To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

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

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

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

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

RENESAS

User's Manual

μPD789862 Subseries

8-Bit Single-Chip Microcontrollers

μPD789862
μPD78E9862

Document No. U15852EJ4V0UD00 (4th edition)
Date Published December 2003 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.

These commodities, technology or software, must be exported in accordance with the export administration regulations of the exporting country.
Diversion contrary to the law of that country is prohibited.

- **The information in this document is current as of May, 2003. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC Electronics sales representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC Electronics does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.
- NEC Electronics products are classified into the following three quality grades: "Standard", "Special" and "Specific".
  The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of an NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product before using it in a particular application.
  "Standard": Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots.
  "Special": Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support).
  "Specific": Aircraft, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems and medical equipment for life support, etc.

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

(Note)
(1) "NEC Electronics" as used in this statement means NEC Electronics Corporation and also includes its majority-owned subsidiaries.
(2) "NEC Electronics products" means any product developed or manufactured by or for NEC Electronics (as defined above).

M8E 02. 11-1

Regional Information

Some information contained in this document may vary from country to country. Before using any NEC Electronics product in your application, please contact the NEC Electronics 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.

[GLOBAL SUPPORT]
http://www.necel.com/en/support/support.html

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

NEC Electronics (Europe) GmbH
Duesseldorf, Germany
Tel: 0211-65 03 01

- **Sucursal en España**
  Madrid, Spain
  Tel: 091-504 27 87
- **Succursale Française**
  Vélizy-Villacoublay, France
  Tel: 01-30-67 58 00
- **Filiale Italiana**
  Milano, Italy
  Tel: 02-66 75 41
- **Branch The Netherlands**
  Eindhoven, The Netherlands
  Tel: 040-244 58 45
- **Tyskland Filial**
  Taeby, Sweden
  Tel: 08-63 80 820
- **United Kingdom Branch**
  Milton Keynes, UK
  Tel: 01908-691-133

NEC Electronics Hong Kong Ltd.
Hong Kong
Tel: 2886-9318

NEC Electronics Hong Kong Ltd.
Seoul Branch
Seoul, Korea
Tel: 02-558-3737

NEC Electronics Shanghai, Ltd.
Shanghai, P.R. China
Tel: 021-6841-1138

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

NEC Electronics Singapore Pte. Ltd.
Novena Square, Singapore
Tel: 6253-8311

J03.4

Major Revisions in This Edition

Page	Description
	CHAPTER 1 GENERAL
p. 24	• Modification of **Note** in **1.3 Ordering Information**
p. 26	• Revision of **1.5 78K/0S Series Lineup**
	CHAPTER 4 EEPROM (DATA MEMORY)
p. 62	• Modification of **Table 4-1 EEPROM Write Time**
p. 65	• Modification of program example of **(3)** in **4.5 Notes for EEPROM Writing**
	CHAPTER 7 16-BIT TIMER/EVENT COUNTER 0
p. 114	• Addition of **7.5 (2) Prohibition of compare register change during timer count operation**
	CHAPTER 11 SERIAL INTERFACE 2
p. 155	• Addition of **11.3 (4) (c) Generation of serial clock from system clock in 3-wire serial I/O mode**
	CHAPTER 12 POWER-ON-CLEAR CIRCUITS
p. 181	• Modification of description and addition of **Note** in **12.4.1 Power-on-clear (POC) circuit operation**
p. 184	• Addition of **Caution** to **12.4.2 Operation of low-voltage detector (LVI)**
p. 185	• Modification of **Figure 12-9 LVI Circuit Operation Timing**
	CHAPTER 13 BIT SEQUENTIAL BUFFER
p. 187	• Addition of **13.3 (2) Port mode register 2 (PM2)**
	CHAPTER 18 μPD78E9862
p. 221	• Modification of **Note 1** in **Table 18-1 Differences Between μPD78E9862 and Mask ROM Version**
p. 223	• Modification of **Table 18-2 Communication Mode List**
	CHAPTER 22 ELECTRICAL SPECIFICATIONS (μPD78E9862)
p. 255	• Addition of **Caution**
p. 265	• Modification of **Note 2** in **AC Characteristics (3) EEPROM**
	CHAPTER 24 RECOMMENDED SOLDERING CONDITIONS
p. 268	• Change of **Table 24-1 Surface Mounting Type Soldering Conditions**

The mark ★ shows major revised points.

INTRODUCTION

Target Readers

This manual is intended for user engineers who wish to understand the functions of the μPD789862 Subseries in order to design and develop its application systems and programs.

Purpose

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

Organization

Two manuals are available for the μPD789862 Subseries: this manual and the Instruction Manual (common to the 78K/0S Series).

μPD789862 Subseries User's Manual	78K/0S Series Instructions User's Manual
• Pin functions • Internal block functions • Interrupts • Other internal peripheral functions • Electrical specifications	• CPU function • Instruction set • Instruction description

How to Use This Manual

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

◊ To understand the overall functions of the μPD789862 Subseries
→ Read this manual in the order of the **CONTENTS**.
◊ How to read register formats
→ The name of a bit whose number is enclosed with <> is reserved in the assembler and is defined in the C compiler by the header file sfrbit.h.
◊ To learn the detailed functions of a register whose register name is known
→ See **APPENDIX B REGISTER INDEX**.
◊ To learn the details of the instruction functions of the 78K/0S Series
→ Refer to **78K/0S Series Instructions User's Manual (U11047E)** separately available.
◊ To confirm the electrical specifications of the μPD789862 Subseries
→ Refer to each chapter of electrical specifications.

Conventions

Data significance:	Higher digits on the left and lower digits on the right
Active low representation:	$\overline{\times\times\times}$ (overscore over pin or signal name)
Note:	Footnote for item marked with **Note** in the text
Caution:	Information requiring particular attention
Remark:	Supplementary information
Numerical representation:	Binary ... ×××× or ××××B
	Decimal ... ××××
	Hexadecimal ... ××××H

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

Documents Related to Devices

Document Name	Document No.
μPD789862 Subseries User's Manual	This manual
78K/0S Series Instructions User's Manual	U11047E

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

Document Name		Document No.
RA78K0S Assembler Package	Operation	U14876E
	Language	U14877E
	Structured Assembly Language	U11623E
CC78K0S C Compiler	Operation	U14871E
	Language	U14872E
ID78K Series Integrated Debugger Ver. 2.30 or Later	Operation (Windows™ Based)	U15185E
Project Manager Ver. 3.12 or Later (Windows Based)		U14610E

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

Document Name	Document No.
IE-78K0S-NS In-Circuit Emulator	U13549E
IE-78K0S-NS-A In-Circuit Emulator	U15207E
IE-789862-NS-EM1 Emulation Board	U16297E

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

Documents Related to Flash Memory Writing

Document Name	Document No.
PG-FP3 Flash Memory Programmer User's Manual	U13502E
PG-FP4 Flash Memory Programmer User's Manual	U15260E

Other Related Documents

Document Name	Document No.
SEMICONDUCTOR SELECTION GUIDE - Products and Packages -	X13769X
Semiconductor Device Mount Manual	**Note**
Quality Grades on NEC Semiconductor Devices	C11531E
NEC Semiconductor Device Reliability/Quality Control System	C10983E
Guide to Prevent Damage for Semiconductor Devices by Electrostatic Discharge (ESD)	C11892E

Note See the "Semiconductor Device Mount Manual" website (http://www.necel.com/pkg/en/mount/index.html)

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

CONTENTS

CHAPTER 1 GENERAL 23

1.1 Features 23
1.2 Applications 23
1.3 Ordering Information 24
1.4 Pin Configuration (Top View) 25
1.5 78K/0S Series Lineup 26
1.6 Block Diagram 29
1.7 Overview of Functions 30

CHAPTER 2 PIN FUNCTIONS 31

2.1 Pin Function List 31
2.2 Description of Pin Functions 33
2.2.1 P00 to P03 (Port 0) 33
2.2.2 P10, P11 (Port 1) 33
2.2.3 P20 to P24 (Port 2) 33
2.2.4 P30 to P33 (Port 3) 33
2.2.5 P40 to P45 (Port 4) 34
2.2.6 P70 (Port 7) 34
2.2.7 EEWE 34
2.2.8 KRREN 35
2.2.9 $\overline{\text{RESET}}$ 35
2.2.10 X1, X2 35
2.2.11 V_{DD} 35
2.2.12 V_{SS} 35
2.2.13 V_{PP} (μPD78E9862 only) 35
2.2.14 IC (mask ROM version only) 35
2.3 Pin I/O Circuits and Recommended Connection of Unused Pins 36

CHAPTER 3 CPU ARCHITECTURE 38

3.1 Memory Space 38
3.1.1 Internal program memory space 40
3.1.2 Internal data memory space 40
3.1.3 Special function register (SFR) area 41
3.1.4 Data memory addressing 41
3.2 Processor Registers 43
3.2.1 Control registers 43
3.2.2 General-purpose registers 46
3.2.3 Special function registers (SFRs) 47
3.3 Instruction Address Addressing 50
3.3.1 Relative addressing 50

3.3.2 Immediate addressing....51
3.3.3 Table indirect addressing....52
3.3.4 Register addressing....52
3.4 Operand Address Addressing....53
3.4.1 Direct addressing....53
3.4.2 Short direct addressing....54
3.4.3 Special function register (SFR) addressing....55
3.4.4 Register addressing....56
3.4.5 Register indirect addressing....57
3.4.6 Based addressing....58
3.4.7 Stack addressing....58

CHAPTER 4 EEPROM (DATA MEMORY)....59

4.1 Memory Space....59
4.2 EEPROM Configuration....59
4.3 EEPROM Control Register....61
4.4 8-Bit Timer 80....63
4.4.1 Configuration of 8-bit timer 80....63
4.4.2 Control registers of 8-bit timer 80....64
4.5 Notes for EEPROM Writing....65

CHAPTER 5 PORT FUNCTIONS....68

5.1 Functions of Ports....68
5.2 Port Configuration....70
5.2.1 Port 0....70
5.2.2 Port 1....71
5.2.3 Port 2....72
5.2.4 Port 3....75
5.2.5 Port 4....78
5.2.6 Port 7....79
5.3 Port Function Control Registers....80
5.4 Operation of Port Functions....82
5.4.1 Writing to I/O port....82
5.4.2 Reading from I/O port....82
5.4.3 Arithmetic operation of I/O port....82

CHAPTER 6 CLOCK GENERATOR....83

6.1 Clock Generator Functions....83
6.2 Clock Generator Configuration....83
6.3 Clock Generator Control Register....84
6.4 System Clock Oscillator....85
6.4.1 System clock oscillator....85
6.4.2 Examples of incorrect resonator connection....86
6.4.3 Frequency divider....87

6.5 Clock Generator Operation 88
6.6 Changing Setting of CPU Clock 89
6.6.1 Time required for switching CPU clock 89
6.6.2 Switching CPU clock 89

CHAPTER 7 16-BIT TIMER/EVENT COUNTER 0 90

7.1 Functions of 16-Bit Timer/Event Counter 0 90
7.2 Configuration of 16-Bit Timer/Event Counter 0 91
7.3 Control Registers of 16-Bit Timer/Event Counter 0 94
7.4 Operation of 16-Bit Timer/Event Counter 0 100
7.4.1 Interval timer operation 100
7.4.2 PPG output operations 102
7.4.3 Pulse width measurement operations 104
7.4.4 External event counter operation 111
7.4.5 Square-wave output operation 113
7.5 Cautions on 16-Bit Timer/Event Counter 0 114

CHAPTER 8 16-BIT TIMER D 118

8.1 Features 118
8.2 Function Overview 118
8.3 Basic Configuration of 16-Bit Timer D 118
8.4 16-Bit Timer D 119
8.5 16-Bit Timer D Control Register 122
8.6 16-Bit Timer D Operation 123
8.7 Application Examples 125
8.8 Cautions 125

CHAPTER 9 8-BIT TIMER/EVENT COUNTER 5 126

9.1 Functions of 8-Bit Timer/Event Counter 5 126
9.2 Configuration of 8-Bit Timer/Event Counter 5 127
9.3 Control Registers of 8-Bit Timer/Event Counter 5 128
9.4 Operations of 8-Bit Timer/Event Counter 5 131
9.4.1 Interval timer (8-bit) operation 131
9.4.2 External event counter operation 133
9.4.3 Square-wave output (8-bit resolution) operation 134
9.4.4 PWM output operation 135
9.5 Cautions on 8-Bit Timer/Event Counter 5 137

CHAPTER 10 WATCHDOG TIMER 138

10.1 Watchdog Timer Functions 138
10.2 Watchdog Timer Configuration 139
10.3 Watchdog Timer Control Registers 140
10.4 Operation of Watchdog Timer 142
10.4.1 Operation as watchdog timer 142
10.4.2 Operation as interval timer 143

CHAPTER 11 SERIAL INTERFACE 2 144

11.1 Functions of Serial Interface 2 144
11.2 Configuration of Serial Interface 2 145
11.3 Control Registers of Serial Interface 2 149
11.4 Operation of Serial Interface 2 156
11.4.1 Operation stopped mode 156
11.4.2 Asynchronous serial interface (UART) mode 157
11.4.3 3-wire serial I/O mode 171

CHAPTER 12 POWER-ON-CLEAR CIRCUITS 177

12.1 Power-on-Clear Circuit Functions 177
12.2 Configuration of Power-on-Clear Circuit 177
12.3 Power-on-Clear Circuit Control Registers 179
12.4 Power-on-Clear Circuit Operation 181
12.4.1 Power-on-clear (POC) circuit operation 181
12.4.2 Operation of low-voltage detector (LVI) 184

CHAPTER 13 BIT SEQUENTIAL BUFFER 186

13.1 Bit Sequential Buffer Functions 186
13.2 Bit Sequential Buffer Configuration 186
13.3 Bit Sequential Buffer Control Registers 187
13.4 Bit Sequential Buffer Operation 188
13.5 Notes 190

CHAPTER 14 KEY RETURN CIRCUIT 191

14.1 Key Return Circuit Function 191
14.2 Key Return Circuit Configuration and Operation 192

CHAPTER 15 INTERRUPT FUNCTIONS 194

15.1 Interrupt Function Types 194
15.2 Interrupt Sources and Configuration 194
15.3 Interrupt Function Control Registers 197
15.4 Interrupt Servicing Operation 202
15.4.1 Non-maskable interrupt request acknowledgement operation 202
15.4.2 Maskable interrupt request acknowledgement operation 204
15.4.3 Multiple interrupt servicing 206
15.4.4 Interrupt request pending 208

CHAPTER 16 STANDBY FUNCTION 209

16.1 Standby Function and Configuration 209
16.1.1 Standby function 209
16.1.2 Standby function control register 210
16.2 Standby Function Operation 211
16.2.1 HALT mode 211

16.2.2 STOP mode 214

CHAPTER 17 RESET FUNCTION 217

CHAPTER 18 μPD78E9862 221

18.1 Features of EEPROM (Program Memory) 222

18.1.1 Programming environment 222
18.1.2 Communication mode 223
18.1.3 On-board pin processing 226
18.1.4 Connection on EEPROM writing adapter 229

CHAPTER 19 MASK OPTIONS 232

CHAPTER 20 INSTRUCTION SET OVERVIEW 233

20.1 Operation 233

20.1.1 Operand identifiers and description methods 233
20.1.2 Description of “Operation” column 234
20.1.3 Description of “Flag” column 234

20.2 Operation List 235
20.3 Instructions Listed by Addressing Type 240

CHAPTER 21 ELECTRICAL SPECIFICATIONS (μPD789862) 243

CHAPTER 22 ELECTRICAL SPECIFICATIONS (μPD78E9862) 255

CHAPTER 23 PACKAGE DRAWING 267

CHAPTER 24 RECOMMENDED SOLDERING CONDITIONS 268

APPENDIX A DEVELOPMENT TOOLS 270

A.1 Software Package 272
A.2 Language Processing Software 272
A.3 Control Software 273
A.4 EEPROM (Program Memory) Writing Tools 274
A.5 Debugging Tools 274

A.5.1 Hardware 274
A.5.2 Software 275

A.6 Notes on Target System Design 276

APPENDIX B REGISTER INDEX 278

B.1 Register Name Index (in Alphabetical Order) 278

B.2 Register Symbol Index (in Alphabetical Order) 280

APPENDIX C REVISION HISTORY 282

LIST OF FIGURES (1/5)

Figure No.	Title	Page
2-1	Pin I/O Circuits	37
3-1	Memory Map (μPD789862)	38
3-2	Memory Map (μPD78E9862)	39
3-3	Data Memory Addressing (μPD789862)	41
3-4	Data Memory Addressing (μPD78E9862)	42
3-5	Program Counter Configuration	43
3-6	Program Status Word Configuration	43
3-7	Stack Pointer Configuration	45
3-8	Data to Be Saved to Stack Memory	45
3-9	Data to Be Restored from Stack Memory	45
3-10	General-Purpose Register Configuration	46
4-1	EEPROM Block Diagram	59
4-2	Block Diagram of 8-Bit Timer 80	60
4-3	Format of EEPROM Write Control Register 10	61
4-4	Format of 8-Bit Timer Mode Control Register 80	64
5-1	Port Types	68
5-2	Block Diagram of P00 to P03	70
5-3	Block Diagram of P10 and P11	71
5-4	Block Diagram of P20 and P21	72
5-5	Block Diagram of P22 and P23	73
5-6	Block Diagram of P24	74
5-7	Block Diagram of P30	75
5-8	Block Diagram of P31	76
5-9	Block Diagram of P32 and P33	77
5-10	Block Diagram of P40 to P45	78
5-11	Block Diagram of P70	79
5-12	Port Mode Register Format	81
5-13	Format of Pull-Up Resistor Option Register	81
6-1	Block Diagram of Clock Generator	83
6-2	Format of Processor Clock Control Register	84
6-3	External Circuit of System Clock Oscillator	85
6-4	Examples of Incorrect Resonator Connection	86
6-5	Switching Between System Clock and CPU Clock	89
7-1	Block Diagram of 16-Bit Timer/Event Counter 0	91

LIST OF FIGURES (2/5)

Figure No.	Title	Page
7-2	Format of 16-Bit Timer Mode Control Register 0 (TMC0)	95
7-3	Format of Capture/Compare Control Register 0 (CRC0)	96
7-4	Format of 16-Bit Timer Output Control Register 0 (TOC0)	97
7-5	Format of Prescaler Mode Register 0 (PRM0)	98
7-6	Format of Port Mode Register 2 (PM2)	99
7-7	Control Register Settings for Interval Timer Operation	100
7-8	Interval Timer Configuration Diagram	101
7-9	Timing of Interval Timer Operation	101
7-10	Control Register Settings for PPG Output Operation	102
7-11	PPG Output Configuration Diagram	103
7-12	PPG Output Operation Timing	103
7-13	Control Register Settings for Pulse Width Measurement with Free-Running Counter and One Capture Register	104
7-14	Configuration Diagram for Pulse Width Measurement by Free-Running Counter	105
7-15	Timing of Pulse Width Measurement Operation by Free-Running Counter and One Capture Register (with Both Edges Specified)	105
7-16	Control Register Settings for Measurement of Two Pulse Widths with Free-Running Counter	106
7-17	CR01 Capture Operation with Rising Edge Specified	107
7-18	Timing of Pulse Width Measurement Operation by Free-Running Counter (with Both Edges Specified)	107
7-19	Control Register Settings for Pulse Width Measurement with Free-Running Counter and Two Capture Registers	108
7-20	Timing of Pulse Width Measurement Operation by Free-Running Counter and Two Capture Registers (with Rising Edge Specified)	109
7-21	Control Register Settings for Pulse Width Measurement by Means of Restart	110
7-22	Timing of Pulse Width Measurement Operation by Means of Restart (with Rising Edge Specified)	110
7-23	Control Register Settings in External Event Counter Mode	111
7-24	External Event Counter Configuration Diagram	112
7-25	External Event Counter Operation Timing (with Rising Edge Specified)	112
7-26	Control Register Settings in Square-Wave Output Mode	113
7-27	Square-Wave Output Operation Timing	113
7-28	Start Timing of 16-Bit Timer Counter 0 (TM0)	114
7-29	Capture Register Data Retention Timing	114
7-30	Operation Timing of OVF0 Flag	115
8-1	Block Diagram of 16-Bit Timer D	118
8-2	Example of Timing During TMD Operation	121
8-3	Format of 16-Bit Timer Mode Control Register D (TMCD)	122
8-4	TMD Compare Operation Example	123

LIST OF FIGURES (3/5)

Figure No.	Title	Page
9-1	Block Diagram of 8-Bit Timer/Event Counter 5	126
9-2	Format of Timer Clock Select Register 5 (TCL5)	128
9-3	Format of 8-Bit Timer Mode Control Register 5 (TMC5)	129
9-4	Format of Port Mode Register 2 (PM2)	130
9-5	Interval Timer Operation Timing	131
9-6	External Event Counter Operation Timing (with Rising Edge Specified)	133
9-7	Square-Wave Output Operation Timing	134
9-8	PWM Output Operation Timing	136
9-9	Timing of Operation by CR5 Transition	137
9-10	Start Timing of 8-Bit Timer Counter 5	137
10-1	Block Diagram of Watchdog Timer	139
10-2	Format of Watchdog Timer Clock Select Register	140
10-3	Format of Watchdog Timer Mode Register	141
11-1	Block Diagram of Serial Interface 2	146
11-2	Block Diagram of Baud Rate Generator	147
11-3	Format of Serial Operating Mode Register 2	149
11-4	Format of Asynchronous Serial Interface Mode Register 2	150
11-5	Format of Asynchronous Serial Interface Status Register 2	152
11-6	Format of Baud Rate Generator Control Register 2	153
11-7	Format of Asynchronous Serial Interface Transmit/Receive Data	164
11-8	Asynchronous Serial Interface Transmission Completion Interrupt Timing	166
11-9	Asynchronous Serial Interface Reception Completion Interrupt Timing	167
11-10	Receive Error Timing	168
11-11	3-Wire Serial I/O Mode Timing	175
12-1	Block Diagram of Power-on-Clear Circuit	178
12-2	Block Diagram of Low-Voltage Detection Circuit	178
12-3	Format of Power-on-Clear Register	179
12-4	Format of Low-Voltage Detection Register	180
12-5	Format of Low-Voltage Detection Level Selection Register	180
12-6	Timing of Internal Reset Signal Generation When POC Circuit Normally Operating	182
12-7	Timing of Internal Reset Signal Generation When POC Circuit Normally Halted	182
12-8	Timing of Internal Reset Signal Generation in POC Switching Circuit	183
12-9	LVI Circuit Operation Timing	185
13-1	Block Diagram of Bit Sequential Buffer	186
13-2	Format of Bit Sequential Buffer 1 Output Control Register	187

LIST OF FIGURES (4/5)

Figure No.	Title	Page
13-3	Format of Port Mode Register 2 (PM2)	187
13-4	Operation Timing of Bit Sequential Buffer	189
13-5	Match Signal of 16-Bit Timer D	190
14-1	Block Diagram of Key Return Circuit	192
14-2	Format of Key Return Edge Detection Register	193
15-1	Basic Configuration of Interrupt Function	196
15-2	Format of Interrupt Request Flag Register	198
15-3	Format of Interrupt Mask Flag Register	199
15-4	Format of External Interrupt Mode Register 0	200
15-5	Program Status Word Configuration	201
15-6	Flowchart from Non-Maskable Interrupt Request Generation to Acknowledgement (INTWDT)	203
15-7	Timing of Non-Maskable Interrupt Request Acknowledgement	203
15-8	Acknowledgement of Non-Maskable Interrupt Request	203
15-9	Interrupt Request Acknowledgement Processing Algorithm	205
15-10	Interrupt Request Acknowledgement Timing (Example of MOV A, r)	206
15-11	Interrupt Request Acknowledgement Timing (When Interrupt Request Flag Is Set at Last Clock During Instruction Execution)	206
15-12	Example of Multiple Interrupts	207
16-1	Format of Oscillation Stabilization Time Selection Register	210
16-2	Releasing HALT Mode by Interrupt	212
16-3	Releasing HALT Mode by $\overline{\text{RESET}}$ Input	213
16-4	Releasing STOP Mode by Interrupt	215
16-5	Releasing STOP Mode by $\overline{\text{RESET}}$ Input	216
17-1	Block Diagram of Reset Function	217
17-2	Reset Timing by $\overline{\text{RESET}}$ Input	218
17-3	Reset Timing by Watchdog Timer Overflow	218
17-4	Reset Timing by $\overline{\text{RESET}}$ Input in STOP Mode	218
18-1	Environment for Writing Program to EEPROM (Program Memory)	222
18-2	Communication Mode Selection Format	223
18-3	Example of Connection with Dedicated Flash Programmer	224
18-4	V_{PP} Pin Connection Example	226
18-5	Signal Conflict (Input Pin of Serial Interface)	227
18-6	Abnormal Operation of Other Device	227

LIST OF FIGURES (5/5)

Figure No.	Title	Page
18-7	Signal Conflict ($\overline{\text{RESET}}$ Pin)	228
18-8	Wiring Example for EEPROM Writing Adapter in 3-Wire Serial I/O Mode	229
18-9	Wiring Example for EEPROM Writing Adapter in UART Mode	230
18-10	Wiring Example for EEPROM Writing Adapter in Pseudo 3-Wire Mode	231
A-1	Development Tools	271
A-2	Distance Between In-Circuit Emulator and Conversion Adapter	276
A-3	Connection Condition of Target System	277

LIST OF TABLES (1/2)

Table No.	Title	Page
2-1	Types of Pin I/O Circuits and Recommended Connection of Unused Pins	36
3-1	Internal ROM Capacity	40
3-2	Vector Table	40
3-3	Special Function Registers	48
4-1	EEPROM Write Time	62
4-2	Configuration of 8-Bit Timer 80	63
5-1	Port Functions	69
5-2	Configuration of Port	70
5-3	Port Mode Register and Output Latch Settings When Using Alternate Functions	80
6-1	Configuration of Clock Generator	83
6-2	Maximum Time Required for Switching CPU Clock	89
7-1	Configuration of 16-Bit Timer/Event Counter 0	91
7-2	TI00/TO0/P23 Pin Valid Edge and CR00, CR01 Capture Trigger	92
7-3	TI01/P24/INTP0 Pin Valid Edge and CR00 Capture Trigger	92
8-1	Configuration of 16-Bit Timer D	118
9-1	Configuration of 8-Bit Timer/Event Counter 5	127
10-1	Inadvertent Loop Detection Time of Watchdog Timer	138
10-2	Interval Time	138
10-3	Configuration of Watchdog Timer	139
10-4	Inadvertent Loop Detection Time of Watchdog Timer	142
10-5	Interval Time of Interval Timer	143
11-1	Configuration of Serial Interface 2	145
11-2	Operation Mode Settings for Serial Interface 2	151
11-3	Example of Relationship Between System Clock and Baud Rate	154
11-4	Relationship Between ASCK2 Pin Input Frequency and Baud Rate (When BRGC2 Is Set to 80H)	155
11-5	Example of Relationship Between System Clock and Baud Rate	161
11-6	Relationship Between ASCK2 Pin Input Frequency and Baud Rate (When BRGC2 Is Set to 80H)	162
11-7	Receive Error Causes	168
13-1	Configuration of Bit Sequential Buffer	186

LIST OF TABLES (2/2)

Table No.	Title	Page
13-2	Count Clock and Match Signal High Level Width of 16-Bit Timer D	190
15-1	Interrupt Source List	195
15-2	Flags Corresponding to Interrupt Request Signal	197
15-3	Time from Generation of Maskable Interrupt Request to Servicing	204
16-1	Operation Statuses in HALT Mode	211
16-2	Operation After Releasing HALT Mode	213
16-3	Operation Statuses in STOP Mode	214
16-4	Operation After Releasing STOP Mode	216
17-1	Status of Hardware After Reset	219
18-1	Differences Between μPD78E9862 and Mask ROM Version	221
18-2	Communication Mode List	223
18-3	Pin Connection List	225
20-1	Operand Identifiers and Description Methods	233
24-1	Surface Mounting Type Soldering Conditions	268

CHAPTER 1 GENERAL

1.1 Features

- ROM and RAM capacity

Item / Product Name	Program Memory (ROM)		Data Memory	
			Internal High-Speed RAM	EEPROM™
μPD789862	Mask ROM	16 KB	512 bytes	256 bytes
μPD78E9862	EEPROM	16 KB		

- System clock: Ceramic/crystal oscillation
- Minimum instruction execution time can be changed from high-speed (0.4 μs) to low-speed (1.6 μs) at 5.0 MHz operation with system clock.
- I/O ports: 22
- Timer: 4 channels
  - 16-bit timer: 2 channels
  - 8-bit timer: 1 channel
  - Watchdog timer: 1 channel
- Serial interface: 1 channel
  3-wire serial I/O mode/UART mode selectable
- On-chip key return circuit
- On-chip power-on-clear circuit
- On-chip bit sequential buffer
- Vectored interrupt sources: 13
- Power supply voltage: V_{DD} = 1.8 to 5.5 V
- Operating ambient temperature: T_A = −40 to +85°C (μPD789862)
  T_A = −40 to +70°C (μPD78E9862)

1.2 Applications

Keyless entry and other automotive electrical equipment

1.3 Ordering Information

Part Number	Package	Internal ROM
μPD789862MC-×××-5A4	30-pin plastic SSOP (7.62 mm (300))	Mask ROM
μPD78E9862MC-5A4[Note]	30-pin plastic SSOP (7.62 mm (300))	EEPROM

★ **Note** The μPD78E9862 is provided exclusively for program development; the product life and reliability are not guaranteed. Use the μPD789862 for reliability testing and mass production.

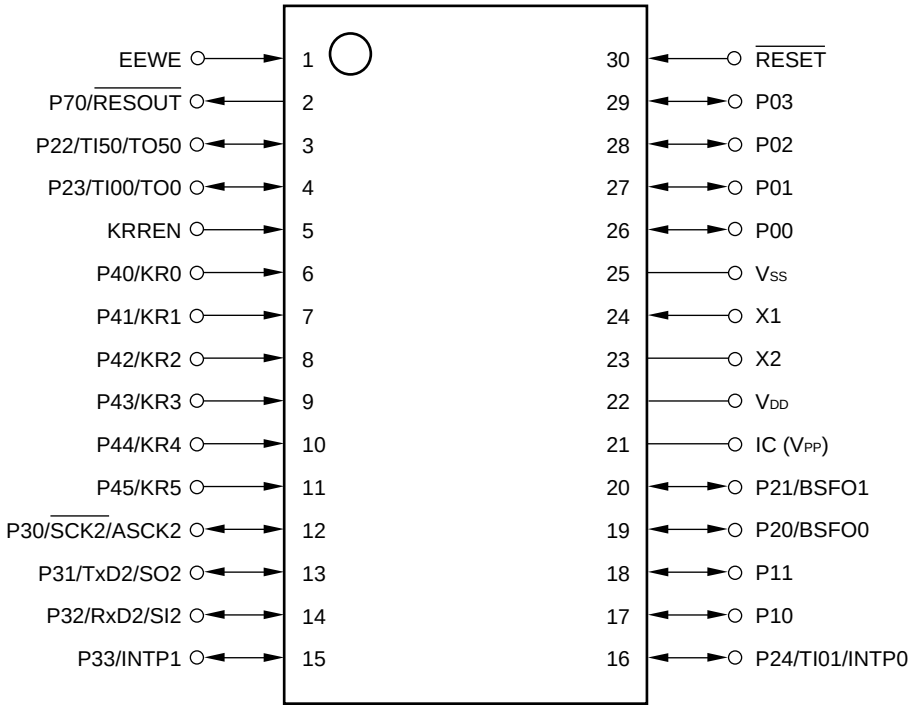
Remark ××× indicates ROM code suffix.

1.4 Pin Configuration (Top View)

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

μPD789862MC-×××-5A4

μPD78E9862MC-5A4



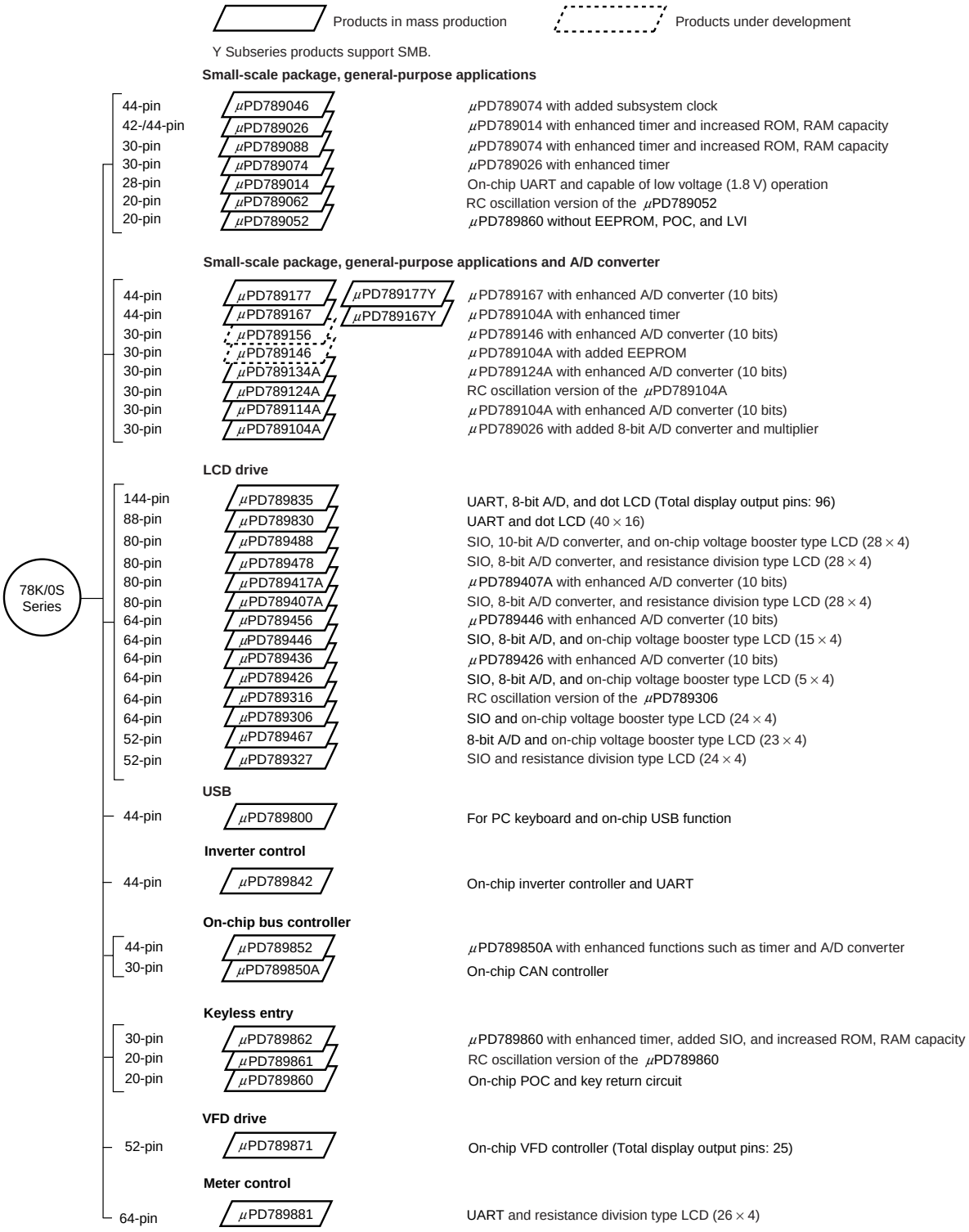
Caution Connect the IC (Internally Connected) pin directly to V_{SS}.

Remark Pin connections in parentheses are intended for the μPD78E9862.

ASCK2:	Asynchronous serial input	RESET:	Reset
BSFO0, BSFO1:	Bit sequential buffer output	RESOUT:	Reset output
EEWE:	EEPROM write enable	RxD2:	Receive data
IC:	Internally connected	SCK2:	Serial clock
INTP0, INTP1:	Interrupt from peripherals	SI2:	Serial input
KR0 to KR5:	Key return	SO2:	Serial output
KRREN:	Key return reset enable	TI00, TI01, TI50:	Timer input
P00 to P03:	Port 0	TO0, TO50:	Timer output
P10, P11:	Port 1	TxD2:	Transmit data
P20 to P24:	Port 2	V_{DD}:	Power supply
P30 to P33:	Port 3	V_{PP}:	Programming power supply
P40 to P45:	Port 4	V_{SS}:	Ground
P70:	Port 7	X1, X2:	Crystal/ceramic oscillator

★ 1.5 78K/0S Series Lineup

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



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 between the subseries are listed below.

Series for General-purpose applications and LCD drive

Subseries Name \ Function		ROM Capacity	Timer				8-Bit A/D	10-Bit A/D	Serial Interface	I/O	V_{DD} MIN. Value	Remarks
			8-Bit	16-Bit	Watch	WDT						
Small-scale package, general-purpose applications	μPD789046	16 KB	1 ch	1 ch	1 ch	1 ch	–	–	1 ch (UART: 1 ch)	34	1.8 V	–
	μPD789026	4 KB to 16 KB			–							
	μPD789088	16 KB to 32 KB	3 ch							24		
	μPD789074	2 KB to 8 KB	1 ch									
	μPD789014	2 KB to 4 KB	2 ch	–						22		
	μPD789062	4 KB							–	14		RC oscillation version
	μPD789052											–
Small-scale package, general-purpose applications and A/D converter	μPD789177	16 KB to 24 KB	3 ch	1 ch	1 ch	1 ch	–	8 ch	1 ch (UART: 1 ch)	31	1.8 V	–
	μPD789167						8 ch	–				
	μPD789156	8 KB to 16 KB	1 ch		–		–	4 ch		20		On-chip EEPROM
	μPD789146						4 ch	–				
	μPD789134A	2 KB to 8 KB					–	4 ch				RC oscillation version
	μPD789124A						4 ch	–				
	μPD789114A						–	4 ch				–
	μPD789104A						4 ch	–				
LCD drive	μPD789835	24 KB to 60 KB	6 ch	–	1 ch	1 ch	3 ch	–	1 ch (UART: 1 ch)	37	1.8 V[Note]	Dot LCD supported
	μPD789830	24 KB	1 ch	1 ch			–			30	2.7 V	
	μPD789488	32 KB to 48 KB	3 ch					8 ch	2 ch (UART: 1 ch)	45	1.8 V	–
	μPD789478	24 KB to 48 KB					8 ch	–				
	μPD789417A	12 KB to 24 KB					–	7 ch	1 ch (UART: 1 ch)	43		
	μPD789407A						7 ch	–				
	μPD789456	12 KB to 16 KB	2 ch				–	6 ch		30		
	μPD789446						6 ch	–				
	μPD789436						–	6 ch		40		
	μPD789426						6 ch	–				
	μPD789316	8 KB to 16 KB					–		2 ch (UART: 1 ch)	23		RC oscillation version
	μPD789306											–
	μPD789467	4 KB to 24 KB		–			1 ch		–	18		
	μPD789327						–		1 ch	21		

Note Flash memory version: 3.0 V

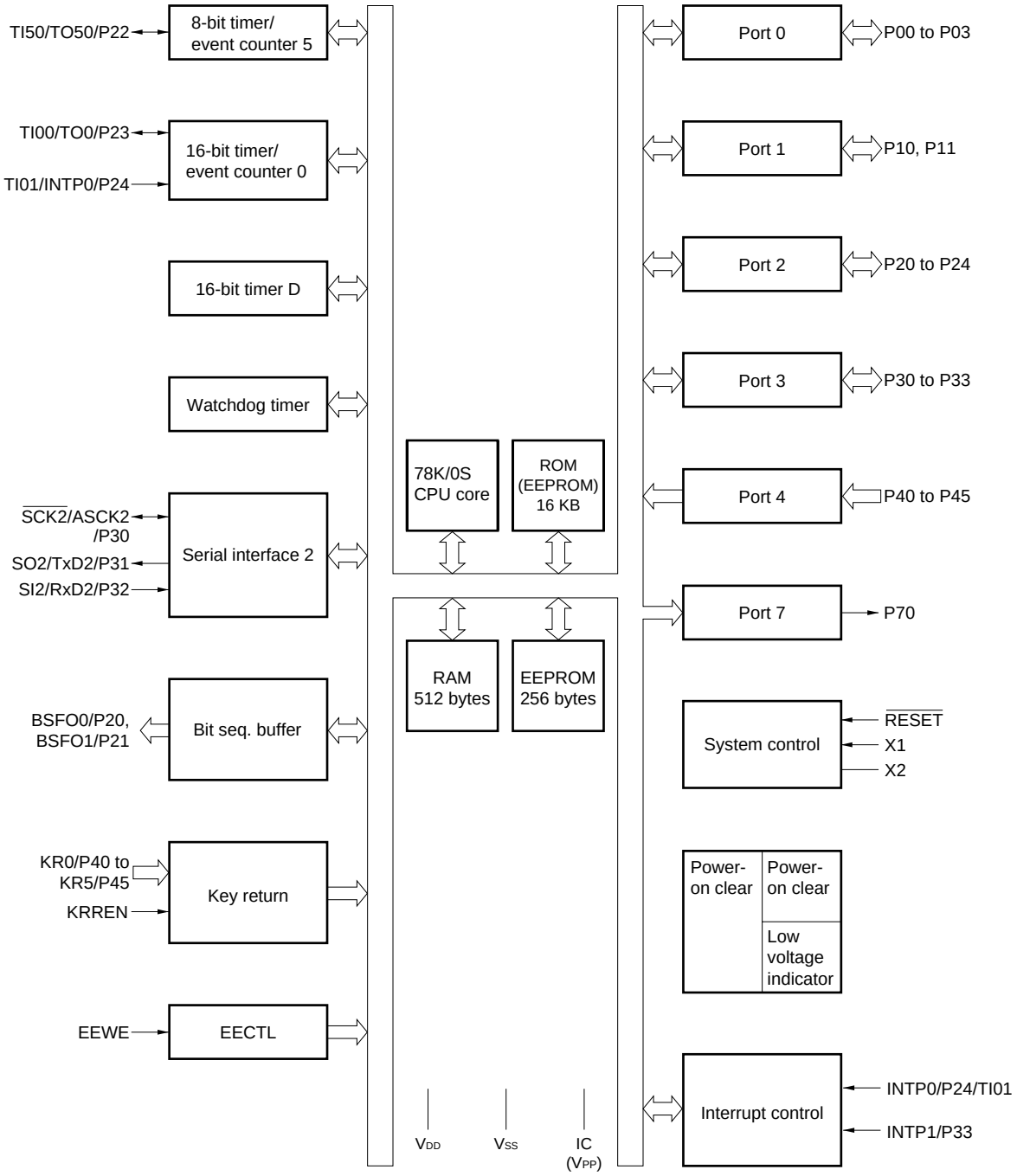
Series for ASSP

Subseries Name \ Function		ROM Capacity	Timer				8-Bit A/D	10-Bit A/D	Serial Interface	I/O	V_{DD} MIN. Value	Remarks
			8-Bit	16-Bit	Watch	WDT						
USB	μPD789800	8 KB	2 ch	–	–	1 ch	–	–	2 ch (USB: 1 ch)	31	4.0 V	–
Inverter control	μPD789842	8 KB to 16 KB	3 ch	**Note 1**	1 ch	1 ch	8 ch	–	1 ch (UART: 1 ch)	30	4.0 V	–
On-chip bus controller	μPD789852	24 KB to 32 KB	3 ch	1 ch	–	1 ch	–	8 ch	3 ch (UART: 2 ch)	31	4.0 V	–
	μPD789850A	16 KB	1 ch				4 ch	–	2 ch (UART: 1 ch)	18		
Keyless entry	μPD789861	4 KB	2 ch	–	–	1 ch	–	–	–	14	1.8 V	RC oscillation version, on-chip EEPROM
	μPD789860											On-chip EEPROM
	μPD789862	16 KB	1 ch	2 ch					1 ch (UART: 1 ch)	22		
VFD drive	μPD789871	4 KB to 8 KB	3 ch	–	1 ch	1 ch	–	–	1 ch	33	2.7 V	–
Meter control	μPD789881	16 KB	2 ch	1 ch	–	1 ch	–	–	1 ch (UART: 1 ch)	28	2.7 V[Note 2]	–

Notes 1. 10-bit timer: 1 channel

2. Flash memory version: 3.0 V

1.6 Block Diagram



Remark Pin connections in parentheses are intended for the μPD78E9862.

1.7 Overview of Functions

Item \ Part Number		μPD789862	μPD78E9862
Internal memory	ROM	Mask ROM	EEPROM
		16 KB	
	High-speed RAM	512 bytes	
	EEPROM	256 bytes	
Oscillator		Ceramic/crystal oscillator	
Minimum instruction execution time		0.4/0.8/1.6 μs (@5.0 MHz operation with system clock)	
General-purpose registers		8 bits × 8 registers	
Instruction set		• 16-bit operations • Bit manipulations (such as set, reset, and test)	
I/O ports		Total: 22 CMOS I/O: 13 CMOS input: 6 CMOS output: 1 N-ch open drain: 2	
Timers		• 16-bit timer: 2 channels • 8-bit timer: 1 channel • Watchdog timer: 1 channel	
Serial interface		3-wire serial I/O mode/UART mode Selectable: 1 channel	
Power-on-clear circuit	POC circuit	Generates internal reset signal according to comparison of detection voltage with power supply voltage	
	LVI circuit	Generates interrupt request signal according to comparison of detection voltage with power supply voltage	
Bit sequential buffer		8 bits + 8 bits = 16 bits	
Key return function		Generates key return signal according to falling edge detection (only one edge selectable)	
Vectored interrupt sources	Maskable	Internal: 9, external: 2	
	Non-maskable	Internal: 1, external: 1	
Power supply voltage		V_{DD} = 1.8 to 5.5 V	
Operating ambient temperature		T_A = –40 to +85°C	T_A = –40 to +70°C
Package		30-pin plastic SSOP (7.62 mm (300))	

CHAPTER 2 PIN FUNCTIONS

2.1 Pin Function List

(1) Port pins

Pin Name	I/O	Function	After Reset	Alternate Function
P00 to P03	I/O	Port 0 4-bit I/O port Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be used by setting pull-up resistor option register 0 (PUB0).	Input	–
P10, P11	I/O	Port 1 2-bit I/O port Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be used by setting pull-up resistor option register 1 (PUB1).	Input	–
P20	I/O	Port 2 5-bit I/O port (P20 and P21 are N-ch open drain I/O ports.) Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be used by setting pull-up resistor option register 2 (PUB2).	Input	BSFO0
P21				BSFO1
P22				TI50/TO50
P23				TI00/TO0
P24				TI01/INTP0
P30	I/O	Port 3 4-bit I/O port Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be used by setting pull-up resistor option register 3 (PUB3).	Input	$\overline{\text{SCK2}}$/ASCK2
P31				TxD2/SO2
P32				RxD2/SI2
P33				INTP1
P40 to P45	Input	Port 4 6-bit input-only port On-chip pull-up resistor	Input	KR0 to KR5
P70	Output	Port 7 1-bit output-only port	Output	$\overline{\text{RESOUT}}$

(2) Non-port pins

Pin Name	I/O	Function	After Reset	Alternate Function
BSFO0, BSFO1	Output	Bit sequential buffer output	Input	P20, P21
TI50	Input	8-bit timer (TM5) input	Input	P22/TO50
TO50	Output	8-bit timer (TM5) output	Input	P22/TI50
TI00	Input	16-bit timer (TM0) input	Input	P23/TO0
TI01				P24/INTP0
TO0	Output	16-bit timer (TM0) output	Input	P23/TI00
$\overline{SCK2}$	I/O	Serial clock input/output of serial interface	Input	P30/ASCK2
SI2	Input	Serial data input of serial interface	Input	P32/RxD2
SO2	Output	Serial data output of serial interface	Input	P31/TxD2
ASCK2	Input	Serial clock input for asynchronous serial interface	Input	P30/$\overline{SCK2}$
RxD2	Input	Serial data input for asynchronous serial interface	Input	P32/SI2
TxD2	Output	Serial data output for asynchronous serial interface	Input	P31/SO2
KR0 to KR5	Input	Key return input	Input	P40 to P45
KRREN	Input	Non-maskable interrupt/reset switch input for key return	Input	–
INTP0	Input	External interrupt input for which the valid edge can be specified (rising, falling, or both rising and falling edges)	Input	P24/TI01
INTP1				P33
$\overline{RESOUT}$	Output	Reset signal output	Output	P70
EEWE	Input	Write enable input of EEPROM master data area	Input	–
X1	Input	Connecting ceramic/crystal resonator for system clock oscillation	–	–
X2	–		–	–
$\overline{RESET}$	Input	System reset input	Input	–
V_{DD}	–	Positive supply voltage	–	–
V_{SS}	–	Ground potential	–	–
IC	–	Internally connected. Connect directly to V_{SS}.	–	–
V_{PP}	–	This pin is used to set the EEPROM programming mode and applies a high voltage when a program is written or verified.	–	–

2.2 Description of Pin Functions

2.2.1 P00 to P03 (Port 0)

These pins constitute a 4-bit I/O port and can be set to input or output port mode in 1-bit units by setting port mode register 0 (PM0). When used as an input port, an on-chip pull-up resistor can be used by setting pull-up resistor option register 0 (PUB0).

