + n	A, CY ← A – (addr16) – CY	×	×	×
		A, [HL]	1	4	5 + n	A, CY ← A – (HL) – CY	×	×	×
		A, [HL + byte]	2	8	9 + n	A, CY ← A – (HL + byte) – CY	×	×	×
		A, [HL + B]	2	8	9 + n	A, CY ← A – (HL + B) – CY	×	×	×
		A, [HL + C]	2	8	9 + n	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 + n	A ← A∧(addr16)	×		
		A, [HL]	1	4	5 + n	A ← A∧[HL]	×		
		A, [HL + byte]	2	8	9 + n	A ← A∧[HL + byte]	×		
		A, [HL + B]	2	8	9 + n	A ← A∧[HL + B]	×		
		A, [HL + C]	2	8	9 + n	A ← A∧[HL + C]	×		

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Except “r = A”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to internal ROM program.

3. n is the number of waits when external memory expansion area is read from.

Instruction Group	Mnemonic	Operands	Byte	Clock		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 + n	A ← A∨(addr16)	×		
		A, [HL]	1	4	5 + n	A ← A∨(HL)	×		
		A, [HL + byte]	2	8	9 + n	A ← A∨(HL + byte)	×		
		A, [HL + B]	2	8	9 + n	A ← A∨(HL + B)	×		
		A, [HL + C]	2	8	9 + n	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 + n	A ← A∀(addr16)	×		
		A, [HL]	1	4	5 + n	A ← A∀(HL)	×		
		A, [HL + byte]	2	8	9 + n	A ← A∀(HL + byte)	×		
		A, [HL + B]	2	8	9 + n	A ← A∀(HL + B)	×		
		A, [HL + C]	2	8	9 + n	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 + n	A – (addr16)	×	×	×
		A, [HL]	1	4	5 + n	A – (HL)	×	×	×
		A, [HL + byte]	2	8	9 + n	A – (HL + byte)	×	×	×
		A, [HL + B]	2	8	9 + n	A – (HL + B)	×	×	×
		A, [HL + C]	2	8	9 + n	A – (HL + C)	×	×	×

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Except “r = A”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to internal ROM program.

3. n is the number of waits when external memory expansion area is read from.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit operation	**ADDW**	AX, #word	3	6	–	AX, CY ← AX + word	×	×	×
	SUBW	AX, #word	3	6	–	AX, CY ← AX – word	×	×	×
	CMPW	AX, #word	3	6	–	AX – word	×	×	×
Multiply/ divide	**MULU**	X	2	16	–	AX ← A × X			
	DIVUW	C	2	25	–	AX (Quotient), C (Remainder) ← AX ÷ C			
Increment/ decrement	**INC**	r	1	2	–	r ← r + 1	×	×	
		saddr	2	4	6	(saddr) ← (saddr) + 1	×	×	
	DEC	r	1	2	–	r ← r – 1	×	×	
		saddr	2	4	6	(saddr) ← (saddr) – 1	×	×	
	INCW	rp	1	4	–	rp ← rp + 1			
	DECW	rp	1	4	–	rp ← rp – 1			
Rotate	**ROR**	A, 1	1	2	–	(CY, A_7 ← A_0, A_{m-1} ← A_m) × 1 time			×
	ROL	A, 1	1	2	–	(CY, A_0 ← A_7, A_{m+1} ← A_m) × 1 time			×
	RORC	A, 1	1	2	–	(CY ← A_0, A_7 ← CY, A_{m-1} ← A_m) × 1 time			×
	ROLC	A, 1	1	2	–	(CY ← A_7, A_0 ← CY, A_{m+1} ← A_m) × 1 time			×
	ROR4	[HL]	2	10	12 + n + m	A_{3-0} ← $(HL)_{3-0}$, $(HL)_{7-4}$ ← A_{3-0}, $(HL)_{3-0}$ ← $(HL)_{7-4}$			
	ROL4	[HL]	2	10	12 + n + m	A_{3-0} ← $(HL)_{7-4}$, $(HL)_{3-0}$ ← A_{3-0}, $(HL)_{7-4}$ ← $(HL)_{3-0}$			
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 ← (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← sfr.bit			×
		CY, A.bit	2	4	–	CY ← A.bit			×
		CY, PSW.bit	3	–	7	CY ← PSW.bit			×
		CY, [HL].bit	2	6	7 + n	CY ← (HL).bit			×
		saddr.bit, CY	3	6	8	(saddr.bit) ← CY			
		sfr.bit, CY	3	–	8	sfr.bit ← CY			
		A.bit, CY	2	4	–	A.bit ← CY			
		PSW.bit, CY	3	–	8	PSW.bit ← CY	×	×	
		[HL].bit, CY	2	6	8 + n + m	(HL).bit ← CY			

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to internal ROM program.

3. n is the number of waits when external memory expansion area is read from.

4. m is the number of waits when external memory expansion area is written to.

Instruction Group	Mnemonic	Operands	Byte	Clock		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 + n	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 + n	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 + n	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 + n + m	(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 + n + m	(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

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to internal ROM program.

3. n is the number of waits when external memory expansion area is read from.

4. m is the number of waits when external memory expansion area is written to.

Instruction Group	Mnemonic	Operands	Byte	Clock		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 manipulate	**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			
Unconditional 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

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to internal ROM program.

Instruction Group	Mnemonic	Operands	Byte	Clock		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 + n	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 + n	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 + n + m	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

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to internal ROM program.

3. n is the number of waits when external memory expansion area is read from.

4. m is the number of waits when external memory expansion area is written to.

24.3 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 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

[MEMO]

APPENDIX A DIFFERENCES BETWEEN μPD780024A, 780034A, AND 780078 SUBSERIES

Table A-1 shows the major differences between μPD780024A, 780034A, and 780078 Subseries.

Table A-1. Major Differences between μPD780024A, 780034A, and 780078 Subseries

Item / Name		μPD780024A, 780034A Subseries	μPD780078 Subseries
Internal I^{2}C bus version (Y subseries)		Provided	
Flash memory version		μPD78F0034A	μPD78F0078
Supply voltage		V_{DD} = 1.8 to 5.5 V	
ROM		• μPD780021A, 780031A: 8 Kbytes • μPD780022A, 780032A: 16 Kbytes • μPD780023A, 780033A: 24 Kbytes • μPD780024A, 780034A: 32 Kbytes	• μPD780076: 48 Kbytes • μPD780078: 60 Kbytes
Internal high-speed RAM		• μPD780021A, 780031A • μPD780022A, 780032A } 512 bytes • μPD780023A, 780033A • μPD780024A, 780034A } 1024 bytes	1024 bytes
Internal expansion RAM		None	1024 bytes
Minimum instruction execution time		0.24 μs (8.38 MHz)	
Number of I/O port		51	52
A/D converter		• μPD780024A subseries: 8 bits × 8 • μPD780034A subseries: 10 bits × 8	10 bits × 8
Serial interface	Subseries without suffix Y	• 3-wire: 2 • UART: 1	• 3-wire: 1 • UART: 1 • 3-wire/UART: 1
	Subseries with suffix Y	• 3-wire: 2 • UART: 1 • Multi-master I^{2}C: 1	• 3-wire: 1 • UART: 1 • 3-wire/UART: 1 • Multi-master I^{2}C: 1
Timer		• 16-bit timer/event counter × 1 • 8-bit timer/event counter × 2 • Watch timer × 1 • Watchdog timer × 1	• 16-bit timer/event counter × 2 • 8-bit timer/event counter × 2 • Watch timer × 1 • Watchdog timer × 1
Timer output		3 (8-bit PWM output possible: 2)	4 (8-bit PWM output possible: 2)
Package		• 64-pin plastic SDIP (19.05 mm (750)) • 64-pin plastic QFP (14 × 14 mm) • 64-pin plastic LQFP (12 × 12 mm)	• 64-pin plastic QFP (14 × 14 mm) • 64-pin plastic TQFP (12 × 12 mm)
Device file		DF780034	DF780078
Emulation board		IE-780034-NS-EM1	IE-780078-NS-EM1
Electrical specifications Recommended soldering conditions		Refer to the data sheet of each product.	

[MEMO]

APPENDIX B   DEVELOPMENT TOOLS

The following development tools are available for the development of systems that employ the μPD780078, and 780078Y Subseries.

Figure B-1 shows the development tool configuration.

- **Support for PC98-NX series**

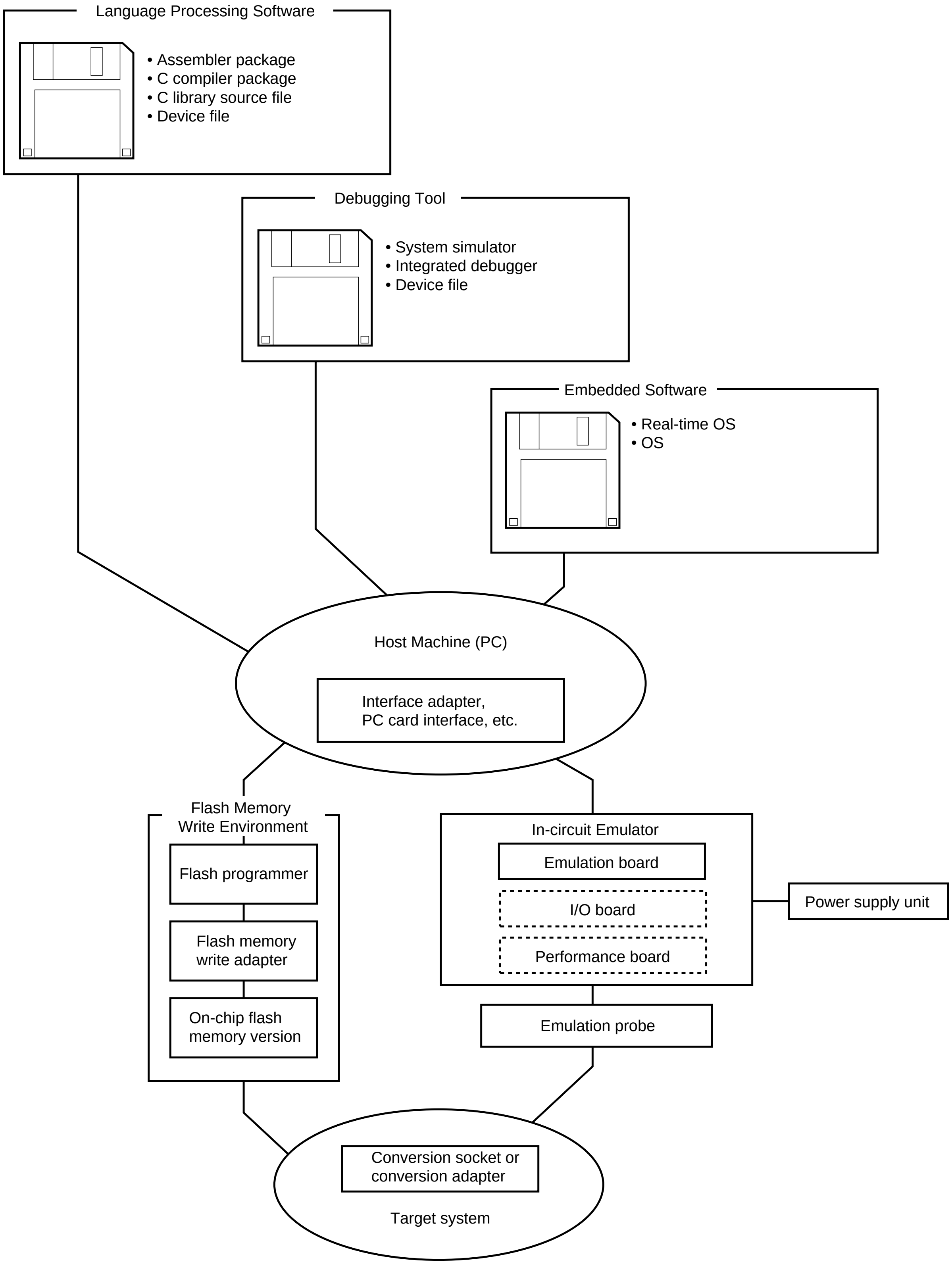
  Unless otherwise specified, products compatible with IBM PC/AT™ computers are compatible with PC98-NX series computers. When using PC98-NX series computers, refer to the explanation for IBM PC/AT computers.