2.2.2 P10, P11 (Port 1)

These pins constitute a 2-bit I/O port and can be set to input or output port mode in 1-bit units by setting port mode register 1 (PM1). When used as an input port, an on-chip pull-up resistor can be used by setting pull-up resistor option register 1 (PUB1).

2.2.3 P20 to P24 (Port 2)

These pins constitute a 5-bit I/O port. In addition, these pins function as the timer I/O, external interrupt input, and bit sequential buffer output.

P20 and P21 are the N-ch open drain I/O ports.

(1) Port mode

In port mode, P20 to P24 function as a 5-bit I/O port and can be specified as an input or output port in 1-bit units by setting port mode register 2 (PM2). When used as an input port, an on-chip pull-up resistor can be used by setting pull-up resistor option register 2 (PUB2).

(2) Control mode

In control mode, P20 to P24 function as the timer I/O, external interrupt input, and bit sequential buffer output.

(a) BSFO0, BSFO1

These pins are the bit sequential buffer output pins.

(b) TI50, TO50

These pins are the 8-bit timer/event counter 5 I/O pins.

(c) TI00, TI01, TO0

These pins are the 16-bit timer/event counter 0 I/O pins.

(d) INTP0

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

2.2.4 P30 to P33 (Port 3)

These pins constitute a 4-bit I/O port. In addition, these pins function as the external interrupt input, data I/O of the serial interface, and clock I/O.

(1) Port mode

In port mode, P30 to P33 function as a 4-bit I/O port and can be specified as an input or output port by setting port mode register 3 (PM3). When used as an input port, an on-chip pull-up resistor can be used by setting pull-up resistor option register 3 (PUB3).

(2) Control mode

In control mode, P30 to P33 function as the external interrupt input, data I/O of the serial interface, and clock I/O.

(a) SI2, SO2

SI2 and SO2 are the serial data I/O pins of the serial interface.

(b) $\overline{\text{SCK2}}$

$\overline{\text{SCK2}}$ is the serial clock I/O pin of the serial interface.

(c) RxD2, TxD2

RxD2 and TxD2 are the serial data I/O pins for the asynchronous serial interface.

(d) ASCK2

ASCK2 is the serial clock input pin for the asynchronous serial interface.

Caution When a pin is used as a serial interface pin, I/O and the output latch must be set in accordance with the function. For how to make these settings, refer to Table 11-2 Operation Mode Settings for Serial Interface 2.

(e) INTP1

INTP1 is the external interrupt input pin for which the valid edges can be specified (rising edge, falling edge, or both rising and falling edges).

2.2.5 P40 to P45 (Port 4)

These pins constitute a 6-bit input-only port, which functions as the key return input as well as a general-purpose input port.

An on-chip pull-up resistor is provided.

(1) Port mode

In port mode, P40 to P45 function as a 6-bit input-only port.

(2) Control mode

In control mode, P40 to P45 function as the key return input (KR0 to KR5).

2.2.6 P70 (Port 7)

This pin constitutes a 1-bit output-only port, which functions as the reset output as well as a general-purpose output port.

(1) Port mode

In port mode, P70 functions as a 1-bit output-only port.

(2) Control mode

In control mode, P70 functions as the reset output ($\overline{\text{RESOUT}}$).

2.2.7 EEWE

This pin is an input pin to enable writing to the EEPROM master data area.

2.2.8 KRREN

This pin is an input pin to switch between non-maskable interrupt and reset for the key return function.

2.2.9 $\overline{\text{RESET}}$

An active-low system reset signal is input to this pin.

2.2.10 X1, X2

These pins are used to connect a crystal or ceramic resonator for system clock oscillation.

To supply an external clock, input the clock to X1 and input the inverted signal to X2.

2.2.11 V_{DD}

This pin supplies positive power.

2.2.12 V_{SS}

This pin is the ground potential pin.

2.2.13 V_{PP} (μPD78E9862 only)

A high voltage should be applied to this pin when the EEPROM programming mode is set and when the program is written or verified.

Perform either of the following pin handling.

- Independently connect a 10 kΩ pull-down resistor.
- Connect to a dedicated flash programmer in the programming mode and directly to V_{SS} in the normal operation mode by using a jumper on the board.

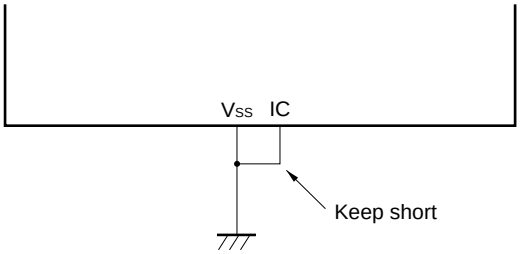
If the wiring length between the V_{PP} and V_{SS} pins is too long or if external noise is superimposed on the V_{PP} pin, the user program may not run correctly.

2.2.14 IC (mask ROM version only)

The IC (Internally Connected) pin is used to set the μPD789862 to test mode before shipment. In normal operation mode, directly connect this pin to the V_{SS} pin with as short a wiring length as possible.

If a potential difference is generated between the IC pin and the V_{SS} pin due to a long wiring length between these pins or an external noise superimposed on the IC pin, the user program may not run correctly.

- **Directly connect the IC pin to the V_{SS} pin.**



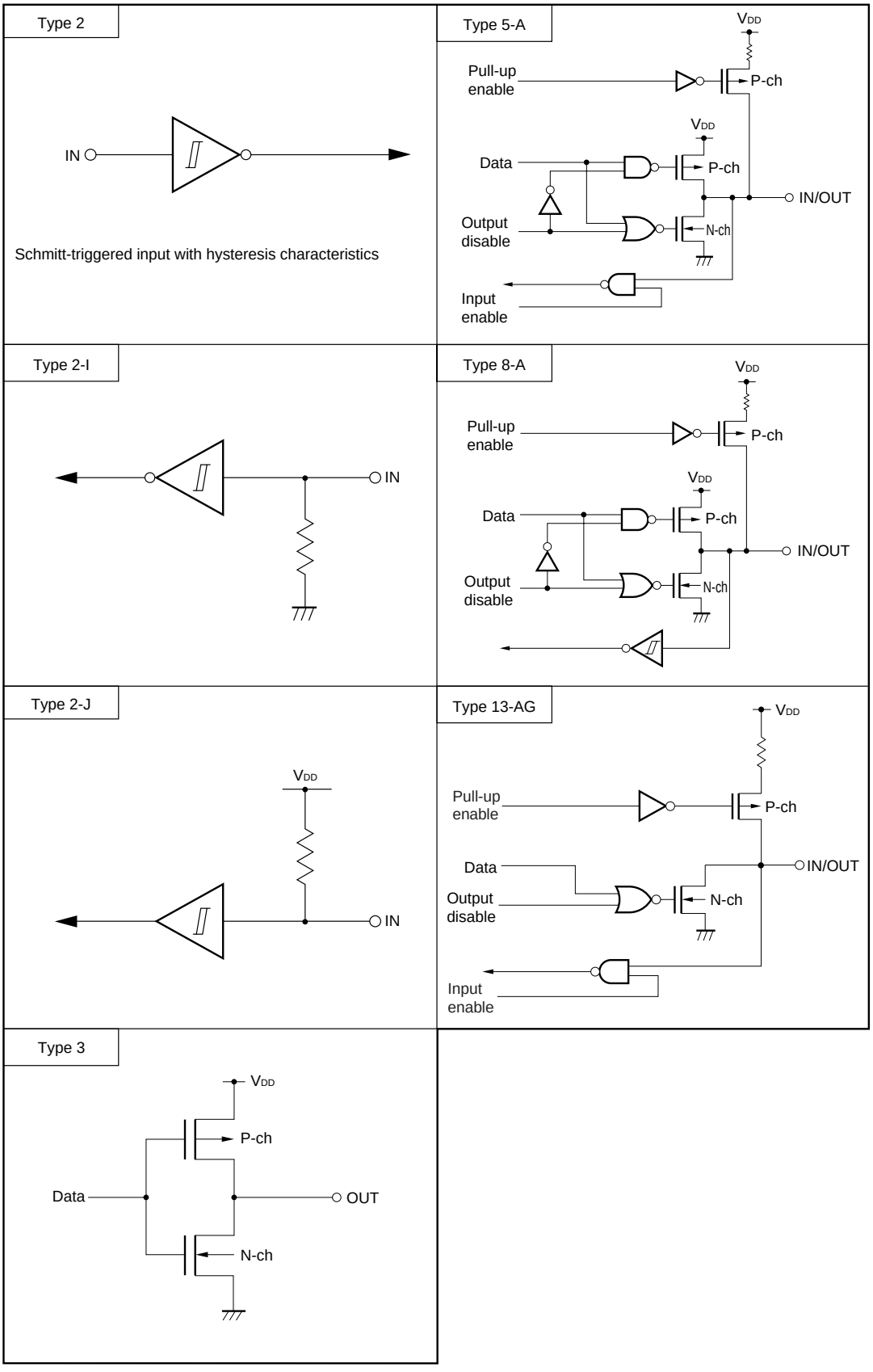
2.3 Pin I/O Circuits and Recommended Connection of Unused Pins

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

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

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00 to P03	5-A	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P10, P11			
P20/BSFO0	13-AG		
P21/BSFO1			
P22/TI50/TO50	8-A		
P23/TI00/TO0			
P24/TI01/INTP0			Input: Independently connect to V_{SS} via a resistor. Output: Leave open.
P30/$\overline{SCK2}$/ASCK2			Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P31/TxD2/SO2	5-A		
P32/RxD2/SI2	8-A		
P33/INTP1			Input: Independently connect to V_{SS} via a resistor. Output: Leave open.
P40/KR0 to P45/KR5	2-J	Input	Connect directly to V_{DD}.
P70/$\overline{RESOUT}$	3	Output	Leave open.
EEWE	2-I	Input	Connect directly to V_{SS}.
KRREN	2	Input	
$\overline{RESET}$	2-J	Input	–
IC	–	–	Connect directly to V_{SS}.
V_{PP}			Independently connect 10 kΩ pull-down resistor or connect directly to V_{SS}.

Figure 2-1. Pin I/O Circuits



CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

The μPD789862 Subseries can access up to 64 KB of memory space. Figures 3-1 and 3-2 show the memory maps.

Figure 3-1. Memory Map (μPD789862)

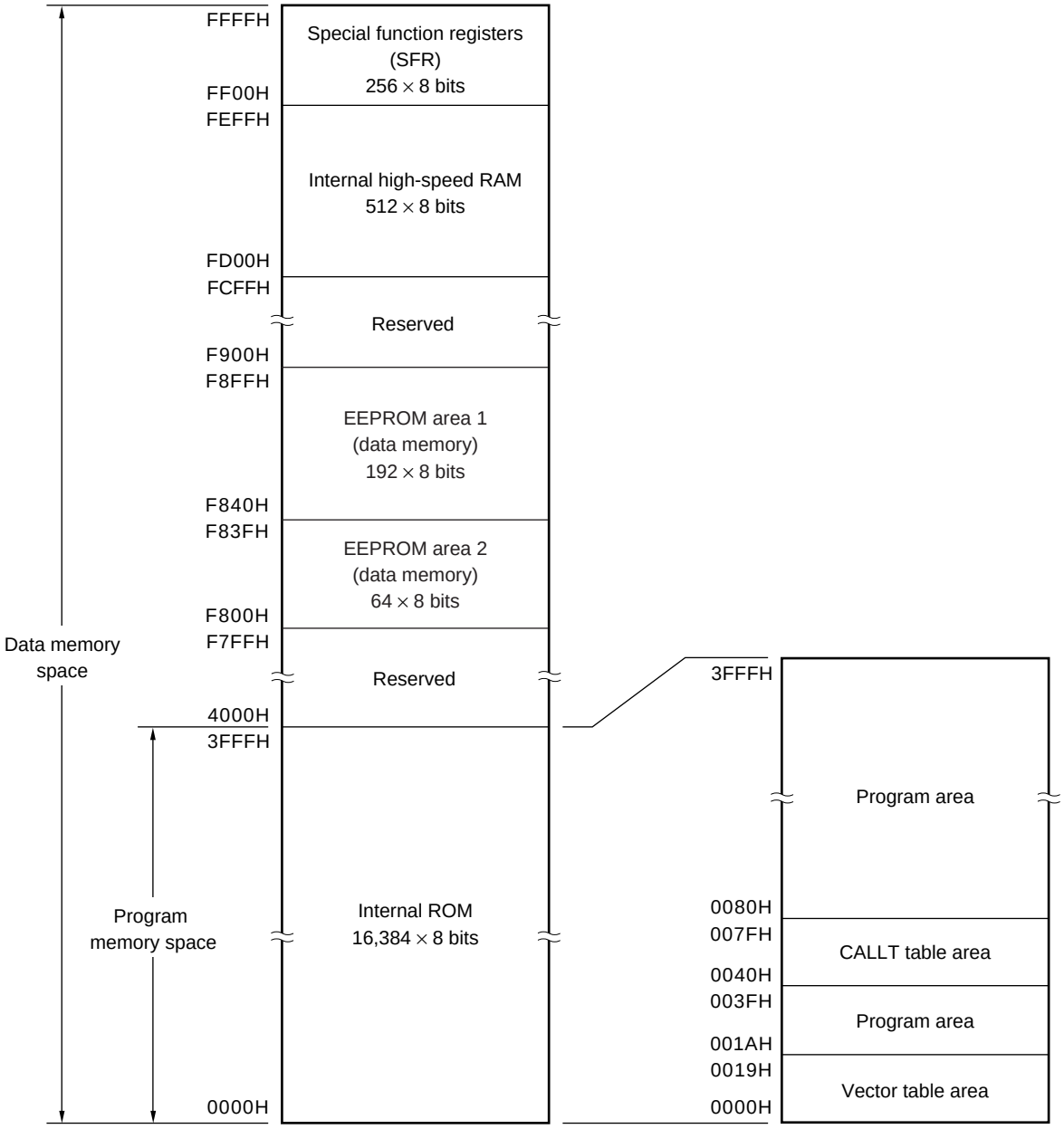
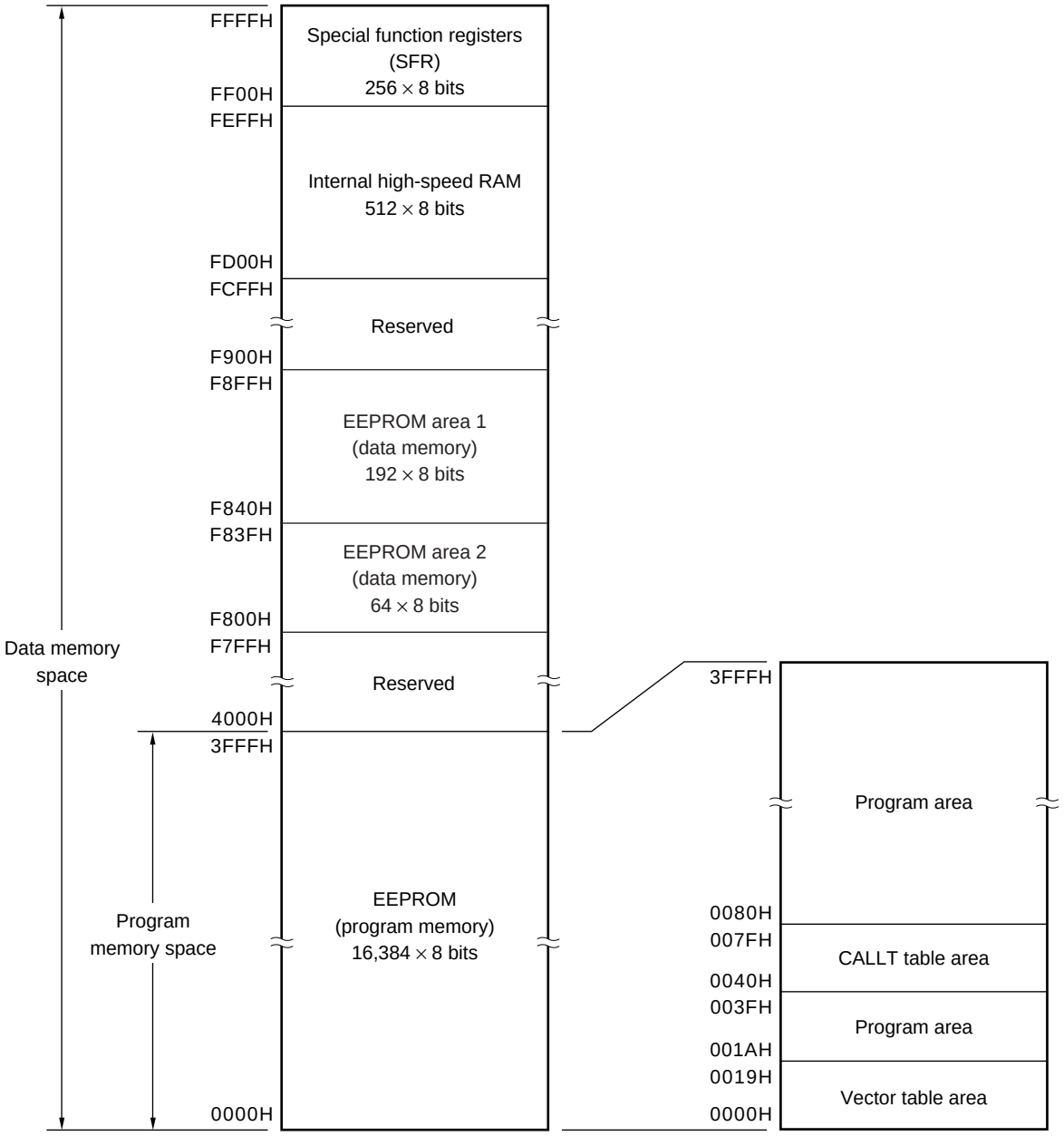


Figure 3-2. Memory Map (μPD78E9862)



3.1.1 Internal program memory space

The internal program memory space stores programs and table data. This space is usually addressed by the program counter (PC).

The μPD789862 Subseries provide the following internal ROMs (or EEPROM) containing the following capacities.

Table 3-1. Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
μPD789862	Mask ROM	16,384 × 8 bits
μPD78E9862	EEPROM	

The following areas are allocated to the internal program memory space.

(1) Vector table area

The 26-byte area of addresses 0000H to 0019H is reserved as a vector table area. This area stores program start addresses to be used when branching by $\overline{\text{RESET}}$ input or interrupt request generation. Of a 16-bit address, the lower 8 bits are stored in an even address, and the higher 8 bits are stored in an odd address.

Table 3-2. Vector Table

Vector Table Address	Interrupt Request	Vector Table Address	Interrupt Request
0000H	$\overline{\text{RESET}}$ input	000EH	INTTM01
0002H	INTKR00	0010H	INTTM50
0004H	INTWDT	0012H	INTSR20/INTCSI20
0006H	INTP0	0014H	INTST20
0008H	INTP1	0016H	INTLVI1
000AH	INTCMD	0018H	INTEE0
000CH	INTTM00		

(2) CALLT instruction table area

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

3.1.2 Internal data memory space

The μPD789862 Subseries provide the following RAMs.

(1) Internal high-speed RAM

The internal high-speed RAM is provided in the area of FD00H to FEFFH.

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

(2) EEPROM

The EEPROM is provided in the area of F800H to F8FFH.

For details of EEPROM, refer to **CHAPTER 4 EEPROM (DATA MEMORY)**.

3.1.3 Special function register (SFR) area

Special function registers (SFRs) of on-chip peripheral hardware are allocated to the area of FF00H to FFFFH (see **Table 3-3**).

3.1.4 Data memory addressing

The μPD789862 Subseries is provided with a wide range of addressing modes to make memory manipulation as efficient as possible. The data memory area (FD00H to FFFFH) can be accessed using a unique addressing mode according to its use, such as a special function register (SFR). Figures 3-3 and 3-4 illustrate the data memory addressing.

Figure 3-3. Data Memory Addressing (μPD789862)

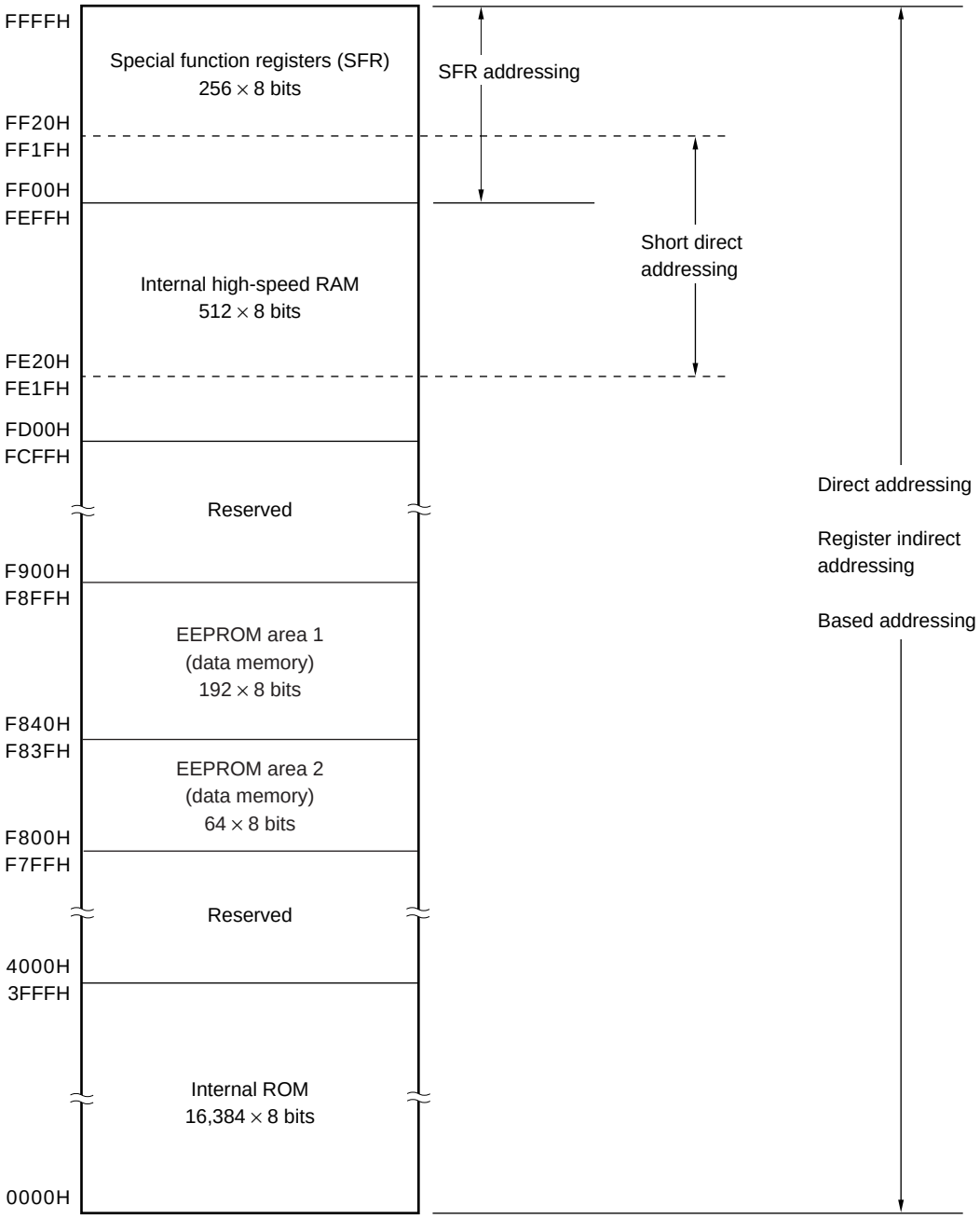
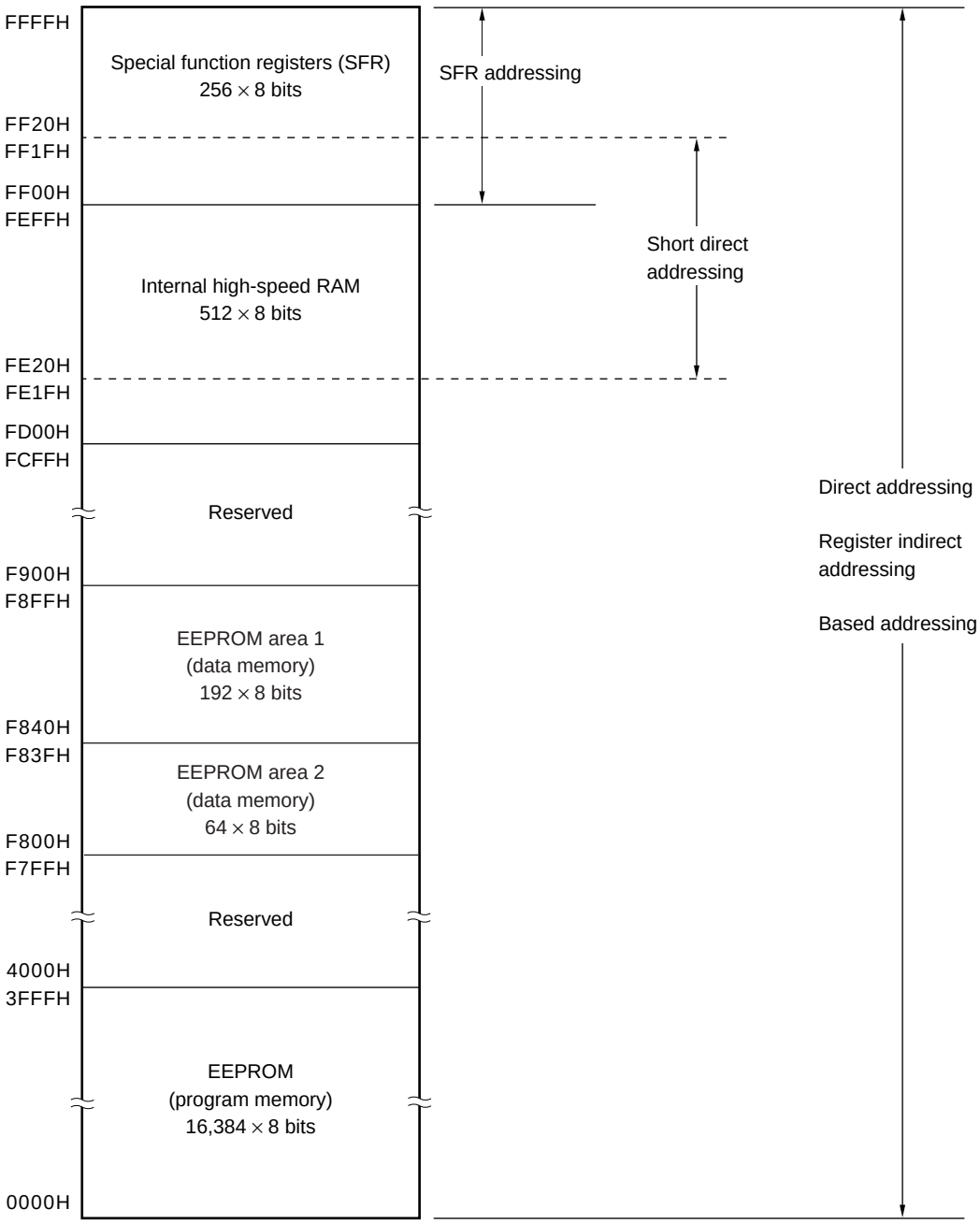


Figure 3-4. Data Memory Addressing (μPD78E9862)



3.2 Processor Registers

The μPD789862 Subseries provide the following on-chip processor registers.

3.2.1 Control registers

The control registers have special functions to control the program sequence statuses and stack memory. The control registers include a program counter, a program status word, and a stack pointer.

(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 or register contents are set.

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

Figure 3-5. Program Counter Configuration

	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 restored upon execution of the RETI and POP PSW instructions.

$\overline{\text{RESET}}$ input sets PSW to 02H.

Figure 3-6. Program Status Word Configuration

	7							0
PSW	IE	Z	0	AC	0	0	1	CY

(a) Interrupt enable flag (IE)

This flag controls interrupt request acknowledge operations of the CPU.

When IE = 0, the interrupt disabled (DI) status is set. All interrupt requests except non-maskable interrupt are disabled.

When IE = 1, the interrupt enabled (EI) status is set. Interrupt request acknowledgment is controlled with an interrupt mask flag for various interrupt sources.

This flag is reset to 0 upon DI instruction execution or interrupt acknowledgment and is set to 1 upon EI instruction execution.

(b) Zero flag (Z)

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

(c) Auxiliary carry flag (AC)

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

(d) Carry flag (CY)

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

(3) Stack pointer (SP)

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

Figure 3-7. Stack Pointer Configuration

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

The SP is decremented before writing (saving) to the stack memory and is incremented after reading (restoring) from the stack memory.

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

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

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

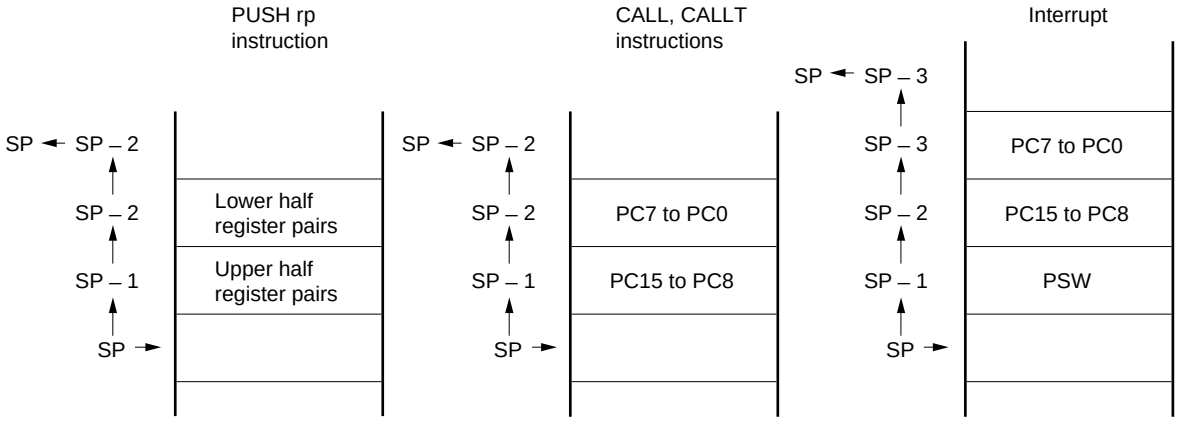
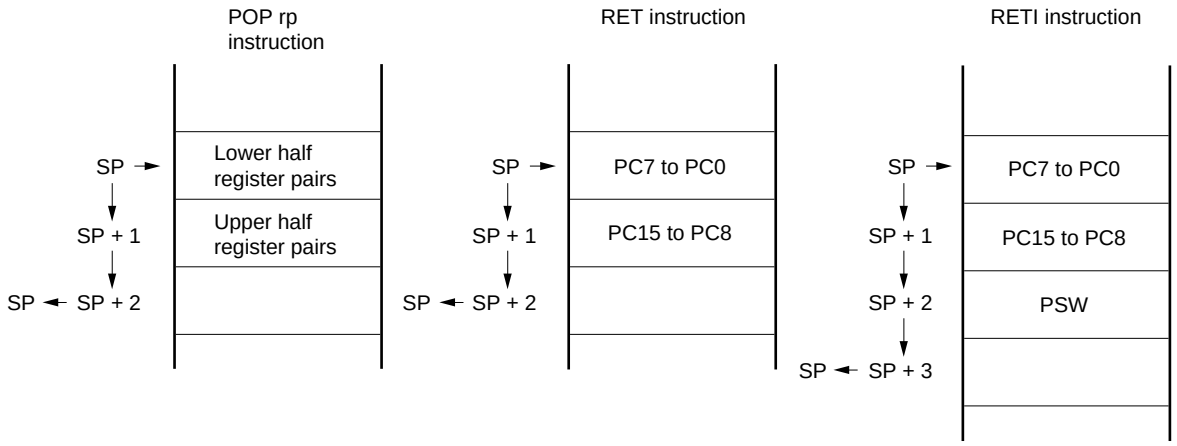


Figure 3-9. Data to Be Restored from Stack Memory



3.2.2 General-purpose registers

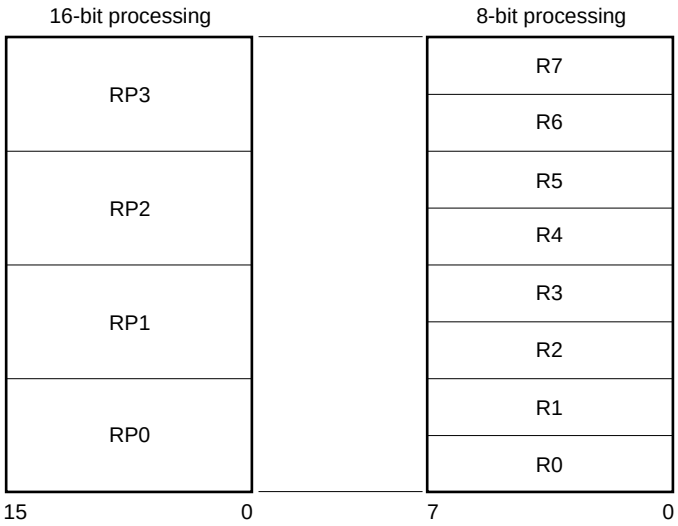
The general-purpose registers consist of eight 8-bit registers (X, A, C, B, E, D, L, and H).

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

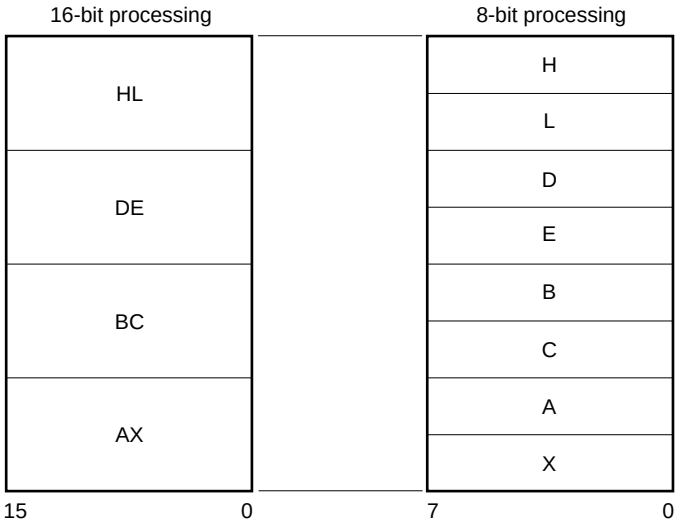
Registers can be described in terms of function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL) and absolute names (R0 to R7 and RP0 to RP3).

Figure 3-10. General-Purpose Register Configuration

(a) Absolute names



(b) Function names



3.2.3 Special function registers (SFRs)

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

The special function registers are allocated to the 256-byte area FF00H to FFFFH.

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

Each manipulation bit unit can be specified as follows.

- 1-bit manipulation
  Describes a symbol reserved by the assembler for the 1-bit manipulation instruction operand (sfr.bit). This manipulation can also be specified with an address.

- 8-bit manipulation
  Describes a symbol reserved by the assembler for the 8-bit manipulation instruction operand (sfr). This manipulation can also be specified with an address.

- 16-bit manipulation
  Describes a symbol reserved by the assembler for the 16-bit manipulation instruction operand. When specifying an address, describe an even address.

Table 3-3 lists the special function registers. The meanings of the symbols in this table are as follows.

- Symbol
  Indicates the addresses of the implemented special function registers. The symbols shown in this column are reserved words in the assembler, and have already been defined in a header file called “sfrbit.h” in the C compiler. Therefore, these symbols can be used as instruction operands if an assembler or integrated debugger is used.

- R/W
  Indicates whether the special function register can be read or written.
  R/W: Read/write
  R: Read only
  W: Write only

- Number of bits manipulated simultaneously
  Indicates the bit units (1, 8, and 16) in which the special function register can be manipulated.

- After reset
  Indicates the status of the special function register when the $\overline{\text{RESET}}$ signal is input.

Table 3-3. Special Function Registers (1/2)

Address	Special Function Register (SFR) Name	Symbol		R/W	Number of Bits Manipulated Simultaneously			After Reset
					1 Bit	8 Bits	16 Bits	
FF00H	Port 0	P0		R/W	√	√	–	00H
FF01H	Port 1	P1			√	√	–	
FF02H	Port 2	P2			√	√	–	
FF03H	Port 3	P3			√	√	–	
FF04H	Port 4	P4		R	√	√	–	
FF07H	Port 7	P7		R/W	√	√	–	
FF10H	Bit sequential buffer 1 data register L	BSFRL10	BSFR10	W	–	√	√	Undefined
FF11H	Bit sequential buffer 1 data register H	BSFRH10			–	√		
FF12H	16-bit timer counter 0	TM0		R	–	–	√	0000H
FF13H								
FF14H	16-bit timer capture/compare register 00	CR00		R/W	–	–	√	
FF15H								
FF16H	16-bit timer capture/compare register 01	CR01			–	–	√	
FF17H								
FF18H	16-bit timer counter D	TMD		R	–	–	√	
FF19H								
FF1AH	16-bit compare register D	CMD		R/W	–	–	√	
FF1BH								
FF20H	Port mode register 0	PM0			√	√	–	FFH
FF21H	Port mode register 1	PM1			√	√	–	
FF22H	Port mode register 2	PM2			√	√	–	
FF23H	Port mode register 3	PM3			√	√	–	
FF30H	Pull-up resistor option register 0	PUB0			√	√	–	00H
FF31H	Pull-up resistor option register 1	PUB1			√	√	–	
FF32H	Pull-up resistor option register 2	PUB2			√	√	–	
FF33H	Pull-up resistor option register 3	PUB3			√	√	–	
FF42H	Watchdog timer clock select register	WDCS			–	√	–	
FF50H	8-bit timer counter 5	TM5		R	–	√	–	
FF51H	8-bit timer compare register 5	CR5		R/W	–	√	–	
FF52H	Timer clock select register 5	TCL5			–	√	–	
FF53H	8-bit timer mode control register 5	TMC5			√	√	–	
FF5CH	16-bit timer mode control register D	TMCD			√	√	–	
FF60H	Bit sequential buffer 1 output control register	BSF1C			√	√	–	
FF70H	Asynchronous serial interface mode register 2	ASIM2			√	√	–	
FF71H	Asynchronous serial interface status register 2	ASIS2		R	√	√	–	
FF72H	Serial operating mode register 2	CSIM2		R/W	√	√	–	
FF73H	Baud rate generator control register 2	BRGC2			–	√	–	
FF74H	Transmit shift register 2	TXS2	SIO2[Note 1]	W	–	√	–	FFH
	Receive buffer register 2	RXB2		R	–	√	–	Undefined
FF80H	Power-on-clear register	POCF		R/W	√	√	–	00H[Note 2]

Notes 1. The SIO2 register can be read and written.

2. This value is 04H only after a power-on-clear reset.

Table 3-3. Special Function Registers (2/2)

Address	Special Function Register (SFR) Name	Symbol	R/W	Number of Bits Manipulated Simultaneously			After Reset
				1 Bit	8 Bits	16 Bits	
FF81H	Low-voltage detection register	LVIF	R/W	√	√	–	00H
FF82H	Low-voltage detection level selection register	LVIS		√	√	–	
FF90H	Key return edge detection register	EDG		√	√	–	**Note 1**
FFA6H	16-bit timer mode control register 0	TMC0		√	√	–	00H
FFA7H	Prescaler mode register 0	PRM0		–	√	–	
FFA8H	Capture/compare control register 0	CRC0		√	√	–	
FFA9H	16-bit timer output control register 0	TOC0		√	√	–	
FFD7H	8-bit timer mode control register 80	TMC80		√	√	–	
FFD8H	8-bit timer compare register 80	CR80	W	–	√	–	Undefined
FFD9H	8-bit timer counter 80	TM80	R	–	√	–	00H
FFDCH	EEPROM write control register 10	EEWC10	R/W	√	√	–	08H[Note 2]
FFE0H	Interrupt request flag register 0	IF0		√	√	–	00H
FFE1H	Interrupt request flag register 1	IF1		√	√	–	
FFE4H	Interrupt mask flag register 0	MK0		√	√	–	FFH
FFE5H	Interrupt mask flag register 1	MK1		√	√	–	
FFECH	External interrupt mode register 0	INTM0		–	√	–	00H
FFF9H	Watchdog timer mode register	WDTM		√	√	–	
FFFAH	Oscillation stabilization time selection register	OSTS		–	√	–	04H
FFFBH	Processor clock control register	PCC		√	√	–	02H

Notes 1. The value is 1000000XB after a key return reset (bit 0 retains the value before the key return reset), and 00H after other resets.

2. The default value of bit 7 (EEWEM10) depends on the condition of the EEWE pin.

3.3 Instruction Address Addressing

An instruction address is determined by the program counter (PC) contents. The PC contents are 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 address information is set to the PC and branched by the following types of addressing (for details of each instruction, refer to **78K/0S Series Instructions User's Manual (U11047E)**).

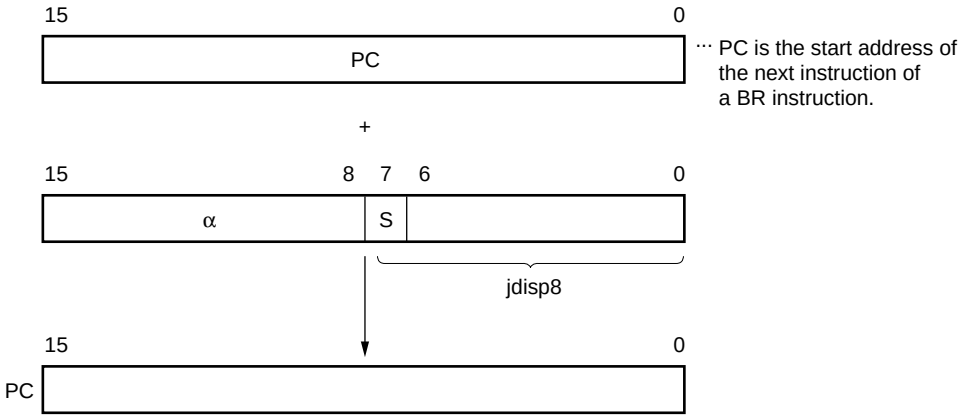
3.3.1 Relative addressing

[Function]

The value obtained by adding 8-bit immediate data (displacement value: jdisp8) of an instruction code to the start address of the following instruction is transferred to the program counter (PC) and branched. The displacement value is treated as signed two's complement data (–128 to +127) and bit 7 becomes the sign bit. In other words, the range of branch in relative addressing is between –128 and +127 of the start address of the following instruction.

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

[Illustration]



When S = 0, α indicates that all bits are 0.
When S = 1, α indicates that all bits are 1.

3.3.2 Immediate addressing

[Function]

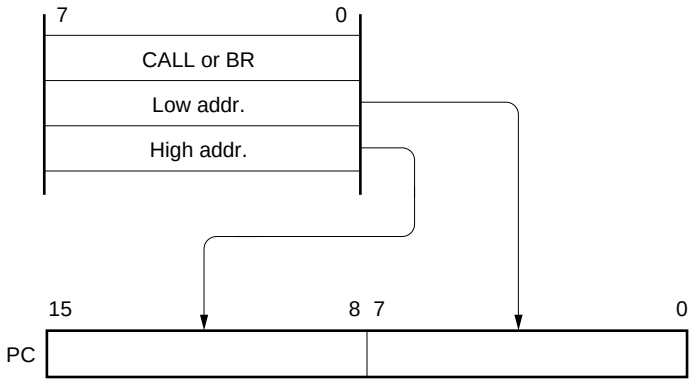
Immediate data in the instruction word is transferred to the program counter (PC) and branched.

This function is carried out when the CALL !addr16 and BR !addr16 instructions are executed.

CALL !addr16 and BR !addr16 instructions can be used to branch to all the memory spaces.

[Illustration]

In case of CALL !addr16 and BR !addr16 instructions



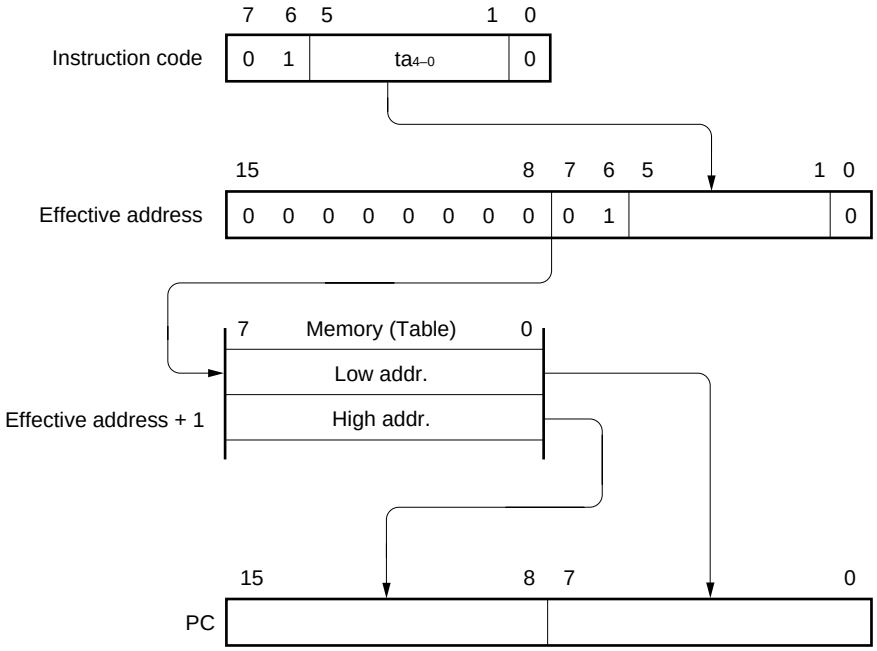
3.3.3 Table indirect addressing

[Function]

The table contents (branch destination address) of the particular location to be addressed by the immediate data of an instruction code from bit 1 to bit 5 are transferred to the program counter (PC) and branched.

Table indirect addressing is carried out when the CALLT [addr5] instruction is executed. This instruction can be used to branch to all the memory spaces according to the address stored in the memory table 40H to 7FH.

[Illustration]



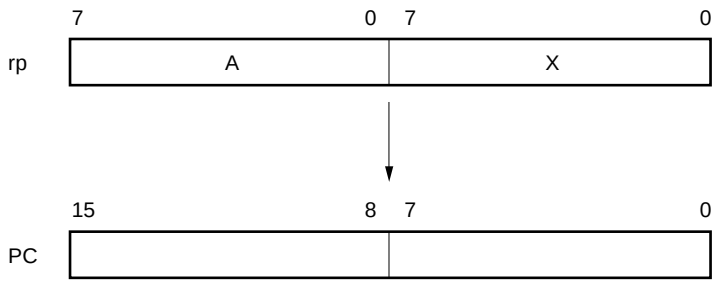
3.3.4 Register addressing

[Function]

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

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

[Illustration]



3.4 Operand Address Addressing

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

3.4.1 Direct addressing

[Function]

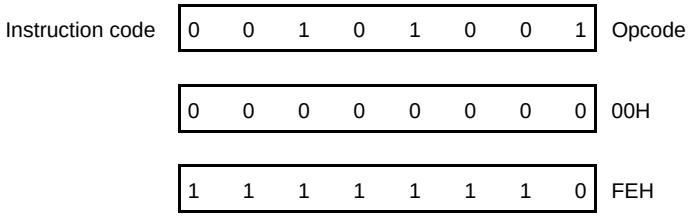
The memory indicated by immediate data in an instruction word is directly addressed.

[Operand format]

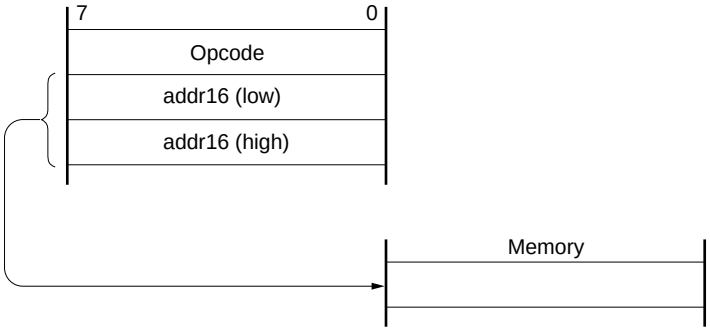
Identifier	Description
addr16	Label or 16-bit immediate data

[Description example]

MOV A, !FE00H; When setting !addr16 to FE00H



[Illustration]



3.4.2 Short direct addressing

[Function]

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

The fixed space where this addressing is applied is the 256-byte space FE20H to FF1FH. An internal high-speed RAM is mapped at FE20H to FEFFH and the special function registers (SFR) are mapped at FF00H to FF1FH.

The SFR area where short direct addressing is applied (FF00H to FF1FH) is a part of the total SFR area. In this area, ports which are frequently accessed in a program and a compare register of the timer 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. See **[Illustration]** below.

[Operand format]

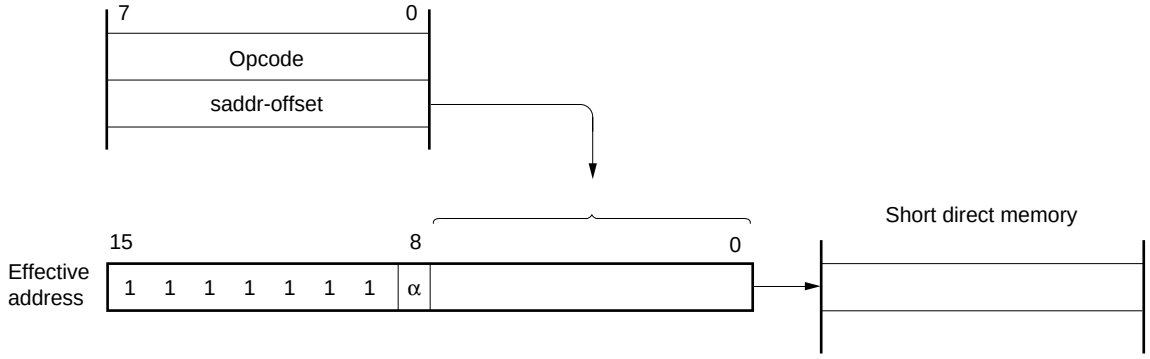
Identifier	Description
saddr	Label or FE20H to FF1FH immediate data
saddrp	Label or FE20H to FF1FH immediate data (even address only)

[Description example]

MOV FE90H, #50H; When setting saddr to FE90H and the immediate data to 50H

Instruction code	1	1	1	1	0	1	0	1	Opcode
	1	0	0	1	0	0	0	0	90H (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, $\alpha = 1$.

3.4.3 Special function register (SFR) addressing

[Function]

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

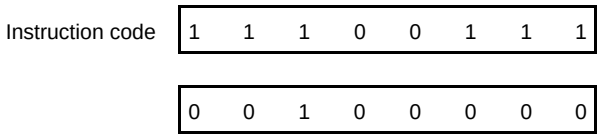
This addressing is applied to the 256-byte space FF00H to FFFFH. However, SFRs mapped at FF00H to FF1FH can also be accessed with short direct addressing.

[Operand format]

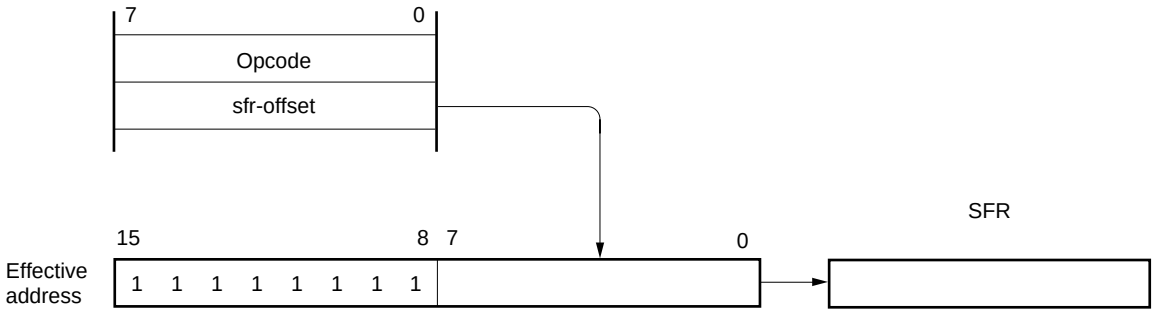
Identifier	Description
sfr	Special function register name

[Description example]

MOV PM0, A; When selecting PM0 for sfr



[Illustration]



3.4.4 Register addressing

[Function]

A general-purpose register is accessed as an operand.

The general-purpose register to be accessed is specified with the register specify code and functional name in the instruction code.

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

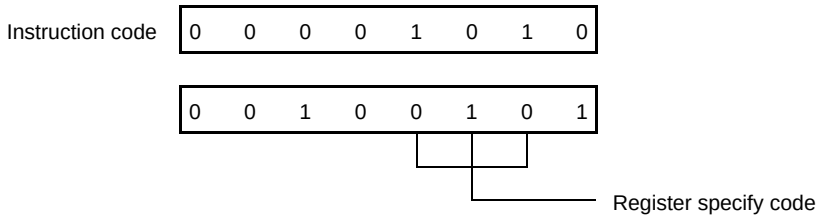
[Operand format]

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

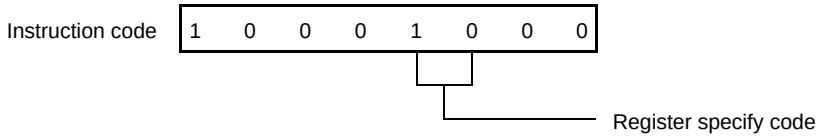
r and rp can be described with absolute names (R0 to R7 and RP0 to RP3) as well as function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL).

[Description example]

MOV A, C; When selecting the C register for r



INCW DE; When selecting the DE register pair for rp



3.4.5 Register indirect addressing

[Function]

The memory is addressed with the contents of the register pair specified as an operand. The register pair to be accessed is specified with the register pair specify code in the instruction code. This addressing can be carried out for all the memory spaces.

[Operand format]

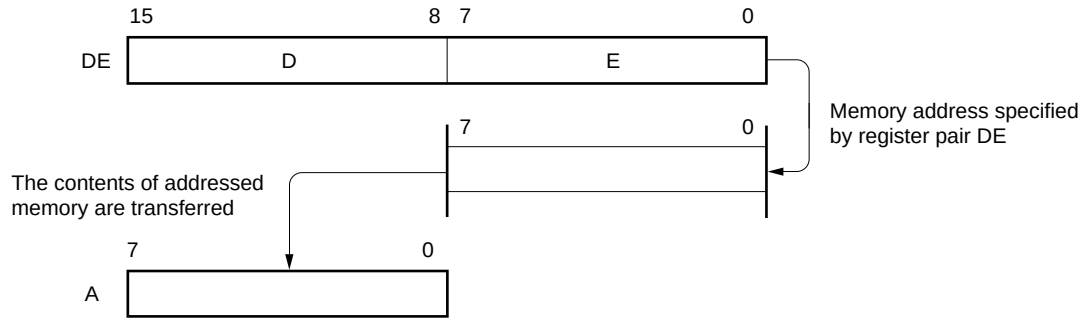
Identifier	Description
–	[DE], [HL]

[Description example]

MOV A, [DE]; When selecting register pair [DE]

Instruction code	0	0	1	0	1	0	1	1

[Illustration]



3.4.6 Based addressing

[Function]

8-bit immediate data is added to the contents of the base register, that is, the HL register pair, 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

Instruction code	0	0	1	0	1	1	0	1
	0	0	0	1	0	0	0	0

3.4.7 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/restored upon interrupt request generation.

Stack addressing can be used to access the internal high-speed RAM area only.

[Description example]

In the case of PUSH DE

Instruction code	1	0	1	0	1	0	1	0

CHAPTER 4 EEPROM (DATA MEMORY)

4.1 Memory Space

Besides internal high-speed RAM, the μPD789862 Subseries has 256 × 8 bits of electrically erasable PROM (EEPROM) on-chip as data memory.

Unlike normal RAM, EEPROM can maintain its contents even if its power supply is cut. In addition, unlike EPROM, its contents can be erased electrically.

4.2 EEPROM Configuration

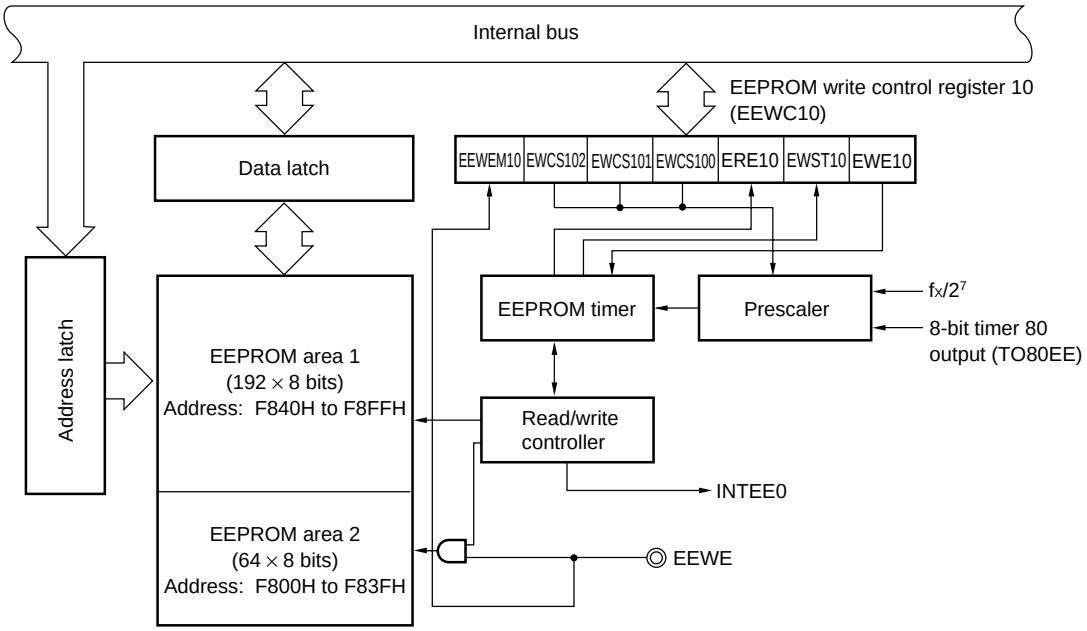
EEPROM consists of the EEPROM itself and a control block.

The control block consists of EEPROM write control register 10 (EEWC10) which controls EEPROM writing and a part that detects the termination of writing and generates an interrupt request signal (INTEE0).

Also, 8-bit timer 80 is provided as a timer dedicated to writing EEPROM. For details, refer to **4.4 8-Bit Timer 80**.

The block diagrams of EEPROM and 8-bit timer 80 are shown in Figures 4-1 and 4-2, respectively.

Figure 4-1. EEPROM Block Diagram

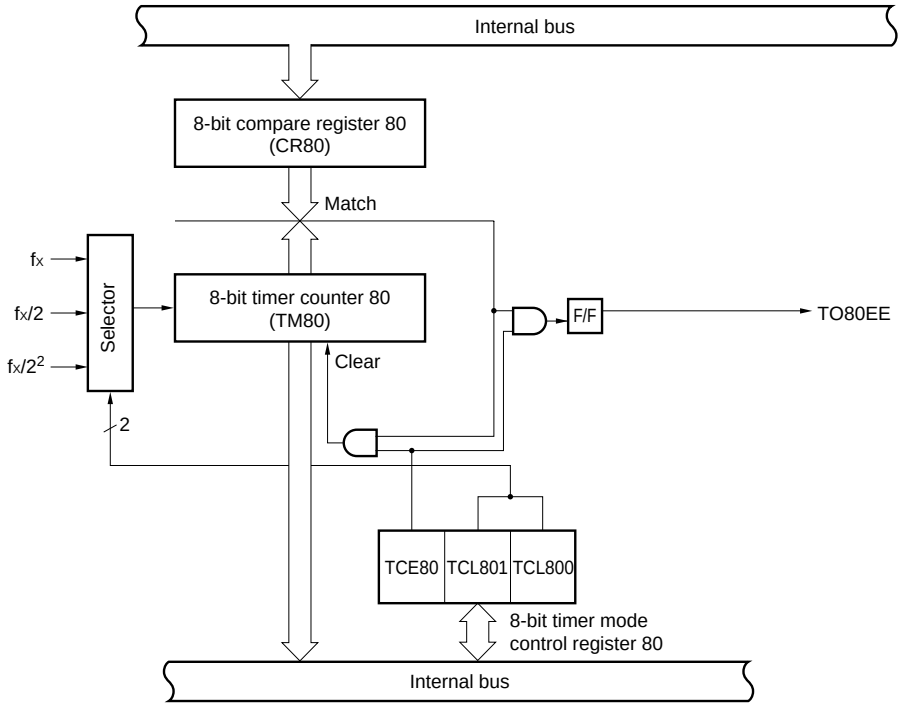


Caution When the status of the EEWE pin is low level, writing to EEPROM area 2 (64 bytes) is disabled. Only reading is enabled. If writing to EEPROM area 2 (64 bytes) is executed when the EEWE pin is low level, the instruction is ignored.

Writing to EEPROM area 2 (64 bytes) is enabled only when the status of the EEWE pin is high level. EEPROM area 1 (192 bytes) can be read/written independently of the status of the EEWE pin.

Keep the EEWE pin high level while EEPROM area 2 (64 bytes) is being written. If the EEWE pin changes to low level during writing, the EEPROM cell value of the address will be undefined.

Figure 4-2. Block Diagram of 8-Bit Timer 80



4.3 EEPROM Control Register

EEPROM is controlled by EEPROM write control register 10 (EEWC10).

EEWC10 is the register that sets the EEPROM count clock selection and EEPROM write control.

EEWC10 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets this register to 08H.

Figure 4-3 shows the format of EEPROM write control register 10. Table 4-1 shows the EEPROM write time.

Figure 4-3. Format of EEPROM Write Control Register 10

Symbol	7	6	5	4	3	<2>	<1>	<0>	Address	After reset	R/W
EEWC10	EEWEM10	EWCS102	EWCS101	EWCS100	1	ERE10	EWST10	EWE10	FFDCH	08H Note 1	R/W Note 2

EWCS102	EWCS101	EWCS100	EEPROM timer count clock selection		
				When operating at f_X = 5.0 MHz	When operating at f_X = 3.0 MHz
1	0	0	$f_X/2^7$	39.1 kHz	23.4 kHz
1	1	1	Output of 8-bit timer 80		
Other than above			Setting prohibited		

ERE10	EWE10	Write	Read	Remarks
0	0	Disabled	Disabled	EEPROM is in standby state (low power consumption mode)
0	1	Setting prohibited		
1	0	Disabled	Enabled	
1	1	Enabled	Enabled	

EWST10	EEPROM write status flag
0	Not writing to EEPROM (EEPROM can be read or written. However, writing is disabled if EWE10 = 0.)
1	Writing to EEPROM (EEPROM cannot be read or written.)

EEWEM10	EEWE pin status flag
0	Low level (Writing to EEPROM area 2 (64 bytes) is disabled.)
1	High level (Writing to EEPROM area 2 (64 bytes) is enabled.)

Notes 1. The initial value of bit 7 (EEWEM10) depends on the status of the EEWE pin.

2. Bits 1 and 7 are read only.

Caution Be sure to set bit 3 to 1.

Remark f_X: System clock oscillation frequency

★ **Table 4-1. EEPROM Write Time**

EWCS102	EWCS101	EWCS100	EEPROM Timer Count Clock			EEPROM Data Write Time[Note]		
				When operating at f_X = 5.0 MHz	When operating at f_X = 3.0 MHz		When operating at f_X = 5.0 MHz	When operating at f_X = 3.0 MHz
1	0	0	$f_X/2^7$	39.1 kHz	23.4 kHz	$2^7/f_X \times 145$	3.71 ms	6.18 ms
1	1	0	Output of 8-bit timer 80			Output of 8-bit timer 80 × 145		
Other than above			Setting prohibited					

Note Be sure to set the EEPROM write time within the range of 3.3 to 6.6 ms.

Remark f_X: System clock oscillation frequency

4.4 8-Bit Timer 80

4.4.1 Configuration of 8-bit timer 80

8-bit timer 80 includes the following hardware.

Table 4-2. Configuration of 8-Bit Timer 80

Item	Configuration
Timer counter	8 bits × 1 (TM80)
Register	Compare register: 8 bits × 1 (CR80)
Control register	8-bit timer mode control register 80 (TMC80)

(1) 8-bit timer compare register 80 (CR80)

This is an 8-bit register that compares the value set to CR80 with the 8-bit timer counter 80 (TM80) count value, and if they match, generates the inverted signal of the TO80 F/F.

CR80 is set by an 8-bit memory manipulation instruction. The values 00H to FFH can be set.

$\overline{\text{RESET}}$ input makes CR80 undefined.

Caution When rewriting CR80 in timer counter operation mode, be sure to stop the timer operation beforehand. If CR80 is rewritten in the timer operation-enabled state, a match interrupt request signal may occur at the moment of rewrite.

(2) 8-bit timer counter 80 (TM80)

This is an 8-bit register used to count count pulses.

TM80 is read with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears TM80 to 00H.

4.4.2 Control registers of 8-bit timer 80

The following register is used to control 8-bit timer 80.

- 8-bit timer mode control register 80 (TMC80)

(1) 8-bit timer mode control register 80 (TMC80)

This register controls enabling/stopping the operation of 8-bit timer counter 80 (TM80), and sets the counter clock of TM80.

TMC80 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears TMC80 to 00H.

Figure 4-4. Format of 8-Bit Timer Mode Control Register 80

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
TMC80	TCE80	0	0	0	0	TCL801	TCL800	0	FFD7H	00H	R/W

TCE80	8-bit timer counter 80 operation control
0	Operation stopped (TM80 cleared to 0)
1	Operation enabled

TCL801	TCL800	8-bit timer counter 80 count clock selection
0	0	f_X
0	1	$f_X/2$
1	0	$f_X/2^2$
1	1	Setting prohibited

Cautions 1. Be sure to set TMC80 after stopping timer operation.

2. Bits 0 and 3 to 6 must be set to 0.

Remark f_X: System clock oscillation frequency

4.5 Notes for EEPROM Writing

The following caution points pertain to writing to EEPROM.

(1) When fetching an instruction from EEPROM or stopping the system clock oscillator, be sure to do so after setting EEPROM to write-disabled (EWE10 = 0).

(2) Set the count clock in a state in which the selected clock is operating (oscillating). If the selected count clock is stopped, there is no transition to the state in which writing is possible even if the clock operation is subsequently started and EEPROM is set to write-enabled (EWE10 = 1).

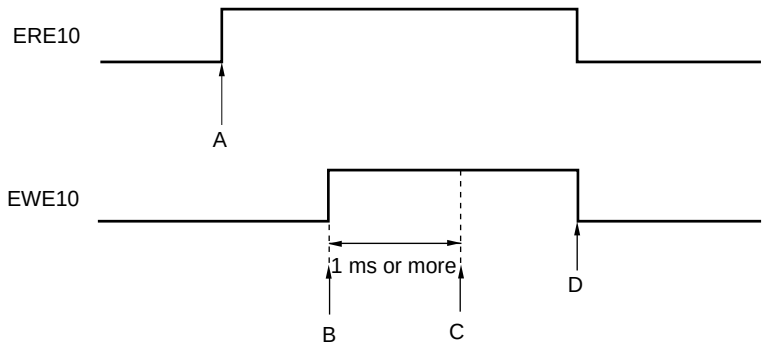
(3) Be sure to set the EEPROM write time within the range of 3.3 to 6.6 ms.

Example When setting the write time to EEPROM to 3.4 ms using 8-bit timer 80 (Condition: f_X = 4 MHz)

```
MOV  TMC80,#00H          ; Set count clock = fX
MOV  CR80,#2EH           ; TM80 cycle = 1/4 MHz × 47 = 11.75 μs
                           The clock to EEPROM is double that of the TM80 cycle (flip-flop
                           output). Therefore, the count clock to the EEPROM timer = 11.75
                           μs × 2 = 23.5 μs
MOV  EEWC10,#01101000B   ; Write time to EEPROM = 23.5 μs × 145 ≅ 3.4 ms
```

(4) When setting ERE10 and EWE10, be sure to use the following procedure. If you set these using other than the following procedure, there is no transition to the state in which writing to EEPROM is possible.

<1> Set ERE10 to 1 (In a state in which EWE10 = 0)
<2> Set EWE10 to 1 (In a state in which ERE10 = 1)
<3> Wait 1 ms or more using software
<4> Shift to state in which writing to EEPROM is possible



A (ERE10 = 1): Transition to state in which reading is possible
B (EWE10 = 1): Set count clock before this point.
C: Transition to state in which writing is possible
D: When ERE10 is cleared (ERE10 = 0), EWE10 is also cleared (EWE10 = 0).
Reading or writing is not possible in this state.

(5) When writing to EEPROM, write after confirming that EWST10 = 0.
If EEPROM is written when EWST10 = 1, the instruction is ignored.

(6) Do not execute the following operations while writing to EEPROM, as execution will cause the EEPROM cell value at that address to become undefined.

- Turn off the power
- Execute a reset
- Set ERE10 to 0
- Set EWE10 to 0
- Switch the EEPROM timer count clock

(7) Do not execute the following operation while writing to EEPROM after selecting system clock division for the EEPROM timer count clock, as execution will cause the EEPROM cell value at that address to become undefined.

- Execute a STOP instruction

(8) Do not execute the following operations while writing to EEPROM after selecting 8-bit timer 80 output for the EEPROM timer count clock, as execution will cause the EEPROM cell value at that address to become undefined.

- Execute a STOP instruction
- Stop 8-bit timer 80 timer output
- Stop 8-bit timer 80 operation

(9) Do not execute the following operations while writing to or reading from EEPROM, as execution will cause the EEPROM data read next to become undefined, and a CPU inadvertent program loop could result.

- Set ERE10 to 0
- Execute a write to EEPROM

(10) When not writing to or reading from EEPROM, it is possible to enter low-power-consumption mode by setting ERE10 to 0. In the ERE10 = 1 state, a current of about 0.27 mA (V_{DD} = 3.6 V) is always flowing. If an instruction to read from EEPROM is then executed, a further 0.9 mA current will flow, increasing the total current flow at this time to approximately 1.17 mA (V_{DD} = 3.6 V).
In the ERE10 = 1, EWE10 = 1 state, a current of about 0.3 mA (V_{DD} = 3.6 V) is always flowing. If an instruction to write to EEPROM is then executed, a further 0.7 mA current will flow, and if an instruction to read from EEPROM is executed, a further 0.9 mA current will flow, increasing the total current flow at this time to approximately 1.0 mA (V_{DD} = 3.6 V) for the former case and 1.2 mA (V_{DD} = 3.6 V) for the latter.

(11) Execution of a STOP instruction causes an automatic change to low-power-consumption mode, regardless of the ERE10 and EWE10 settings. The states of ERE10 and EWE10 at that time are maintained. During the wait time following STOP mode release, a current of approximately 300 μA (V_{DD} = 3.6 V) flows.
Executing a HALT instruction does not change the mode to low-power-consumption mode.

(12) When the status of the EEWE pin is low level, writing to EEPROM area 2 (64 bytes) is disabled. Only reading is enabled. If writing to EEPROM area 2 (64 bytes) is executed when the EEWE pin is low level, the instruction is ignored.

Writing to EEPROM area 2 (64 bytes) is enabled only when the status of the EEWE pin is high level. EEPROM area 1 (192 bytes) can be read/written independently of the status of the EEWE pin.

Keep the EEWE pin high level while EEPROM area 2 (64 bytes) is being written. If the EEWE pin changes to low level during writing, the EEPROM cell value of the address will be undefined.

CHAPTER 5 PORT FUNCTIONS

5.1 Functions of Ports

The μPD789862 Subseries provides the ports shown in Figure 5-1, enabling various methods of control.

Numerous other functions are provided that can be used in addition to the digital I/O port function. For more information on these additional functions, refer to **CHAPTER 2 PIN FUNCTIONS**.

Figure 5-1. Port Types

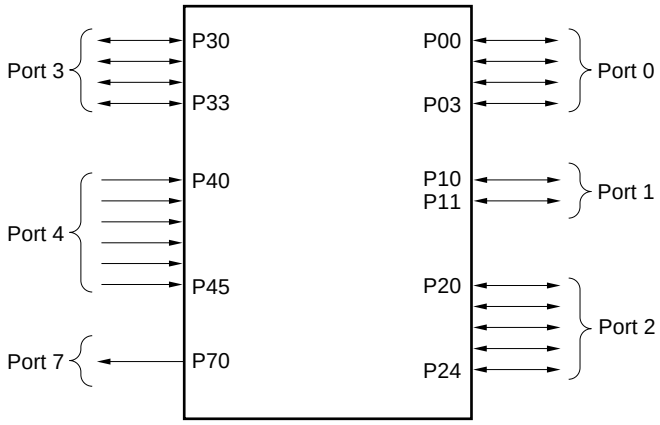


Table 5-1. Port Functions

Pin Name	I/O	Function	After Reset	Alternate Function
P00 to P03	I/O	Port 0 4-bit I/O port Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be used by setting pull-up resistor option register 0 (PUB0).	Input	–
P10, P11	I/O	Port 1 2-bit I/O port Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be used by setting pull-up resistor option register 1 (PUB1).	Input	–
P20	I/O	Port 2 5-bit I/O port (P20 and P21 are N-ch open drain I/O ports.) Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be used by setting pull-up resistor option register 2 (PUB2).	Input	BSFO0
P21				BSFO1
P22				TI50/TO50
P23				TI00/TO0
P24				TI01/INTP0
P30	I/O	Port 3 4-bit I/O port Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be used by setting pull-up resistor option register 3 (PUB3).	Input	$\overline{\text{SCK2}}$/ASCK2
P31				TxD2/SO2
P32				RxD2/SI2
P33				INTP1
P40 to P45	Input	Port 4 6-bit input-only port On-chip pull-up resistor	Input	KR0 to KR5
P70	Output	Port 7 1-bit output-only port	Output	$\overline{\text{RESOUT}}$

5.2 Port Configuration

A port includes the following hardware.

Table 5-2. Configuration of Port

Item	Configuration
Control registers	Port mode register (PMm: m = 0 to 3) Pull-up resistor option register (PUBm: m = 0 to 3)
Ports	Total: 22 (input: 6, output: 1, I/O: 15)
Pull-up resistors	Total: 15 (software control only)

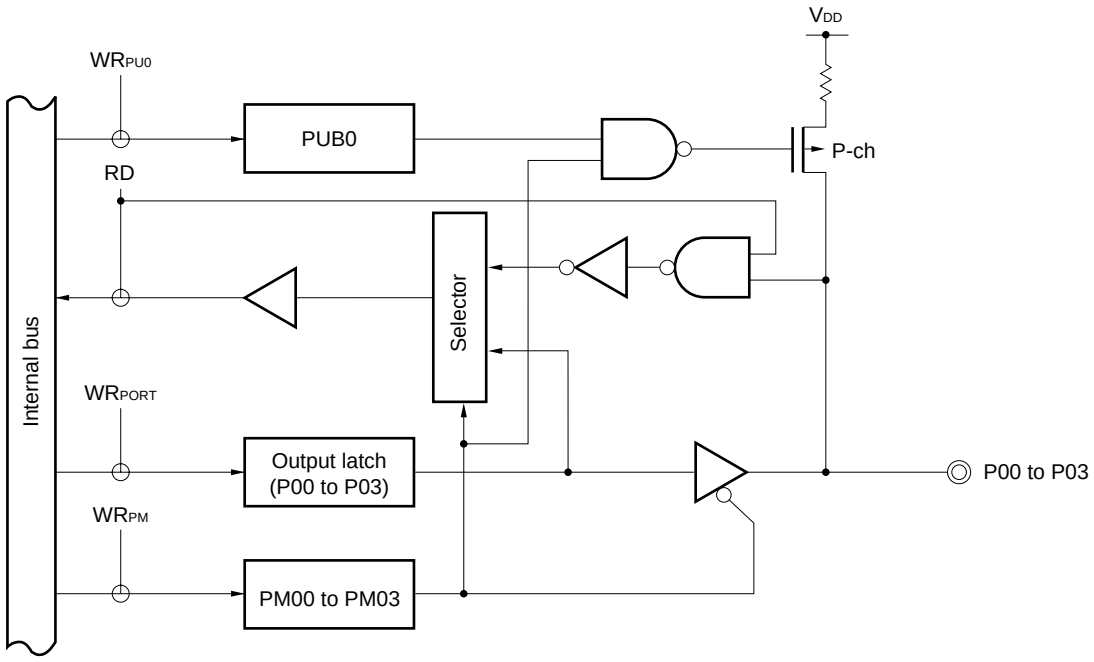
5.2.1 Port 0

This is a 4-bit I/O port with output latches. Port 0 can be specified as an input or output port in 1-bit units by using port mode register 0 (PM0). When pins P00 to P03 are used as input port pins, on-chip pull-up resistors can be connected in 4-bit units by using pull-up resistor option register 0 (PUB0).

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

Figure 5-2 shows the block diagram of port 0.

Figure 5-2. Block Diagram of P00 to P03



PUB0: Pull-up resistor option register 0
PM: Port mode register
RD: Port 0 read signal
WR: Port 0 write signal

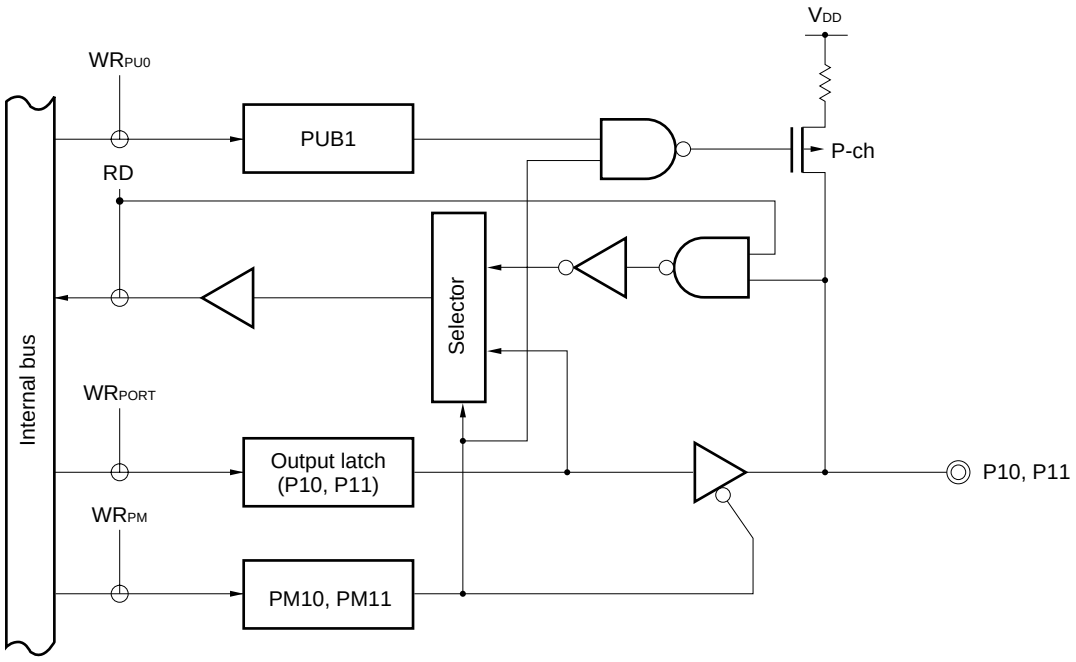
5.2.2 Port 1

This is a 2-bit I/O port with output latches. Port 1 can be specified as an input or output port in 1-bit units by using port mode register 1 (PM1). When pins P10 and P11 are used as input port pins, on-chip pull-up resistors can be connected in 2-bit units by using pull-up resistor option register 1 (PUB1).

$\overline{\text{RESET}}$ input sets port 1 to input mode.

Figure 5-3 shows the block diagram of port 1.

Figure 5-3. Block Diagram of P10 and P11



PUB1: Pull-up resistor option register 1
PM: Port mode register
RD: Port 1 read signal
WR: Port 1 write signal

5.2.3 Port 2

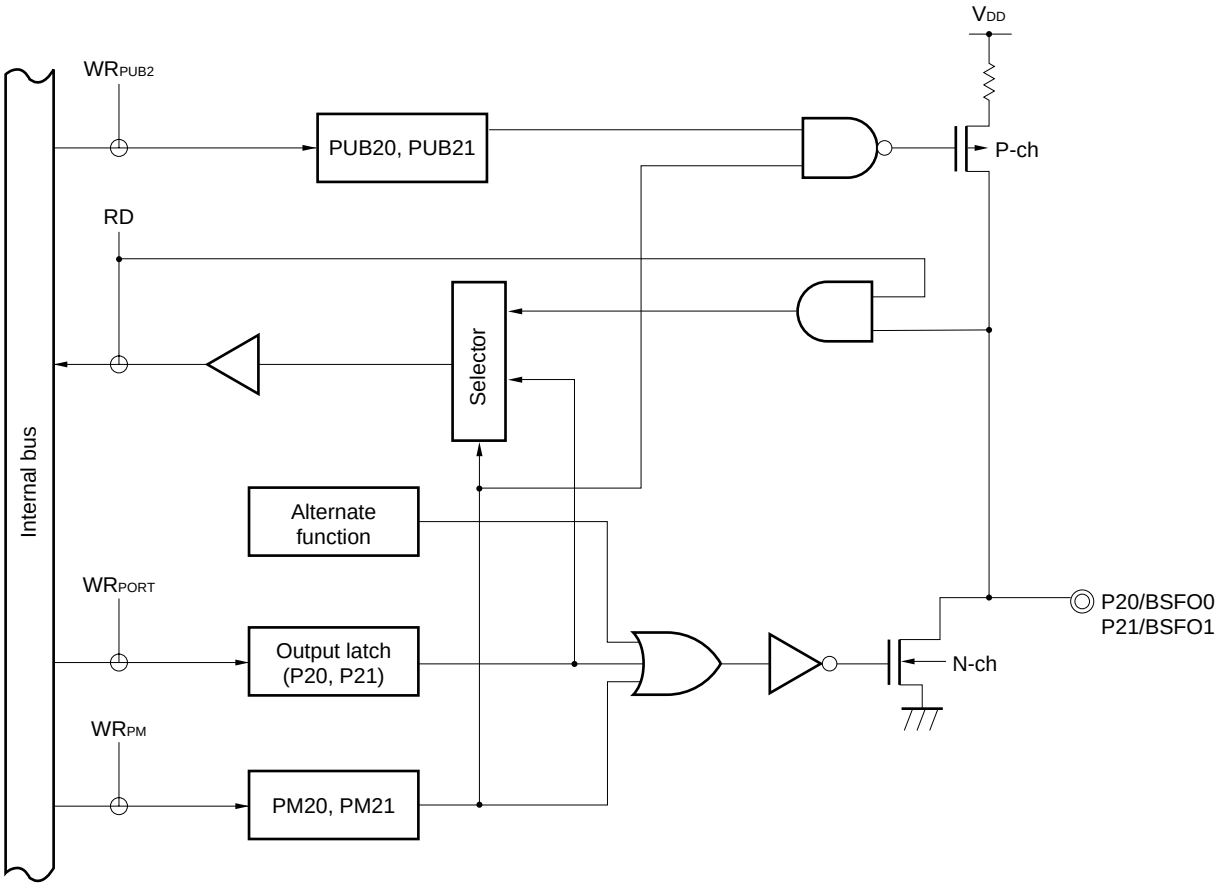
This is a 5-bit I/O port with output latches. P20 and P21 are N-ch open drain I/O port pins. Port 2 can be specified as an input or output port in 1-bit units by using port mode register 2 (PM2). When pins P20 to P24 are used as input port pins, use of on-chip pull-up resistors can be specified in 1-bit units by using pull-up resistor option register 2 (PUB2).

The port is also used for timer I/O, external interrupt input, and bit sequential buffer output.

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

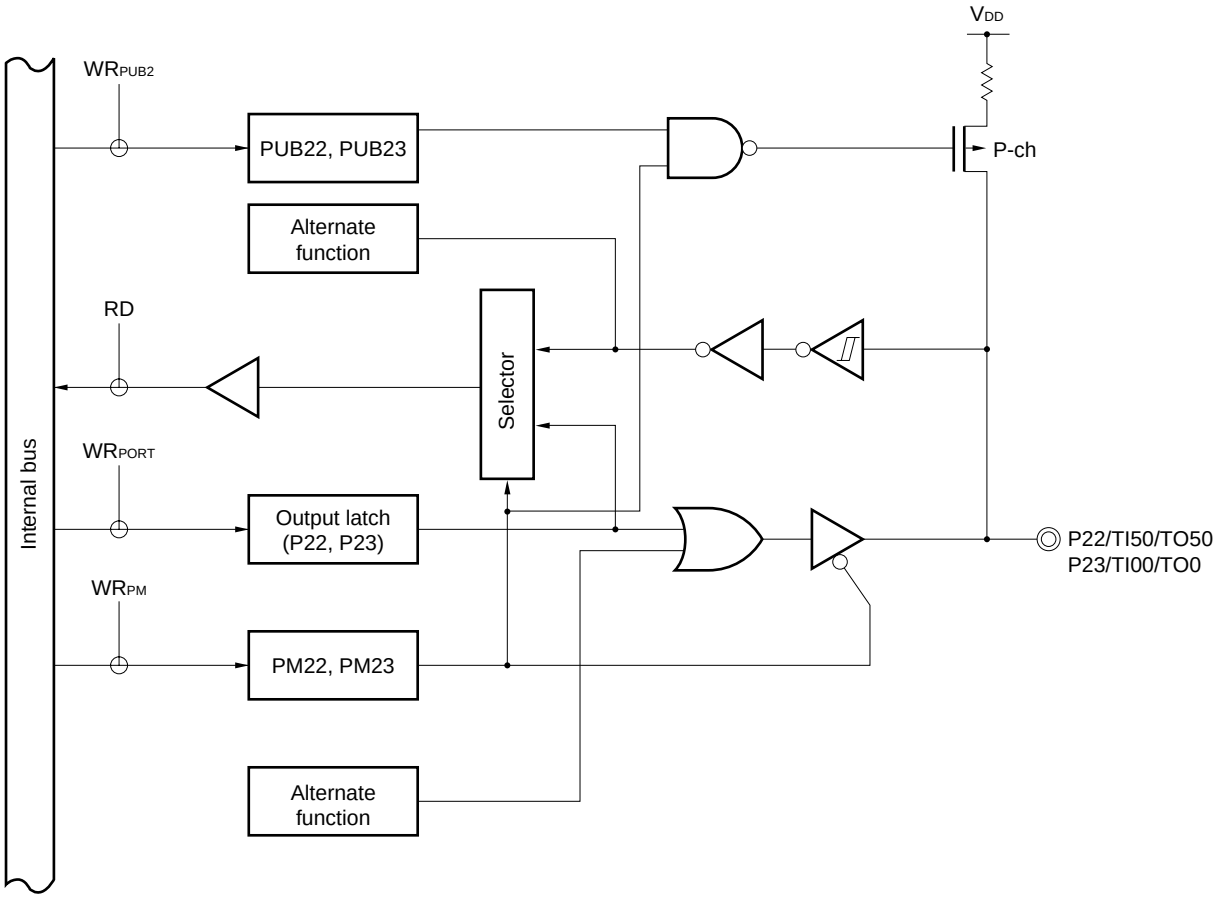
Figures 5-4 to 5-6 show block diagrams of port 2.

Figure 5-4. Block Diagram of P20 and P21



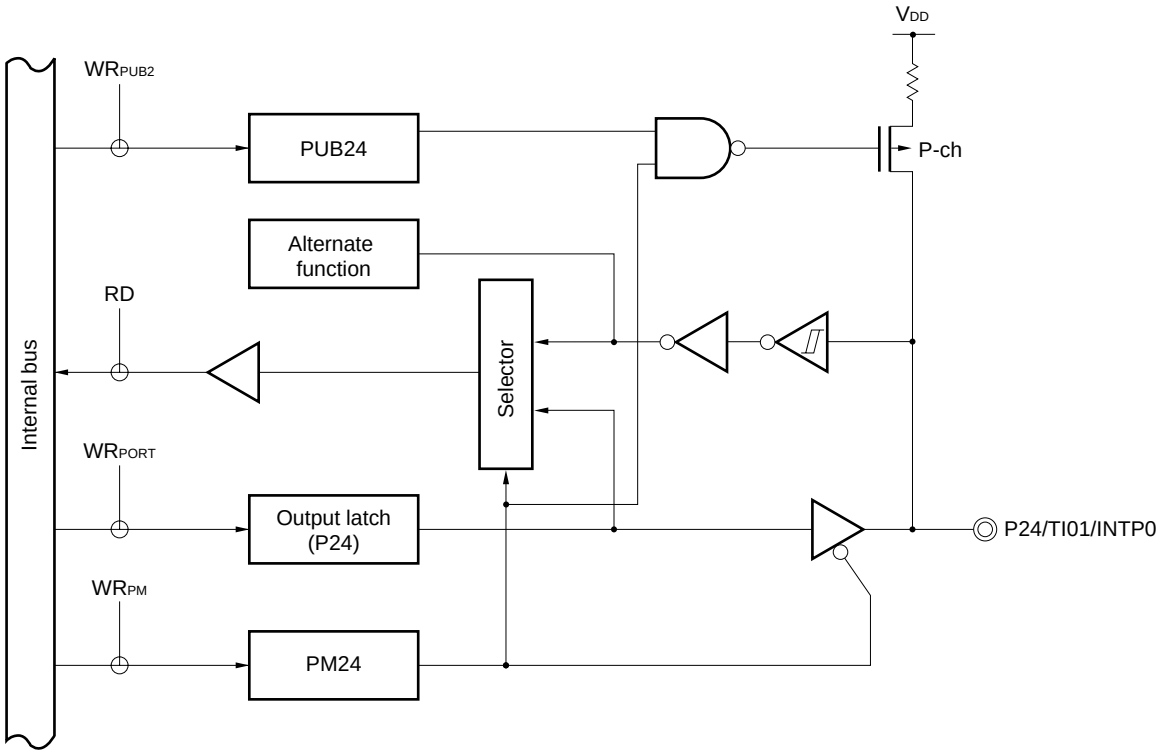
PUB2: Pull-up resistor option register 2
PM: Port mode register
RD: Port 2 read signal
WR: Port 2 write signal

Figure 5-5. Block Diagram of P22 and P23



PUB2: Pull-up resistor option register 2
PM: Port mode register
RD: Port 2 read signal
WR: Port 2 write signal

Figure 5-6. Block Diagram of P24



PUB2: Pull-up resistor option register 2
PM: Port mode register
RD: Port 2 read signal
WR: Port 2 write signal

5.2.4 Port 3

This is a 4-bit I/O port with output latches. Port 3 can be specified as an input or output port in 1-bit units by using port mode register 3 (PM3). Use of on-chip pull-up resistors can be specified for pins P30 to P33 in 1-bit units by using pull-up resistor option register 3 (PUB3).