- **Windows**

  Unless otherwise specified, "Windows" means the following OSs.
  - Windows 3.1
  - Windows95
  - WindowsNT™ Ver 4.0

Figure B-1. Development Tool Configuration (1/2)

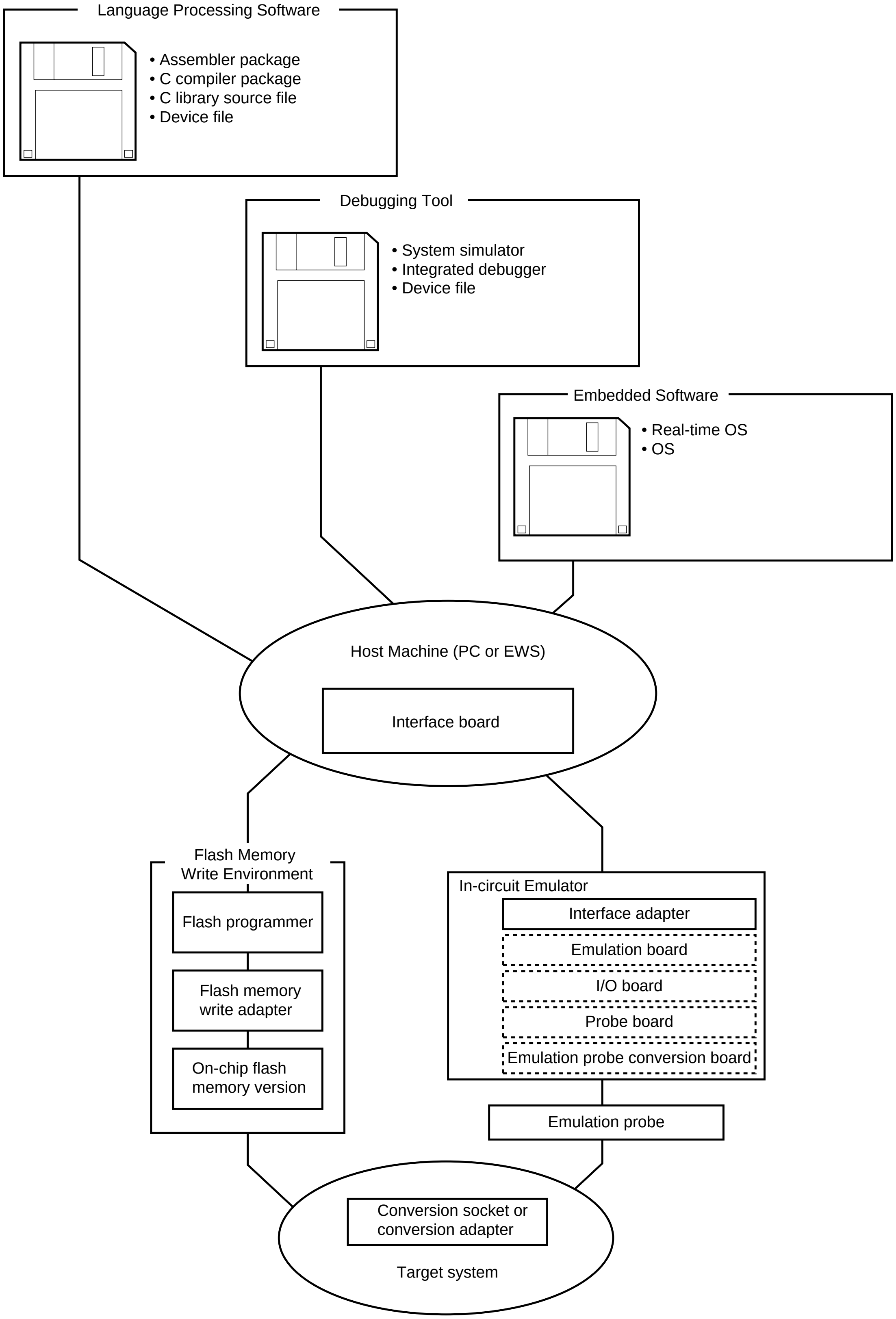
(1) When using the in-circuit emulator IE-78K0-NS



Remark Items in broken line boxes differ according to the development environment. See **B.3.1 Hardware**.

Figure B-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. See **B.3.1 Hardware**.

B.1 Language Processing Software

RA78K0 Assembler Package	This assembler converts programs written in mnemonics into an object codes executable with 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 an optical device file (DF780078). **<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 assembler package) on Windows.
	Part number: µS××××RA78K0
CC78K0 C Compiler Package	This compiler converts programs written in C language into object codes executable with a microcontroller. This compiler should be used in combination with an optical assembler package and device file. **<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 assembler package) on Windows.
	Part number: µS××××CC78K0
DF780078[Note] Device File	This file contains information peculiar to the device. This device file should be used in combination with an optical tool (RA78K0, CC78K0, SM78K0, ID78K0-NS, and ID78K0). Corresponding OS and host machine differ depending on the tool to be used with.
	Part number: µS××××DF780078
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 C compiler package to the customer’s specifications.
	Part number: µS××××CC78K0-L

Note The DF780078 can be used in common with the RA78K0, CC78K0, SM78K0, ID78K0-NS, and ID78K0.

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××RA78K0
μS××××CC78K0
μS××××DF780078
μS××××CC78K0-L

××××	Host Machine	OS	Supply Medium
AA13	PC-9800 series	Windows (Japanese version)[Note]	3.5-inch 2HD FD
AB13	IBM PC/AT compatibles	Windows (Japanese version)[Note]	3.5-inch 2HC FD
BB13		Windows (English version)[Note]	
3P16	HP9000 series 700$^{\text{TM}}$	HP-UX$^{\text{TM}}$ (Rel. 10.10)	DAT (DDS)
3K13	SPARCstation$^{\text{TM}}$	SunOS$^{\text{TM}}$ (Rel. 4.1.4)	3.5-inch 2HC FD
3K15		Solaris$^{\text{TM}}$ (Rel. 2.5.1)	1/4-inch CGMT
3R13	NEWS$^{\text{TM}}$ (RISC)	NEWS-OS$^{\text{TM}}$ (Rel. 6.1)	3.5-inch 2HC FD

Note Can be operated in DOS environment.

B.2 Flash Memory Writing Tools

Flashpro II (part number: FL-PR2) Flashpro III (part number: FL-PR3, PG-FP3) Flash Programmer	Flash programmer dedicated to microcontrollers with on-chip flash memory.
FA-64GC FA-64GK [Note] Flash Memory Writing Adapter	Flash memory writing adapter used connected to the Flashpro II and Flashpro III. • FA-64GC : 64-pin plastic QFP (GC-AB8 type) • FA-64GK : 64-pin plastic TQFP (GK-9ET type)

Note Under development.

Remark FL-PR2, FL-PR3, FA-64CW, FA-64GC, and FA-64GK are products of Naito Densei Machida Mfg. Co., Ltd.
Phone: (044) 822-3813 Naito Densei Machida Mfg. Co., Ltd.

B.3 Debugging Tools

B.3.1 Hardware (1/2)

(1) When using the in-circuit emulator IE-78K0-NS

IE-78K0-NS In-circuit Emulator		The in-circuit emulator serves to debug hardware and software when developing application systems using a 78K/0 Series product. It corresponds to integrated debugger (ID78K0-NS). This emulator should be used in combination with power supply unit, emulation probe, and interface adapter which is required to connect this emulator to the host machine.
IE-78K0-NS-PA Performance Board		This board is connected to the IE-78K0-NS to expand its functions. Adding this board adds a coverage function and enhances debugging functions such as tracer and timer functions.
IE-70000-MC-PS-B Power Supply Unit		This adapter is used for supplying power from a receptacle of 100-V to 240-V AC.
IE-70000-98-IF-C Interface Adapter		This adapter is required when using the PC-9800 series computer (except notebook type) as the IE-78K0-NS host machine (C bus compatible).
IE-70000-CD-IF-A PC Card Interface		This is PC card and interface cable required when using notebook-type computer as the IE-78K0-NS host machine (PCMCIA socket compatible).
IE-70000-PC-IF-C Interface Adapter		This adapter is required when using the IBM PC compatible computers as the IE-78K0-NS host machine (ISA bus compatible).
IE-70000-PCI-IF Interface Adapter		This adapter is required when using a computer with PCI bus as the IE-78K0-NS host machine.
IE-780078-NS-EM1[Note] Emulation 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.
NP-64GC Emulation Probe		This probe is used to connect the in-circuit emulator to a target system and is designed for use with 64-pin plastic QFP (GC-AB8 type).
	EV-9200GC-64 Conversion Socket (See **Figures B-2** and **B-3**)	This conversion socket connects the NP-64GC to a target system board designed for a 64-pin plastic QFP (GC-AB8 type).
NP-64GC-TQ Emulation Probe		This probe is used to connect the in-circuit emulator to a target system and is designed for use with 64-pin plastic QFP (GC-AB8 type).
	TGC-064SAP Conversion Adapter	This conversion adapter connects the NP-64GC-TQ to a target system board designed for a 64-pin plastic LQFP (GK-8A8 type).
NP-64GK Emulation Probe		This probe is used to connect the in-circuit emulator to a target system and is designed for use with 64-pin plastic TQFP (GK-9ET type).
	TGK-064SBP Conversion Adapter (See **Figure B-4**)	This conversion socket connects the NP-64GK to a target system board designed for a 64-pin plastic TQFP (GK-9ET type).

Remarks 1. NP-64CW, NP-64GC, NP-64GC-TQ, and NP-64GK are products of Naito Densei Machida Mfg. Co., Ltd.
Phone: (044) 822-3813 Naito Densei Machida Mfg. Co., Ltd.

2. TGK-064SBP and TGC-064SAP are products of TOKYO ELETECH CORPORATION.
Inquiry: Daimaru Kogyo, Ltd. Phone: Tokyo (03) 3820-7112 Electronics Dept.
Osaka (06) 6244-6672 Electronics 2nd Dept.

3. EV-9200GC-64 is sold in five units.

4. TGK-064SBP and TGC-064SAP are sold in one units.

B.3.1 Hardware (2/2)

(2) When using the in-circuit emulator IE-78001-R-A

IE-78001-R-A In-circuit Emulator	The in-circuit emulator serves to debug hardware and software when developing application systems using a 78K/0 Series product. It corresponds to integrated debugger (ID78K0). This emulator should be used in combination with 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 the PC-9800 series computer (except notebook type) as the IE-78001-R-A host machine (C bus compatible).
IE-7000-PC-IF-C Interface Adapter	This adapter is required when using the IBM PC/AT compatible computers as the IE-78001-R-A host machine (ISA bus compatible).
IE-70000-PCI-IF Interface Adapter	This board is required when using a computer with PCI bus as the IE-78001-R-A host machine.
IE-78000-R-SV3 Interface Adapter	This is adapter and cable required when using an EWS computer as the IE-78001-R-A host machine, and is used connected to the board in the IE-78000-R-A. As Ethernet^{TM}, 10Base-5 is supported. With the other method, a commercially available conversion adapter is necessary.
IE-780078-NS-EM1 Emulation 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 emulation conversion board.
IE-78K0-R-EX1 Emulation Probe Conversion Board	This board is required when using the IE-780078-NS-EM1 on the IE-78001-R-A.
EP-78240GC-R Emulation Probe	This probe is used to connect the in-circuit emulator to a target system and is designed for use with 64-pin plastic QFP (GC-AB8 type).
EV-9200GC-64 Conversion Socket (See **Figures B-2** and **B-3**)	This conversion socket connects the EP-78240GC-R to a target system board designed for a 64-pin plastic QFP (GC-AB8 type).
EP-78012GK-R Emulation Probe	This probe is used to connect the in-circuit emulator to a target system and is designed for use with 64-pin plastic TQFP (GK-9ET type).
TGK-064SBP Conversion Adapter (See **Figure B-4**)	This conversion socket connects the EP-78012GK-R to a target system board designed for a 64-pin plastic TQFP (GK-9ET type).

Note Under development

Remarks 1. TGK-064SBP is a product of TOKYO ELETECH CORPORATION.
Inquiry: Daimaru Kogyo, Ltd. Phone: Tokyo (03) 3820-7112 Electronics Dept.
Osaka (06) 244-6672 Electronics 2nd Dept.