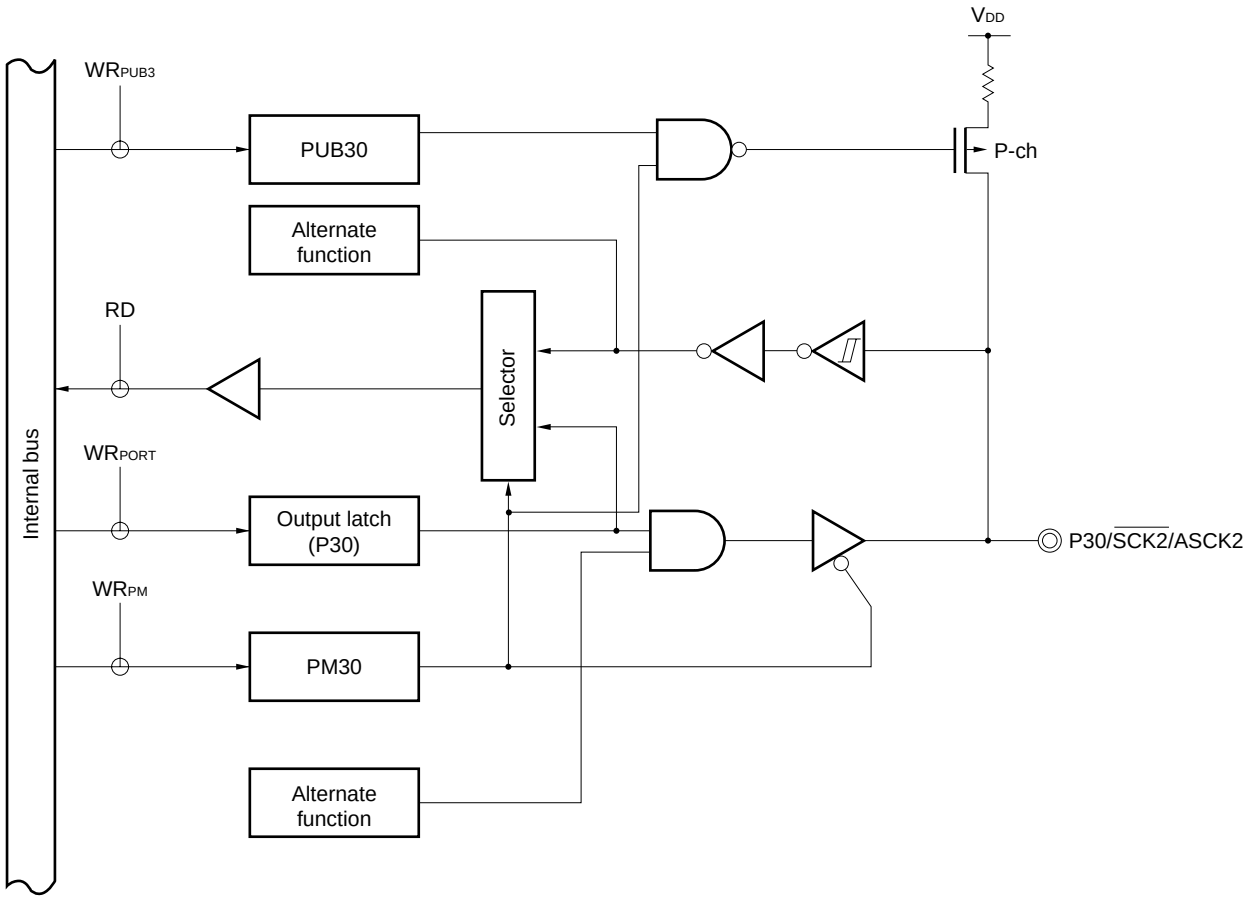
The port is also used for serial interface I/O, clock I/O, and external interrupt input.

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

Figures 5-7 to 5-9 show block diagrams of port 3.

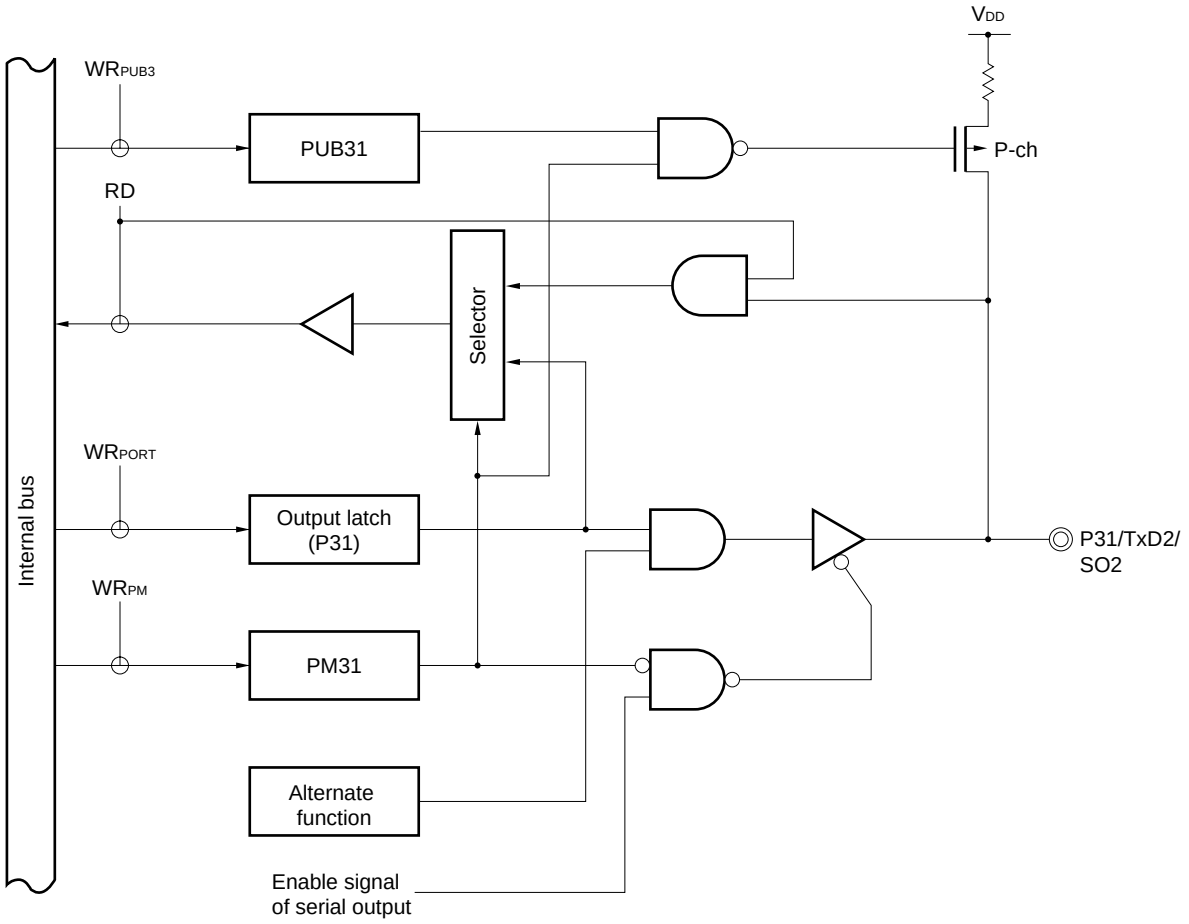
Caution When using the pins of port 3 for the serial interface, the I/O or output latch must be set according to the function to be used. For how to set the latches, see Table 11-2 Operation Mode Settings for Serial Interface 2.

Figure 5-7. Block Diagram of P30



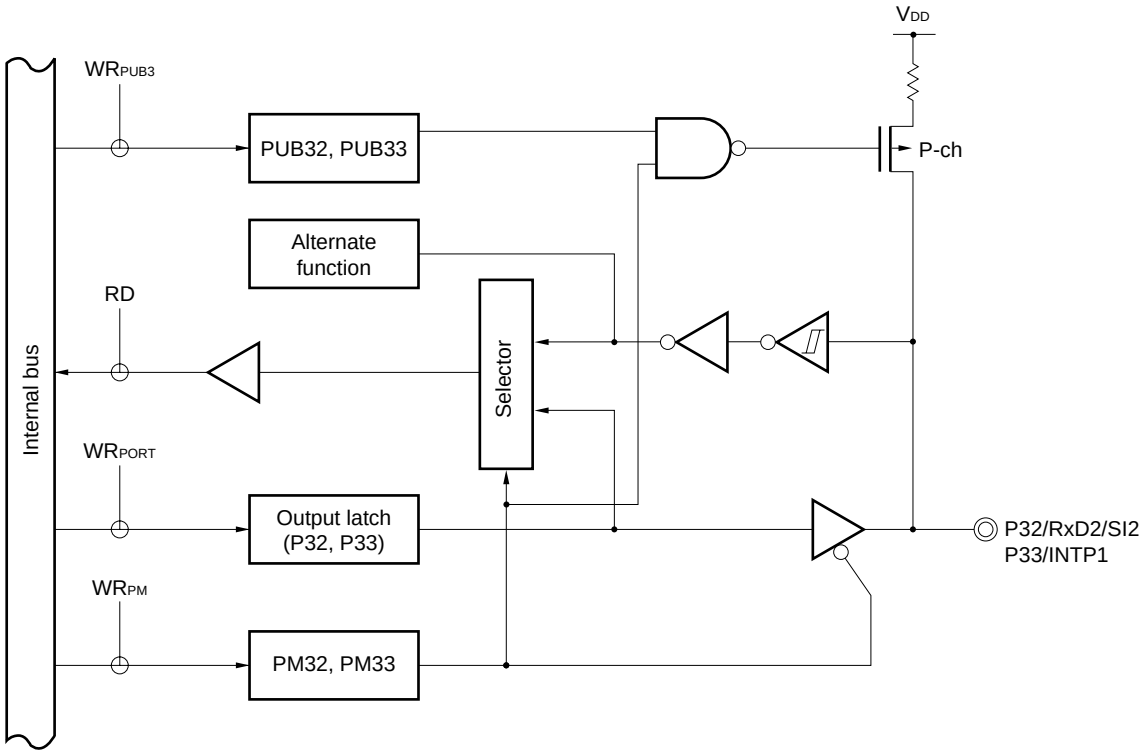
PUB3: Pull-up resistor option register 3
PM: Port mode register
RD: Port 3 read signal
WR: Port 3 write signal

Figure 5-8. Block Diagram of P31



PUB3: Pull-up resistor option register 3
PM: Port mode register
RD: Port 3 read signal
WR: Port 3 write signal

Figure 5-9. Block Diagram of P32 and P33



PUB3: Pull-up resistor option register 3
PM: Port mode register
RD: Port 3 read signal
WR: Port 3 write signal

5.2.5 Port 4

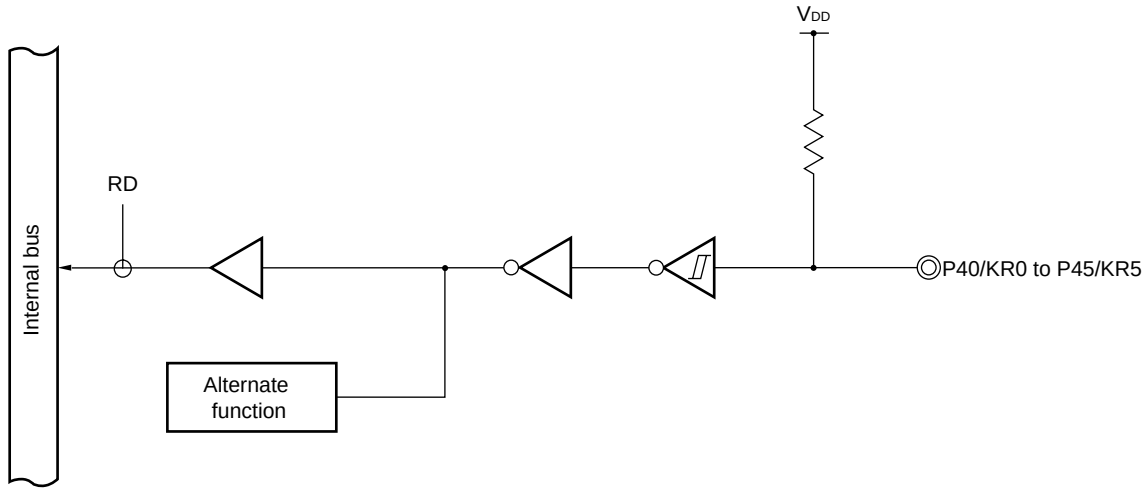
This is a 6-bit input port. It incorporates a pull-up resistor.

The port is also used for key return input.

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

Figure 5-10 shows a block diagram of port 4.

Figure 5-10. Block Diagram of P40 to P45



RD: Port 4 read signal

5.2.6 Port 7

This is a 1-bit output port.

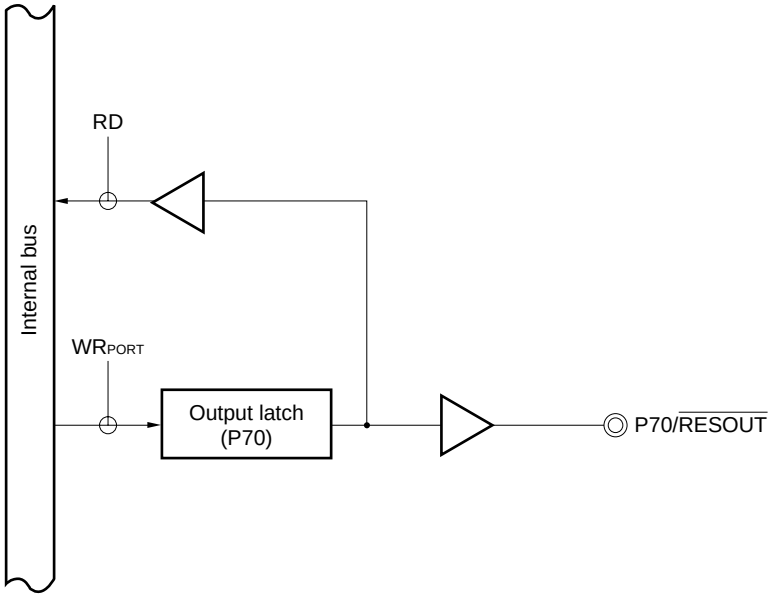
This port is also used for reset output.

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

When read, the value of the output latch can be read.

Figure 5-11 shows a block diagram of port 7.

Figure 5-11. Block Diagram of P70



RD: Port 7 read signal

WR: Port 7 write signal

5.3 Port Function Control Registers

The following two types of registers control the ports.

- Port mode registers (PM0 to PM3)
- Pull-up resistor option registers (PUB0 to PUB3)

(1) Port mode registers (PM0 to PM3)

These registers are used to set port input/output in 1-bit units.

Port mode registers are independently set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets these registers to FFH.

When port pins are used as alternate-function pins, set the port mode register and output latch according to Table 5-3.

Caution As ports 2 and 3 have an alternate function as external interrupt inputs, 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.

Table 5-3. Port Mode Register and Output Latch Settings When Using Alternate Functions

Pin Name	Alternate Function		PM××	P××
	Name	I/O		
P20	BSFO0	Output	0	0
P21	BSFO1	Output	0	0
P22	TI50	Input	1	×
	TO50	Output	0	0
P23	TI00	Input	1	×
	TO0	Output	0	0
P24	TI01	Input	1	×
	INTP0	Input	1	×
P33	INTP1	Input	1	×

Caution When port 3 is used for the serial interface, the I/O latch or output latch must be set according to its function. For the setting method, refer to Table 11-2 Operation Mode Settings for Serial Interface 2.

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

Figure 5-12. Port Mode Register Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	1	1	1	PM03	PM02	PM01	PM00	FF20H	FFH	R/W
PM1	1	1	1	1	1	1	PM11	PM10	FF21H	FFH	R/W
PM2	1	1	1	PM24	PM23	PM22	PM21	PM20	FF22H	FFH	R/W
PM3	1	1	1	1	PM33	PM32	PM31	PM30	FF23H	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0 to 3, n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

(2) Pull-up resistor option registers (PUBm: m = 0 to 3)

The pull-up resistor option registers (PUBm) set whether to use on-chip pull-up resistors at each port or not.

At a port where use of on-chip pull-up resistors has been specified by PUBm, the pull-up resistors can be internally used only for the bits set to input mode. No on-chip pull-up resistors can be used for the bits set to output mode, in spite of the setting of PUBm. On-chip pull-up resistors can also not be used when the pins are used as the alternate-function output pins.

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

$\overline{\text{RESET}}$ input clears PUBm to 00H.

Figure 5-13. Format of Pull-Up Resistor Option Register

Symbol	7	6	5	4	<3>	<2>	<1>	<0>	Address	After reset	R/W
PUB0	0	0	0	0	PUB03	PUB02	PUB01	PUB00	FF30H	00H	R/W
	7	6	5	4	3	2	<1>	<0>			
PUB1	0	0	0	0	0	0	PUB11	PUB10	FF31H	00H	R/W
	7	6	5	<4>	<3>	<2>	<1>	<0>			
PUB2	0	0	0	PUB24	PUB23	PUB22	PUB21	PUB20	FF32H	00H	R/W
	7	6	5	4	<3>	<2>	<1>	<0>			
PUB3	0	0	0	0	PUB33	PUB32	PUB31	PUB30	FF33H	00H	R/W

PUBm	Pm on-chip pull-up resistor selection (m = 0 to 3)
0	On-chip pull-up resistor not used
1	On-chip pull-up resistor used

5.4 Operation of Port Functions

The operation of a port differs depending on whether the port is set to input or output mode, as described below.

5.4.1 Writing to I/O port

(1) In output mode

A value can be written to the output latch by using a transfer instruction. The contents of the output latch can be output from the pins of the port.

The data once written to the output latch is retained until new data is written to the output latch.

(2) In input mode

A value can be written to the output latch by using a transfer instruction. However, the status of the port pin is not changed because the output buffer is off.

The data once written to the output latch is retained until new data is written to the output latch.

Caution A 1-bit memory manipulation instruction is executed to manipulate 1 bit of a port. However, this instruction accesses the port in 8-bit units. When this instruction is executed to manipulate a bit of an input/output port, therefore, the contents of the output latch of the pin that is set in the input mode and not subject to manipulation become undefined.

5.4.2 Reading from I/O port

(1) In output mode

The contents of the output latch can be read by using a transfer instruction. The contents of the output latch are not changed.

(2) In input mode

The status of a pin can be read by using a transfer instruction. The contents of the output latch are not changed.

5.4.3 Arithmetic operation of I/O port

(1) In output mode

An arithmetic operation can be performed with the contents of the output latch. The result of the operation is written to the output latch. The contents of the output latch are output from the port pins.

The data once written to the output latch is retained until new data is written to the output latch.

(2) In input mode

The contents of the output latch become undefined. However, the status of the pin is not changed because the output buffer is off.

Caution A 1-bit memory manipulation instruction is executed to manipulate 1 bit of a port. However, this instruction accesses the port in 8-bit units. When this instruction is executed to manipulate a bit of an input/output port, therefore, the contents of the output latch of the pin that is set in the input mode and not subject to manipulation become undefined.

CHAPTER 6 CLOCK GENERATOR

6.1 Clock Generator Functions

The clock generator generates the clock to be supplied to the CPU and peripheral hardware.
The following type of system clock oscillator is used.

- System clock (crystal/ceramic) oscillator
  This circuit oscillates at 1.0 to 5.0 MHz. Oscillation can be stopped by executing the STOP instruction.

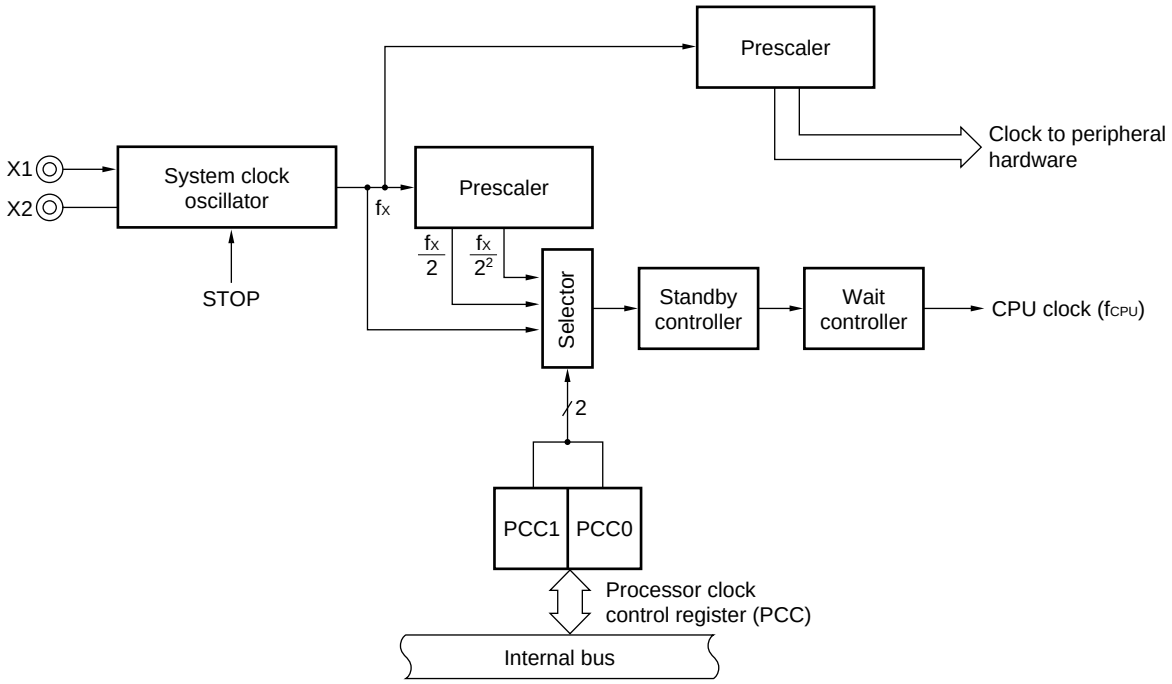
6.2 Clock Generator Configuration

The clock generator includes the following hardware.

Table 6-1. Configuration of Clock Generator

Item	Configuration
Control register	Processor clock control register (PCC)
Oscillator	Crystal/ceramic oscillator

Figure 6-1. Block Diagram of Clock Generator



6.3 Clock Generator Control Register

The clock generator is controlled by the following register.

- Processor clock control register (PCC)

(1) Processor clock control register (PCC)

PCC selects the CPU clock and the division ratio.

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

$\overline{\text{RESET}}$ input sets PCC to 02H.

Figure 6-2. Format of Processor Clock Control Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PCC	0	0	0	0	0	0	PCC1	PCC0	FFFBH	02H	R/W

PCC1	PCC0	CPU clock (f_{CPU}) selection	Minimum instruction execution time: $2/f_{CPU}$
			f_X = 5.0 MHz
0	0	f_X	0.4 μs
0	1	$f_X/2$	0.8 μs
1	0	$f_X/2^2$	1.6 μs
1	1	Setting prohibited	

Caution Bits 2 to 7 must be set to 0.

Remark f_X: System clock oscillation frequency

6.4 System Clock Oscillator

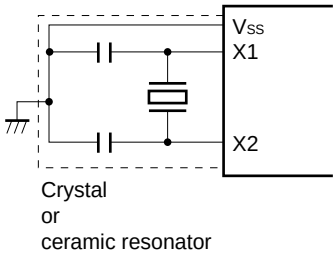
6.4.1 System clock oscillator

The system clock oscillator is oscillated by the crystal or ceramic resonator (5.0 MHz TYP.) connected across the X1 and X2 pins.

Figure 6-3 shows the external circuit of the system clock oscillator.

Figure 6-3. External Circuit of System Clock Oscillator

Crystal or ceramic oscillation



Caution When using the system clock oscillator, wire as follows in the area enclosed by the broken lines in Figure 6-3 to avoid an adverse effect from wiring capacitance.

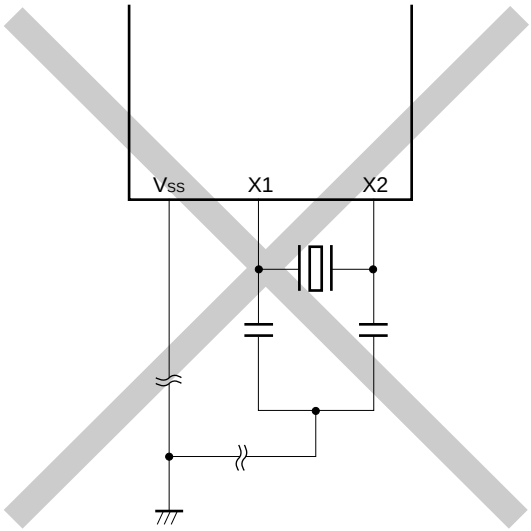
- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as V_{SS}. Do not ground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

6.4.2 Examples of incorrect resonator connection

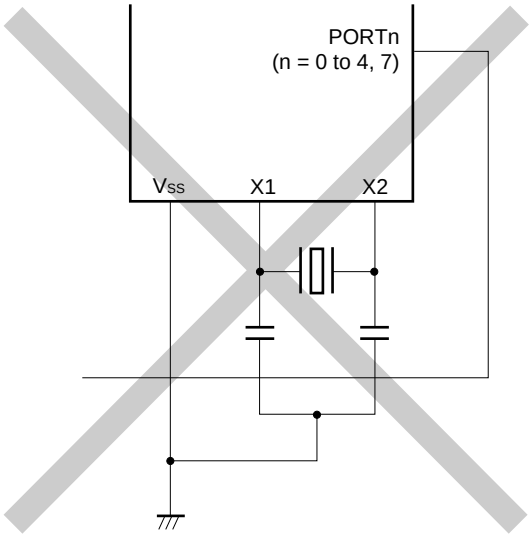
Figure 6-4 shows examples of incorrect resonator connection.

Figure 6-4. Examples of Incorrect Resonator Connection (1/2)

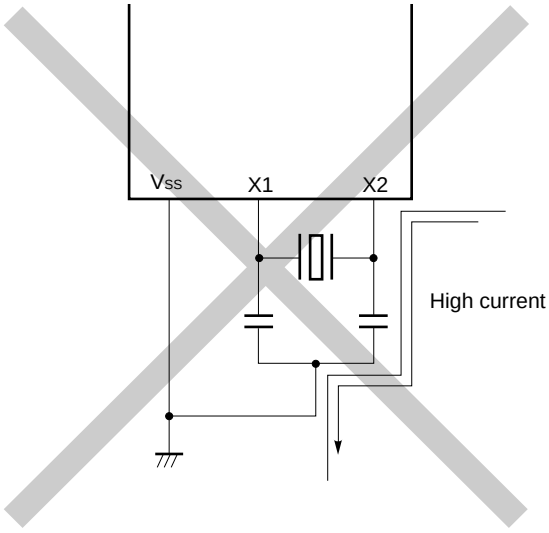
(a) Wiring too long



(b) Crossed signal line



(c) Wiring near high fluctuating current



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

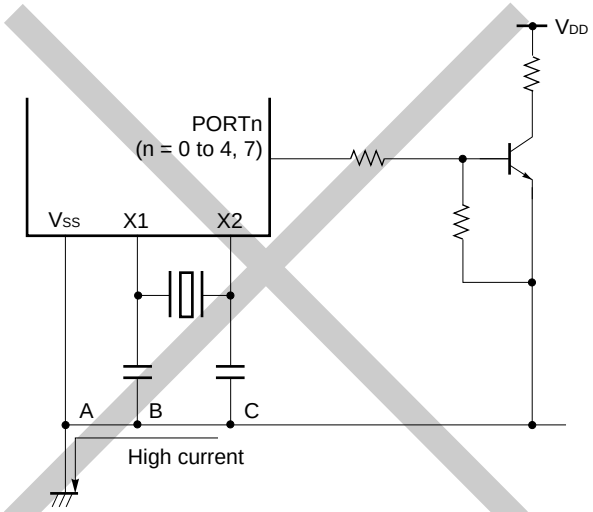
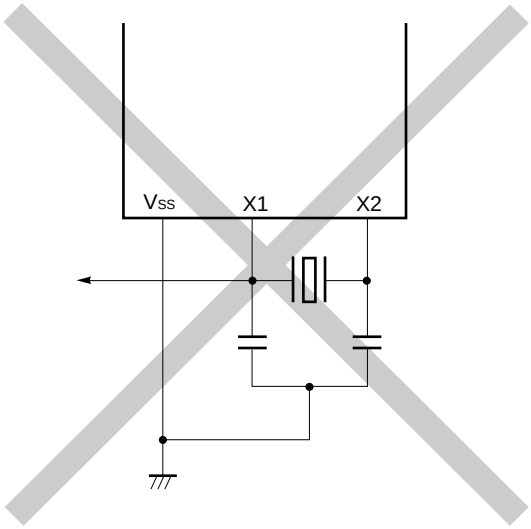


Figure 6-4. Examples of Incorrect Resonator Connection (2/2)

(e) Signal is fetched



6.4.3 Frequency divider

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

6.5 Clock Generator Operation

The clock generator generates the following clocks and controls the operation modes of the CPU, such as standby mode.

- System clock f_X
- CPU clock f_{CPU}
- Clock to peripheral hardware

The operation of the clock generator is determined by the processor clock control register (PCC) as follows.

(a) The slow mode (1.6 μs: at 5.0 MHz operation) of the system clock is selected when the $\overline{\text{RESET}}$ signal is generated (PCC = 02H). While a low level is being input to the $\overline{\text{RESET}}$ pin, oscillation of the system clock is stopped.

(b) Three types of minimum instruction execution time (0.4 μs, 0.8 μs, 1.6 μs: at 5.0 MHz operation) can be selected by the PCC setting.

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

(d) The clock for the peripheral hardware is generated by dividing the frequency of the system clock. Therefore, the peripheral hardware stops when the system clock stops (except for an externally input clock).

6.6 Changing Setting of CPU Clock

6.6.1 Time required for switching CPU clock

The CPU clock can be selected by using bits 0 and 1 (PCC0 and PCC1) of the processor clock control register (PCC).

Actually, the specified clock is not selected immediately after the setting of PCC has been changed, and the old clock is used for the duration of several instructions after that (see **Table 6-2**).

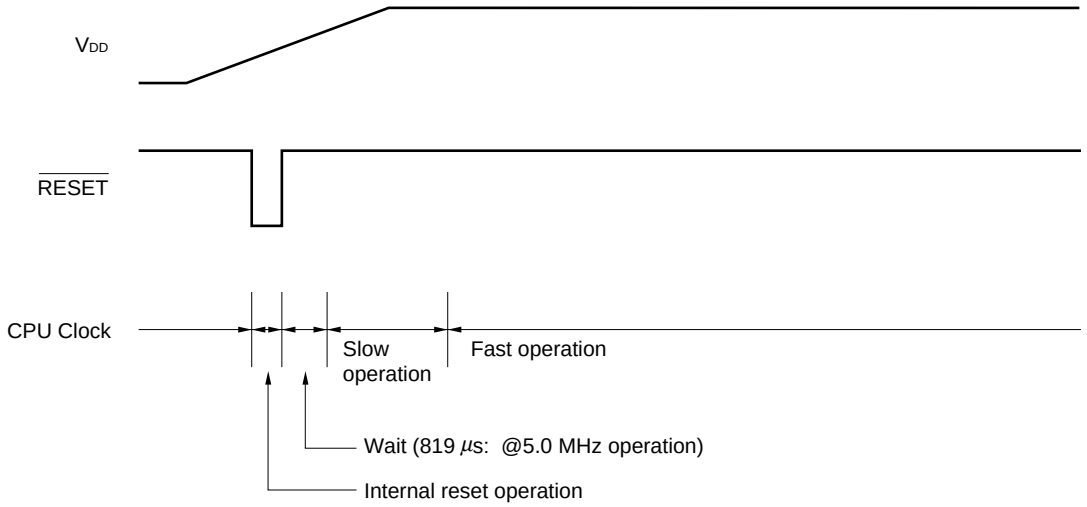
Table 6-2. Maximum Time Required for Switching CPU Clock

Set Value Before Switching		Set Value After Switching					
PCC1	PCC0	PCC1	PCC0	PCC1	PCC0	PCC1	PCC0
		0	0	0	1	1	0
0	0			4 instructions		4 instructions	
0	1	2 instructions				2 instructions	
1	0	1 instruction		1 instruction			

6.6.2 Switching CPU clock

The following figure illustrates how the CPU clock is switched.

Figure 6-5. Switching Between System Clock and CPU Clock



<1> The CPU is reset when the $\overline{\text{RESET}}$ pin is made low on power application. The effect of resetting is released when the $\overline{\text{RESET}}$ pin is later made high, and the system clock starts oscillating. At this time, the oscillation stabilization time ($2^{12}/f_X$) is automatically secured.

After that, the CPU starts instruction execution at the slow speed of the system clock (1.6 μs: @ 5.0 MHz operation).

<2> After the time required for the V_{DD} voltage to rise to the level at which the CPU can operate at the high speed has elapsed, the processor clock control register (PCC) is rewritten so that the high-speed operation can be selected.

CHAPTER 7 16-BIT TIMER/EVENT COUNTER 0

7.1 Functions of 16-Bit Timer/Event Counter 0

16-bit timer/event counter 0 has the following functions.

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

(1) Interval timer

16-bit timer/event counter 0 generates interrupt requests at the preset time interval.

(2) PPG output

16-bit timer/event counter 0 can output a square wave whose frequency and output pulse can be set freely.

(3) Pulse width measurement

16-bit timer/event counter 0 can measure the pulse width of an externally input signal.

(4) External event counter

16-bit timer/event counter 0 can measure the number of pulses of an externally input signal.

(5) Square-wave output

16-bit timer/event counter 0 can output a square wave with any selected frequency.

7.2 Configuration of 16-Bit Timer/Event Counter 0

16-bit timer/event counter 0 includes the following hardware.

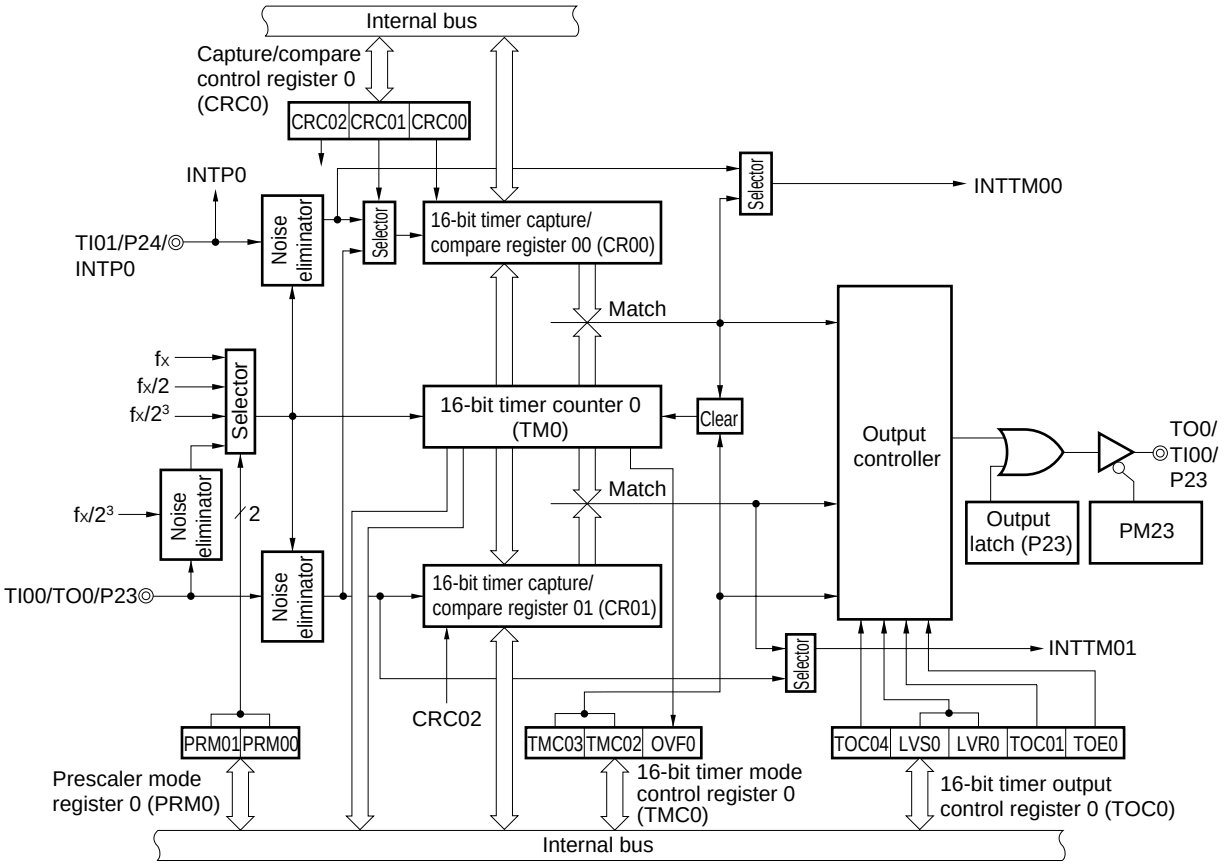
Table 7-1. Configuration of 16-Bit Timer/Event Counter 0

Item	Configuration
Timer/counter	16 bits × 1 (TM0)
Register	16-bit timer capture/compare register: 16 bits × 2 (CR00, CR01)
Timer output	1 (TO0)
Control registers	16-bit timer mode control register 0 (TMC0) Capture/compare control register 0 (CRC0) 16-bit timer output control register 0 (TOC0) Prescaler mode register 0 (PRM0) Port mode register 2 (PM2)[Note] Port 2 (P2)[Note]

Note See **Figure 5-5 Block Diagram of P22 and P23** and **Figure 5-6 Block Diagram of P24**.

Figure 7-1 shows a block diagram of 16-bit timer/event counter 0.

Figure 7-1. Block Diagram of 16-Bit Timer/Event Counter 0



(1) 16-bit timer counter 0 (TM0)

TM0 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. The count value is reset to 0000H in the following cases.

<1> At $\overline{\text{RESET}}$ input
<2> If TMC03 and TMC02 are cleared
<3> If the valid edge of TI00 is input in the clear & start mode entered by inputting the valid edge of TI00
<4> If TM0 and CR00 match in the clear & start mode on a match between TM0 and CR00

(2) 16-bit timer capture/compare register 00 (CR00)

CR00 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 0 (CRC00) of capture/compare control register 0 (CRC0).

- **When CR00 is used as a compare register**

  The value set in the CR00 is constantly compared with the 16-bit timer counter 0 (TM0) count value, and an interrupt request (INTTM00) is generated if they match. It can also be used as the register which holds the interval time when TM0 is set to interval timer operation.

- **When CR00 is used as a capture register**

  It is possible to select the valid edge of the TI00/TO0/P23 pin or the TI01/P24/INTP0 pin as the capture trigger. Setting of the TI00 or TI01 valid edge is performed by means of prescaler mode register 0 (PRM0).

  If CR00 is specified as a capture register and capture trigger is specified to be the valid edge of the TI00/TO0/P23 pin, the situation is as shown in Table 7-2. On the other hand, when capture trigger is specified to be the valid edge of the TI01/P24/INTP0 pin, the situation is as shown in Table 7-3.

Table 7-2. TI00/TO0/P23 Pin Valid Edge and CR00, CR01 Capture Trigger

ES01	ES00	TI00/TO0/P23 Pin Valid Edge	CR00 Capture Trigger	CR01 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 7-3. TI01/P24/INTP0 Pin Valid Edge and CR00 Capture Trigger

ES11	ES10	TI01/P24/INTP0 Pin Valid Edge	CR00 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

CR00 is set by a 16-bit memory manipulation instruction.

After $\overline{\text{RESET}}$ input, the value of CR00 is set to 0000H.

Cautions 1. Set a value other than 0000H in CR00 in the clear & start mode entered on a match between TM0 and CR00. However, in the free-running mode and in the clear mode using the valid edge of TI00, if 0000H is set to CR00, an interrupt request (INTTM00) is generated following an overflow (FFFFH) when the value changes from 0000H to 0001H.

2. When P23 is used as the input pin of the valid edge of TI00, it cannot be used as timer output (TO0). Moreover, when P23 is used as TO0, it cannot be used as the input pin of the valid edge of TI00.

3. If the register read period and the input of the capture trigger conflict when CR00 is used as a capture register, the read data is undefined (the capture data itself is a normal value).

If the count stop input and capture trigger input conflict, the capture data is undefined.

4. CR00 must not be rewritten during the TM0 operation.

(3) 16-bit timer capture/compare register 01 (CR01)

CR01 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 (CRC02) of capture/compare control register 0 (CRC0).

- **When CR01 is used as a compare register**

  The value set in the CR01 is constantly compared with the 16-bit timer counter 0 (TM0) count value, and an interrupt request (INTTM01) is generated if they match.

- **When CR01 is used as a capture register**

  It is possible to select the valid edge of the TI00/TO0/P23 pin as the capture trigger. The TI00/TO0/P23 valid edge is set by means of prescaler mode register 0 (PRM0).

CR01 is set by a 16-bit memory manipulation instruction.
After $\overline{\text{RESET}}$ input, the value of CR01 is set to 0000H.

Cautions 1. CR01 can be set to 0000H.

However, in the free-running mode and in the clear mode using the valid edge of TI00, if CR01 is set to 0000H, an interrupt request (INTTM01) is generated following an overflow (FFFFH) when the value changes from 0000H to 0001H.

2. If the register read period and the input of the capture trigger conflict when CR01 is used as a capture register, the read data is undefined (the capture data itself is a normal value).

If the count stop input and capture trigger input conflict, the capture data is undefined.

3. CR01 can be rewritten during the TM0 operation. For details, refer to Remark 2 in Figure 7-12.

7.3 Control Registers of 16-Bit Timer/Event Counter 0

The following six registers are used to control 16-bit timer/event counter 0.

- 16-bit timer mode control register 0 (TMC0)
- Capture/compare control register 0 (CRC0)
- 16-bit timer output control register 0 (TOC0)
- Prescaler mode register 0 (PRM0)
- Port mode register 2 (PM2)
- Port 2 (P2)

(1) 16-bit timer mode control register 0 (TMC0)

This register sets the 16-bit timer operating mode, the 16-bit timer counter 0 (TM0) clear mode, and output timing, and detects an overflow.

TMC0 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TMC0 value to 00H.

Caution 16-bit timer counter 0 (TM0) starts operation at the moment TMC02 and TMC03 are set to values other than 0, 0 (operation stop mode), respectively. Set TMC02 and TMC03 to 0, 0 to stop the operation.

Figure 7-2. Format of 16-Bit Timer Mode Control Register 0 (TMC0)

Address: FFA6H After reset: 00H R/W

Symbol	7	6	5	4	<3>	<2>	1	<0>
TMC0	0	0	0	0	TMC03	TMC02	0	OVF0

TMC03	TMC02	Operation mode and clear mode selection	TO0 output timing selection	Interrupt request generation
0	0	Operation stop (TM0 cleared to 0)	No change	Not generated
0	1	Free-running mode	Match between TM0 and CR00 or match between TM0 and CR01	Generated on match between TM0 and CR00, or match between TM0 and CR01
1	0	Clear & start on TI00 valid edge	–	
1	1	Clear & start on match between TM0 and CR00	Match between TM0 and CR00 or match between TM0 and CR01	

OVF0	16-bit timer counter 0 (TM0) overflow detection
0	Overflow not detected
1	Overflow detected

Cautions
1. **Timer operation must be stopped before writing to bits other than the OVF0 flag.**
2. **Set the valid edge of the TI00/TO0/P23 pin using prescaler mode register 0 (PRM0).**
3. **If clear & start mode entered on a match between TM0 and CR00, clear & start mode entered on the valid edge of TI00, or the free-running mode is selected, when the set value of CR00 is FFFFH and the TM0 value changes from FFFFH to 0000H, the OVF0 flag is set to 1.**
4. **Bits 1 and 4 to 7 must be set to 0.**

Remarks
1. TO0: 16-bit timer/event counter 0 output pin
2. TI00: 16-bit timer/event counter 0 input pin
3. TM0: 16-bit timer counter 0
4. CR00: 16-bit timer capture/compare register 00
5. CR01: 16-bit timer capture/compare register 01

(2) Capture/compare control register 0 (CRC0)

This register controls the operation of the 16-bit timer capture/compare registers (CR00, CR01).

CRC0 is set by a 1-bit or 8-bit memory manipulation instruction.

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

Figure 7-3. Format of Capture/Compare Control Register 0 (CRC0)

Address: FFA8H After reset: 00H R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
CRC0	0	0	0	0	0	CRC02	CRC01	CRC00

CRC02	CR01 operation mode selection
0	Operates as compare register
1	Operates as capture register

CRC01	CR00 capture trigger selection
0	Captures on valid edge of TI01
1	Captures on valid edge of TI00 by reverse phase[Note]

CRC00	CR00 operation mode selection
0	Operates as compare register
1	Operates as capture register

Note If both the rising and falling edges have been selected as the valid edges of TI00, CR00 does not perform capture.

Cautions 1. Timer operation must be stopped before setting CRC0.

2. When clear & start mode entered on a match between TM0 and CR00 is selected by 16-bit timer mode control register 0 (TMC0), CR00 should not be specified as a capture register.

3. To ensure the reliability of the capture operation, the capture trigger requires a pulse two cycles longer than the count clock selected by prescaler mode register 0 (PRM0).

(3) 16-bit timer output control register 0 (TOC0)

This register controls the operation of the 16-bit timer/event counter 0 output controller. It controls R-S type flip-flop (LV0) setting/resetting, output inversion enabling/disabling, and 16-bit timer/event counter 0 timer output enabling/disabling.

TOC0 is set by a 1-bit or 8-bit memory manipulation instruction.

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

Figure 7-4 shows the format of TOC0.

Figure 7-4. Format of 16-Bit Timer Output Control Register 0 (TOC0)

Address: FFA9H After reset: 00H R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
TOC0	0	0	0	TOC04	LVS0	LVR0	TOC01	TOE0

TOC04	Timer output F/F control by match of CR01 and TM0
0	Inversion operation disabled
1	Inversion operation enabled

LVS0	LVR0	16-bit timer/event counter 0 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

TOC01	Timer output F/F control by match of CR00 and TM0
0	Inversion operation disabled
1	Inversion operation enabled

TOE0	16-bit timer/event counter 0 output control
0	Output disabled (output set to level 0)
1	Output enabled

Cautions 1. Timer operation must be stopped before setting TOC0.

2. If LVS0 and LVR0 are read, they will be 0.

★ **3. Do not set LVS0 to 1 before setting TOE0, and do not set LVS0 and TOE0 to 1 at the same time.**

(4) Prescaler mode register 0 (PRM0)

This register is used to set the 16-bit timer counter 0 (TM0) count clock and TI00, TI01 input valid edges.

PRM0 is set by an 8-bit memory manipulation instruction.

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

Figure 7-5. Format of Prescaler Mode Register 0 (PRM0)

Address: FFA7H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	2	<1>	<0>
PRM0	ES11	ES10	ES01	ES00	0	0	PRM01	PRM00

ES11	ES10	TI01 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES01	ES00	TI00 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

PRM01	PRM00	Count clock selection
0	0	f_X (0.2 μs)
0	1	$f_X/2$ (0.4 μs)
1	0	$f_X/2^3$ (1.6 μs)
1	1	TI00 valid edge[Note]

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

Cautions
1. **Always set data to PRM0 after stopping the timer operation.**
2. **If the valid edge of TI00 is to be set for the count clock, do not set the clear & start mode and the capture trigger at the valid edge of TI00.**
3. **If the TI00 or TI01 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 TI00 pin or TI01 pin to enable the operation of the 16-bit timer counter 0 (TM0). Care is therefore needed when pulling up the TI00 pin or the TI01 pin. However, when re-enabling operation after the operation has been stopped once, the rising edge is not detected.**
4. **When P23 is used as the input pin of the valid edge of TI00, it cannot be used as timer output (TO0). Moreover, when P23 is used as TO0, it cannot be used as the input pin of the valid edge of TI00.**

Remarks
1. f_X: System clock oscillation frequency
2. TI00, TI01: 16-bit timer/event counter 0 input pin
3. Figures in parentheses are for operation with f_X = 5.0 MHz.