2. EV-9200GC-64 is sold in five units.

3. TGK-064SBP is sold in one units.

B.3.2 Software (1/2)

SM78K0 System Simulator	This system simulator is used to perform debugging at 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 the optical device file (DF780078).
	Part number: μS××××SM78K0

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××SM78K0

××××	Host Machine	OS	Supply Medium
AA13	PC-9800 series	Windows (Japanese version)	3.5-inch 2HD FD
AB13	IBM PC/AT compatibles	Windows (Japanese version)	3.5-inch 2HC FD
BB13		Windows (English version)	

B.3.2 Software (2/2)

ID78K0-NS Integrated Debugger (supporting in-circuit emulator IE-78K0-NS)	This debugger is a control program to debug 78K/0 Series microcontrollers. It adopts 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 in 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 task debugger and system performance analyzer, the efficiency of debugging programs, which run on real-time OSs can be improved. It should be used in combination with the optional device file.
ID78K0 Integrated Debugger (supporting in-circuit emulator IE-78001-R-A)	
	Part number: μS××××ID78K0-NS, μS××××ID78K0

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××ID78K0-NS

××××	Host Machine	OS	Supply Medium
AA13	PC-9800 series	Windows (Japanese version)	3.5-inch 2HD FD
AB13	IBM PC/AT compatibles	Windows (Japanese version)	3.5-inch 2HC FD
BB13		Windows (English version)	

μS××××ID78K0

××××	Host Machine	OS	Supply Medium
AA13	PC-9800 series	Windows (Japanese version)	3.5-inch 2HD FD
AB13	IBM PC/AT compatibles	Windows (Japanese version)	3.5-inch 2HC FD
BB13		Windows (English version)	
3P16	HP9000 series 700	HP-UX (Rel. 10.10)	DAT (DDS)
3K13	SPARCstation	SunOS (Rel. 4.1.4), Solaris (Rel. 2.5.1)	3.5-inch 2HC FD
3K15			1/4-inch CGMT
3R13	NEWS^{TM} (RISC)	NEWS-OS (Rel. 6.1)	3.5-inch 2HC FD

B.4 System Upgrade from Former In-circuit Emulator for 78K/0 Series to IE-78001-R-A

If you already have a former in-circuit emulator for 78K/0 Series microcontrollers (IE-78000-R or IE-78000-R-A), that in-circuit emulator can operate as an equivalent to the IE-78001-R-A by replacing its internal break board with the IE-78001-R-BK.

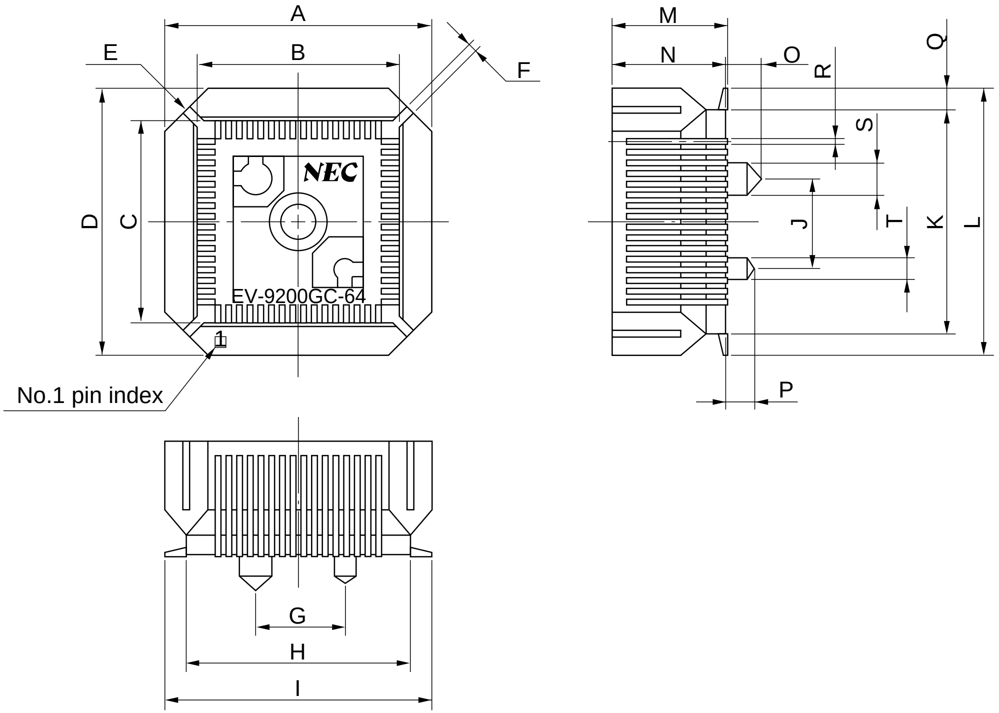
Table B-1. System Upgrade Method from Former In-Circuit Emulator for 78K/0 Series To the IE-78001-R-A

In-circuit Emulator Owned	In-circuit Emulator Cabinet Upgrade[Note]	Board To Be Purchased
IE-78000-R	Required	IE-78001-R-BK
IE-78000-R-A	Not required	

Note For upgrading of a cabinet, send your in-circuit emulator to NEC.

Conversion Socket (EV-9200GC-64) Package Drawing and Recommended Board Mounting Pattern

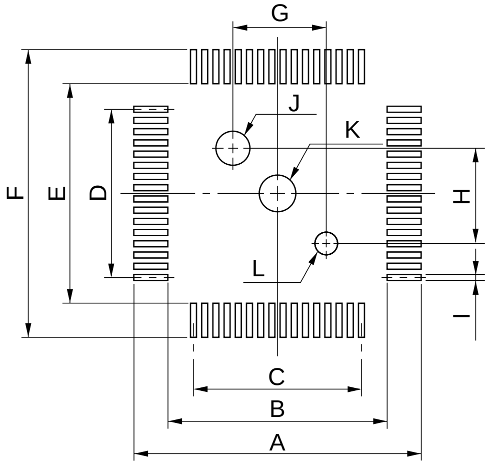
Figure B-2. EV-9200GC-64 Package Drawing (for reference only)



EV-9200GC-64-G0E

ITEM	MILLIMETERS	INCHES
A	18.8	0.74
B	14.1	0.555
C	14.1	0.555
D	18.8	0.74
E	4-C 3.0	4-C 0.118
F	0.8	0.031
G	6.0	0.236
H	15.8	0.622
I	18.5	0.728
J	6.0	0.236
K	15.8	0.622
L	18.5	0.728
M	8.0	0.315
N	7.8	0.307
O	2.5	0.098
P	2.0	0.079
Q	1.35	0.053
R	0.35±0.1	$0.014^{+0.004}_{-0.005}$
S	ϕ2.3	ϕ0.091
T	ϕ1.5	ϕ0.059

Figure B-3. EV-9200GC-64 Recommended Board Mounting Pattern (for reference only)