(5) Port mode register 2 (PM2)

This register sets port 2 input/output in 1-bit units.

When using the P23/TO0/TI00 pin for timer output, set PM23 and the output latch of P23 to 0.

When using the P23/TO0/TI00 pin for timer input, set PM23 to 0. At this time, the output latch of P23 may be either 0 or 1.

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

$\overline{\text{RESET}}$ input sets PM2 value to FFH.

Figure 7-6. Format of Port Mode Register 2 (PM2)

Address: FF22H After reset: FFH R/W

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

PM2n	P2n pin I/O mode selection (n = 0 to 4)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

7.4 Operation of 16-Bit Timer/Event Counter 0

7.4.1 Interval timer operation

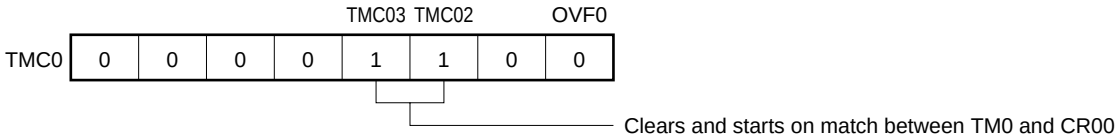
Setting 16-bit timer mode control register 0 (TMC0) and capture/compare control register 0 (CRC0) as shown in Figure 7-7 allows operation as an interval timer. Interrupt requests are generated repeatedly using the count value set in 16-bit timer capture/compare register 00 (CR00) beforehand as the interval.

When the count value of 16-bit timer counter 0 (TM0) matches the value set to CR00, counting continues with the TM0 value cleared to 0 and the interrupt request signal (INTTM00) is generated.

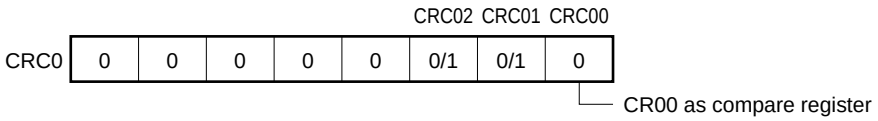
The count clock of the 16-bit timer/event counter can be selected with bits 0 and 1 (PRM00, PRM01) of prescaler mode register 0 (PRM0).

Figure 7-7. Control Register Settings for Interval Timer Operation

(a) 16-bit timer mode control register 0 (TMC0)

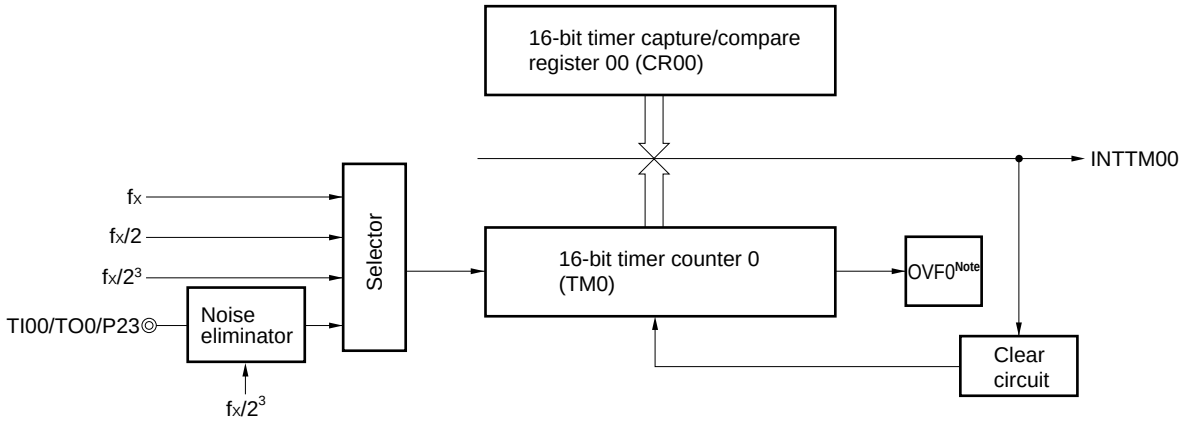


(b) Capture/compare control register 0 (CRC0)



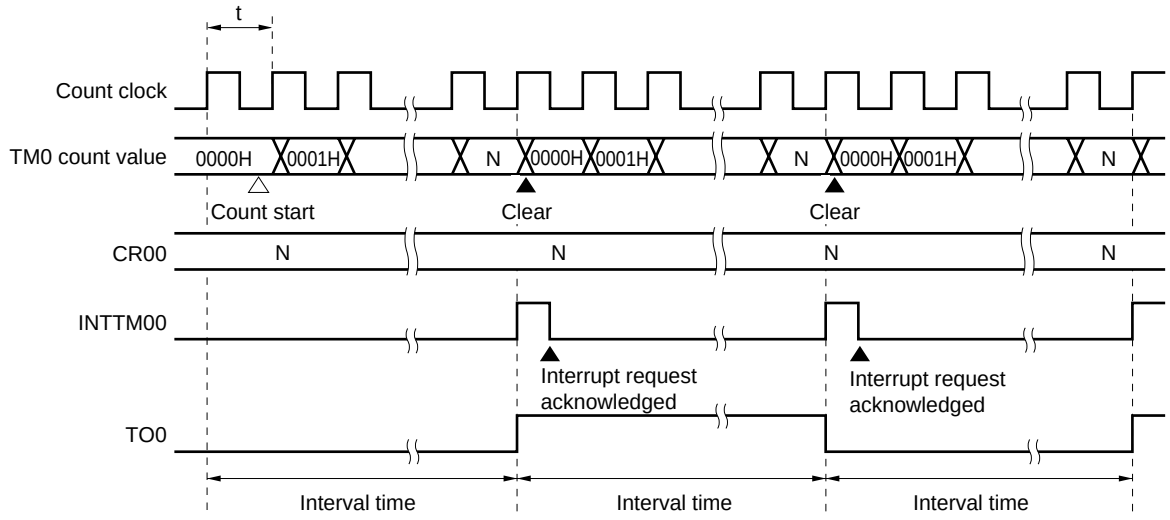
Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with the interval timer. See **Figure 7-3**.

Figure 7-8. Interval Timer Configuration Diagram



Note OVF0 becomes 1 only when CR00 is set to FFFFH.

Figure 7-9. Timing of Interval Timer Operation



Remark Interval time = (N + 1) × t
N = 0001H to FFFFH

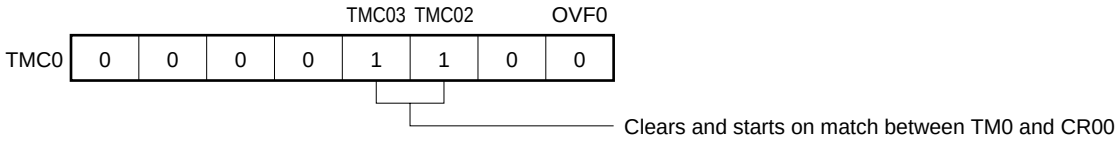
7.4.2 PPG output operations

Setting 16-bit timer mode control register 0 (TMC0) and capture/compare control register 0 (CRC0) as shown in Figure 7-10 allows operation as PPG (Programmable Pulse Generator) output.

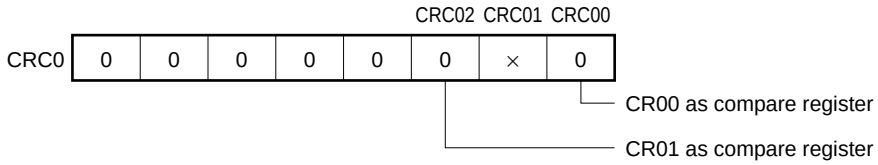
In the PPG output operation, square waves are output from the TO0/TI00/P23 pin with the pulse width and the cycle that correspond to the count values set beforehand in 16-bit timer capture/compare register 01 (CR01) and in 16-bit timer capture/compare register 00 (CR00), respectively.

Figure 7-10. Control Register Settings for PPG Output Operation

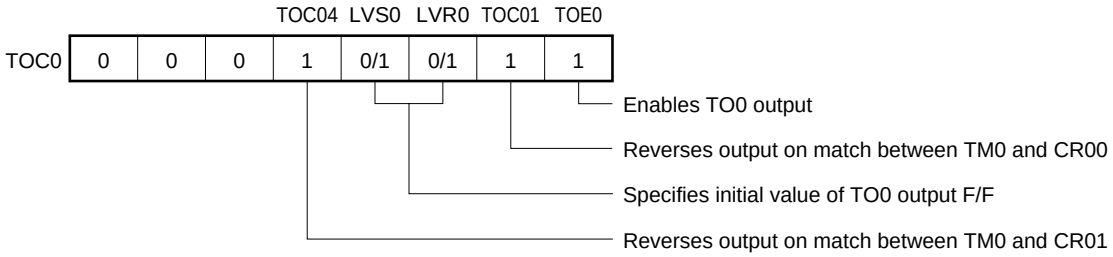
(a) 16-bit timer mode control register 0 (TMC0)



(b) Capture/compare control register 0 (CRC0)



(c) 16-bit timer output control register 0 (TOC0)



Cautions 1. Values in the following range should be set in CR00 and CR01:
0000H ≤ CR01 < CR00 ≤ FFFFH

2. The cycle of the pulse generated through PPG output (CR00 setting value + 1) has a duty of (CR01 setting value + 1)/(CR00 setting value + 1).

Remark ×: Don't care

Figure 7-11. PPG Output Configuration Diagram

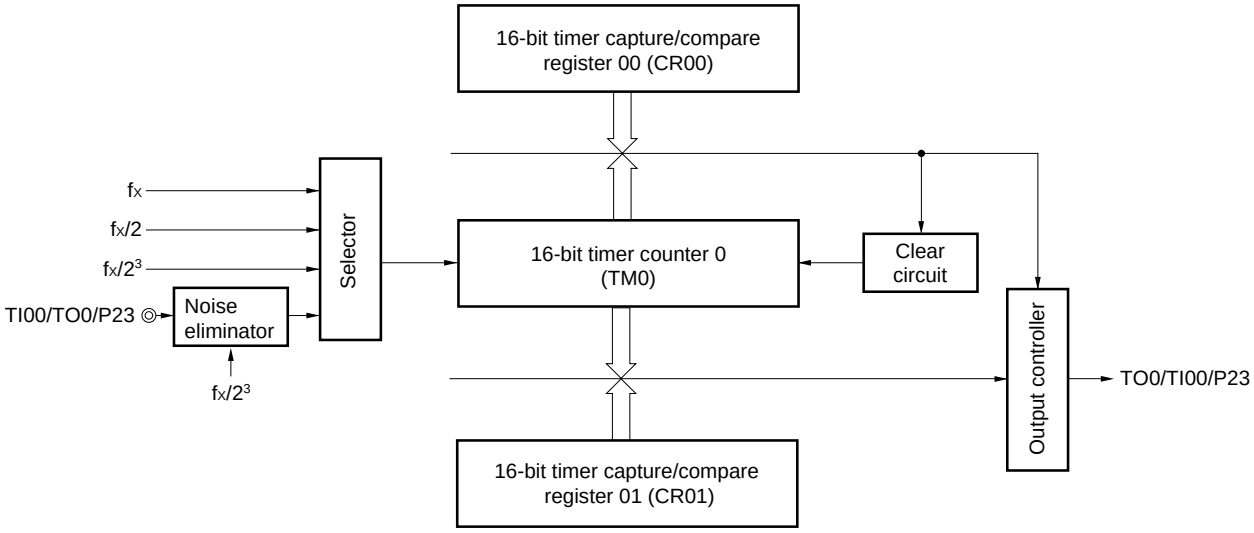
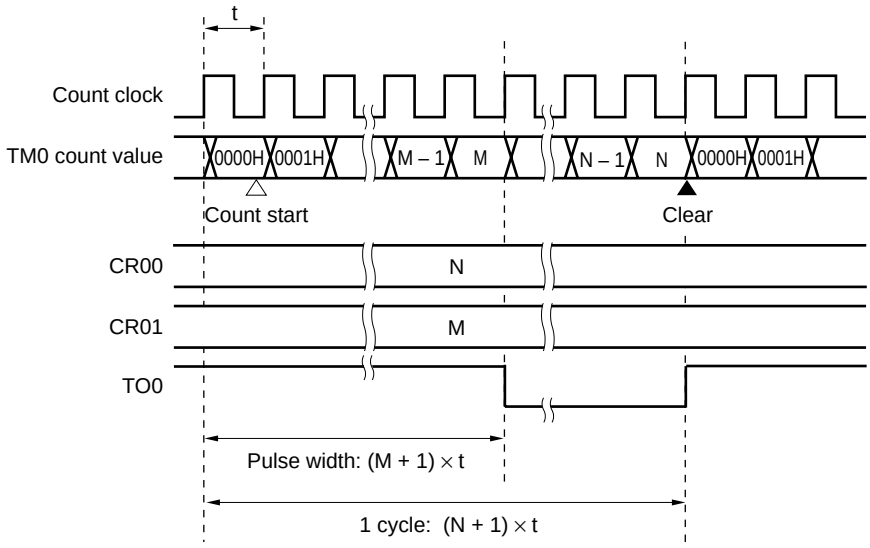


Figure 7-12. PPG Output Operation Timing



Caution CR00 cannot be rewritten during TM0 operation.

Remarks 1. 0000H ≤ M < N ≤ FFFFH

2. In the PPG output operation, change the pulse width (rewrite CR01) during TM0 operation using the following procedure.

<1> Disable the timer output inversion operation by match of TM0 and CR01 (TOC04 = 0)

<2> Disable the INTTM01 interrupt (TMMK01 = 1)

<3> Rewrite CR01

<4> Wait for 1 cycle of the TM0 count clock

<5> Enable the timer output inversion operation by match of TM0 and CR01 (TOC04 = 1)

<6> Clear the interrupt request flag of INTTM01 (TMIF01 = 0)

<7> Enable the INTTM01 interrupt (TMMK01 = 0)

7.4.3 Pulse width measurement operations

It is possible to measure the pulse width of the signals input to the TI00/TO0/P23 pin and TI01/P24/INTP0 pin using 16-bit timer counter 0 (TM0).

There are two measurement methods: measuring with TM0 used in free-running mode, and measuring by restarting the timer in synchronization with the edge of the signal input to the TI00/TO0/P23 pin.

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

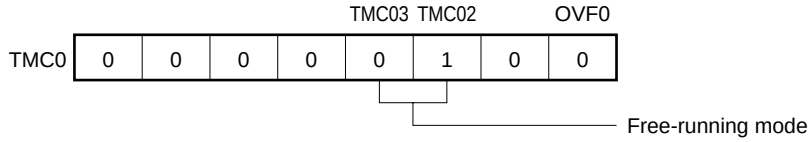
When 16-bit timer counter 0 (TM0) is operated in free-running mode, and the edge specified by prescaler mode register 0 (PRM0) is input to the TI00/TO0/P23 pin, the value of TM0 is taken into 16-bit timer capture/compare register 01 (CR01) and an external interrupt request signal (INTTM01) is set.

Any of three edges can be selected–rising, falling, or both edges–specified by bits 4 and 5 (ES00 and ES01) of PRM0.

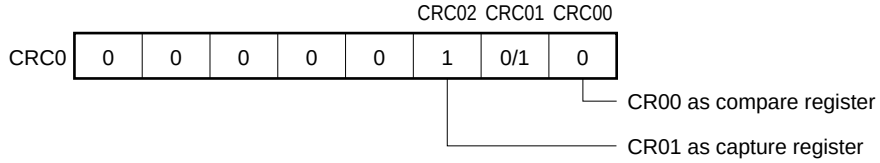
Sampling is performed at the count clock selected by PRM0, and a capture operation is only performed when a valid level of the TI00/TO0/P23 pin is detected twice, thus eliminating noise with a short pulse width.

Figure 7-13. Control Register Settings for Pulse Width Measurement with Free-Running Counter and One Capture Register

(a) 16-bit timer mode control register 0 (TMC0)



(b) Capture/compare control register 0 (CRC0)



Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. See **Figures 7-2** and **7-3**.

Figure 7-14. Configuration Diagram for Pulse Width Measurement by Free-Running Counter

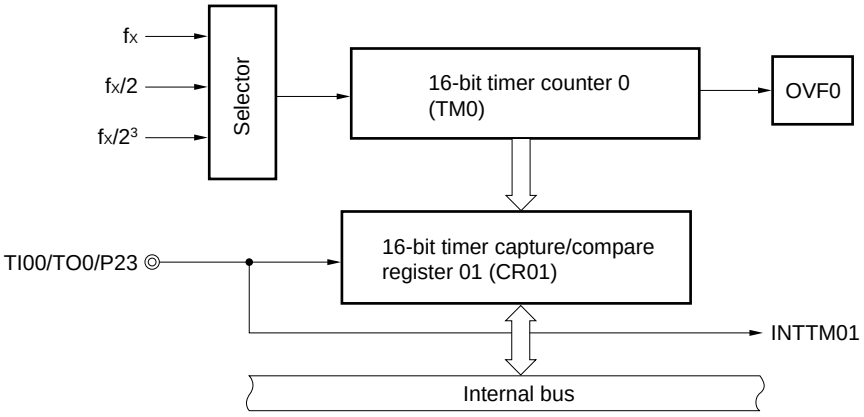
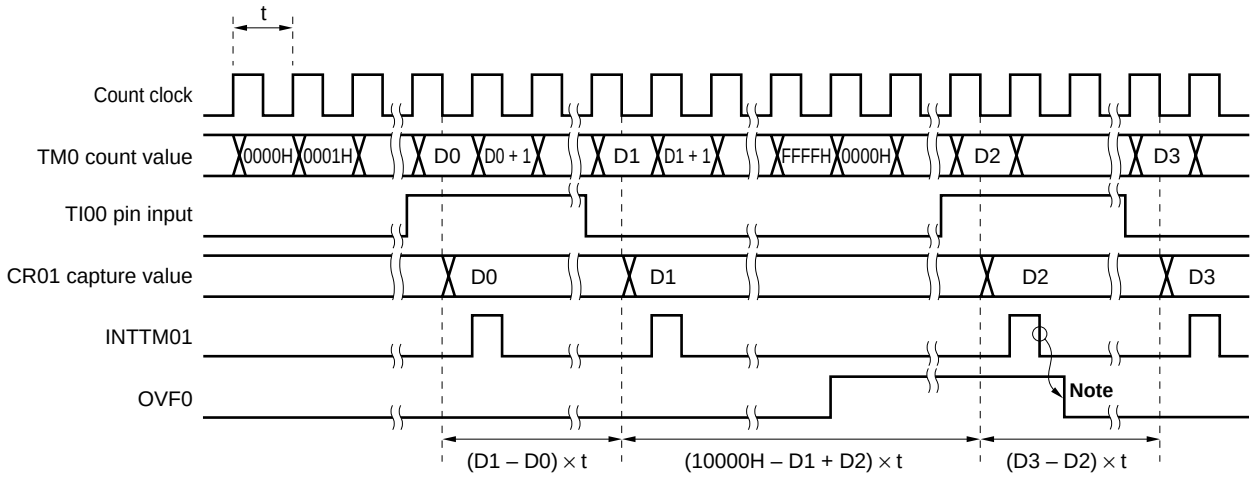


Figure 7-15. Timing of Pulse Width Measurement Operation by Free-Running Counter and One Capture Register (with Both Edges Specified)



Note Clear OVF0 by software.

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

When 16-bit timer counter 0 (TM0) is operated in free-running mode, it is possible to simultaneously measure the pulse widths of the two signals input to the TI00/TO0/P23 pin and the TI01/P24/INTP0 pin.

When the edge specified by bits 4 and 5 (ES00 and ES01) of prescaler mode register 0 (PRM0) is input to the TI00/TO0/P23 pin, the value of TM0 is taken into 16-bit timer capture/compare register 01 (CR01) and an interrupt request signal (INTTM01) is set.

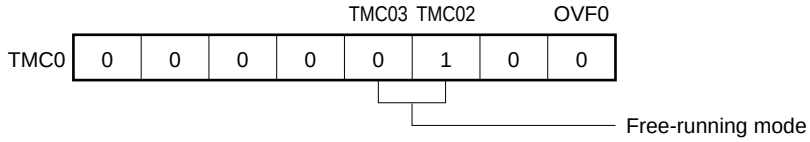
Also, when the edge specified by bits 6 and 7 (ES10 and ES11) of PRM0 is input to the TI01/P24/INTP0 pin, the value of TM0 is taken into 16-bit timer capture/compare register 00 (CR00) and an interrupt request signal (INTTM00) is set.

Any of three edges can be selected–rising, falling, or both edges–as the valid edges for the TI00/TO0/P23 pin and the TI01/P24/INTP0 pin specified by bits 4 and 5 (ES00 and ES01) and bits 6 and 7 (ES10 and ES11) of PRM0, respectively.

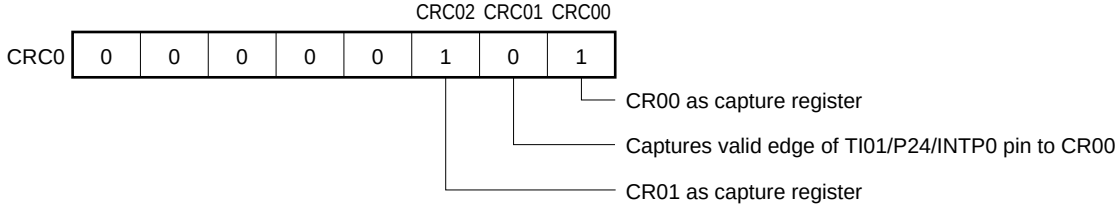
Sampling is performed at the interval selected by prescaler mode register 0 (PRM0), and a capture operation is only performed when a valid level of the TI00/TO0/P23 pin or TI01/P24/INTP0 pin is detected twice, thus eliminating noise with a short pulse width.

Figure 7-16. Control Register Settings for Measurement of Two Pulse Widths with Free-Running Counter

(a) 16-bit timer mode control register 0 (TMC0)



(b) Capture/compare control register 0 (CRC0)



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

  The capture register operation when the capture trigger is input is shown.

Figure 7-17. CR01 Capture Operation with Rising Edge Specified

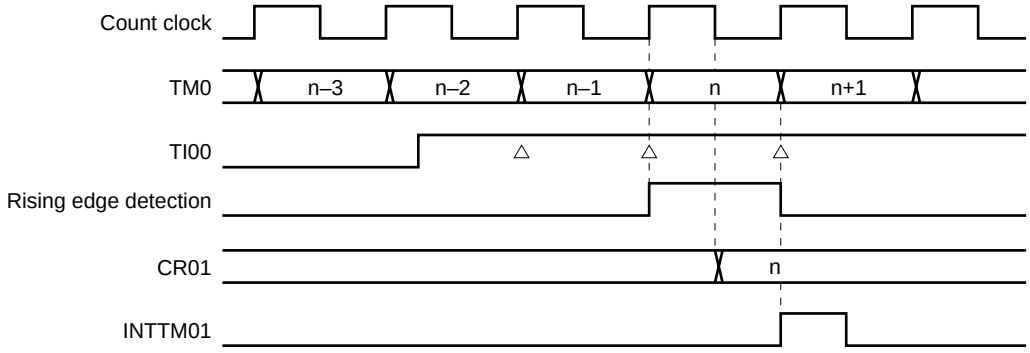
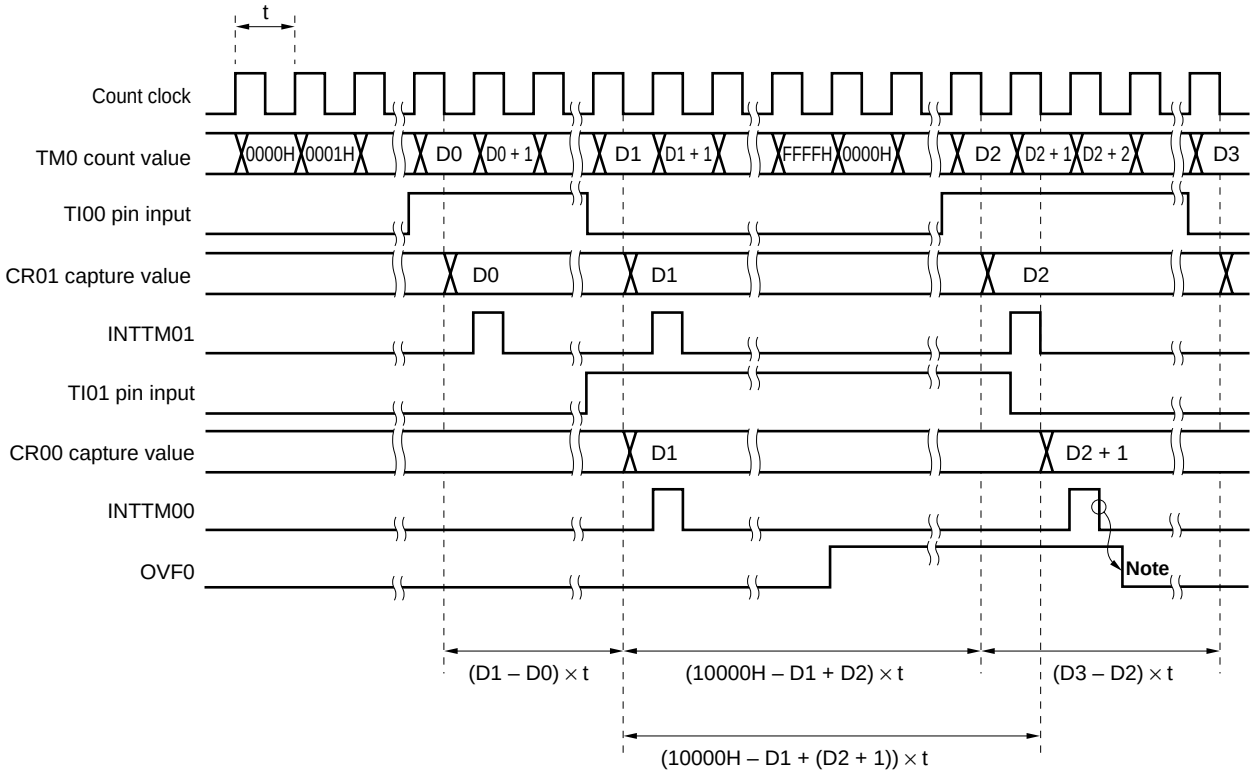


Figure 7-18. Timing of Pulse Width Measurement Operation by Free-Running Counter (with Both Edges Specified)



Note Clear OVF0 by software.

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

When 16-bit timer counter 0 (TM0) is operated in free-running mode, it is possible to measure the pulse width of the signal input to the TI00/TO0/P23 pin.

When the edge specified by bits 4 and 5 (ES00 and ES01) of prescaler mode register 0 (PRM0) is input to the TI00/TO0/P23 pin, the value of TM0 is taken into 16-bit timer capture/compare register 01 (CR01) and an interrupt request signal (INTTM01) is set.

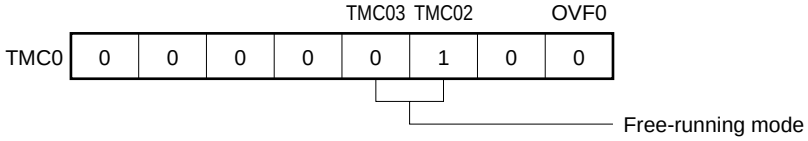
Also, the value of TM0 is taken into 16-bit timer capture/compare register 00 (CR00) on the inverse edge to that input in the capture operation to CR01.

Either of two edges can be selected–rising or falling–as the valid edges for the TI00/TO0/P23 pin specified by bits 4 and 5 (ES00 and ES01) of prescaler mode register 0 (PRM0).

Sampling is performed at the interval selected by prescaler mode register 0 (PRM0), and a capture operation is only performed when a valid level of the TI00/TO0/P23 pin is detected twice, thus eliminating noise with a short pulse width.

Figure 7-19. Control Register Settings for Pulse Width Measurement with Free-Running Counter and Two Capture Registers

(a) 16-bit timer mode control register 0 (TMC0)



(b) Capture/compare control register 0 (CRC0)

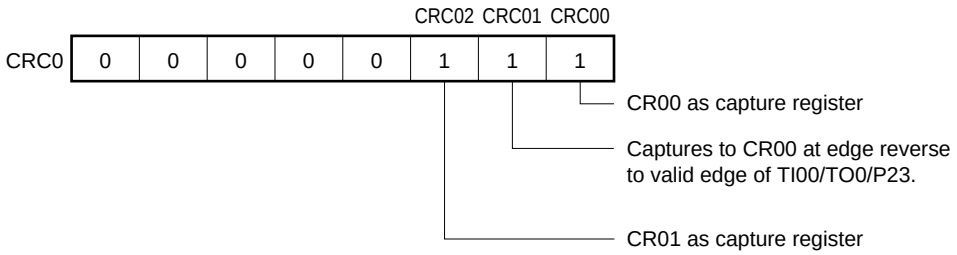
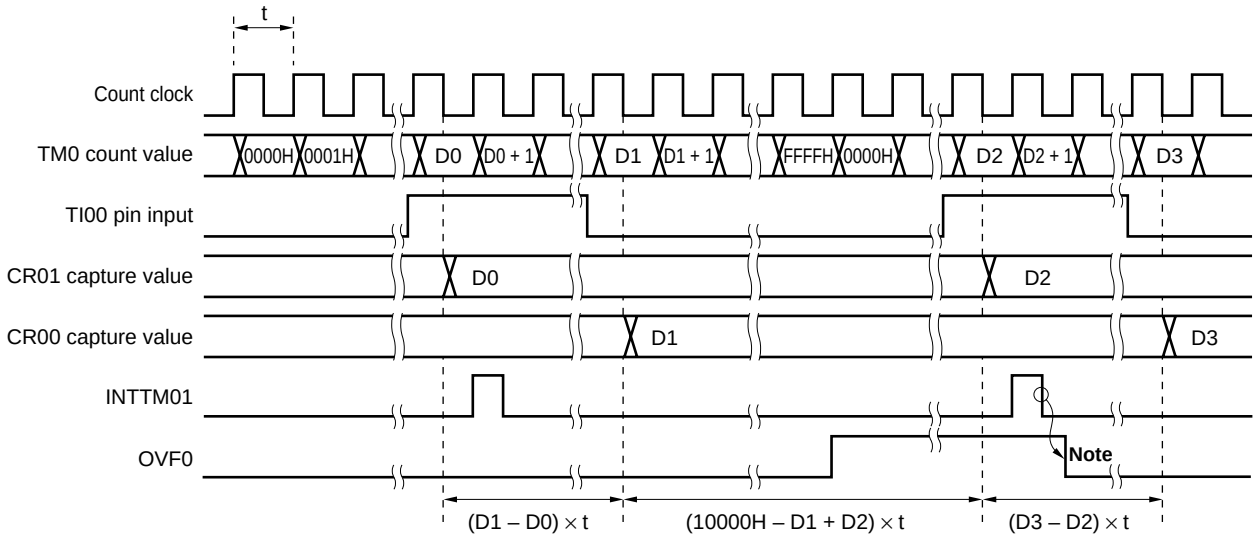


Figure 7-20. Timing of Pulse Width Measurement Operation by Free-Running Counter and Two Capture Registers (with Rising Edge Specified)



Note Clear OVF0 by software.

(4) Pulse width measurement by means of restart

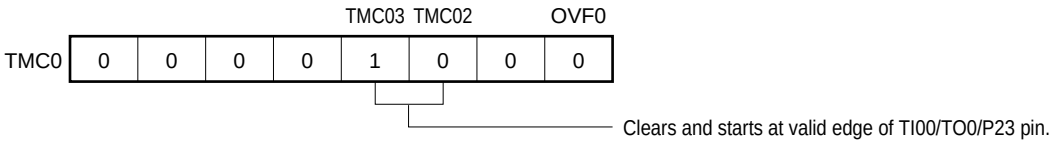
When input of a valid edge to the TI00/TO0/P23 pin is detected, the count value of 16-bit timer counter 0 (TM0) is taken into 16-bit timer capture/compare register 01 (CR01), and then the pulse width of the signal input to the TI00/TO0/P23 pin is measured by clearing TM0 and restarting the count.

Either of two edges can be selected–rising or falling–as the valid edges by bits 4 and 5 (ES00 and ES01) of prescaler mode register 0 (PRM0).

Sampling is performed at the interval selected by prescaler mode register 0 (PRM0), and a capture operation is only performed when a valid level of the TI00/TO0/P23 pin is detected twice, thus eliminating noise with a short pulse width.

Figure 7-21. Control Register Settings for Pulse Width Measurement by Means of Restart

(a) 16-bit timer mode control register 0 (TMC0)



(b) Capture/compare control register 0 (CRC0)

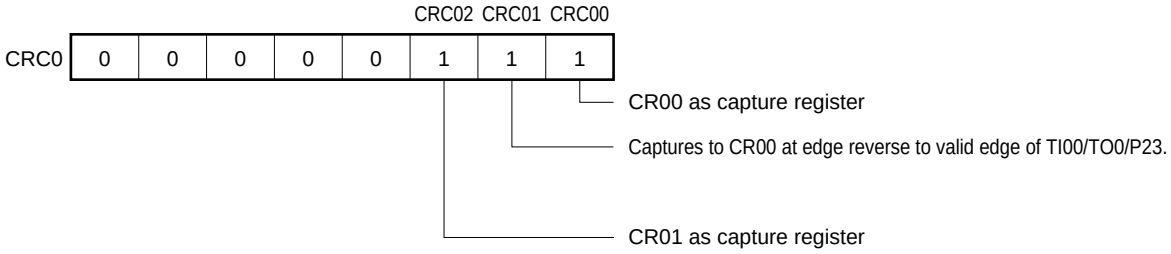
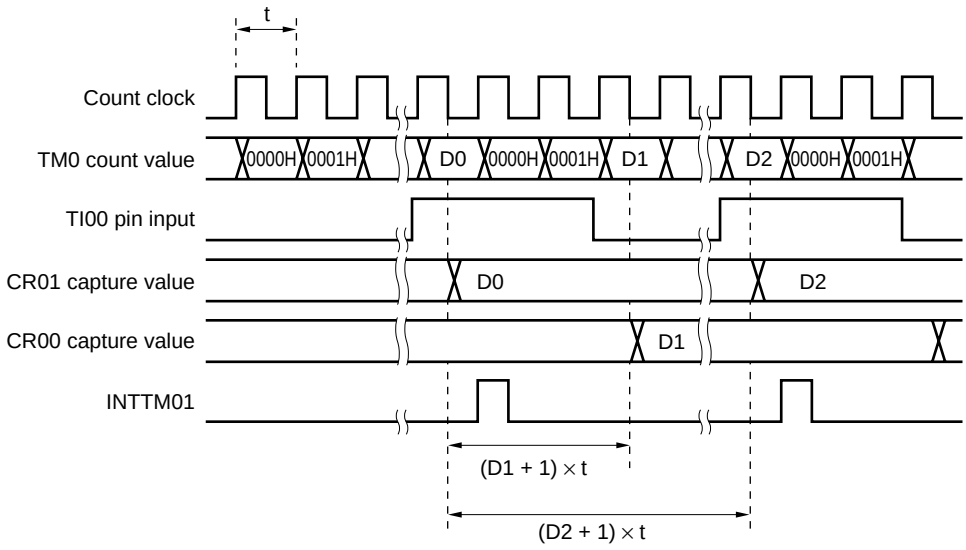


Figure 7-22. Timing of Pulse Width Measurement Operation by Means of Restart (with Rising Edge Specified)



7.4.4 External event counter operation

The external event counter counts the number of external clock pulses to be input to the TI00/TO0/P23 pin with 16-bit timer counter 0 (TM0).

TM0 is incremented each time the valid edge specified with prescaler mode register 0 (PRM0) is input.

When the TM0 count value matches the 16-bit timer capture/compare register 00 (CR00) value, TM0 is cleared to 0 and an interrupt request signal (INTTM00) is generated.

Input a value other than 0000H to CR00 (a 1-pulse count operation cannot be carried out).

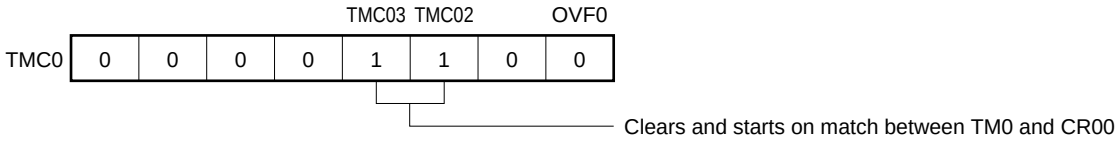
The rising edge, the falling edge, or both edges can be selected by bits 4 and 5 (ES00 and ES01) of prescaler mode register 0 (PRM0).

Sampling is performed with the internal clock ($f_X/2^3$), and an external event counter operation is only performed when a valid level of the TI00/TO0/P23 pin is detected twice, thus eliminating noise with a short pulse width.

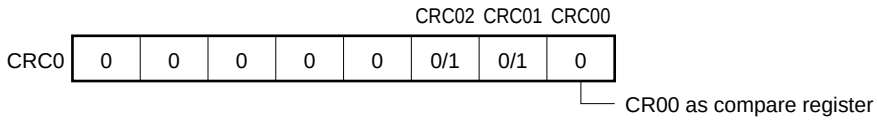
Caution When used as an external event counter, the P23/TI00/TO0 pin cannot be used as a timer output (TO0).

Figure 7-23. Control Register Settings in External Event Counter Mode

(a) 16-bit timer mode control register 0 (TMC0)

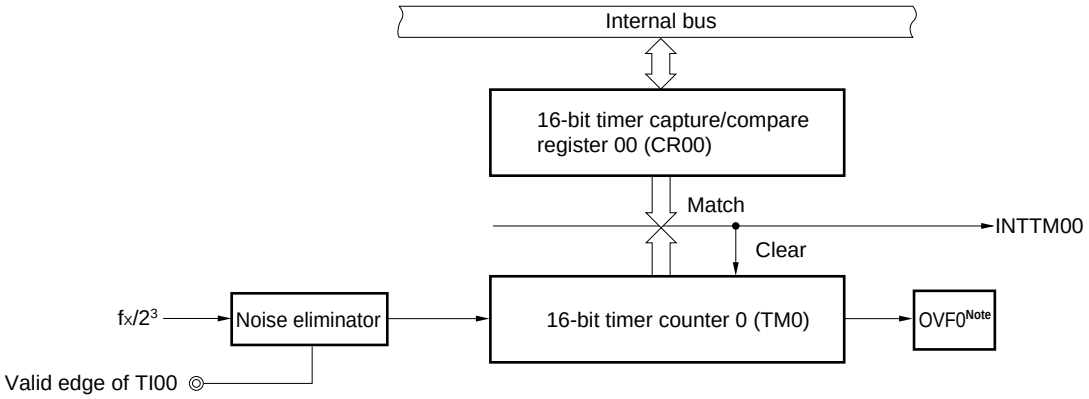


(b) Capture/compare control register 0 (CRC0)



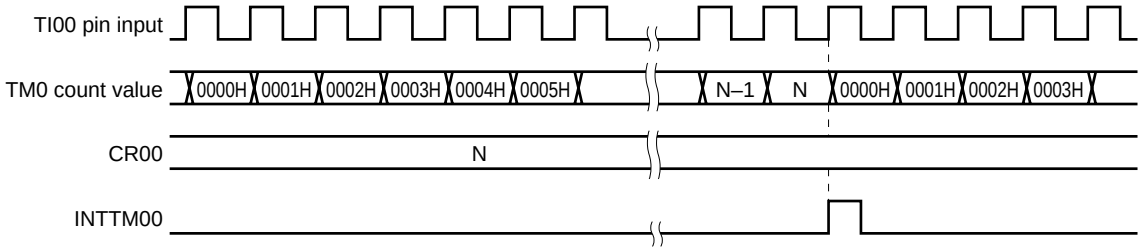
Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with the external event counter. See **Figures 7-2** and **7-3**.

Figure 7-24. External Event Counter Configuration Diagram



Note OVF0 becomes 1 only when CR00 is set to FFFFH.

Figure 7-25. External Event Counter Operation Timing (with Rising Edge Specified)



Caution When reading the external event counter count value, TM0 should be read.

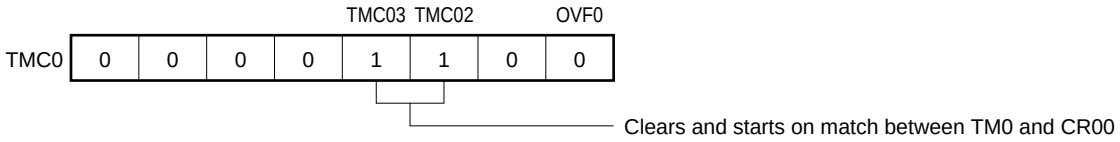
7.4.5 Square-wave output operation

A square wave with any selected frequency can be output at intervals determined by the count value preset to 16-bit timer capture/compare register 00 (CR00).

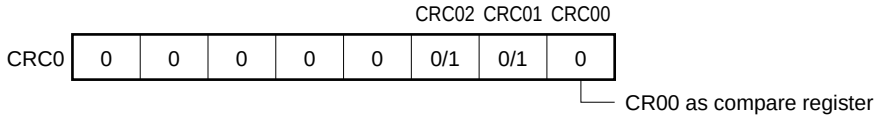
The TO0 pin output status is reversed at intervals determined by the count value preset to CR00 by setting bit 0 (TOE0) and bit 1 (TOC01) of 16-bit timer output control register 0 (TOC0) to 1. This enables a square wave with any selected frequency to be output.

Figure 7-26. Control Register Settings in Square-Wave Output Mode

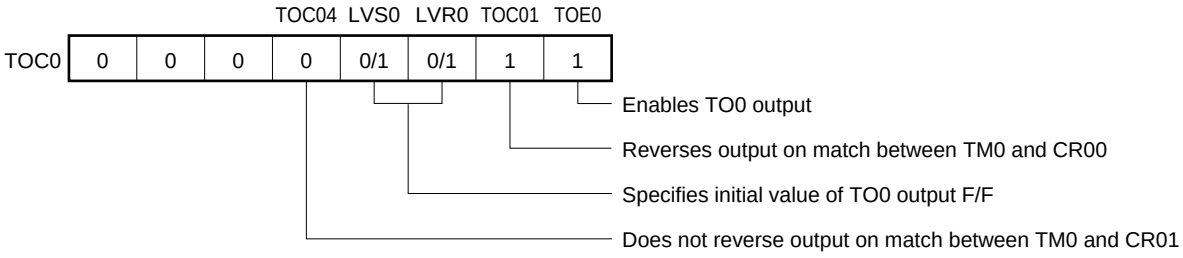
(a) 16-bit timer mode control register 0 (TMC0)



(b) Capture/compare control register 0 (CRC0)

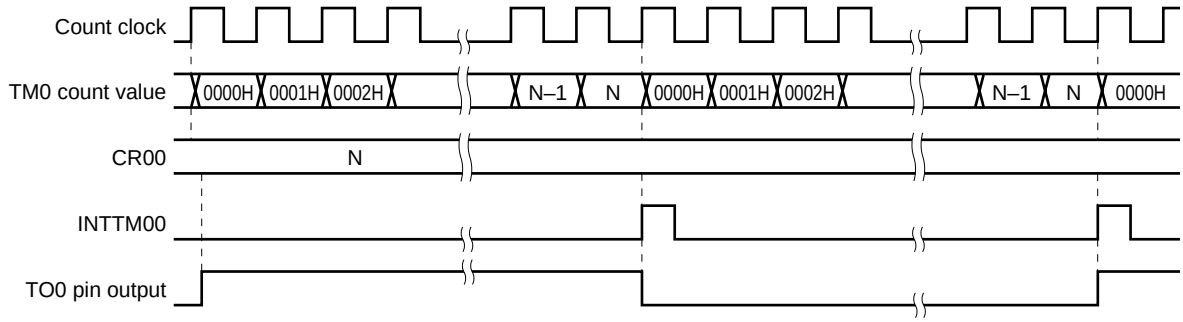


(c) 16-bit timer output control register 0 (TOC0)



Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with square-wave output. See **Figures 7-3** and **7-4**.

Figure 7-27. Square-Wave Output Operation Timing

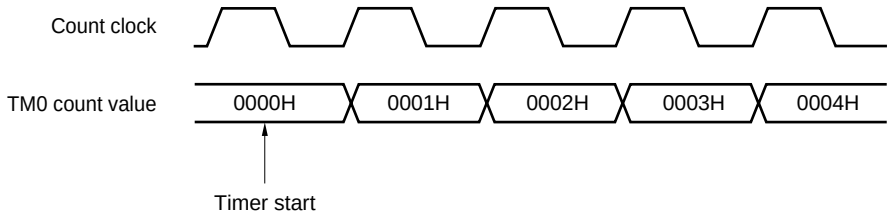


7.5 Cautions on 16-Bit Timer/Event Counter 0

(1) Timer start error

An error of up to one clock may occur in the time required for a match signal to be generated after timer start. This is because 16-bit timer counter 0 (TM0) is started asynchronously with the count clock.

Figure 7-28. Start Timing of 16-Bit Timer Counter 0 (TM0)



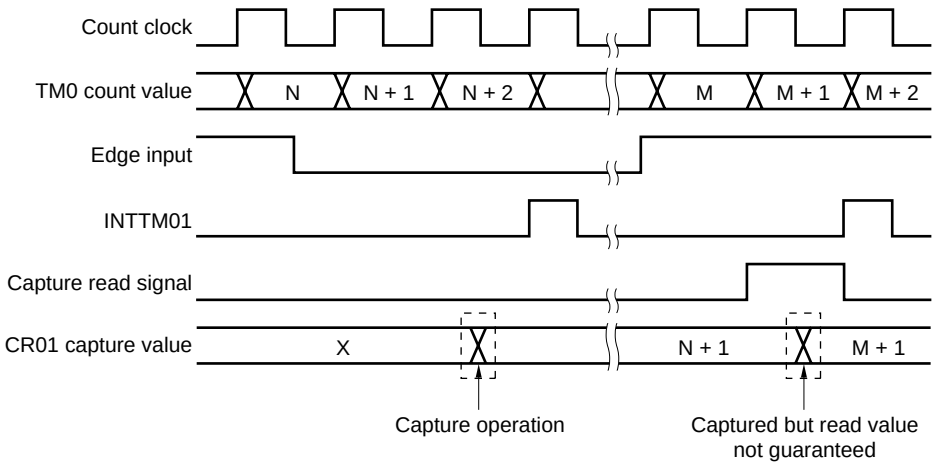
★ **(2) Prohibition of compare register change during timer count operation**

Do not rewrite CR00 during TM0 operation.

(3) Capture register data retention timing

<1> If the valid edge of the TI00/TO0/P23 pin is input while 16-bit timer capture/compare register 01 (CR01) is being read, CR01 carries out the capture operation but the read value at this time is not guaranteed. However, the interrupt request signal (INTTM01) is generated upon detection of the valid edge.

Figure 7-29. Capture Register Data Retention Timing



<2> The value of 16-bit timer capture/compare registers 00 and 01 (CR00 and CR01) is not guaranteed after 16-bit timer/event counter 0 is stopped.

(4) Valid edge setting

Set the valid edge of the TI00/TO0/P23 pin after setting bits 2 and 3 (TMC02 and TMC03) of 16-bit timer mode control register 0 (TMC0) to 0, 0, respectively, and then stopping timer operation. The valid edge is set with bits 4 and 5 (ES00 and ES01) of prescaler mode register 0 (PRM0).

(5) Operation of OVF0 flag

<1> The OVF0 flag is also set to 1 in the following case.

Either the clear & start mode entered on a match between TM0 and CR00, clear & start mode entered on the valid edge of TI00, or the free-running mode is selected.

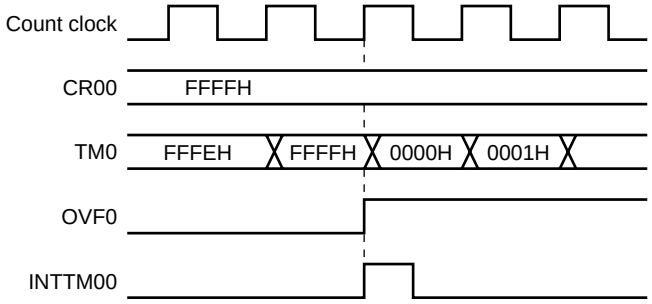
↓

CR00 is set to FFFFH.

↓

When TM0 is counted up from FFFFH to 0000H.

Figure 7-30. Operation Timing of OVF0 Flag



<2> Even if the OVF0 flag is cleared before the next count clock is counted (TM0 becomes 0001H) after the occurrence of a TM0 overflow, the OVF0 flag is reset newly and the clear is disabled.

(6) Timer operation

<1> Even if 16-bit timer counter 0 (TM0) is read, the value is not captured by 16-bit timer capture/compare register 01 (CR01).

<2> Regardless of the CPU's operation mode, when the timer stops, the signals input to pins TI00/TI01 are not acknowledged.

(7) Capture operation

<1> If TI00 is specified as the valid edge of the count clock, a capture operation by the capture register specified as the trigger for TI00 is not possible.

<2> If both the rising and falling edges are selected as the valid edge of TI00, CR00 does not perform capture.

<3> To ensure the reliability of the capture operation, the capture trigger requires a pulse two cycles longer than the count clock selected by prescaler mode register 0 (PRM0).

<4> The capture operation is performed at the fall of the count clock. An interrupt request input (INTTM0n), however, is generated at the rise of the next count clock.

(8) Compare operation

A capture operation may not be performed for CR00/CR01 set in compare mode even if a capture trigger has been input.

(9) Edge detection

<1> If the TI00 pin or the TI01 pin is high level immediately after system reset and the rising edge or both the rising and falling edges are specified as the valid edge for the TI00 pin or TI01 pin to enable the 16-bit timer counter 0 (TM0) operation, a rising edge is detected immediately. Be careful when pulling up the TI00 pin or the TI01 pin. However, the rising edge is not detected at restart after the operation has been stopped once.

<2> The sampling clock used to eliminate noise differs when the TI00 valid edge is used as the 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 0 (PRM0). The capture operation is started when a valid edge is detected twice by sampling, thus eliminating noise with a short pulse width.

CHAPTER 8 16-BIT TIMER D

8.1 Features

16-bit timer D functions as a 16-bit interval timer.

8.2 Function Overview

- 16-bit interval timer
- Compare registers: 1
- Interrupt request sources: 1 source
- Count clock selected from divisions of system clock

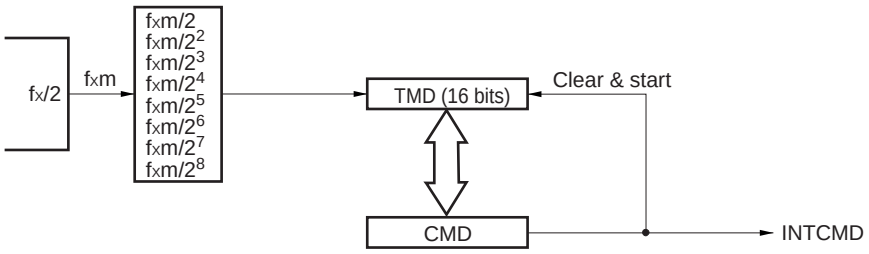
8.3 Basic Configuration of 16-Bit Timer D

Table 8-1. Configuration of 16-Bit Timer D

Timer	Count Clock	Register	Read/Write	Generated Interrupt Signal	Capture Trigger	Timer Output S/R	Other Functions
16-bit timer D	$f_X/2$, $f_X/2^2$, $f_X/2^3$, $f_X/2^4$, $f_X/2^5$, $f_X/2^6$, $f_X/2^7$, $f_X/2^8$	TMD	Read	–	–	–	–
		CMD	Read/write	INTCMD	–	–	–

Remark f_X: System clock
S/R: Set/reset

Figure 8-1. Block Diagram of 16-Bit Timer D



Remark f_X: System clock

8.4 16-Bit Timer D

(1) 16-bit timer counter D (TMD)

TMD is a 16-bit timer. It is mainly used as an interval timer for software.

Starting and stopping TMD is controlled by the CE bit of timer mode control register D (TMCD).

Division by the prescaler can be selected for the count clock from among $f_X/2$, $f_X/2^2$, $f_X/2^3$, $f_X/2^4$, $f_X/2^5$, $f_X/2^6$, $f_X/2^7$, and $f_X/2^8$ by the CS0 to CS2 bits of the TMCD register (f_X: system clock).

TMD is read-only in 16-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
TMD																	FF18H	0000H

The conditions for which the TMD register becomes 0000H are shown below.

- $\overline{\text{RESET}}$ input
- CAE bit = 0
- CE bit = 0
- Match of TMD register and CMD register
- Overflow

Cautions 1. If the CAE bit of the TMCD register is cleared (0), a reset is performed asynchronously.

2. If the CE bit of the TMCD register is cleared (0), a reset is performed, in synchronization with the internal clock. Similarly, a synchronized reset is performed after a match with the CMD register and after an overflow.

3. The count clock must not be changed during a timer operation. If it is to be overwritten, it should be overwritten after the CE bit is cleared (0).

4. Up to 4 clocks are required after a value is set in the CE bit until the set value is transferred to internal units. When a count operation begins, the count cycle from 0000H to 0001H differs from subsequent cycles.

5. After a compare match is generated, the timer is cleared at the next count clock. Therefore, if the division ratio is large, the timer value may not be zero even if the timer value is read immediately after a match interrupt is generated.

(2) 16-bit compare register D (CMD)

The CMD and TMD register count values are compared, and an interrupt request signal (INTCMD) is generated when a match occurs. TMD is cleared in synchronization with this match. If the CAE bit of the TMCD register is set to 0, a reset is performed asynchronously, and the registers are initialized.

The CMD register is configured with a master/slave configuration. When the CMD register is written, data is first written to the master register and then the master register data is transferred to the slave register. In a compare operation, the slave register value is compared with the count value of the TMD register. When the CMD register is read, data in the master side is read out.

CMD can be read or written in 16-bit units.

Cautions 1. A write operation to the CMD register requires 4 clocks until the value that was set in the CMD register is transferred to internal units. When writing continuously to the CMD register, be sure to secure a time interval of at least 4 clocks.

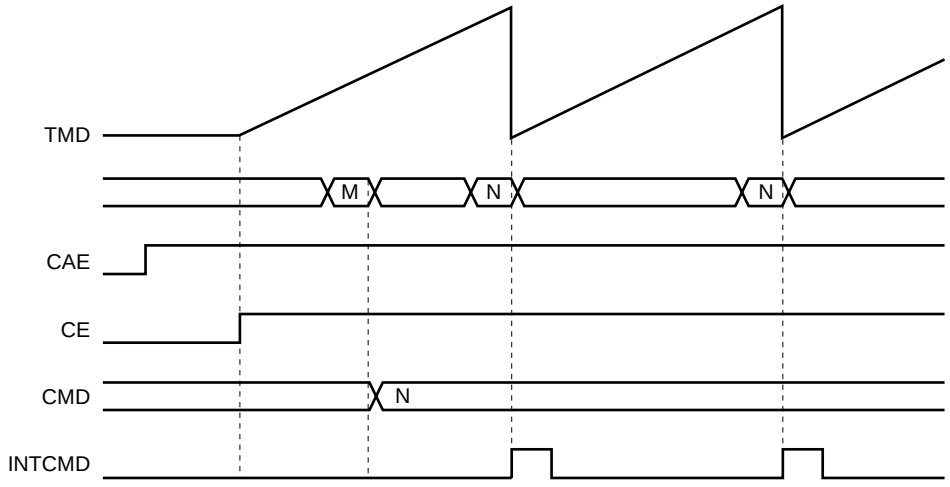
2. The CMD register can be overwritten only once in a single TMD register cycle (from 0000H until an INTCMD interrupt is generated due to a match of the TMD register and CMD register). If this cannot be secured by the application, make sure that the CMD register is not overwritten during timer operation.

3. Note that an INTCMD interrupt will be generated after an overflow if a value less than the counter value is written in the CMD register during TMD register operation (Figure 8-2).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
CMD																	FF1AH	0000H

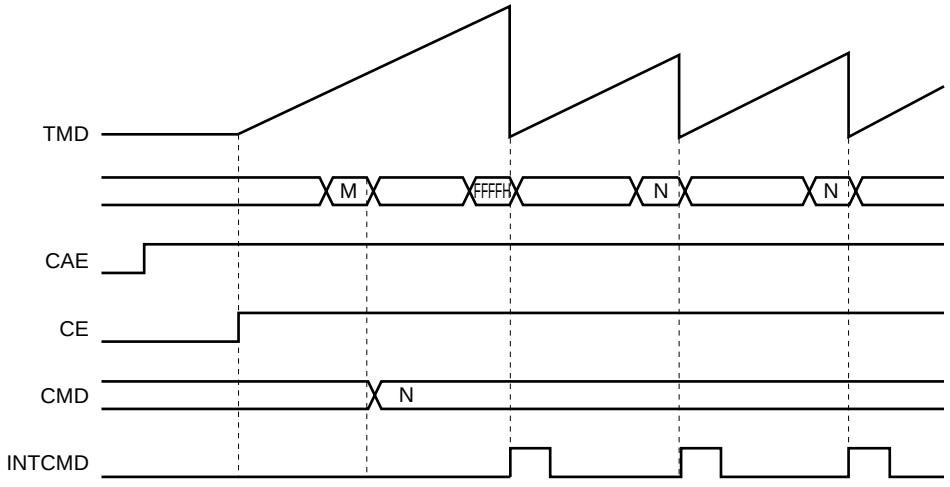
Figure 8-2. Example of Timing During TMD Operation

(a) When TMD < CMD



Remark M = TMD value when overwritten
N = CMD value when overwritten
M < N

(b) When TMD > CMD



Remark M = TMD value when overwritten
N = CMD value when overwritten
M > N

8.5 16-Bit Timer D Control Register

(1) 16-bit timer mode control register D (TMCD)

The TMCD register controls the operation of 16-bit timer D.

This register can be read or written in 8-bit or 1-bit units.

Caution The CAE and other bits cannot be set at the same time. The other bits and the registers of the other TMD units should always be set after the CAE bit has been set.

Figure 8-3. Format of 16-Bit Timer Mode Control Register D (TMCD)

Address: FF5CH After reset: 00H R/W

	7	<6>	<5>	<4>	3	2	<1>	<0>
TMCD	0	CS2	CS1	CS0	0	0	CE	CAE

CS2	CS1	CS0	Count clock selection
0	0	0	$f_X/2$ (0.4 μs)
0	0	1	$f_X/2^2$ (0.8 μs)
0	1	0	$f_X/2^3$ (1.6 μs)
0	1	1	$f_X/2^4$ (3.2 μs)
1	0	0	$f_X/2^5$ (6.4 μs)
1	0	1	$f_X/2^6$ (12.8 μs)
1	1	0	$f_X/2^7$ (25.6 μs)
1	1	1	$f_X/2^8$ (51.2 μs)

CE	TMD operation control
0	Count operation disabled (stopped at 0000H and not operated)
1	Count operation enabled

CAE	Count clock control
0	Entire TMD unit is asynchronously reset and clock supply to TMD unit is stopped.
1	Clock supplied to TMD unit.

Cautions 1. Do not change bits CS2 to CS0 during a timer operation. Before changing these bits, set the CE bit to 0. If these bits are changed during an operation, the operation is not guaranteed.

2. The CE bit will not be cleared even if a mach is detected during a compare operation. To stop the count operation, clear the CE bit.

3. When the CAE bit is 0, the TMD unit is in the reset status. Therefore, to operate TMD, set the CAE bit to 1.

4. If the CAE bit is changed from 1 to 0, all the registers in the TMD unit will be initialized. To set the CAE bit to 1 again, all the registers in the TMD unit must be set again after CAE has been set to 1.

Remarks 1. f_X: System clock oscillation frequency

2. Figures in parentheses are for operation with f_X = 5.0 MHz.

8.6 16-Bit Timer D Operation

(1) Compare operation

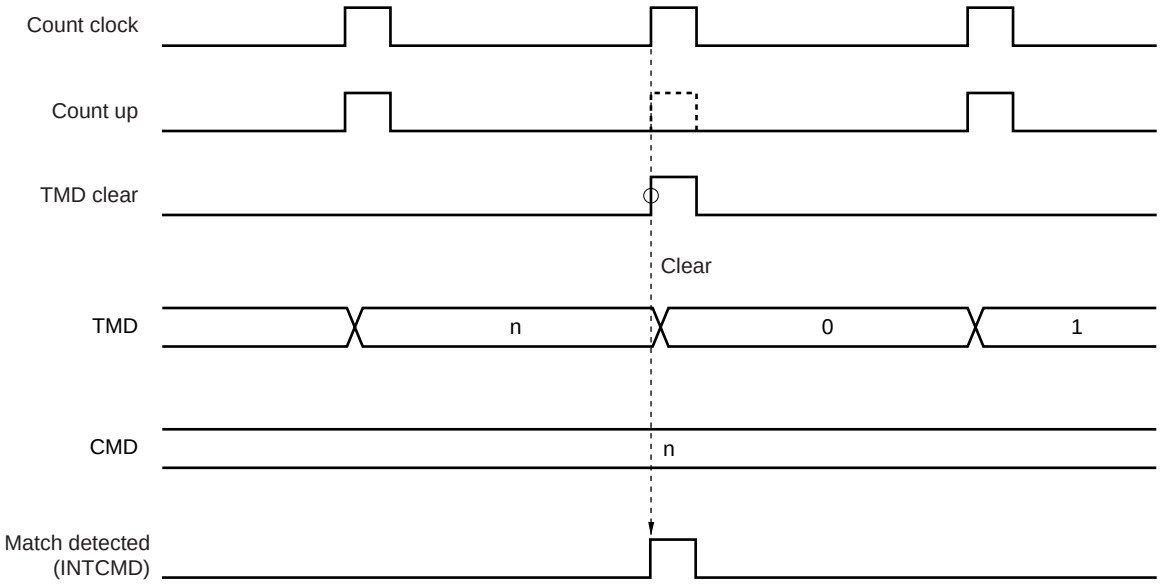
TMD can be used for a compare operation in which the value that was set in the compare register (CMD) is compared with the TMD count value.

If a match is detected by the compare operation, an interrupt (INTCMD) is generated. The generation of the interrupt causes TMD to be cleared (0) at the next count timing. This function enables 16-bit timer D to be used as an interval timer.

CMD can also be set to 0. In this case, when an overflow occurs and TMD becomes 0, a match is detected and INTCMD is generated. Although the TMD value is cleared (0) at the next count timing, INTCMD is not generated by this match.

Figure 8-4. TMD Compare Operation Example (1/2)

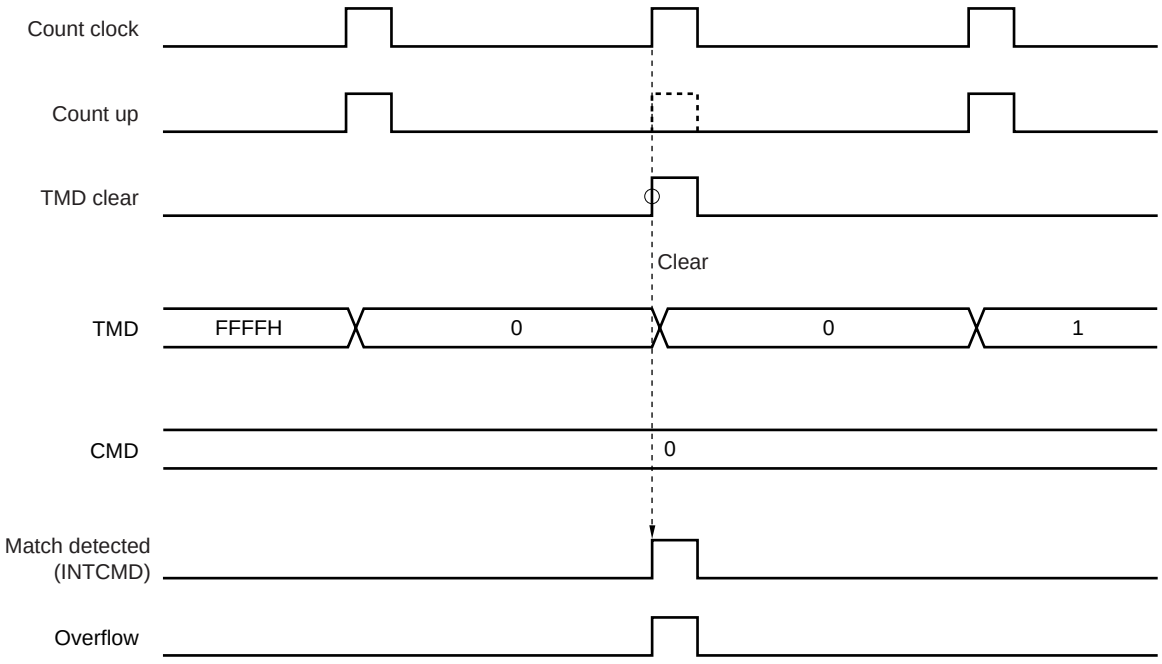
(a) When CMD is set to n (except zero)



Remark Interval time = (n + 1) × (Count clock cycle)
n = 1 to 65,536 (FFFFH)

Figure 8-4. TMD Compare Operation Example (2/2)

(b) When CMD is set to 0



Remark Interval time = (FFFFH + 2) × (Count clock cycle)

8.7 Application Examples

(1) Interval timer

This section explains an example in which 16-bit timer D is used as an interval timer with 16-bit precision.

An interrupt request (INTCMD) is output at equal intervals (see **Figure 8-4 TMD Compare Operation Example**). The setup procedure is shown below.

<1> Set (1) the CAE bit.

<2> Set each register.

- Select the count clock using the CS0 to CS2 bits of the TMCD register.
- Set the compare value in the CMD register.

<3> Start counting by setting (1) the CE bit.

<4> If the TMD register and CMD register values match, the INTCMD interrupt is generated.

<5> The INTCMD interrupt is generated thereafter at equal intervals.

8.8 Cautions

Various cautions concerning 16-bit timer D are shown below.

(1) To operate TMD, first set (1) the CAE bit.

(2) Up to 4 clocks are required after a value is set in the CE bit until the set value is transferred to internal units. When a count operation begins, the count cycle from 0000H to 0001H differs from subsequent cycles.

(3) To initialize the TMD register status and start counting again, clear (0) the CE bit and then set (1) the CE bit after an interval of 4 clocks has elapsed.

(4) Up to 4 clocks are required until the value that was set in the CMD register is transferred to internal units. When writing continuously to the CMD register, be sure to secure a time interval of at least 4 clocks.

(5) The CMD register can be overwritten only once during a timer/counter operation (from 0000H until the INTCMD interrupt is generated due to a match of the TMD register and CMD register). If this cannot be secured by the application, make sure that the CMD register is not overwritten during a timer/counter operation.

(6) The count clock must not be changed during a timer operation. If it is to be overwritten, it should be overwritten after the CE bit is cleared (0). If the count clock is overwritten during a timer operation, operation cannot be guaranteed.

(7) The INTCMD interrupt will occur after an overflow if a value less than the counter value is written in the CMD register during TMD register operation.

CHAPTER 9 8-BIT TIMER/EVENT COUNTER 5

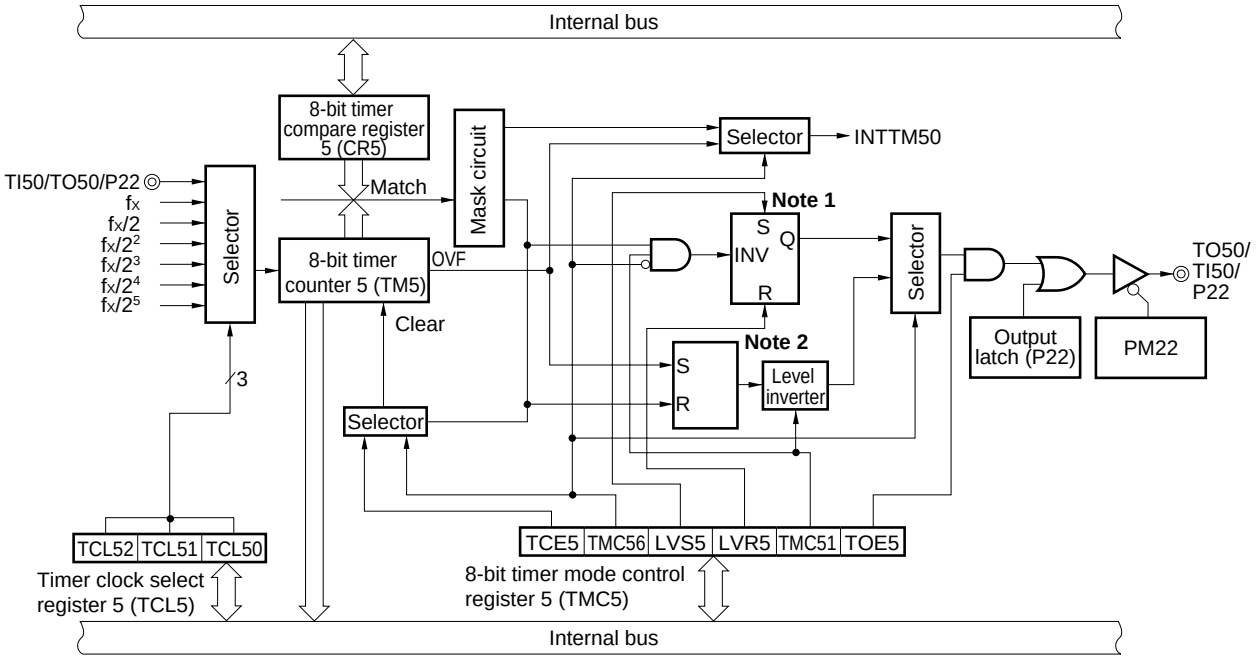
9.1 Functions of 8-Bit Timer/Event Counter 5

8-bit timer/event counter 5 has the following functions.

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

Figure 9-1 shows the block diagram of 8-bit timer/event counter 5.

Figure 9-1. Block Diagram of 8-Bit Timer/Event Counter 5



Notes 1. Timer output F/F

2. PWM output F/F

9.2 Configuration of 8-Bit Timer/Event Counter 5

8-bit timer/event counter 5 includes the following hardware.

Table 9-1. Configuration of 8-Bit Timer/Event Counter 5

Item	Configuration
Timer register	8-bit timer counter 5 (TM5)
Register	8-bit timer compare register 5 (CR5)
Timer output	2 (TO5)
Control registers	Timer clock select register 5 (TCL5) 8-bit timer mode control register 5 (TMC5) Port mode register 2 (PM2)[Note] Port 2 (P2)[Note]

Note See **Figure 5-5 Block Diagram of P22 and P23**.

(1) 8-bit timer counter 5 (TM5)

TM5 is an 8-bit read-only register which counts the count pulses.

The counter is incremented in synchronization with the rising edge of the count clock.

When the count value is read during operation, count clock input is temporary stopped, and then the count value is read.

In the following situations, the count value is set to 00H.

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

<2> When TCE5 is cleared

<3> When TM5 and CR5 match in clear & start mode entered on a match between TM5 and CR5.

(2) 8-bit timer compare register 5 (CR5)

CR5 is a register that can be read/written by 8-bit memory manipulation.

In other than the PWM mode, the value set in CR5 is constantly compared with the 8-bit timer counter 5 (TM5) count value, and an interrupt request (INTTM50) is generated if they match.

In the PWM mode, the TO50 pin is at an active level when TM5 overflows, while the TO50 pin is at an inactive level when the values of TM5 and CR5 match.

It is possible to rewrite the value of CR5 within 00H to FFH.

$\overline{\text{RESET}}$ input sets CR5 to 00H.

Cautions 1. Do not write another value to CR5 during operation in the clear & start mode entered on a match between TM5 and CR5 (TMC56 = 0).

2. In the PWM mode, set the interval between each write operation to CR5 to at least three times the count clock selected by TCL5.

9.3 Control Registers of 8-Bit Timer/Event Counter 5

The following four registers are used to control 8-bit timer/event counter 5.

- Timer clock select register 5 (TCL5)
- 8-bit timer mode control register 5 (TMC5)
- Port mode register 2 (PM2)
- Port 2 (P2)

(1) Timer clock select register 5 (TCL5)

This register sets the count clock of 8-bit timer/event counter 5 and the valid edge of TI50 input.

TCL5 is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TCL5 to 00H.

Figure 9-2. Format of Timer Clock Select Register 5 (TCL5)

Address: FF52H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TCL5	0	0	0	0	0	TCL52	TCL51	TCL50

TCL52	TCL51	TCL50	Count clock selection
0	0	0	TI50 falling edge
0	0	1	TI50 rising edge
0	1	0	f_X (0.2 μs)
0	1	1	$f_X/2$ (0.4 μs)
1	0	0	$f_X/2^2$ (0.8 μs)
1	0	1	$f_X/2^3$ (1.6 μs)
1	1	0	$f_X/2^4$ (3.2 μs)
1	1	1	$f_X/2^5$ (6.4 μs)

Cautions 1. When rewriting other data to TCL5, stop the timer operation beforehand.

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

Remarks 1. f_X: System clock oscillation frequency

2. Figures in parentheses are for operation with f_X = 5.0 MHz.

(2) 8-bit timer mode control register 5 (TMC5)

TMC5 is a register that makes the following five types of settings.

<1> 8-bit timer counter 5 (TM5) count operation control
<2> 8-bit timer counter 5 (TM5) operating mode selection
<3> Timer output F/F (flip-flop) status setting
<4> Active level selection in timer F/F control or PWM (free-running) mode
<5> Timer output control

TMC5 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TMC5 to 00H.

Figure 9-3 shows the TMC5 format.

Figure 9-3. Format of 8-Bit Timer Mode Control Register 5 (TMC5)

Address: FF53H After reset: 00H R/W[Note]

Symbol	<7>	<6>	5	4	<3>	<2>	<1>	<0>
TMC5	TCE5	TMC56	0	0	LVS5	LVR5	TMC51	TOE5

TCE5	TM5 count operation control
0	After clearing the counter to 0, count operation disabled (counter stopped)
1	Count operation starts

TMC56	TM5 operating mode selection
0	Clear and start mode by matching of TM5 and CR5
1	PWM (free-running) mode

LVS5	LVR5	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

TMC51	In other modes (TMC56 = 0)	In PWM mode (TMC56 = 1)
	Timer F/F control	Active level selection
0	Inversion operation disabled	Active high
1	Inversion operation enabled	Active low

TOE5	Timer output control
0	Output disabled (TM50 outputs low level)
1	Output enabled

Note Bits 2 and 3 are write only.

Cautions 1. The settings of LVS5 and LVR5 are valid except in the PWM mode.
2. TMC51 and TOE5 must not be rewritten at the same time.
3. When switching to the PWM mode, TMC56, LVS5, and LVR5 must not be rewritten at the same time.
4. Before rewriting TMC56, the operation must be stopped.

Remarks 1. In PWM mode, PWM output will be inactive because of TCE5 = 0.
2. If LVS5 and LVR5 are read after data is set, 0 is read.
3. The TO50 pin reflects each value of the TMC56, LVS5, LVR5, TMC51, and TOE5 bits regardless of the value of TCE5.

(3) Port mode register 2 (PM2)

This register sets port 2 input/output in 1-bit units.

When using the P22/TO50/TI50 pin for timer output, set PM22 and the output latch of P22 to 0.

When using the P22/TO50/TI50 pin for timer input, set PM22 to 1. At this time, the output latch of P22 may be either 0 or 1.

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

$\overline{\text{RESET}}$ input sets PM2 to FFH.

Figure 9-4. Format of Port Mode Register 2 (PM2)

Address: FF22H After reset: FFH R/W

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

PM2n	P2n pin I/O mode selection (n = 0 to 4)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

9.4 Operations of 8-Bit Timer/Event Counter 5

9.4.1 Interval timer (8-bit) operation

The 8-bit timer/event counter operates as an interval timer that generates interrupt requests repeatedly at intervals of the count value preset to 8-bit timer compare register 5 (CR5).

When the count value of 8-bit timer counter 5 (TM5) matches the value set to CR5, counting continues with the TM5 value cleared to 0 and an interrupt request signal (INTTM50) is generated.

The count clock of TM5 can be selected with bits 0 to 2 (TCL50 to TCL52) of timer clock select register 5 (TCL5).

[Setting]

<1> Set the registers.

- TCL5: Select count clock
- CR5: Compare value
- TMC5: Select count operation stop, clear and start mode by match of TM5 and CR5, and timer output disabled
  (TMC5 = 0000×××0B × = don't care)

<2> After TCE5 = 1 is set, count operation starts.

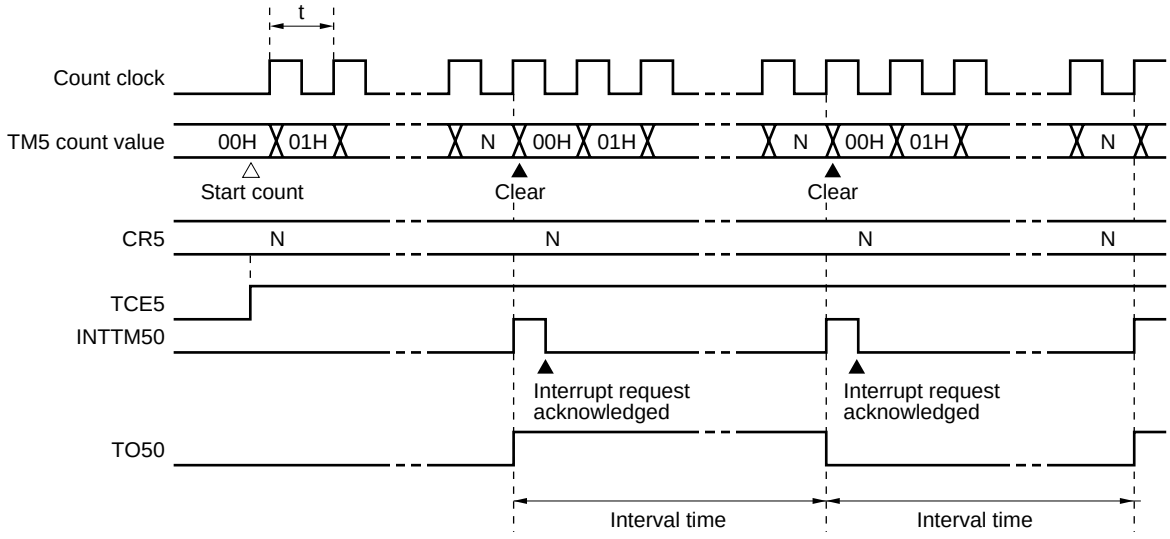
<3> If the values of TM5 and CR5 match, INTTM50 is generated (TM5 is cleared to 00H).

<4> INTTM50 is generated repeatedly at the same interval. Set TCE5 to 0 to stop the count operation.

Caution Do not write another value to CR5 during operation.

Figure 9-5. Interval Timer Operation Timing (1/2)

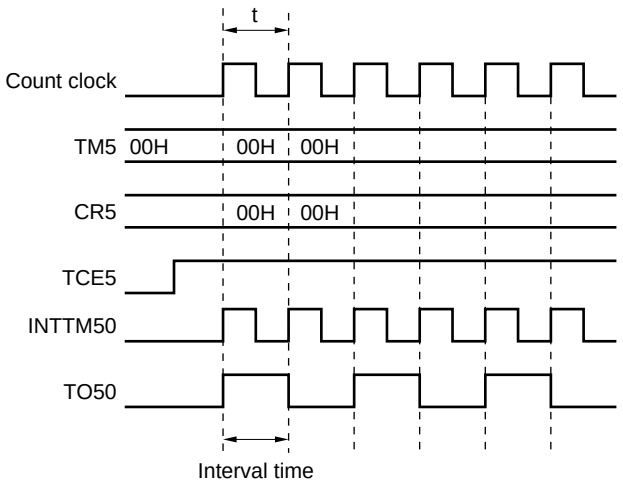
(a) Basic operation



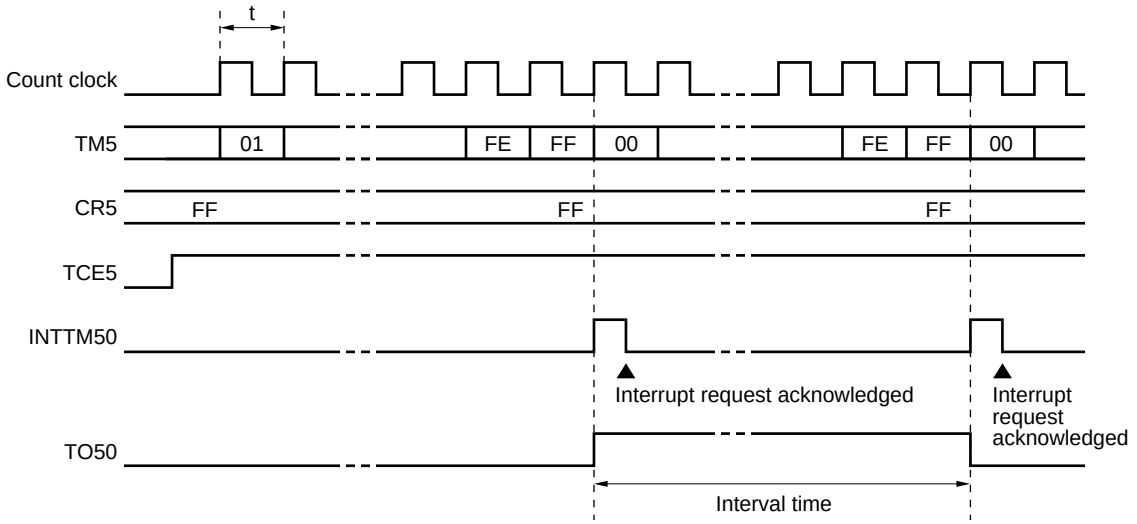
Remark Interval time = (N + 1) × t
N = 00H to FFH

Figure 9-5. Interval Timer Operation Timing (2/2)

(b) When CR5 = 00H



(c) When CR5 = FFH



9.4.2 External event counter operation

The external event counter counts the number of external clock pulses to be input to TI50 by 8-bit timer counter 5 (TM5).

TM5 is incremented each time the valid edge specified by timer clock select register 5 (TCL5) is input. Either the rising or falling edge can be selected.

When the TM5 count value matches the value of 8-bit timer compare register 5 (CR5), TM5 is cleared to 0 and an interrupt request signal (INTTM50) is generated.

INTTM50 is subsequently generated, whenever the TM5 count value matches the value of CR5.

[Setting]

<1> Set the registers.

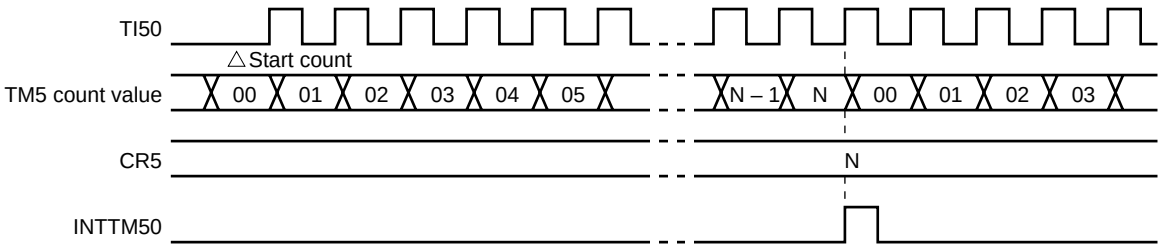
- TCL5: Select edge of TI50 input
  Fall of TI50 → TCL5 = 00H
  Rise of TI50 → TCL5 = 01H
- CR5: Compare value
- TMC5: Select count operation stop, clear and start mode by match of TM5 and CR5, timer F/F reverse operation disabled, and timer output disabled
  (TMC5 = 0000××00B × = don't care)

<2> After TCE5 = 1 is set, an operation to count the number of pulses input by TI50 starts.

<3> If the values of TM5 and CR5 match, INTTM50 is generated (TM5 is cleared to 00H).

<4> After this, INTTM50 is generated each time the values of TM5 and CR5 match.

Figure 9-6. External Event Counter Operation Timing (with Rising Edge Specified)



N = 00H to FFH

9.4.3 Square-wave output (8-bit resolution) operation

A square wave with any selected frequency is output at intervals determined by the value preset to 8-bit timer compare register 5 (CR5).

The TO50 pin output status is reversed at intervals determined by the count value preset to CR5 by setting bit 0 (TOE5) of 8-bit timer mode control register 5 (TMC5) to 1. This enables a square wave with any selected frequency to be output (duty = 50%).

[Setting]

<1> Set each register.

- Set port latch (P22) and port mode register (PM22) to 0
- TCL5: Select count clock
- CR5: Compare value
- TMC5: Count operation stopped, clear and start mode by match of TM5 and CR5

LVS5	LVR5	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
(TMC5 = 00001011B or 00000111B)

<2> After TCE5 = 1 is set, count operation starts.

<3> Timer output F/F is reversed by match of TM5 and CR5.
After INTTM50 is generated, TM5 is cleared to 00H.

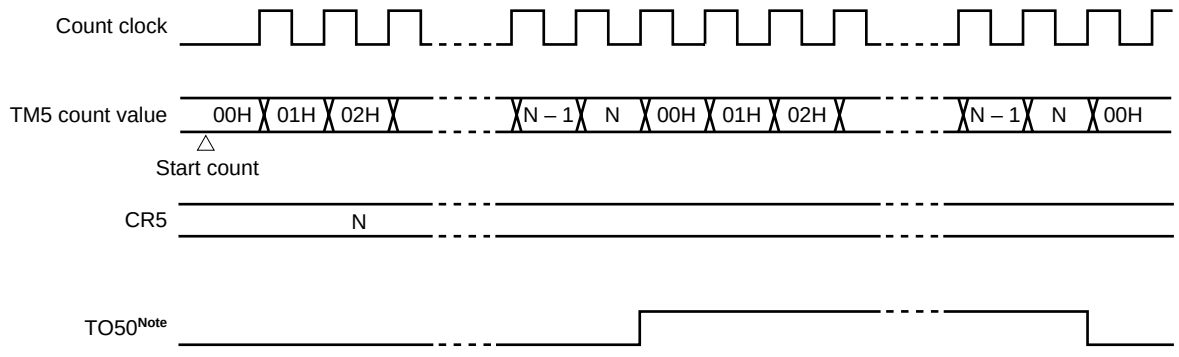
<4> Timer output F/F is reversed at the same interval and square wave is output from TO50.

The frequency is as follows.

- Frequency = $f_{CNT}/2\ (N + 1)$
  (N: 00 to FFH, f_{CNT}: Count clock)

Caution Do not write another value to CR5 during operation.

Figure 9-7. Square-Wave Output Operation Timing



Note The TO50 output initial value can be set by bits 2 and 3 (LVR5, LVS5) of 8-bit timer mode control register 5 (TMC5).

9.4.4 PWM output operation

The 8-bit timer/event counter operates as PWM output when bit 6 (TMC56) of 8-bit timer mode control register 5 (TMC5) is set to 1.

The duty rate pulse determined by the value set to 8-bit timer compare register 5 (CR5) is output from TO50.

Set the active level width of the PWM pulse to CR5, and select the active level using bit 1 of TMC5 (TMC51).

The count clock can be selected with bits 0 to 2 (TCL50 to TCL52) of timer clock select register 5 (TCL5).

PWM output enable/disable can be selected with bit 0 of TMC5 (TOE5).

Caution In the PWM mode, set the rewrite interval of CR5 to 3 clocks or more of the count clock (clock selected by TCL5).

(1) PWM output basic operation

[Setting]

<1> Set the registers.

- Set the port latch (P22) and port mode register 2 (PM22) to 0.
- TCL5: Select count clock
- CR5: Compare value
- TMC5: Count operation stopped, PWM mode selected, timer output F/F does not change

TMC51	Active Level Selection
0	Active high
1	Active low

Timer output enable
(TMC5 = 01000001B or 01000011B)

<2> When TCE5 is set to 1, the count operation starts.
Set TCE5 to 0 to stop count operation.

[PWM output operation]

<1> PWM output (output from TO50) outputs the inactive level from when the count operation starts until an overflow is generated.

<2> When an overflow is generated, the active level is output.
The active level is output until CR5 matches the count value of 8-bit timer counter 5 (TM5).

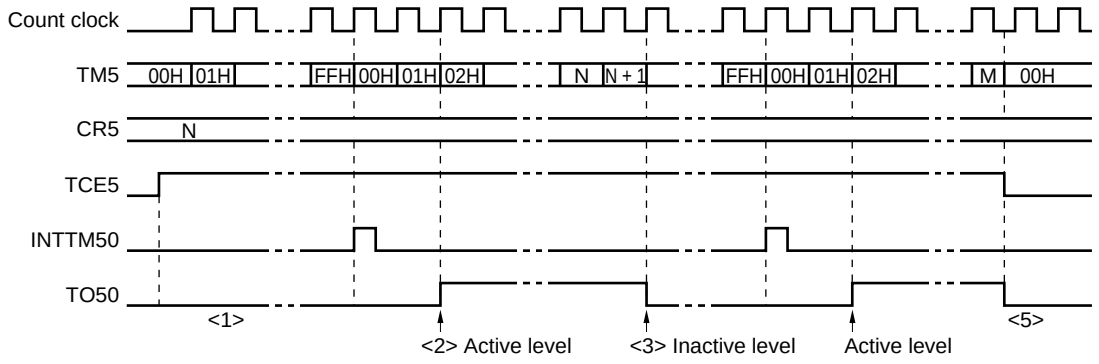
<3> When CR5 matches the count value, the inactive level is output, and continues being output until an overflow is generated again.

<4> Operations <2> and <3> are repeated until the count operation stops.

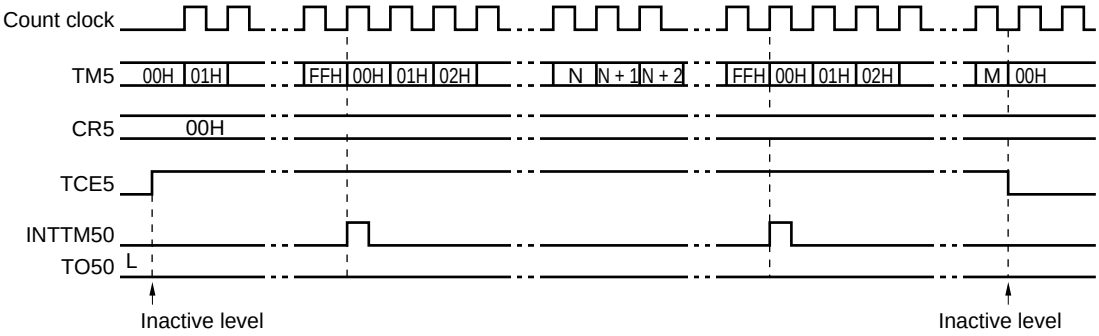
<5> When the count operation is stopped with TCE5 = 0, PWM output changes to the inactive level.

Figure 9-8. PWM Output Operation Timing

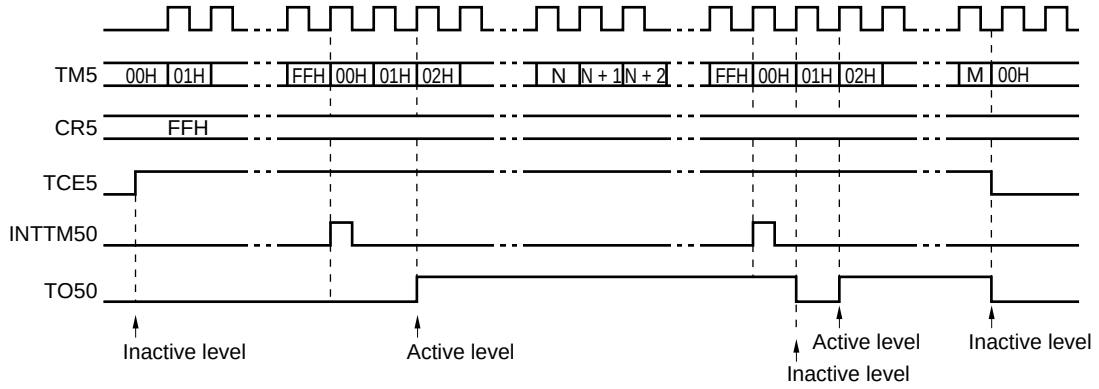
(a) Basic operation (active level = H)



(b) CR5 = 00H



(c) CR5 = FFH

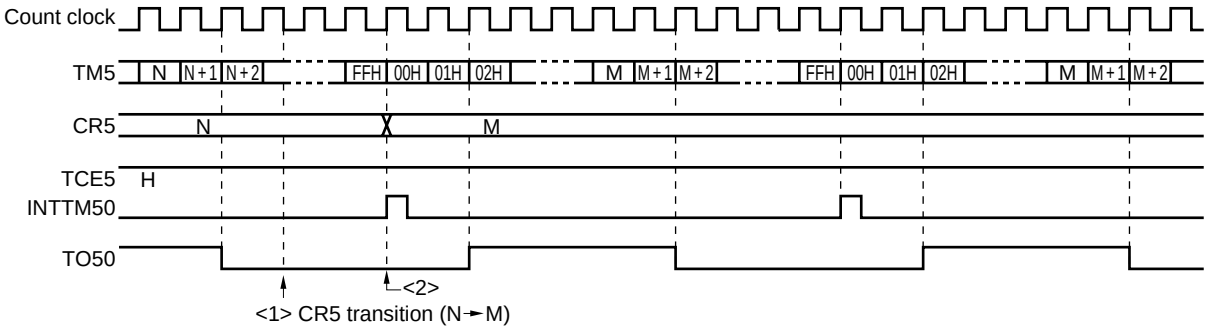


Remark <1> to <3> and <5> in Figure 9-8 (a) correspond to <1> to <3> and <5> in **9.4.4 (1) PWM output basic operation [PWM output operation]**.