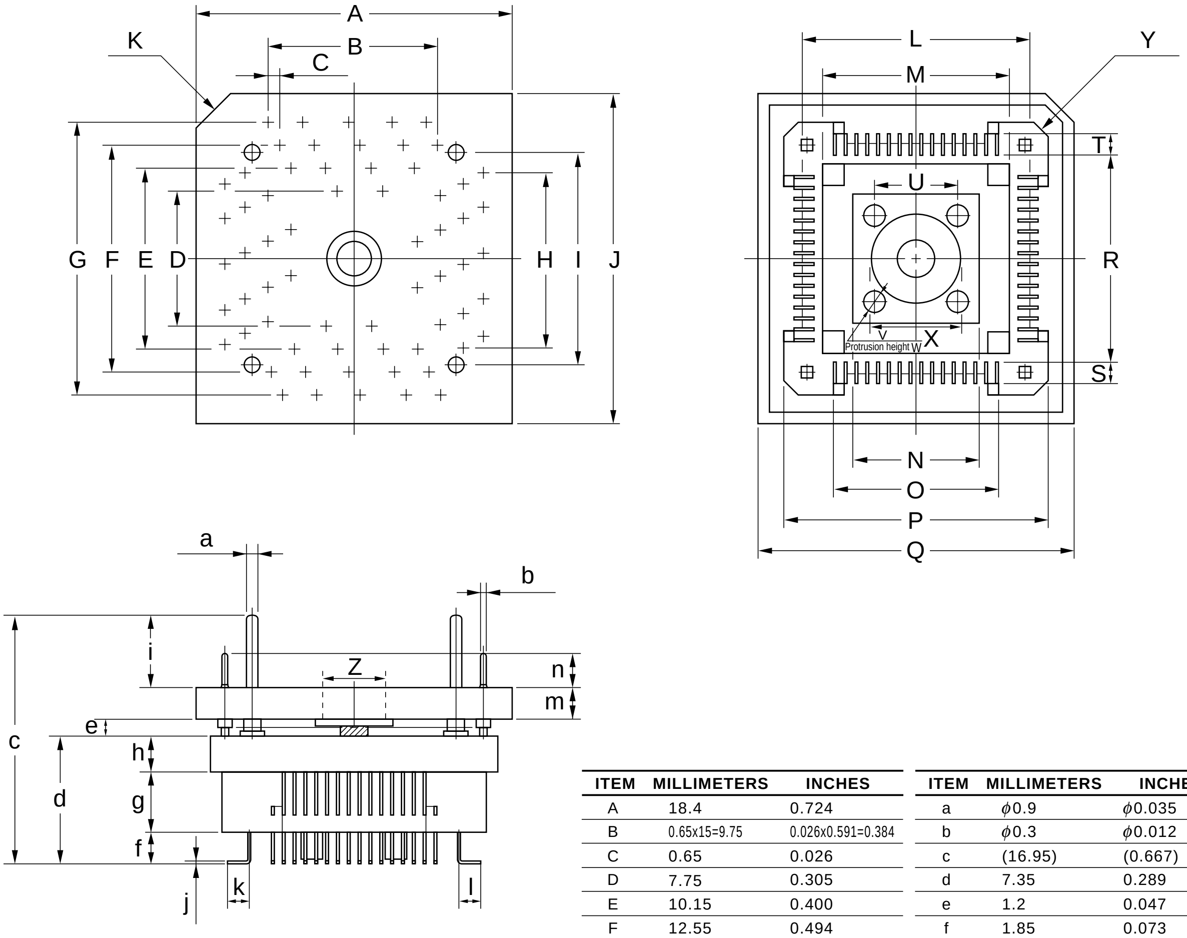
EV-9200GC-64-P1E

ITEM	MILLIMETERS	INCHES
A	19.5	0.768
B	14.8	0.583
C	0.8±0.02 × 15=12.0±0.05	$0.031^{+0.002}_{-0.001} \times 0.591 = 0.472^{+0.003}_{-0.002}$
D	0.8±0.02 × 15=12.0±0.05	$0.031^{+0.002}_{-0.001} \times 0.591 = 0.472^{+0.003}_{-0.002}$
E	14.8	0.583
F	19.5	0.768
G	6.00±0.08	$0.236^{+0.004}_{-0.003}$
H	6.00±0.08	$0.236^{+0.004}_{-0.003}$
I	0.5±0.02	$0.197^{+0.001}_{-0.002}$
J	ϕ2.36±0.03	$\phi 0.093^{+0.001}_{-0.002}$
K	ϕ2.2±0.1	$\phi 0.087^{+0.004}_{-0.005}$
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).

Conversion Adapter Package Drawing (TGK-064SBP)

Figure B-4. TGK-064SBP Package Drawing (for reference only)



ITEM	MILLIMETERS	INCHES
A	18.4	0.724
B	0.65x15=9.75	0.026x0.591=0.384
C	0.65	0.026
D	7.75	0.305
E	10.15	0.400
F	12.55	0.494
G	14.95	0.589
H	0.65x15=9.75	0.026x0.591=0.384
I	11.85	0.467
J	18.4	0.724
K	C 2.0	C 0.079
L	12.45	0.490
M	10.25	0.404
N	7.7	0.303
O	10.02	0.394
P	14.92	0.587
Q	18.4	0.724
R	11.1	0.437
S	1.45	0.057
T	1.45	0.057
U	5.0	0.197
V	4-ϕ1.3	ϕ0.051
W	1.8	0.071
X	ϕ5.3	ϕ0.209
Y	4-C 1.0	4-C 0.039
Z	ϕ3.55	ϕ0.140

ITEM	MILLIMETERS	INCHES
a	ϕ0.9	ϕ0.035
b	ϕ0.3	ϕ0.012
c	(16.95)	(0.667)
d	7.35	0.289
e	1.2	0.047
f	1.85	0.073
g	3.5	0.138
h	2.0	0.079
i	6.0	0.236
j	0.25	0.010
k	1.325	0.052
l	1.325	0.052
m	2.4	0.094
n	2.7	0.106

TGK-064SBP-G0E

note: Product by TOKYO ELETECH CORPORATION.

[MEMO]

APPENDIX C EMBEDDED SOFTWARE

For efficient development and maintenance of the μPD780078 and 780078Y Subseries, the following embedded products are available.

Real-Time OS (1/2)

RX78K/0 Real-time OS	RX78K/0 is a real-time OS conforming to the μITRON specifications. Tool (configurator) for generating nucleus of RX78K/0 and plural information tables is supplied. Used in combination with an optional assembler package (RA78K0) and device file (DF780078). **<Precaution when using RX78K/0 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 RX78K/0, 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 version)[Note]	3.5-inch 2HD FD
AB13	IBM PC/AT compatibles	Windows (Japanese version)[Note]	3.5-inch 2HC FD
BB13		Windows (English version)[Note]	
3P16	HP9000 series 700	HP-UX (Rel. 10.10)	DAT (DDS)
3K13	SPARCstation	SunOS (Rel. 4.1.4), Solaris (Rel. 2.5.1)	3.5-inch 2HC FD
3K15			1/4-inch CGMT
3R13	NEWS (RISC)	NEWS-OS (Rel. 6.1)	3.5-inch 2HC FD

Note Can also be operated in DOS environment.

Real-Time OS (2/2)

MX78K0 OS	MX78K0 is an OS for μITRON specification subsets. A nucleus for the MX78K0 is also included as a companion product. This manages tasks, events, and time. In the task management, determining the task execution order and switching from 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 version)[Note]	3.5-inch 2HD FD
AB13	IBM PC/AT compatibles	Windows (Japanese version)[Note]	3.5-inch 2HC FD
BB13		Windows (English version)[Note]	
3P16	HP9000 series 700	HP-UX (Rel. 10.10)	DAT (DDS)
3K13	SPARCstation	SunOS (Rel. 4.1.4), Solaris (Rel. 2.5.1)	3.5-inch 2HC FD
3K15			1/4-inch CGMT
3R13	NEWS (RISC)	NEWS-OS (Rel. 6.1)	3.5-inch 2HC FD

Note Can also be operated in DOS environment.

APPENDIX D REGISTER INDEX

D.1 Register Index (In Alphabetical Order with Respect to Register Names)

[A]

A/D conversion result register 0 (ADCR0) … 220
A/D converter mode register 0 (ADM0) … 221
Analog input channel specification register 0 (ADS0) … 223
Asynchronous serial interface mode register 0 (ASIM0) … 242
Asynchronous serial interface mode register 2 (ASIM2) … 267
Asynchronous serial interface status register 0 (ASIS0) … 244
Asynchronous serial interface status register 2 (ASIS2) … 270
Asynchronous serial interface transmit status register 2 (ASIF2) ... 272

[B]

Baud rate generator control register 0 (BRGC0) … 244
Baud rate generator control register 2 (BRGC2) ... 271

[C]

Capture/compare control register 00 (CRC00) … 151
Capture/compare control register 01 (CRC01) ... 151
Clock output select register (CKS) … 214
Clock select register 2 (CKSEL2) ... 273

[E]

8-bit timer compare register 50 (CR50) … 181
8-bit timer compare register 51 (CR51) … 181
8-bit timer counter 50 (TM50) … 181
8-bit timer counter 51 (TM51) … 181
8-bit timer mode control register 50 (TMC50) … 183
8-bit timer mode control register 51 (TMC51) … 183
External interrupt falling edge enable register (EGN) … 223, 412
External interrupt rising edge enable register (EGP) … 223, 412

[I]

IIC transfer clock select register 0 (IICCL0) … 360
IIC control register 0 (IICC0) … 353
IIC shift register 0 (IIC0) … 352, 361
IIC status register 0 (IICS0) … 357
Internal expansion RAM size switching register (IXS) ... 449
Interrupt mask flag register 0H (MK0H) … 410
Interrupt mask flag register 0L (MK0L) … 410
Interrupt mask flag register 1L (MK1L) … 410
Interrupt request flag register 0H (IF0H) … 409
Interrupt request flag register 0L (IF0L) … 409
Interrupt request flag register 1L (IF1L) … 409

[M]

Memory expansion mode register (MEM) … 427
Memory expansion wait setting register (MM) … 428
Memory size switching register (IMS) … 448

[O]

Oscillation stabilization time select register (OSTS) … 210, 436

[P]

Port 0 (P0) … 108
Port 1 (P1) … 109
Port 2 (P2) … 110
Port 3 (P3) … 112, 114
Port 4 (P4) … 117
Port 5 (P5) … 119
Port 6 (P6) … 120
Port 7 (P7) … 121
Port 8 (P8) ... 122
Port mode register 0 (PM0) … 123
Port mode register 2 (PM2) … 123
Port mode register 3 (PM3) … 123
Port mode register 4 (PM4) … 123
Port mode register 5 (PM5) … 123
Port mode register 6 (PM6) … 123
Port mode register 7 (PM7) … 123, 157, 186, 216
Port mode register 8 (PM8) ... 123
Prescaler mode register 00 (PRM00) … 155
Prescaler mode register 01 (PRM01) ... 155
Priority specification flag register 0H (PR0H) … 411
Priority specification flag register 0L (PR0L) … 411
Priority specification flag register 1L (PR1L) … 411
Processor clock control register (PCC) … 131
Program status word (PSW) … 83, 413
Pull-up resistor option register 0 (PU0) … 125
Pull-up resistor option register 2 (PU2) … 125
Pull-up resistor option register 3 (PU3) … 125
Pull-up resistor option register 4 (PU4) … 125
Pull-up resistor option register 5 (PU5) … 125
Pull-up resistor option register 6 (PU6) … 125
Pull-up resistor option register 7 (PU7) … 125
Pull-up resistor option register 8 (PU8) ... 125

[R]

Receive buffer register 0 (RXB0) … 241
Receive buffer register 2 (RXB2) ... 265
Receive shift register 0 (RX0) … 241

[S]

Serial clock select register 1 (CSIC1) ... 335
Serial I/O shift register 1 (SIO1) … 333
Serial I/O shift register 3 (SIO3) … 324
Serial operation mode register 1 (CSIM1) … 334
Serial operation mode register 3 (CSIM3) … 325
16-bit timer capture/compare register 000 (CR000) … 146
16-bit timer capture/compare register 001 (CR001) ... 146
16-bit timer capture/compare register 011 (CR011) ... 147
16-bit timer capture/compare register 010 (CR010) … 147
16-bit timer counter 01 (TM01) ... 146
16-bit timer mode control register 00 (TMC00) … 148
16-bit timer mode control register 01 (TMC01) ... 148
16-bit timer output control register 00 (TOC00) … 153
16-bit timer output control register 01 (TOC01) ... 153
16-bit timer counter 00 (TM00) … 146
Slave address register 0 (SVA0) … 352, 361

[T]