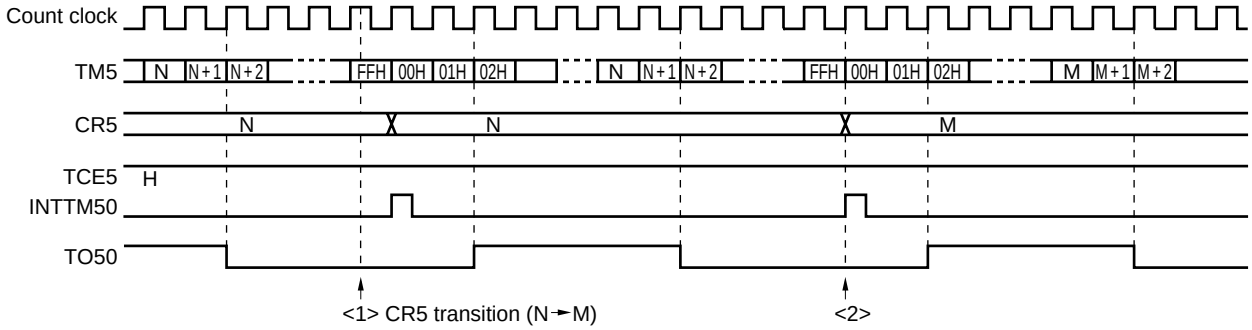
(2) Operation by CR5 transition

Figure 9-9. Timing of Operation by CR5 Transition

(a) When CR5 value changes from N to M before rising edge of FFH clock
→ Value is reloaded to CR5 at next overflow



(b) When CR5 value changes from N to M after rising edge of FFH clock
→ Value is reloaded to CR5 at second overflow



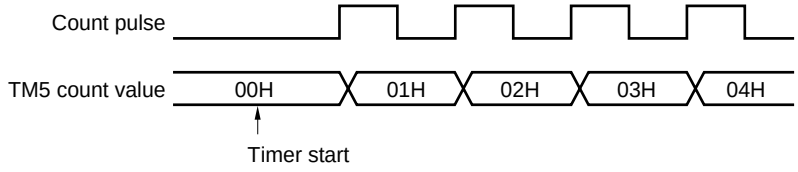
Caution If CR5 is read between <1> and <2> in Figure 9-9, the read value is different than the actual value during operation (read value: M, actual CR5 value: N).

9.5 Cautions on 8-Bit Timer/Event Counter 5

(1) Timer start errors

An error of up to one clock may occur in the time required for a match signal to be generated after timer start. This is because 8-bit timer counter 5 (TM5) is started asynchronously with the count pulse.

Figure 9-10. Start Timing of 8-Bit Timer Counter 5



CHAPTER 10 WATCHDOG TIMER

10.1 Watchdog Timer Functions

The watchdog timer has the following functions.

- Watchdog timer
- Interval timer

Caution Select the watchdog timer mode or interval timer mode by using the watchdog timer mode register (WDTM).

(1) Watchdog timer

The watchdog timer is used to detect inadvertent program loops. When an inadvertent loop is detected, a non-maskable interrupt or the $\overline{\text{RESET}}$ signal can be generated.

Table 10-1. Inadvertent Loop Detection Time of Watchdog Timer

Inadvertent Loop Detection Time	At f_X = 5.0 MHz
$2^{11} \times 1/f_X$	410 μs
$2^{13} \times 1/f_X$	1.64 ms
$2^{15} \times 1/f_X$	6.55 ms
$2^{17} \times 1/f_X$	26.2 ms

f_X: System clock oscillation frequency

(2) Interval timer

The interval timer generates an interrupt at a given interval set in advance.

Table 10-2. Interval Time

Interval	At f_X = 5.0 MHz
$2^{11} \times 1/f_X$	410 μs
$2^{13} \times 1/f_X$	1.64 ms
$2^{15} \times 1/f_X$	6.55 ms
$2^{17} \times 1/f_X$	26.2 ms

f_X: System clock oscillation frequency

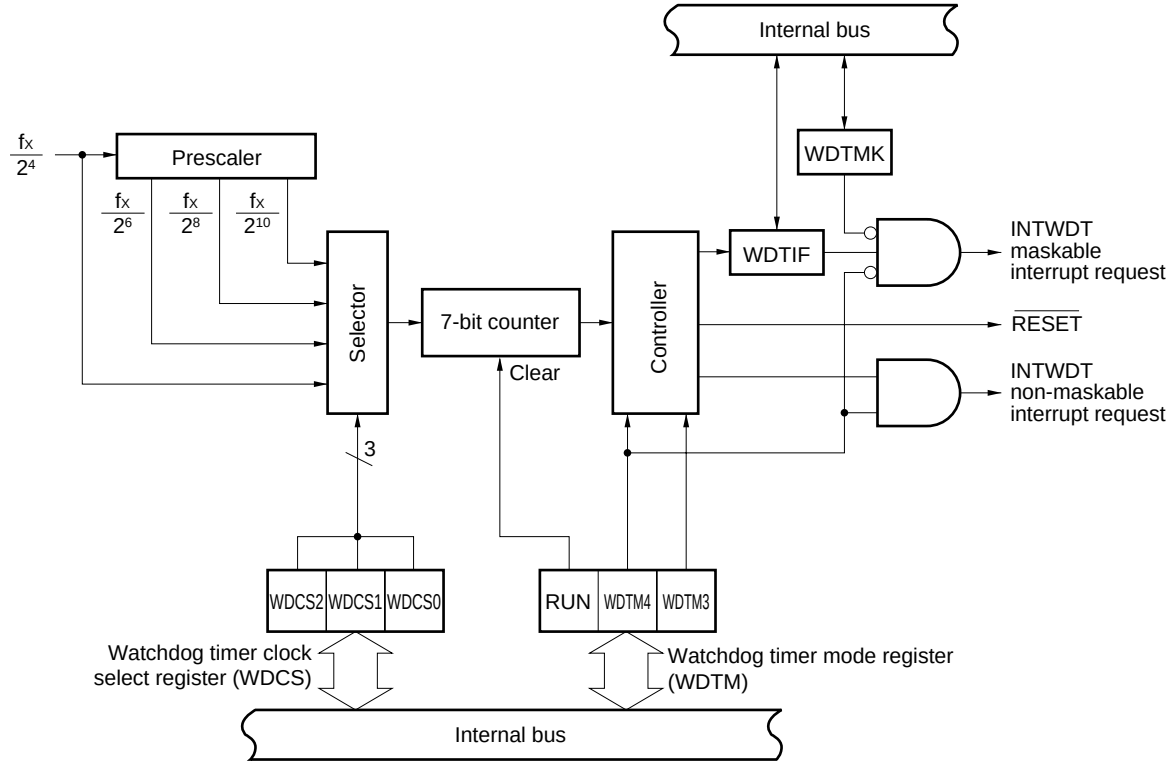
10.2 Watchdog Timer Configuration

The watchdog timer includes the following hardware.

Table 10-3. Configuration of Watchdog Timer

Item	Configuration
Control register	Watchdog timer clock select register (WDCS) Watchdog timer mode register (WDTM)

Figure 10-1. Block Diagram of Watchdog Timer



Remark fx: System clock oscillation frequency

10.3 Watchdog Timer Control Registers

The following two registers are used to control the watchdog timer.

- Watchdog timer clock select register (WDCS)
- Watchdog timer mode register (WDTM)

(1) Watchdog timer clock select register (WDCS)

This register sets the watchdog timer count clock.

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

$\overline{\text{RESET}}$ input clears WDCS to 00H.

Figure 10-2. Format of Watchdog Timer Clock Select Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
WDCS	0	0	0	0	0	WDCS2	WDCS1	WDCS0	FF42H	00H	R/W

WDCS2	WDCS1	WDCS0	Watchdog timer count clock selection	Interval Time
0	0	0	$f_X/2^4$ (313 kHz)	$2^{11}/f_X$ (410 μs)
0	1	0	$f_X/2^6$ (78.1 kHz)	$2^{13}/f_X$ (1.64 ms)
1	0	0	$f_X/2^8$ (19.5 kHz)	$2^{15}/f_X$ (6.55 ms)
1	1	0	$f_X/2^{10}$ (4.88 kHz)	$2^{17}/f_X$ (26.2 ms)
Other than above			Setting prohibited	

Remarks 1. f_X: System clock oscillation frequency

2. Figures in parentheses are for operation with f_X = 5.0 MHz.

(2) Watchdog timer mode register (WDTM)

This register sets the operation mode of the watchdog timer, and enables/disables counting of the watchdog timer.

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

$\overline{\text{RESET}}$ input clears WDTM to 00H.

Figure 10-3. Format of Watchdog Timer Mode Register

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
WDTM	RUN	0	0	WDTM4	WDTM3	0	0	0	FFF9H	00H	R/W

RUN	Watchdog timer operation selection[Note 1]
0	Stops counting
1	Clears counter and starts counting.

WDTM4	WDTM3	Watchdog timer operation mode selection[Note 2]
0	0	Operation stopped
0	1	Interval timer mode (overflow and maskable interrupt occur)[Note 3]
1	0	Watchdog timer mode 1 (overflow and non-maskable interrupt occur)
1	1	Watchdog timer mode 2 (overflow occurs and reset operation started)

Notes 1. Once RUN has been set to 1, it cannot be cleared to 0 by software. Therefore, when counting is started, it cannot be stopped by any means other than $\overline{\text{RESET}}$ input.

2. Once WDTM3 and WDTM4 have been set to 1, they cannot be cleared to 0 by software.

3. The watchdog timer starts operations as an interval timer when RUN is set to 1.

Cautions 1. When the watchdog timer is cleared by setting RUN to 1, the actual overflow time is up to 0.8% shorter than the time set by the watchdog timer clock select register (WDCS).

2. In watchdog timer mode 1 or 2, set WDTM4 to 1 after confirming that WDTIF (bit 0 of interrupt request flag register 0 (IF0)) is set to 0. When watchdog timer mode 1 or 2 is selected under the condition where WDTIF is 1, a non-maskable interrupt occurs at the completion of rewriting.

10.4 Operation of Watchdog Timer

10.4.1 Operation as watchdog timer

The watchdog timer detects an inadvertent program loop when bit 4 (WDTM4) of the watchdog timer mode register (WDTM) is set to 1.

The count clock (inadvertent loop detection time interval) of the watchdog timer can be selected by bits 0 to 2 (WDCS0 to WDCS2) of the watchdog timer clock select register (WDCS). By setting bit 7 (RUN) of WDTM to 1, the watchdog timer is started. Set RUN to 1 within the set inadvertent loop detection time interval after the watchdog timer has been started. By setting RUN to 1, the watchdog timer can be cleared and start counting. If RUN is not set to 1, and the inadvertent loop detection time is exceeded, the system is reset or a non-maskable interrupt is generated, depending on value of bit 3 (WDTM3) of WDTM.

The watchdog timer continues operation in the HALT mode, but stops in the STOP mode. Therefore, set RUN to 1 before entering the STOP mode to clear the watchdog timer, and then execute the STOP instruction.

Caution The actual inadvertent loop detection time may be up to 0.8% shorter than the set time.

Table 10-4. Inadvertent Loop Detection Time of Watchdog Timer

WDCS2	WDCS1	WDCS0	Inadvertent Loop Detection Time	At f_X = 5.0 MHz
0	0	0	$2^{11} \times 1/f_X$	410 μs
0	1	0	$2^{13} \times 1/f_X$	1.64 ms
1	0	0	$2^{15} \times 1/f_X$	6.55 ms
1	1	0	$2^{17} \times 1/f_X$	26.2 ms

f_X: System clock oscillation frequency

10.4.2 Operation as interval timer

When bits 4 and 3 (WDTM4, WDTM3) of the watchdog timer mode register (WDTM) are set to 0 and 1 respectively, the watchdog timer also operates as an interval timer that repeatedly generates an interrupt at time intervals specified by a count value set in advance.

Select the count clock (or interval time) by setting bits 0 to 2 (WDCS0 to WDCS2) of the watchdog timer clock select register (WDCS). The watchdog timer starts operation as an interval timer when the RUN bit (bit 7 of WDTM) is set to 1.

In the interval timer mode, the interrupt mask flag (WDTMK) is valid, and a maskable interrupt (INTWDT) can be generated. The priority of INTWDT is set as the highest of all the maskable interrupts.

The interval timer continues operation in the HALT mode, but stops in the STOP mode. Therefore, set RUN to 1 before entering the STOP mode to clear the interval timer, and then execute the STOP instruction.

Cautions 1. Once bit 4 (WDTM4) of WDTM is set to 1 (when the watchdog timer mode is selected), the interval timer mode is not set, unless the $\overline{\text{RESET}}$ signal is input.

2. The interval time immediately after the setting by WDTM may be up to 0.8% shorter than the set time.

Table 10-5. Interval Time of Interval Timer

WDCS2	WDCS1	WDCS0	Interval Time	At f_X = 5.0 MHz
0	0	0	$2^{11} \times 1/f_X$	410 μs
0	1	0	$2^{13} \times 1/f_X$	1.64 ms
1	0	0	$2^{15} \times 1/f_X$	6.55 ms
1	1	0	$2^{17} \times 1/f_X$	26.2 ms

f_X: System clock oscillation frequency

CHAPTER 11 SERIAL INTERFACE 2

11.1 Functions of Serial Interface 2

Serial interface 2 has the following three modes.

- Operation stopped mode
- Asynchronous serial interface (UART) mode
- 3-wire serial I/O mode

(1) Operation stopped mode

This mode is used when serial transfer is not performed. Power consumption is minimized in this mode.

(2) Asynchronous serial interface (UART) mode

This mode is used to send and receive the one byte of data that follows a start bit. It supports full-duplex communication.

Serial interface 2 contains a dedicated UART baud rate generator, enabling communication over a wide range of baud rates. It is also possible to define baud rates by dividing the frequency of the input clock pulse at the ASCK2 pin.

(3) 3-wire serial I/O mode (switchable between MSB-first and LSB-first transmission)

This mode is used to transmit 8-bit data, using three lines: a serial clock ($\overline{\text{SCK2}}$) line and two serial data lines (SI2 and SO2).

As it supports simultaneous transmission and reception, 3-wire serial I/O mode requires less processing time for data transmission than asynchronous serial interface mode.

Because, in 3-wire serial I/O mode, it is possible to select whether 8-bit data transmission begins with the MSB or LSB, serial interface 2 can be connected to any device regardless of whether that device is designed for MSB-first or LSB-first transmission.

3-wire serial I/O mode can be used to connect IC and display controllers having clock synchronous serial interfaces.

11.2 Configuration of Serial Interface 2

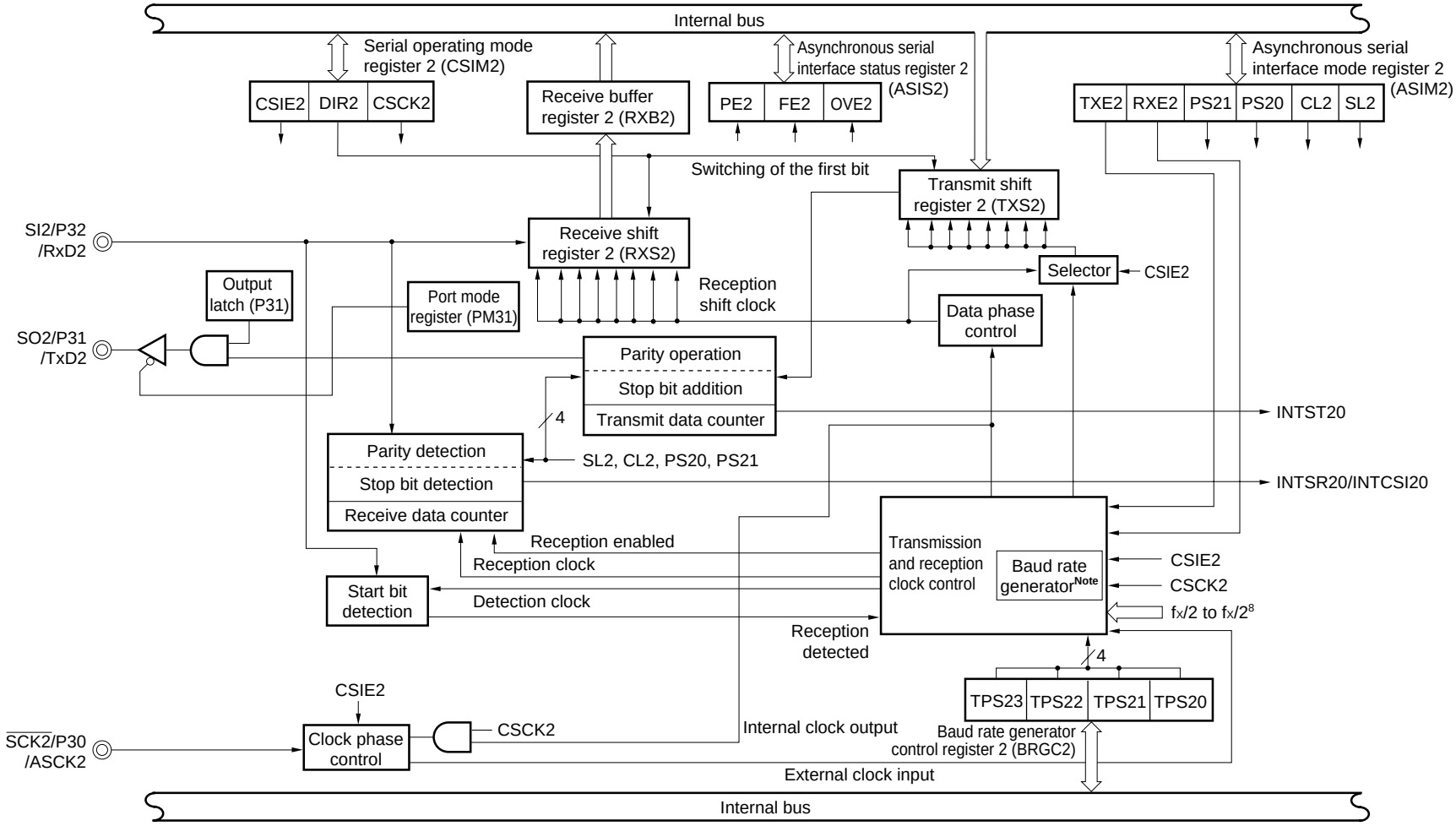
Serial interface 2 includes the following hardware.

Table 11-1. Configuration of Serial Interface 2

Item	Configuration
Registers	Transmit shift register 2 (TXS2) Receive shift register 2 (RXS2) Receive buffer register 2 (RXB2)
Control registers	Serial operating mode register 2 (CSIM2) Asynchronous serial interface mode register 2 (ASIM2) Asynchronous serial interface status register 2 (ASIS2) Baud rate generator control register 2 (BRGC2) Port mode register 3 (PM3)[Note] Port 3 (P3)[Note]

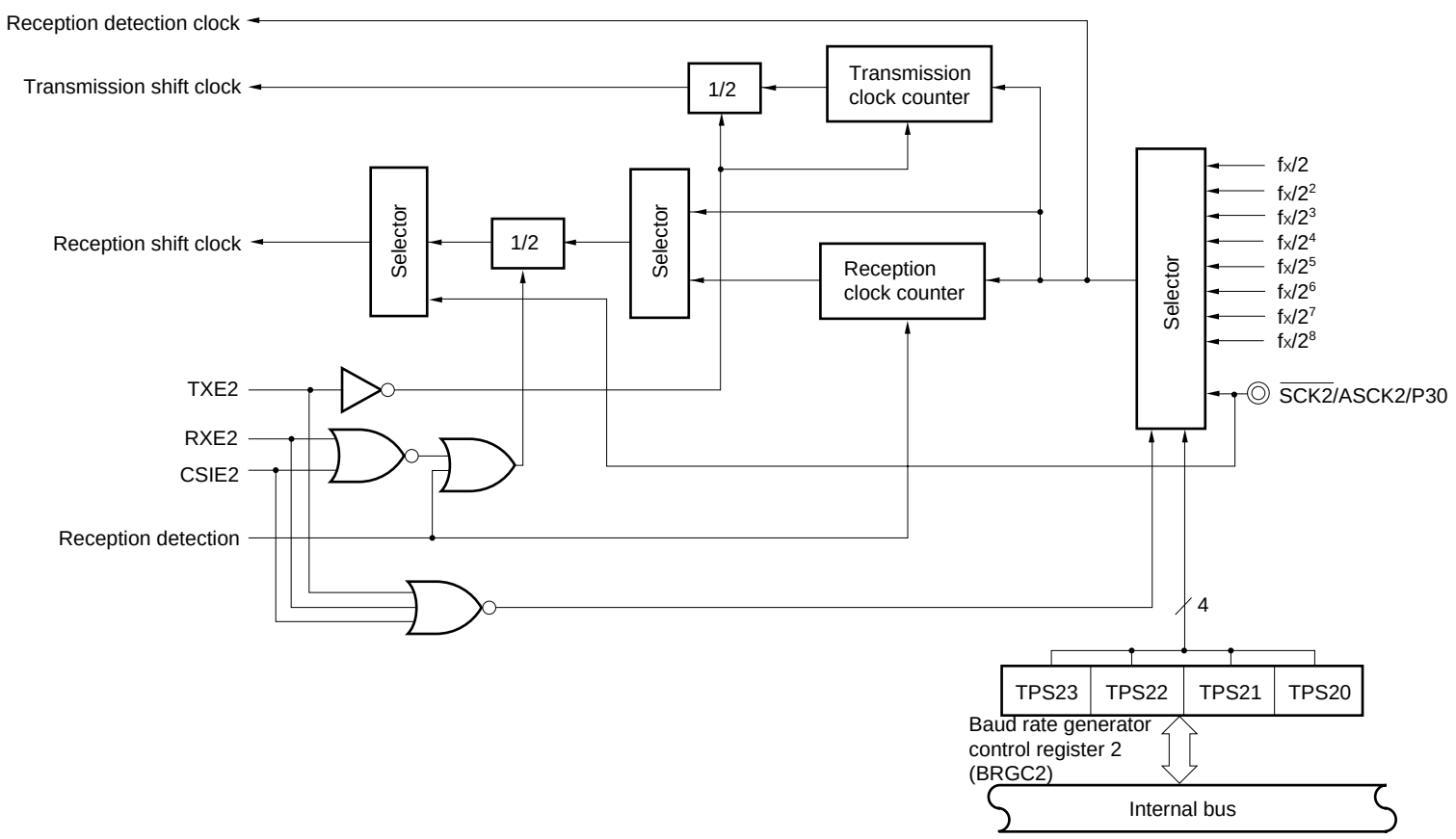
Note See **Figure 5-7 Block Diagram of P30**, **Figure 5-8 Block Diagram of P31**, and **Figure 5-9 Block Diagram of P32 and P33**.

Figure 11-1. Block Diagram of Serial Interface 2



Note See **Figure 11-2** for the configuration of the baud rate generator.

Figure 11-2. Block Diagram of Baud Rate Generator



(1) Transmit shift register 2 (TXS2)

TXS2 is a register in which transmit data is prepared. Data written to TXS2 is transmitted as serial data.

When the data length is seven bits, bits 0 to 6 of the data in TXS2 will be transmit data. Writing data to TXS2 triggers transmission.

TXS2 can be written with an 8-bit memory manipulation instruction, but cannot be read.

$\overline{\text{RESET}}$ input sets TXS2 to FFH.

Caution Do not write to TXS2 during transmission.

TXS2 and receive buffer register 2 (RXB2) are mapped at the same address, so that any attempt to read from TXS2 results in a value being read from RXB2.

(2) Receive shift register 2 (RXS2)

RXS2 is a register in which serial data, received at the RxD2 pin, is converted to parallel data. Once one entire byte has been received, RXS2 transfers the receive data to receive buffer register 2 (RXB2).

RXS2 cannot be manipulated directly by a program.

(3) Receive buffer register 2 (RXB2)

RXB2 holds receive data. New receive data is transferred from receive shift register 2 (RXS2) per byte of data received.

When the data length is specified as seven bits, the receive data is sent to bits 0 to 6 of RXB2, in which the MSB is always fixed to 0.

RXB2 can be read with an 8-bit memory manipulation instruction, but cannot be written to.

$\overline{\text{RESET}}$ input makes RXB2 undefined.

Caution RXB2 and transmit shift register 2 (TXS2) are mapped at the same address, so that any attempt to write to RXB2 results in a value being written to TXS2.

(4) Transmission controller

The transmission controller controls transmission. For example, it adds start, parity, and stop bits to the data in transmit shift register 2 (TXS2), according to the setting of asynchronous serial interface mode register 2 (ASIM2).

(5) Reception controller

The reception controller controls reception according to the setting of asynchronous serial interface mode register 2 (ASIM2). It also checks for errors, such as parity errors, during reception. If an error is detected, asynchronous serial interface status register 2 (ASIS2) is set according to the status of the error.

11.3 Control Registers of Serial Interface 2

Serial interface 2 is controlled by the following six registers.

- Serial operating mode register 2 (CSIM2)
- Asynchronous serial interface mode register 2 (ASIM2)
- Asynchronous serial interface status register 2 (ASIS2)
- Baud rate generator control register 2 (BRGC2)
- Port mode register 3 (PM3)
- Port 3 (P3)

(1) Serial operating mode register 2 (CSIM2)

CSIM2 is used to make the settings related to 3-wire serial I/O mode.

CSIM2 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM2 to 00H.

Figure 11-3. Format of Serial Operating Mode Register 2

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
CSIM2	CSIE2	0	0	0	0	DIR2	CSCK2	0	FF72H	00H	R/W

CSIE2	3-wire serial I/O mode operation control
0	Operation disabled
1	Operation enabled

DIR2	First-bit specification
0	MSB
1	LSB

CSCK2	3-wire serial I/O mode clock selection
0	External clock pulse input to the $\overline{\text{SCK2}}$ pin.
1	Output of the dedicated baud rate generator

Cautions 1. Bits 0 and 3 to 6 must be fixed to 0.

2. CSIM2 must be cleared to 00H, if UART mode is selected.

3. Switch operating modes after halting the serial transmit/receive operation.

(2) Asynchronous serial interface mode register 2 (ASIM2)

ASIM2 is used to make the settings related to asynchronous serial interface mode.

ASIM2 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ASIM2 to 00H.

Figure 11-4. Format of Asynchronous Serial Interface Mode Register 2

Symbol	<7>	<6>	5	4	3	2	1	0	Address	After reset	R/W
ASIM2	TXE2	RXE2	PS21	PS20	CL2	SL2	0	0	FF70H	00H	R/W

TXE2	Transmit operation control
0	Transmit operation stopped
1	Transmit operation enabled

RXE2	Receive operation control
0	Receive operation stopped
1	Receive operation enabled

PS21	PS20	Parity bit specification
0	0	No parity
0	1	Always add 0 parity at transmission. Parity check is not performed at reception (no parity error occurs).
1	0	Odd parity
1	1	Even parity

CL2	Transmit data character length specification
0	7 bits
1	8 bits

SL2	Transmit data stop bit length specification
0	1 bit
1	2 bits

Cautions 1. Bits 0 and 1 must be fixed to 0.

2. If 3-wire serial I/O mode is selected, ASIM2 must be cleared to 00H.

3. Switch operating modes after halting the serial transmit/receive operation.

Table 11-2. Operation Mode Settings for Serial Interface 2

(1) Operation stopped mode

ASIM2		CSIM2			PM32	P32	PM31	P31	PM30	P30	First Bit	Shift Clock	P32/SI2/RxD2 Pin Function	P31/SO2/TxD2 Pin Function	P30/$\overline{SCK2}$/ASCK2 Pin Function
TXE2	RXE2	CSIE2	DIR2	CSCK2											
0	0	0	×	×	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	–	–	P32	P31	P30
Other than above											Setting prohibited				

(2) 3-wire serial I/O mode

ASIM2		CSIM2			PM32	P32	PM31	P31	PM30	P30	First Bit	Shift Clock	P32/SI2/RxD2 Pin Function	P31/SO2/TxD2 Pin Function	P30/$\overline{SCK2}$/ASCK2 Pin Function
TXE2	RXE2	CSIE2	DIR2	CSCK2											
0	0	1	0	0	1Note 2	×Note 2	0	1	1	×	MSB	External clock	SI2Note 2	SO2 (CMOS output)	$\overline{SCK2}$ input
				1					0	1		Internal clock			$\overline{SCK2}$ output
		1	1	0					1	×	LSB	External clock			$\overline{SCK2}$ input
				1					0	1		Internal clock			$\overline{SCK2}$ output
Other than above											Setting prohibited				

(3) Asynchronous serial interface mode

ASIM2		CSIM2			PM32	P32	PM31	P31	PM30	P30	First Bit	Shift Clock	P32/SI2/RxD2 Pin Function	P31/SO2/TxD2 Pin Function	P30/$\overline{SCK2}$/ASCK2 Pin Function
TXE2	RXE2	CSIE2	DIR2	CSCK2											
1	0	0	0	0	×Note 1	×Note 1	0	1	1	×	LSB	External clock	P32	TxD2 (CMOS output)	ASCK2 input
									×Note 1	×Note 1		Internal clock			P30
0	1	0	0	0	1	×	×Note 1	×Note 1	1	×		External clock	RxD2	P31	ASCK2 input
									×Note 1	×Note 1		Internal clock			P30
1	1	0	0	0	1	×	0	1	1	×		External clock		TxD2 (CMOS output)	ASCK2 input
									×Note 1	×Note 1		Internal clock			P30
Other than above											Setting prohibited				

Notes 1. These pins can be used for port functions.

2. When only transmission is used, this pin can be used as P32 (CMOS I/O).

Remark ×: Don't care

(3) Asynchronous serial interface status register 2 (ASIS2)

ASIS2 is used to display the type of a reception error, if it occurs while asynchronous serial interface mode is set.

ASIS2 is read with a 1-bit or 8-bit memory manipulation instruction.

The contents of ASIS2 are undefined in 3-wire serial I/O mode.

$\overline{\text{RESET}}$ input clears ASIS2 to 00H.

Figure 11-5. Format of Asynchronous Serial Interface Status Register 2

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
ASIS2	0	0	0	0	0	PE2	FE2	OVE2	FF71H	00H	R

PE2	Parity error flag
0	No parity error has occurred.
1	A parity error has occurred (parity mismatch in transmission data).

FE2	Framing error flag
0	No framing error has occurred.
1	A framing error has occurred (no stop bit detected)Note 1.

OVE2	Overrun error flag
0	No overrun error has occurred.
1	An overrun error has occurredNote 2. (Before data was read from the receive buffer register, the subsequent reception sequence was completed.)

Notes 1. Even when the stop bit length is set to 2 bits by setting bit 2 (SL2) of asynchronous serial interface mode register 2 (ASIM2), the stop bit detection in the case of reception is performed with 1 bit.

2. Be sure to read receive buffer register 2 (RXB2) when an overrun error occurs. If not, every time the data is received an overrun error occurs.

(4) Baud rate generator control register 2 (BRGC2)

BRGC2 is used to specify the serial clock for serial interface 2.

BRGC2 is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears BRGC2 to 00H.

Figure 11-6. Format of Baud Rate Generator Control Register 2

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
BRGC2	TPS23	TPS22	TPS21	TPS20	0	0	0	0	FF73H	00H	R/W

TPS23	TPS22	TPS21	TPS20	3-bit counter source clock selection	n
0	0	0	0	$f_X/2$ (2.5 MHz)	1
0	0	0	1	$f_X/2^2$ (1.25 MHz)	2
0	0	1	0	$f_X/2^3$ (625 kHz)	3
0	0	1	1	$f_X/2^4$ (313 kHz)	4
0	1	0	0	$f_X/2^5$ (156 kHz)	5
0	1	0	1	$f_X/2^6$ (78.1 kHz)	6
0	1	1	0	$f_X/2^7$ (39.1 kHz)	7
0	1	1	1	$f_X/2^8$ (19.5 kHz)	8
1	0	0	0	External clock pulse input at the ASCK2 pin[Note]	–
Other than above				Setting prohibited	

Note An external clock can only be used in UART mode.

Cautions 1. When writing to BRGC2 is performed during a communication operation, the output of baud rate generator is disrupted and communications cannot be performed normally. Be sure not to write to BRGC2 during communication operations.

2. Do not select n = 1 during operation at f_X > 3 MHz in the UART mode because n = 1 exceeds the baud rate limit.

3. When the external input clock is selected, set port mode register 3 (PM3) to input mode.

Remarks 1. f_X: System clock oscillation frequency

2. n: Value specified by TPS20 to TPS23 ($1 \leq n \leq 8$)

3. Figures in parentheses are for operation with f_X = 5.0 MHz.

The baud rate transmit/receive clock to be generated is either a divided system clock signal, or a division of the clock input from the ASCK2 pin.

(a) Generation of baud rate transmit/receive clock by means of system clock

The transmit/receive clock is generated by dividing the system clock. The baud rate generated from the system clock is estimated by using the following expression.

$$[\text{Baud rate}] = \frac{f_X}{2^{n+1} \times 8} \quad [\text{Hz}]$$

f_X: System clock oscillation frequency

n: Values in Figure 11-6 specified by the setting in TPS20 to TPS23 ($2 \leq n \leq 8$)

Table 11-3. Example of Relationship Between System Clock and Baud Rate

Baud Rate (bps)	n	BRGC2 Set Value	Error (%)	
			f_X = 5.0 MHz	f_X = 4.9152 MHz
1,200	8	70H	1.73	0
2,400	7	60H		
4,800	6	50H		
9,600	5	40H		
19,200	4	30H		
38,400	3	20H		
76,800	2	10H		

Caution Do not select n = 1 during operation at f_X > 3 MHz in the UART mode because n = 1 exceeds the baud rate limit.

(b) Generation of baud rate transmit/receive clock by means of external clock from ASCK2 pin

The transmit/receive clock is generated by dividing the clock input from the ASCK2 pin. The baud rate generated from the clock input from the ASCK2 pin is estimated by using the following expression.

$$[\text{Baud rate}] = \frac{f_{ASCK}}{16} \text{ [Hz]}$$

f_{ASCK}: Frequency of clock input to the ASCK2 pin

Table 11-4. Relationship Between ASCK2 Pin Input Frequency and Baud Rate (When BRGC2 Is Set to 80H)

Baud Rate (bps)	ASCK2 Pin Input Frequency (kHz)
75	1.2
150	2.4
300	4.8
600	9.6
1,200	19.2
2,400	38.4
4,800	76.8
9,600	153.6
19,200	307.2
31,250	500.0
38,400	614.4

★ **(c) Generation of serial clock from system clock in 3-wire serial I/O mode**

The serial clock is generated by dividing the system clock. The serial clock frequency is estimated by using the following expression. BRGC2 does not need to be set when an external serial clock is input to the $\overline{SCK2}$ pin.

$$\text{Serial clock frequency} = \frac{f_X}{2^{n+1}} \text{ [Hz]}$$

f_X: System clock oscillation frequency

n: Value (shown in Figure 11-6) determined by setting TPS20 to TPS23 ($1 \leq n \leq 8$)

11.4 Operation of Serial Interface 2

Serial interface 2 provides the following three modes.

- Operation stopped mode
- Asynchronous serial interface (UART) mode
- 3-wire serial I/O mode

11.4.1 Operation stopped mode

In the operation stopped mode, serial transfer is not executed, allowing a reduction in the power consumption. The P30/$\overline{SCK2}$/ASCK2, P31/SO2/TxD2, and P32/SI2/RxD2 pins can be used as normal I/O ports.

(1) Register setting

Operation stopped mode is set by serial operating mode register 2 (CSIM2) and asynchronous serial interface mode register 2 (ASIM2).

(a) Serial operating mode register 2 (CSIM2)

CSIM2 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{RESET}$ input clears CSIM2 to 00H.

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
CSIM2	CSIE2	0	0	0	0	DIR2	CSCK2	0	FF72H	00H	R/W

CSIE2	3-wire serial I/O mode operation control
0	Operation disabled
1	Operation enabled

Caution Bits 0 and 3 to 6 must be fixed to 0.

(b) Asynchronous serial interface mode register 2 (ASIM2)

ASIM2 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{RESET}$ input clears ASIM2 to 00H.

Symbol	<7>	<6>	5	4	3	2	1	0	Address	After reset	R/W
ASIM2	TXE2	RXE2	PS21	PS20	CL2	SL2	0	0	FF70H	00H	R/W

TXE2	Transmit operation control
0	Transmit operation stopped
1	Transmit operation enabled

RXE2	Receive operation control
0	Receive operation stopped
1	Receive operation enabled

Caution Bits 0 and 1 must be fixed to 0.

11.4.2 Asynchronous serial interface (UART) mode

In this mode, the one-byte data following the start bit is transmitted/received and thus full-duplex communication is possible.

This device incorporates a UART-dedicated baud rate generator that enables communication at a wide range of baud rates. In addition, the baud rate can also be defined by dividing the clock input to the ASCK2 pin.

The UART-dedicated baud rate generator can also output the 31.25 kbps baud rate that complies with the MIDI standard.

(1) Register setting

The UART mode is set by serial operating mode register 2 (CSIM2), asynchronous serial interface mode register 2 (ASIM2), asynchronous serial interface status register 2 (ASIS2), baud rate generator control register 2 (BRGC2), port mode register 3 (PM3), and port 3 (P3).

(a) Serial operating mode register 2 (CSIM2)

CSIM2 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM2 to 00H.

Set CSIM2 to 00H when UART mode is selected.

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
CSIM2	CSIE2	0	0	0	0	DIR2	CSCK2	0	FF72H	00H	R/W

CSIE2	3-wire serial I/O mode operation control
0	Operation disabled
1	Operation enabled

DIR2	First-bit specification
0	MSB
1	LSB

CSCK2	3-wire serial I/O mode clock selection
0	External clock pulse input to the $\overline{\text{SCK2}}$ pin.
1	Output of the dedicated baud rate generator

Cautions 1. Bits 0 and 3 to 6 must be fixed to 0.

2. Switch operating modes after halting the serial transmit/receive operation.

(b) Asynchronous serial interface mode register 2 (ASIM2)

ASIM2 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ASIM2 to 00H.

Symbol	<7>	<6>	5	4	3	2	1	0	Address	After reset	R/W
ASIM2	TXE2	RXE2	PS21	PS20	CL2	SL2	0	0	FF70H	00H	R/W

TXE2	Transmit operation control
0	Transmit operation stopped
1	Transmit operation enabled

RXE2	Receive operation control
0	Receive operation stopped
1	Receive operation enabled

PS21	PS20	Parity bit specification
0	0	No parity
0	1	Always add 0 parity at transmission. Parity check is not performed at reception (no parity error occurs).
1	0	Odd parity
1	1	Even parity

CL2	Character length specification
0	7 bits
1	8 bits

SL2	Transmit data stop bit length specification
0	1 bit
1	2 bits

Cautions 1. Bits 0 and 1 must be fixed to 0.

2. Switch operating modes after halting the serial transmit/receive operation.

(c) Asynchronous serial interface status register 2 (ASIS2)

ASIS2 is read with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ASIS2 to 00H.

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
ASIS2	0	0	0	0	0	PE2	FE2	OVE2	FF71H	00H	R

PE2	Parity error flag
0	No parity error has occurred.
1	A parity error has occurred (parity mismatch in transmission data).

FE2	Framing error flag
0	No framing error has occurred.
1	A framing error has occurred (no stop bit detected)Note 1.

OVE2	Overrun error flag
0	No overrun error has occurred.
1	An overrun error has occurredNote 2. (Before data was read from the receive buffer register, the subsequent reception sequence was completed.)

Notes 1. Even when the stop bit length is set to 2 bits by setting bit 2 (SL2) of asynchronous serial interface mode register 2 (ASIM2), the stop bit detection in the case of reception is performed with 1 bit.

2. Be sure to read receive buffer register 2 (RXB2) when an overrun error occurs. If not, every time the data is received an overrun error occurs.

(d) Baud rate generator control register 2 (BRGC2)

BRGC2 is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears BRGC2 to 00H.

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
BRGC2	TPS23	TPS22	TPS21	TPS20	0	0	0	0	FF73H	00H	R/W

TPS23	TPS22	TPS21	TPS20	3-bit counter source clock selection	n
0	0	0	0	$f_X/2$ (2.5 MHz)	1
0	0	0	1	$f_X/2^2$ (1.25 MHz)	2
0	0	1	0	$f_X/2^3$ (625 kHz)	3
0	0	1	1	$f_X/2^4$ (313 kHz)	4
0	1	0	0	$f_X/2^5$ (156 kHz)	5
0	1	0	1	$f_X/2^6$ (78.1 kHz)	6
0	1	1	0	$f_X/2^7$ (39.1 kHz)	7
0	1	1	1	$f_X/2^8$ (19.5 kHz)	8
1	0	0	0	Input clock from external to ASCK2 pin.	–
Other than above				Setting prohibited	

Cautions 1. When writing to BRGC2 is performed during a communication operation, the output of baud rate generator is disrupted and communications cannot be performed normally. Be sure not to write to BRGC2 during a communication operation.

2. Do not select n = 1 because the operation is performed at $f_X \geq 3$ MHz and n = 1 exceeds the baud rate limit.

3. When the external input clock is selected, set port mode register 3 (PM3) to input mode.

Remarks 1. f_X: System clock oscillation frequency

2. n: Value specified by TPS20 to TPS23 ($1 \leq n \leq 8$)

3. Figures in parentheses are for operation with f_X = 5.0 MHz.

The baud rate transmit/receive clock to be generated is either a divided system clock signal, or a division of the clock input from the ASCK2 pin.

(i) Generation of baud rate transmit/receive clock by means of system clock

The transmit/receive clock is generated by dividing the system clock. The baud rate generated from the system clock is estimated by using the following expression.

$$[\text{Baud rate}] = \frac{f_X}{2^{n+1} \times 8} \text{ [Hz]}$$

f_X: System clock oscillation frequency

n: Values in the above table specified by the setting in TPS20 to TPS23 ($2 \leq n \leq 8$)

Table 11-5. Example of Relationship Between System Clock and Baud Rate

Baud Rate (bps)	n	BRGC2 Set Value	Error (%)	
			f_X = 5.0 MHz	f_X = 4.9152 MHz
1,200	8	70H	1.73	0
2,400	7	60H		
4,800	6	50H		
9,600	5	40H		
19,200	4	30H		
38,400	3	20H		
76,800	2	10H		

Caution Do not select n = 1 because the operation is performed at $f_X \geq 3$ MHz and n = 1 exceeds the baud rate limit.

(ii) Generation of baud rate transmit/receive clock by means of external clock from ASCK2 pin

The transmit/receive clock is generated by dividing the clock input from the ASCK2 pin. The baud rate generated from the clock input from the ASCK2 pin is estimated by using the following expression.

$$[\text{Baud rate}] = \frac{f_{ASCK}}{16} \text{ [Hz]}$$

f_{ASCK}: Frequency of clock input to the ASCK2 pin

Table 11-6. Relationship Between ASCK2 Pin Input Frequency and Baud Rate (When BRGC2 Is Set to 80H)

Baud Rate (bps)	ASCK2 Pin Input Frequency (kHz)
75	1.2
150	2.4
300	4.8
600	9.6
1,200	19.2
2,400	38.4
4,800	76.8
9,600	153.6
19,200	307.2
31,250	500.0
38,400	614.4

(e) Port mode register 3 (PM3)

This register sets port 3 input/output in 1-bit units.

When using the P31/TxD2/SO2 pin for serial interface data output, set PM31 to 0 and the output latch of P31 to 1.

When using the P32/RxD2/SI2 pin for serial interface data input, set PM32 to 1. At this time, the output latch of P32 may be either 0 or 1.

PM3 is set by a 1-bit or 8-bit memory manipulation instruction.

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

Address: FF23H After reset: FFH R/W

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

PM3n	P3n pin I/O mode selection (n = 0 to 3)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

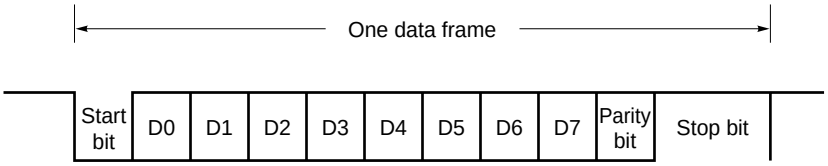
(2) Communication operation

(a) Data format

The transmit/receive data format is as shown in Figure 11-7. One data frame consists of a start bit, character bits, parity bit and stop bit(s).

The specification of character bit length, the parity selection, and the specification of stop bit length for each data frame is carried out using asynchronous serial interface mode register 2 (ASIM2).

Figure 11-7. Format of Asynchronous Serial Interface Transmit/Receive Data



- Start bit ..... 1 bit
- Character bits..... 7 bits/8 bits
- Parity bits ..... Even parity/odd parity/0 parity/no parity
- Stop bits ..... 1 bit/2 bits

When 7 bits are selected as the number of character bits, only the lower 7 bits (bits 0 to 6) are valid; in transmission the most significant bit (bit 7) is ignored, and in reception the most significant bit (bit 7) is always “0”.

The serial transfer rate is selected by means of baud rate generator control register 2 (BRGC2).

If a serial data receive error occurs, the receive error contents can be determined by reading the status of asynchronous serial interface status register 2 (ASIS2).

(b) Parity types and operation

The parity bit is used to detect a bit error in the communication data. Normally, the same kind of parity bit is used on the transmitting side and the receiving side. With even parity and odd parity, a “1” bit (odd number) error can be detected. With 0 parity and no parity, an error cannot be detected.

(i) Even parity

- **Transmission**

  The transmission operation is controlled so that the number of bits with a value of “1” in the transmit data including parity bit may be even. The parity bit value should be as follows.

  The number of bits with a value of “1” is an odd number in transmit data: 1
  The number of bits with a value of “1” is an even number in transmit data: 0

- **Reception**

  The number of bits with a value of “1” in the receive data including parity bit is counted, and if the number is odd, a parity error occurs.

(ii) Odd parity

- **Transmission**

  Conversely to even parity, the transmission operation is controlled so that the number of bits with a value of “1” in the transmit data including parity bit may be odd. The parity bit value should be as follows.

  The number of bits with a value of “1” is an odd number in transmit data: 0
  The number of bits with a value of “1” is an even number in transmit data: 1

- **Reception**

  The number of bits with a value of “1” in the receive data including parity bit is counted, and if the number is even, a parity error occurs.

(iii) 0 parity

When transmitting, the parity bit is set to “0” irrespective of the transmit data.

At reception, a parity bit check is not performed. Therefore, a parity error does not occur, irrespective of whether the parity bit is set to “0” or “1”.

(iv) No parity

A parity bit is not added to the transmit data.

At reception, data is received assuming that there is no parity bit. Since there is no parity bit, a parity error does not occur.