Timer clock select register 50 (TCL50) … 182
Timer clock select register 51 (TCL51) … 182
Transfer mode specification register 2 (TRMC2) ... 275
Transmit buffer register 1 (SOTB1) ... 332
Transmit buffer register 2 (TXB2) ... 265
Transmit shift register 0 (TXS0) … 241

[W]

Watch timer operation mode register (WTM) … 200
Watchdog timer clock select register (WDCS) … 208
Watchdog timer mode register (WDTM) … 209

D.2 Register Index (In Alphabetical Order with Respect to Register Symbol)

[A]

ADCR0 : A/D conversion result register 0 … 220
ADM0 : A/D converter mode register 0 … 221
ADS0 : Analog input channel specification register 0 … 223
ASIF2 : Asynchronous serial interface transmit status register 2 ... 272
ASIM0 : Asynchronous serial interface mode register 0 … 242
ASIM2 : Asynchronous serial interface mode status register 2 ... 267
ASIS0 : Asynchronous serial interface status register 0 … 244
ASIS2 : Asynchronous serial interface status register 2 ... 270

[B]

BRGC0 : Baud rate generator control register 0 … 244
BRGC2 : Baud rate generator control register 2 ... 271

[C]

CKS : Clock output select register … 214
CKSEL2 : Clock select register 2 ... 273
CR000 : 16-bit timer capture/compare register 000 … 146
CR001 : 16-bit timer capture/compare register 001 ... 146
CR011 : 16-bit timer capture/compare register 011 ... 147
CR010 : 16-bit timer capture/compare register 010 … 147
CR50 : 8-bit timer compare register 50 … 181
CR51 : 8-bit timer compare register 51 … 181
CRC00 : Capture/compare control register 00 … 151
CRC01 : Capture/compare control register 01 ... 151
CSIC1 : Serial clock select register 1 ... 335
CSIM1 : Serial operation mode register 1 … 334
CSIM3 : Serial operation mode register 3 … 325

[E]

EGN : External interrupt falling edge enable register … 223, 412
EGP : External interrupt rising edge enable register … 223, 412

[I]

IF0H : Interrupt request flag register 0H … 409
IF0L : Interrupt request flag register 0L … 409
IF1L : Interrupt request flag register 1L … 409
IIC0 : IIC shift register 0 … 352, 361
IICC0 : IIC control register 0 … 353
IICCL0 : IIC transfer clock select register 0 … 360
IICS0 : IIC status register 0 … 357
IMS : Memory size switching register … 448
IXS : Internal expansion RAM size switching register ... 449

[M]

MEM : Memory expansion mode register … 427
MK0H : Interrupt mask flag register 0H … 410
MK0L : Interrupt mask flag register 0L … 410
MK1L : Interrupt mask flag register 1L … 410
MM : Memory expansion wait setting register … 428

[O]

OSTS : Oscillation stabilization time select register … 210, 436

[P]

P0 : Port 0 … 108
P1 : Port 1 … 109
P2 : Port 2 … 110
P3 : Port 3 … 112, 114
P4 : Port 4 … 117
P5 : Port 5 … 119
P6 : Port 6 … 120
P7 : Port 7 … 121
P8 : Port 8 ... 122
PCC : Processor clock control register … 131
PM0 : Port mode register 0 … 123
PM2 : Port mode register 2 … 123
PM3 : Port mode register 3 … 123
PM4 : Port mode register 4 … 123
PM5 : Port mode register 5 … 123
PM6 : Port mode register 6 … 123
PM7 : Port mode register 7 … 123, 157, 186, 216
PM8 : Port mode register 8 ... 123
PR0H : Priority specification flag register 0H … 411
PR0L : Priority specification flag register 0L … 411
PR1L : Priority specification flag register 1L … 411
PRM00 : Prescaler mode register 00 … 155
PRM01 : Prescaler mode register 01 ... 155
PSW : Program status word … 83, 413
PU0 : Pull-up resistor option register 0 … 125
PU2 : Pull-up resistor option register 2 … 125
PU3 : Pull-up resistor option register 3 … 125
PU4 : Pull-up resistor option register 4 … 125
PU5 : Pull-up resistor option register 5 … 125
PU6 : Pull-up resistor option register 6 … 125
PU7 : Pull-up resistor option register 7 … 125
PU8 : Pull-up resistor option register 8 ... 125

[R]

RXB0 : Receive buffer register 0 … 241
RXB2 : Receive buffer register 2 ... 265
RX0 : Receive shift register 0 … 241

[S]

SIO1 : Serial I/O shift register 1 … 333
SIO3 : Serial I/O shift register 3 … 324
SOTB1 : Transmit buffer register 1 ... 332
SVA0 : Slave address register 0 … 352, 361

[T]

TCL50 : Timer clock select register 50 … 182
TCL51 : Timer clock select register 51 … 182
TM00 : 16-bit timer counter 00 … 146
TM01 : 16-bit timer counter 01 ... 146
TM50 : 8-bit timer counter 50 … 181
TM51 : 8-bit timer counter 51 … 181
TMC00 : 16-bit timer mode control register 00 … 148
TMC01 : 16-bit timer mode control register 01 ... 148
TMC50 : 8-bit timer mode control register 50 … 183
TMC51 : 8-bit timer mode control register 51 … 183
TOC00 : 16-bit timer output control register 00 … 153
TOC01 : 16-bit timer output control register 01 ... 153
TRMC2 : Transfer mode specification register 2 ... 275
TXB2 : Transmit buffer register 2 ... 265
TXS0 : Transmit shift register 0 … 241

[W]

WDCS : Watchdog timer clock select register … 208
WDTM : Watchdog timer mode register … 209
WTM : Watch timer operation mode register … 200

NEC

Facsimile Message

From:

Name

Company

Tel. FAX

Address

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.

Thank you for your kind support.

North America
NEC Electronics Inc.
Corporate Communications Dept.
Fax: 1-800-729-9288
1-408-588-6130

Europe
NEC Electronics (Europe) GmbH
Technical Documentation Dept.
Fax: +49-211-6503-274

South America
NEC do Brasil S.A.
Fax: +55-11-6465-6829

Hong Kong, Philippines, Oceania
NEC Electronics Hong Kong Ltd.
Fax: +852-2886-9022/9044

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

Taiwan
NEC Electronics Taiwan Ltd.
Fax: 02-2719-5951

Asian Nations except Philippines
NEC Electronics Singapore Pte. Ltd.
Fax: +65-250-3583

Japan
NEC Semiconductor Technical Hotline
Fax: 044-435-9608

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	❏	❏	❏	❏