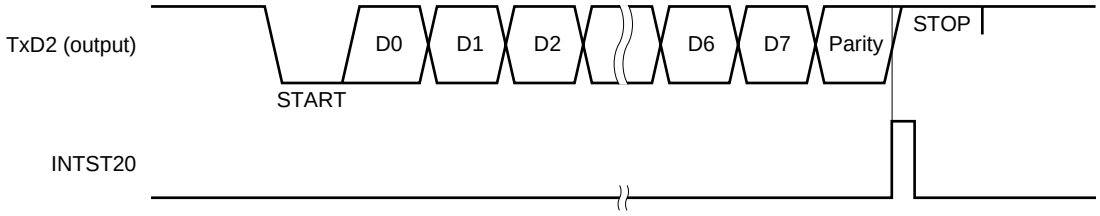
(c) Transmission

A transmit operation is started by writing transmit data to transmit shift register 2 (TXS2). The start bit, parity bit and stop bit(s) are added automatically.

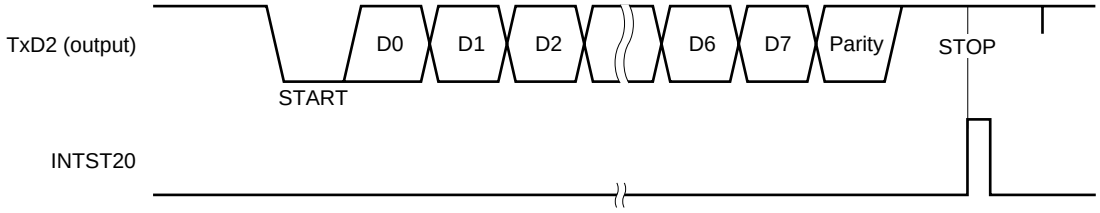
When the transmit operation starts, the data in TXS2 is shifted out, and when TXS2 is empty, a transmission completion interrupt (INTST20) is generated.

Figure 11-8. Asynchronous Serial Interface Transmission Completion Interrupt Timing

(a) Stop bit length: 1



(b) Stop bit length: 2



Caution Do not rewrite asynchronous serial interface mode register 2 (ASIM2) during a transmit operation. If ASIM2 register is rewritten during transmission, subsequent transmission may not be performed (the normal state is restored by $\overline{\text{RESET}}$ input).

It is possible to determine whether transmission is in progress by software by using a transmission completion interrupt (INTST20) or the interrupt request flag (STIF20) set by INTST20.

(d) Reception

When bit 6 (RXE2) of asynchronous serial interface mode register 2 (ASIM2) is set (1), a receive operation is enabled and sampling of the RxD2 pin input is performed.

RxD2 pin input sampling is performed using the serial clock specified by BRGC2.

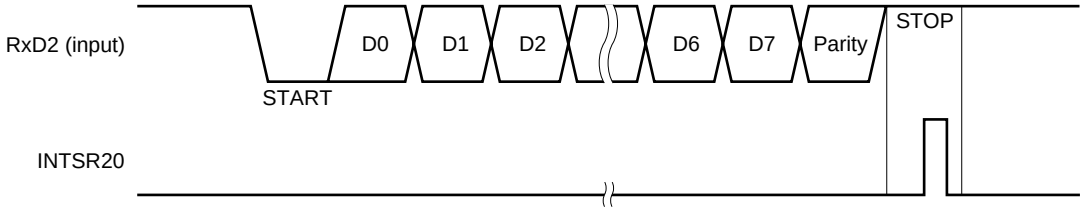
When the RxD2 pin input becomes low, the 3-bit counter starts counting, and when half the time determined by the specified baud rate has passed, the data sampling start timing signal is output. If the RxD2 pin input sampled again as a result of this start timing signal is low, it is identified as a start bit, the 3-bit counter is initialized and starts counting, and data sampling is performed. When character data, a parity bit and one stop bit are detected after the start bit, reception of one frame of data ends.

When one frame of data has been received, the receive data in the shift register is transferred to receive buffer register 2 (RXB2), and a reception completion interrupt (INTSR20) is generated.

If an error occurs, the receive data in which the error occurred is still transferred to RXB2, and INTSR20 is generated.

If the RXE2 bit is reset (0) during the receive operation, the receive operation is stopped immediately. In this case, the contents of RXB2 and asynchronous serial interface status register 2 (ASIS2) are not changed, and INTSR20 is not generated.

Figure 11-9. Asynchronous Serial Interface Reception Completion Interrupt Timing



Caution Be sure to read receive buffer register 2 (RXB2) even if a receive error occurs. If RXB2 is not read, an overrun error will occur when the next data is received, and the receive error state will continue indefinitely.

(e) Receive errors

The following three errors may occur during a receive operation: a parity error, framing error, or overrun error. The data reception result error flag is set in asynchronous serial interface status register 2 (ASIS2). Receive error causes are shown in Table 11-7.

It is possible to determine what kind of error occurred during reception by reading the contents of ASIS2 in the reception error interrupt servicing (see **Figures 11-9** and **11-10**).

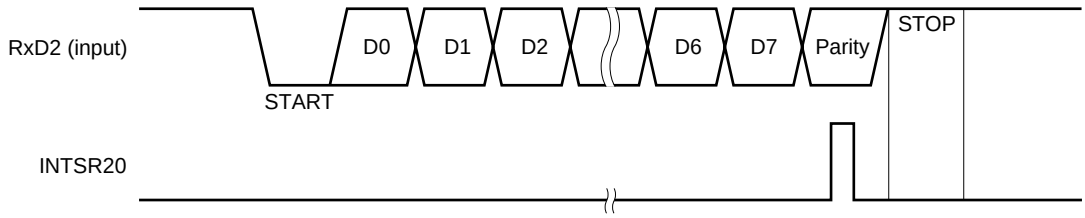
The contents of ASIS2 are reset (0) by reading receive buffer register 2 (RXB2) or receiving the next data (if there is an error in the next data, the corresponding error flag is set).

Table 11-7. Receive Error Causes

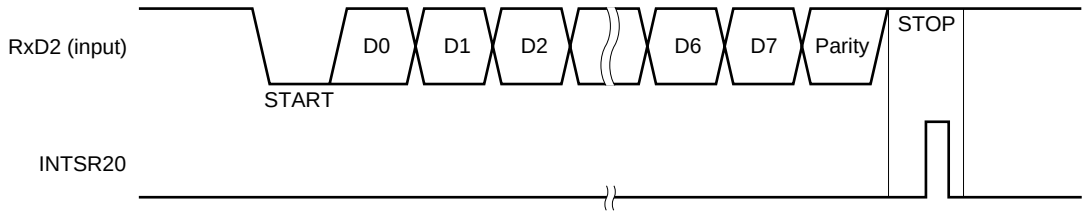
Receive Errors	Cause
Parity error	Transmission-time parity specification and reception data parity do not match
Framing error	Stop bit not detected
Overrun error	Reception of next data is completed before data is read from receive buffer register

Figure 11-10. Receive Error Timing

(a) Parity error occurs



(b) Framing error or overrun error occurs



Cautions 1. The contents of the ASIS2 register are reset (0) by reading receive buffer register 2 (RXB2) or receiving the next data. To ascertain the error contents, read ASIS2 before reading RXB2.

2. Be sure to read receive buffer register 2 (RXB2) even if a receive error occurs. If RXB2 is not read, an overrun error will occur when the next data is received, and the receive error state will continue indefinitely.

(f) Reading receive data

If a reception completion interrupt (INTSR20) is generated, read the receive data by reading the values of receive buffer register 2 (RXB2).

Read the receive data stored in receive buffer register 2 (RXB2) while reception is enabled (RXE2 = 1).

Remark If it is necessary to read the receive data while reception is disabled (RXE2 = 0), read the data using either method below.

(a) Wait one cycle or more of the source clock selected by BRGC2 and set RXE2 to 0, then read the data.

(b) Set bit 2 (DIR2) of serial operating mode register 2 (CSIM2) to 1, and then read the data.

Programming example for (a) (BRGC2 = 00H (source clock = $f_X/2$))

```
INTRXE:                 ;<Reception completion interrupt routine>
    NOP                 ;2 clocks
    CLR1    RXE2        ;Reception operation stopped
    MOV     A, RXB2     ;Read receive data
```

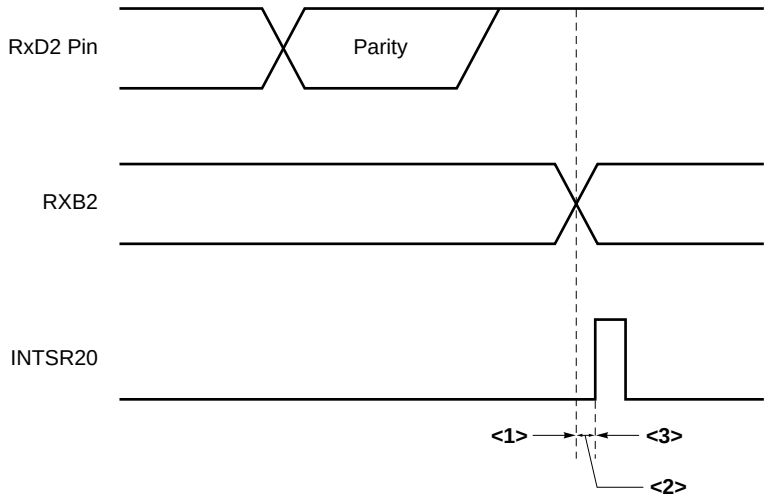
Programming example for (b)

```
INTRXE:                 ;<Reception completion interrupt routine>
    SET1    CSIM2.2     ;Set DIR2 flag to LSB first
    CLR1    RXE2        ;Reception operation stopped
    MOV     A, RXB2     ;Read receive data
```

(3) UART mode cautions

(a) When bit 7 (TXE2) of asynchronous serial interface mode register 2 (ASIM2) is cleared during transmission, be sure to set transmit shift register 2 (TXS2) to FFH, then set TXE2 to 1 before executing the next transmission.

(b) When bit 6 (RXE2) of asynchronous serial interface mode register 2 (ASIM2) is cleared during reception, receive buffer register 2 (RXB2) and receive completion interrupt 20 (INTSR20) are as follows.



When RXE2 is set to 0 at the time indicated by **<1>**, RXB2 holds the previous data and does not generate INTSR20.
When RXE2 is set to 0 at the time indicated by **<2>**, RXB2 renews the data and does not generate INTSR20.
When RXE2 is set to 0 at the time indicated by **<3>**, RXB2 renews the data and generates INTSR20.

11.4.3 3-wire serial I/O mode

The 3-wire serial I/O mode can be used to connect peripheral IC and display controllers, etc. that incorporate a synchronous clocked serial interface.

Communication is performed using three lines: the serial clock ($\overline{SCK2}$), serial output (SO2), and serial input (SI2).

(1) Register setting

3-wire serial I/O mode settings are performed using serial operating mode register 2 (CSIM2), asynchronous serial interface mode register 2 (ASIM2), baud rate generator control register 2 (BRGC2), port mode register 3 (PM3), and port 3 (P3).

(a) Serial operating mode register 2 (CSIM2)

CSIM2 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{RESET}$ input clears CSIM2 to 00H.

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
CSIM2	CSIE2	0	0	0	0	DIR2	CSCK2	0	FF72H	00H	R/W

CSIE2	3-wire serial I/O mode operation control
0	Operation disabled
1	Operation enabled

DIR2	First-bit specification
0	MSB
1	LSB

CSCK2	3-wire serial I/O mode clock selection
0	External clock pulse input to the $\overline{SCK2}$ pin.
1	Output of the dedicated baud rate generator

Cautions 1. Bits 0 and 3 to 6 must be fixed to 0.

2. Switch operating modes after halting the serial transmit/receive operation.

(b) Asynchronous serial interface mode register 2 (ASIM2)

ASIM2 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ASIM2 to 00H.

When the 3-wire serial I/O mode is selected, ASIM2 must be set to 00H.

Symbol	<7>	<6>	5	4	3	2	1	0	Address	After reset	R/W
ASIM2	TXE2	RXE2	PS21	PS20	CL2	SL2	0	0	FF70H	00H	R/W

TXE2	Transmit operation control
0	Transmit operation stopped
1	Transmit operation enabled

RXE2	Receive operation control
0	Receive operation stopped
1	Receive operation enabled

PS21	PS20	Parity bit specification
0	0	No parity
0	1	Always add 0 parity at transmission. Parity check is not performed at reception (no parity error occurs).
1	0	Odd parity
1	1	Even parity

CL2	Character length specification
0	7 bits
1	8 bits

SL2	Transmit data stop bit length specification
0	1 bit
1	2 bits

Cautions 1. Bits 0 and 1 must be fixed to 0.

2. Switch operating modes after halting the serial transmit/receive operation.

(c) Baud rate generator control register 2 (BRGC2)

BRGC2 is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears BRGC2 to 00H.

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
BRGC2	TPS23	TPS22	TPS21	TPS20	0	0	0	0	FF73H	00H	R/W

TPS23	TPS22	TPS21	TPS20	3-bit counter source clock selection	n
0	0	0	0	$f_X/2$ (2.5 MHz)	1
0	0	0	1	$f_X/2^2$ (1.25 MHz)	2
0	0	1	0	$f_X/2^3$ (625 kHz)	3
0	0	1	1	$f_X/2^4$ (313 kHz)	4
0	1	0	0	$f_X/2^5$ (156 kHz)	5
0	1	0	1	$f_X/2^6$ (78.1 kHz)	6
0	1	1	0	$f_X/2^7$ (39.1 kHz)	7
0	1	1	1	$f_X/2^8$ (19.5 kHz)	8
Other than above				Setting prohibited	

Cautions 1. When writing to BRGC2 is performed during a communication operation, the baud rate generator output is disrupted and communication cannot be performed normally. Be sure not to write to BRGC2 during communication operation.

2. When the external input clock is selected, set port mode register 3 (PM3) to input mode.

Remarks 1. f_X: System clock oscillation frequency

2. n: Values specified by TPS20 to TPS23 ($1 \leq n \leq 8$)

3. Figures in parentheses are for operation with f_X = 5.0 MHz.

If the internal clock is used as the serial clock for the 3-wire serial I/O mode, set the TPS20 to TPS23 bits to set the frequency of the serial clock. To obtain the frequency to be set, use the following formula. When the serial clock is input from off-chip, setting BRGC2 is not necessary.

$$\text{Serial clock frequency} = \frac{f_X}{2^{n+1}} \text{ [Hz]}$$

f_X: System clock oscillation frequency

n: Values in the above table specified by the setting in TPS20 to TPS23 ($1 \leq n \leq 8$)

(d) Port mode register 3 (PM3)

This register sets port 3 input/output in 1-bit units.

When using the P30/$\overline{\text{SCK2}}$/ASCK2 pin for clock output and the P31/TxD2/SO2 pin for serial interface data output, set PM30 and PM31 to 0 and the output latch of P30 and P31 to 1.

When using the P30/$\overline{\text{SCK2}}$/ASCK2 pin for clock input and the P32/RxD2/SI2 pin for serial interface data input, set PM30 and PM32 to 1. At this time, the output latch of P30 and P32 may be either 0 or 1.

PM3 is set by a 1-bit or 8-bit memory manipulation instruction.

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

Address: FF23H After reset: FFH R/W

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

PM3n	P3n pin I/O mode selection (n = 0 to 3)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

(2) Communication operation

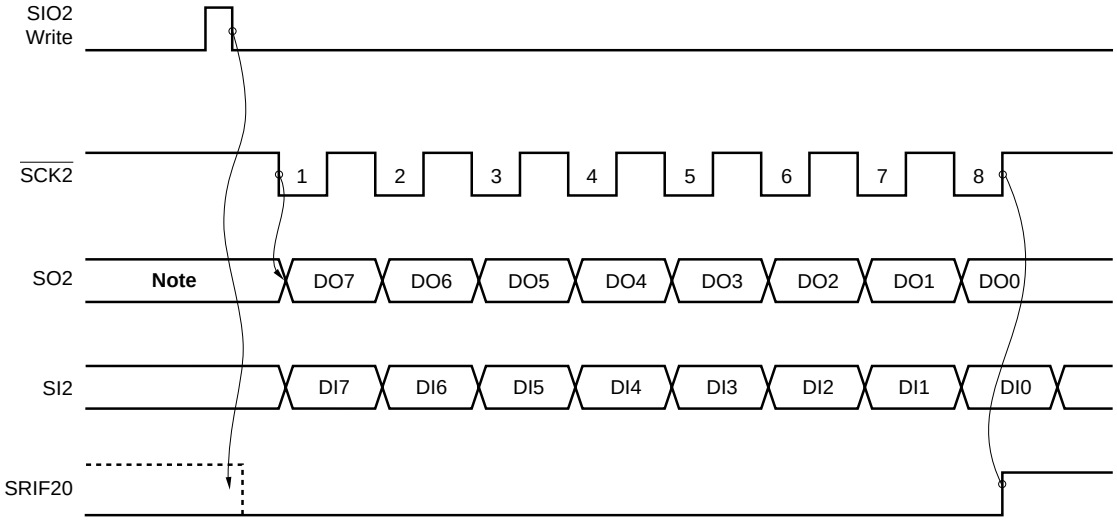
In the 3-wire serial I/O mode, data transmission/reception is performed in 8-bit units. Data is transmitted/received bit by bit in synchronization with the serial clock.

The transmit shift register (TXS2/SIO2) and receive shift register (RXS2) shift operations are performed in synchronization with the fall of the serial clock ($\overline{SCK2}$). Then transmit data is held in the SO2 latch and output from the SO2 pin. Also, receive data input to the SI pin is latched in the receive buffer register (RXB2/SIO2) on the rise of $\overline{SCK2}$.

At the end of an 8-bit transfer, the operation of TXS2/SIO2 or RXS2 stops automatically, and an interrupt request signal (INTCSI20) is generated.

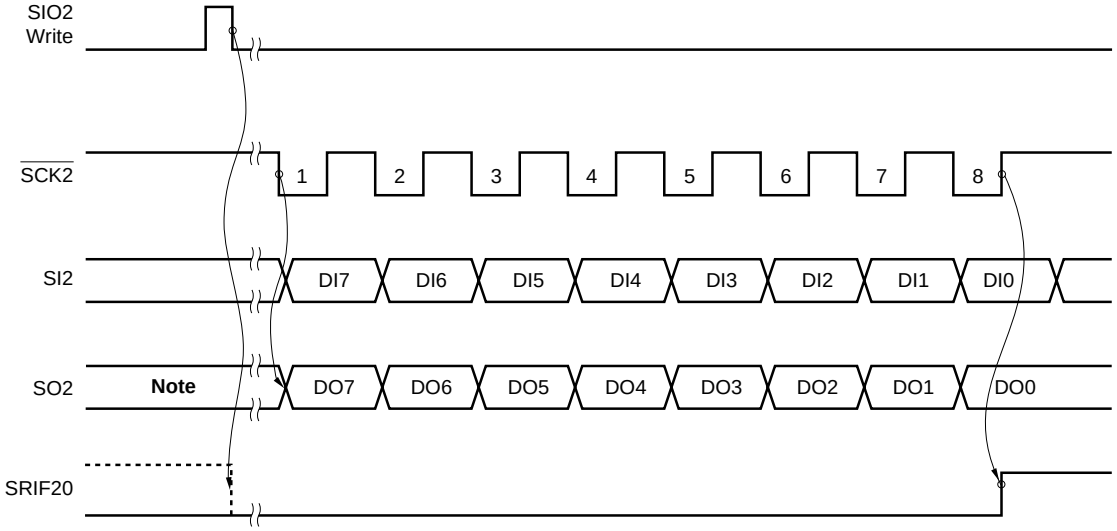
Figure 11-11. 3-Wire Serial I/O Mode Timing

(i) Master operation timing



Note The value of the last bit output previously is output.

(ii) Slave operation timing



Note The value of the last bit output previously is output.

(3) Transfer start

Serial transfer is started by setting transfer data to the transmit shift register (TXS2/SIO2) when the following two conditions are satisfied.

- Serial operating mode register 2 (CSIM2) bit 7 (CSIE2) = 1
- Internal serial clock is stopped or $\overline{SCK2}$ is a high level after 8-bit serial transfer.

Caution If CSIE2 is set to 1 after data is written to TXS2/SIO2, transfer does not start.

Termination of 8-bit transfer stops the serial transfer automatically and generates an interrupt request signal (INTCSI20).

CHAPTER 12 POWER-ON-CLEAR CIRCUITS

12.1 Power-on-Clear Circuit Functions

The power-on-clear circuits include the following two circuits, which have the following functions.

(1) Power-on-clear (POC) circuit

- Compares the detection voltage (V_{POC}) with the power supply voltage (V_{DD}) and generates an internal reset signal if $V_{DD} < V_{POC}$.
- The mask ROM versions can select a POC switching circuit, normally operating POC circuit, or normally halted POC circuit by using a mask option. When a POC switching circuit is selected, POC operation can be controlled by software (refer to **CHAPTER 19 MASK OPTIONS**).
- This circuit can operate even in STOP mode.

(2) Low-voltage detection (LVI) circuit

- Compares the detection voltage (V_{LVI}) with the power supply voltage (V_{DD}) and generates an interrupt request signal (INTLVI1) if $V_{DD} < V_{LVI}$.
- Eight levels of detection voltage can be selected using software.
- This circuit stops operation in STOP mode.

12.2 Configuration of Power-on-Clear Circuit

Figures 12-1 and 12-2 show the block diagrams of the power-on-clear circuits.

Figure 12-1. Block Diagram of Power-on-Clear Circuit

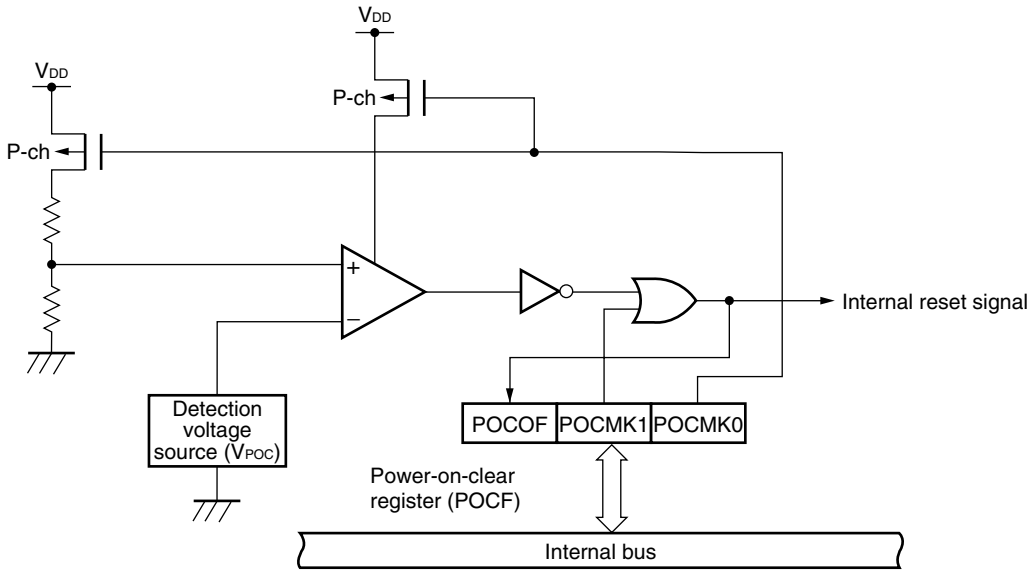
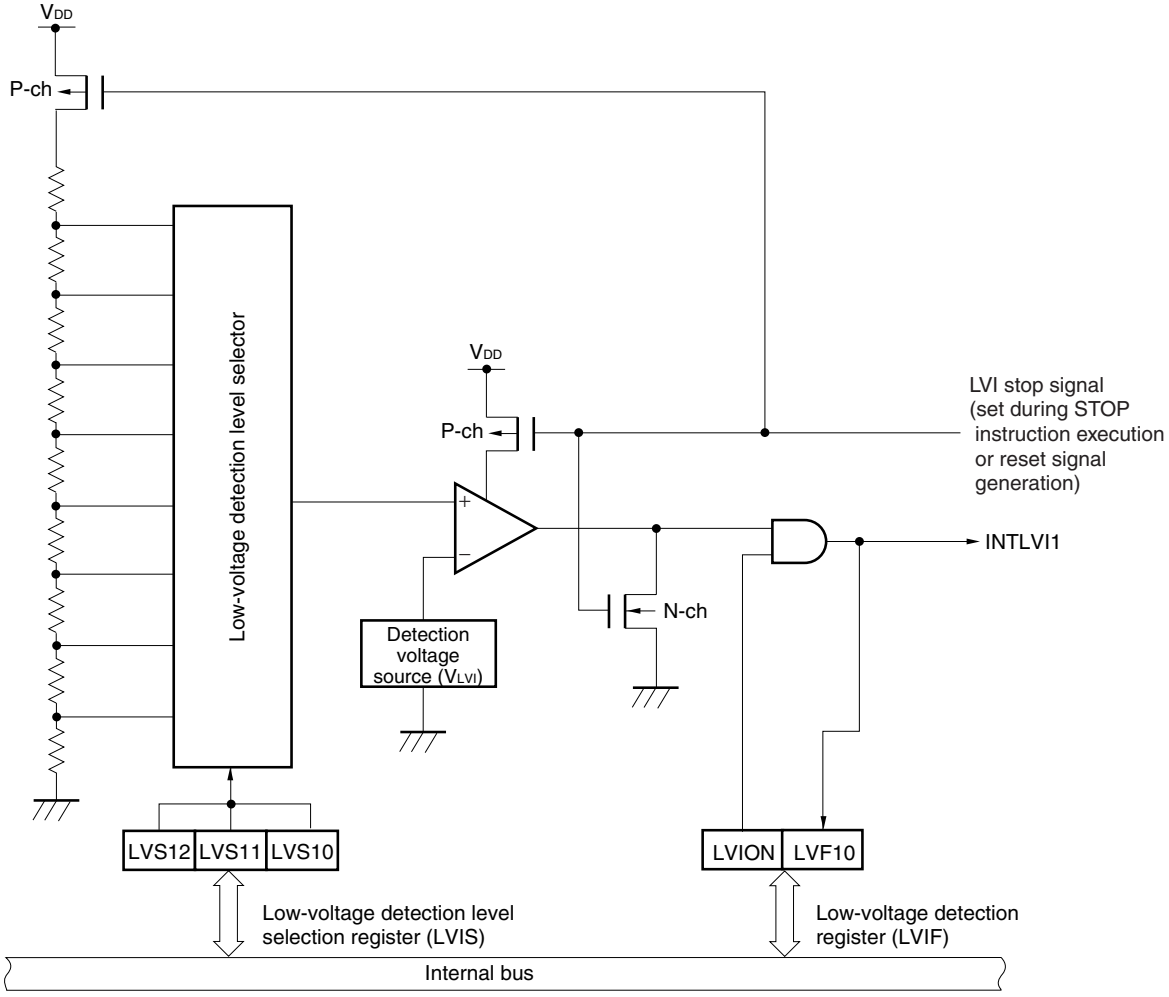


Figure 12-2. Block Diagram of Low-Voltage Detector



12.3 Power-on-Clear Circuit Control Registers

The following three registers control the power-on-clear circuits.

- Power-on-clear register (POCF)
- Low-voltage detection register (LVIF)
- Low-voltage detection level selection register (LVIS)

(1) Power-on-clear register (POCF)

This register controls POC circuit operation.

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

Figure 12-3. Format of Power-on-Clear Register

Symbol	7	6	5	4	3	<2>	<1>	<0>	Address	After reset	R/W
POCF	0	0	0	0	0	POCOF	POCMK1	POCMK0	FF80H	00H[Note]	R/W

POCOF	POC output detection flag
0	Non-generation of reset signal by POC or in cleared state due to a write operation to POCF
1	Generation of reset signal by POC

POCMK1	POC reset control
0	Generation of reset signal by POC enabled
1	Generation of reset signal by POC disabled

POCMK0	POC operation control
0	POC operating
1	POC halted

Note This value is 04H only after a power-on-clear reset.

Caution The POCMK0 and POCMK1 in the mask ROM version are only valid when a POC switching circuit has been selected using a mask option.

(2) Low-voltage detection register (LVIF)

This register controls the operation of the LVI circuit.

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

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

Figure 12-4. Format of Low-Voltage Detection Register

Symbol	<7>	6	5	4	3	2	1	<0>	Address	After reset	R/W
LVIF	LVION	0	0	0	0	0	0	LVF10	FF81H	00H	R/W[Note]

LVION	LVI operation enable flag
0	LVI disabled
1	LVI enabled

LVF10	LVI output detection flag
0	Power supply voltage (V_{DD}) > LVI detection voltage (V_{LVI}) or operation disabled
1	$V_{DD} < V_{LVI}$

Note Bit 0 is read only.

(3) Low-voltage detection level selection register (LVIS)

This register selects the level of the detection voltage (V_{LVI}).

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

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

Figure 12-5. Format of Low-Voltage Detection Level Selection Register

Symbol	7	6	5	4	3	<2>	<1>	<0>	Address	After reset	R/W
LVIS	0	0	0	0	0	LVS12	LVS11	LVS10	FF82H	00H	R/W

LVS12	LVS11	LVS10	Selection of detection voltage (V_{LVI}) level[Note]
0	0	0	V_{LVI0}
0	0	1	V_{LVI1}
0	1	0	V_{LVI2}
0	1	1	V_{LVI3}
1	0	0	V_{LVI4}
1	0	1	V_{LVI5}
1	1	0	V_{LVI6}
1	1	1	V_{LVI7}

Note Refer to each chapter of electrical specifications for detection voltage specifications.

Caution When changing the detection voltage level (V_{LVI}), an operation stabilization time of about 2 ms is required in order for the LVI output to stabilize. Do not, therefore, set the LVI circuit to operation-enabled until the operation has stabilized.

12.4 Power-on-Clear Circuit Operation

12.4.1 Power-on-clear (POC) circuit operation

The POC circuit compares the detection voltage (V_{POC}) with the power supply voltage (V_{DD}) and generates an internal reset signal if $V_{DD} < V_{POC}$.

For mask ROM versions, it is possible to select a POC switching circuit, normally operating POC circuit, or normally halted POC circuit by using a mask option. When a POC switching circuit is selected, POC operation can be controlled by software. In the μPD78E9862, only the POC switching circuit is available (circuits cannot be selected by a mask option).

Observe the following procedure when switching POC operation using the POC switching circuit.

(1) Switching from POC stopped to POC operating

<1> Check that POCMK1 = 1

<2> Set POCMK0 to 0 to put the POC circuit into the operating state

<3> Wait until the operation stabilization time has elapsed (because the output signal is unstable, generation of the reset signal via the POC circuit is set to disabled)

<4> Set POCMK1 to 0 to enable generation of the reset signal via the POC circuit

(2) Switching from POC operating to POC stopped

<1> Set POCMK1 to 1 to disable generation of the reset signal via the POC circuit

<2> Set POCMK0 to 1 to put the POC circuit into the operation stopped state

Generation of the reset signal via the POC circuit can be determined by reading the POCOF flag. When the reset ★ signal is generated via the POC circuit, POCOF is set to 1. POCOF is cleared to 0 by writing to POCF[Note].

When using the POC circuit, clear POCOF beforehand.

★ **Note** POCOF is cleared by writing to one of bits 0 to 2.

Figures 12-6 to 12-8 show the timing of reset signal generation via the POC circuit.

Figure 12-6. Timing of Internal Reset Signal Generation When POC Circuit Normally Operating

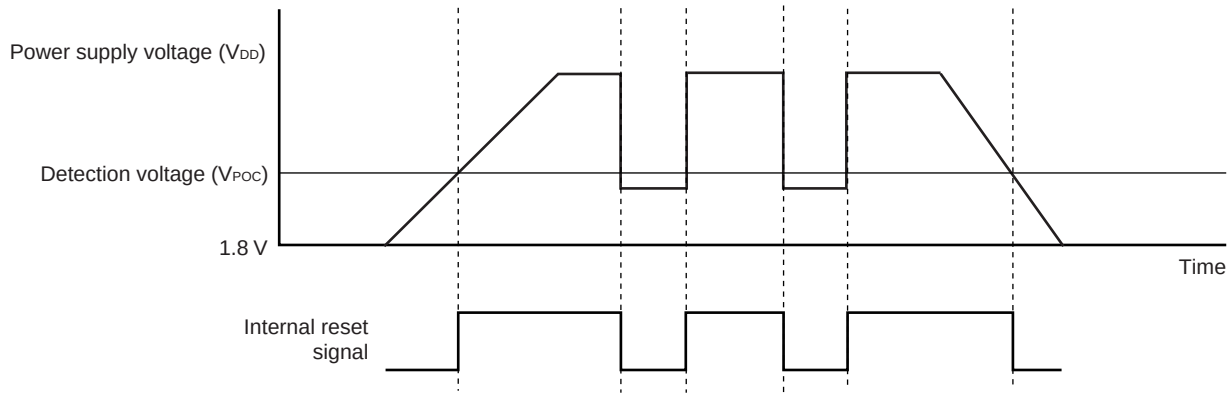


Figure 12-7. Timing of Internal Reset Signal Generation When POC Circuit Normally Halted

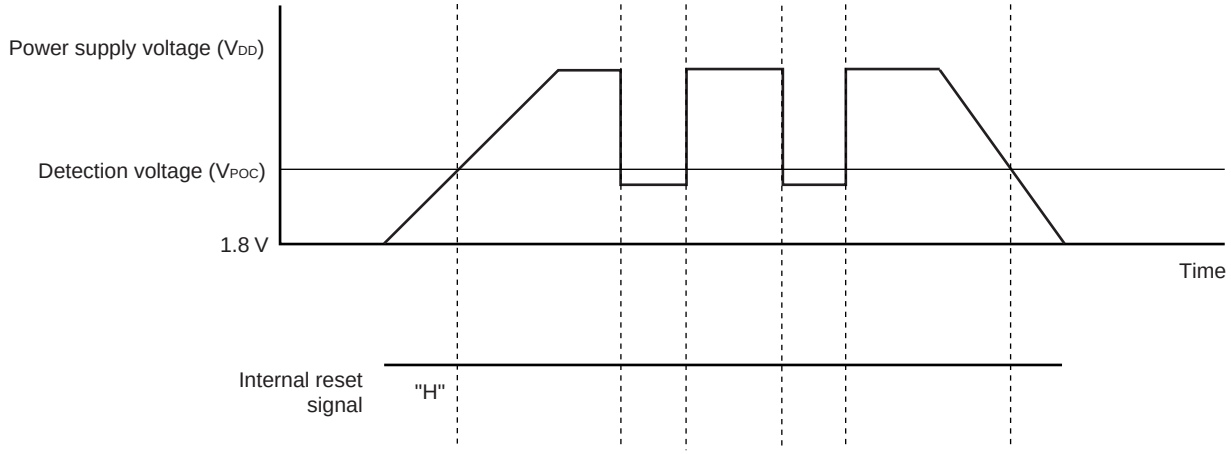
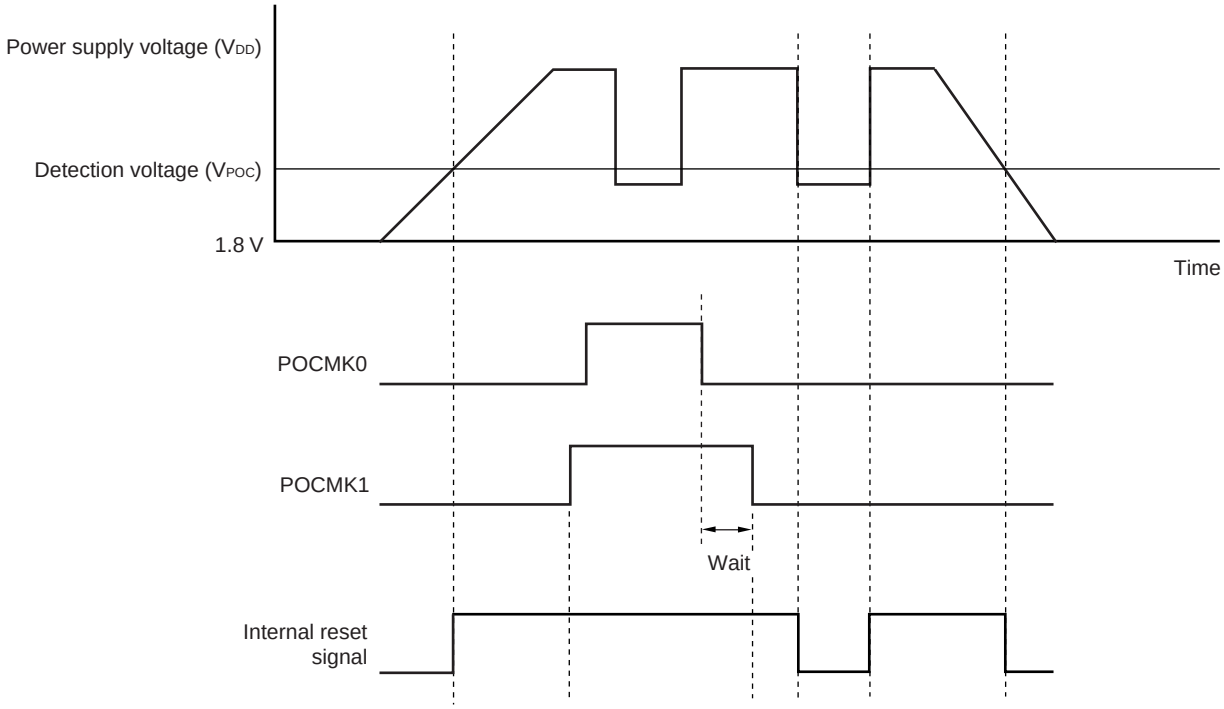


Figure 12-8. Timing of Internal Reset Signal Generation in POC Switching Circuit



12.4.2 Operation of low-voltage detector (LVI)

The LVI circuit compares the detection voltage (V_{LVI}) with the power supply voltage (V_{DD}) and generates an interrupt request signal (INTLVI1) if $V_{DD} < V_{LVI}$ (LVI circuit operating).

As shown in **Figure 12-2 Block Diagram of Low-Voltage Detector,** the divided resistors and comparators of the LVI circuit turn OFF when the reset signal is generated or in STOP mode. After reset is released, LVI operation starts when LVION (bit 7 of the low-voltage detection register (LVIF)) is set. At this time, approximately 2 ms are required until the LVI circuit operation is stabilized.

Once the LVI operation is started, divided resistors and comparators cannot be turned OFF unless the STOP instruction or reset signal is generated, even if LVION is cleared. Low-voltage detection is enabled immediately after LVION is set again.

★ **Caution The divided resistors and comparators of the LVI circuit are turned ON after reset is released.**

Use one of the following methods to constantly monitor low voltage.

<1> Low-voltage monitoring by LVF10 (bit 0 of the low-voltage detection register (LVIF)) without using the LVI detection interrupt.

<2> Low-voltage monitoring using the LVI detection interrupt. In this case, disable the LVI operation once, and then enable it (LVION = 0 → 1) before enabling interrupts (LVIMK1 = 0).

An example of a program in which low voltage is constantly monitored using the LVI detection interrupt is shown below.

(a) Processing when reset mode is released

```
DI
MOV     LVIS, #xxH      ; Setting LVI detection voltage
SET1    LVIMK1          ; LVI interrupt disabled
SET1    LVION           ; LVI operation enabled
CALL    !WAIT_2ms       ; 2 ms wait
CLR1    LVIIF1
CLR1    LVION           ; LVI operation disabled
SET1    LVION           ; LVI operation enabled
CLR1    LVIMK1          ; LVI interrupt enabled
EI
```

(b) Processing when STOP mode is released

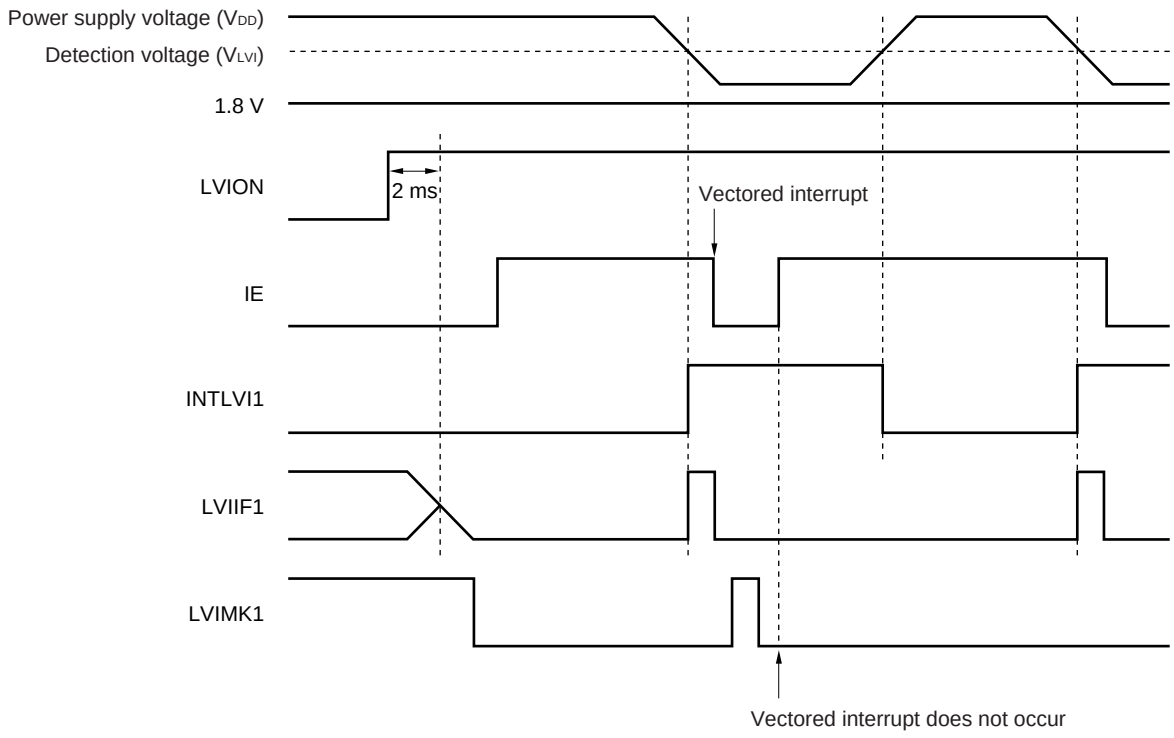
```
SET1    LVIMK1          ; LVI interrupt disabled
STOP
CALL    !WAIT           ; Total 2 ms wait, combined with oscillation stabilization time
CLR1    LVIIF1
CLR1    LVION           ; LVI operation disabled
SET1    LVION           ; LVI operation enabled
CLR1    LVIMK1          ; LVI interrupt enabled
EI
```

(c) Processing to enable LVI interrupt again after LVI interrupt servicing

```
SET1    LVIMK1      ; LVI interrupt disabled
CLR1    LVION       ; LVI operation disabled
SET1    LVION       ; LVI operation enabled
CLR1    LVIMK1      ; LVI interrupt enabled
EI
```

Figure 12-9 shows the LVI circuit operation timing.

★ **Figure 12-9. LVI Circuit Operation Timing**



Caution The low-voltage detection interrupt request flag (LVIIF1) is set at the rising edge of the LVI circuit comparator output signal (INTLVI1). Therefore, the power supply voltage (V_{DD}) becomes lower than the detection voltage (V_{LVI}) during LVI operation, and if that state continues after INTLVI1 generation, LVIIF1 is not set. After low-voltage detection, when set as $V_{DD} > V_{LVI}$ and then $V_{DD} < V_{LVI}$ again, LVIIF1 is set.

CHAPTER 13 BIT SEQUENTIAL BUFFER

13.1 Bit Sequential Buffer Functions

The μPD789862 Subseries has an on-chip bit sequential buffer of 8 bits + 8 bits = 16 bits.
The functions of the bit sequential buffer are shown below.

- If the value of the bit sequential buffer 1 data register (BSFRL10, BSFRH10) is shifted 1 bit to the lower side, the LSB can be output to the port at the same time.
- It is possible to write to BSFRL10 and BSFRH10 using an 8-bit or 16-bit memory manipulation instruction (reading is not possible).
- Overwriting is enabled during a shift operation on the higher 8 bits (BSFRH10) only (the period in which the shift clock is low level).

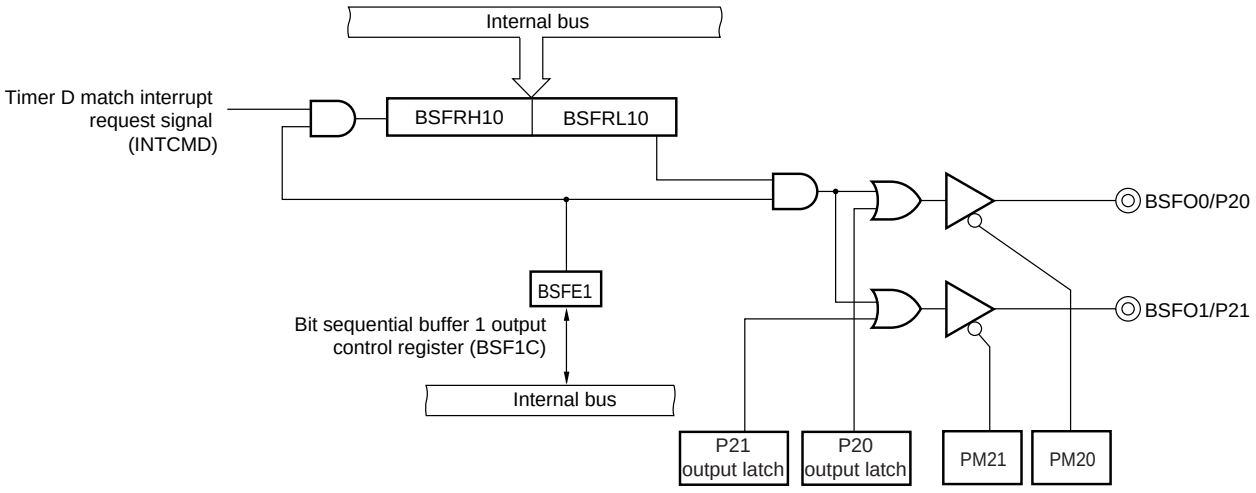
13.2 Bit Sequential Buffer Configuration

The bit sequential buffer includes the following hardware.

Table 13-1. Configuration of Bit Sequential Buffer

Item	Configuration
Data register	Bit sequential buffer: 8 bits × 8 bits = 16 bits
Control registers	Bit sequential buffer 1 output control register (BSF1C) Port mode register 2 (PM2) Port 2 (P2)

Figure 13-1. Block Diagram of Bit Sequential Buffer



13.3 Bit Sequential Buffer Control Registers

The following registers control the bit sequential buffer.

- Bit sequential buffer 1 output control register (BSF1C)
- Port mode register 2 (PM2)
- Port 2 (P2)

(1) Bit sequential buffer 1 output control register (BSF1C)

This register controls the operation of the bit sequential buffer.

BSF1C is set by a 1-bit or 8-bit memory manipulation instruction.

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

Figure 13-2. Format of Bit Sequential Buffer 1 Output Control Register

Symbol	7	6	5	4	3	2	1	<0>	Address	After reset	R/W
BSF1C	0	0	0	0	0	0	0	BSFE1	FF60H	00H	R/W

BSFE1	Bit sequential buffer operation control
0	Operation disabled
1	Operation enabled

★ **(2) Port mode register 2 (PM2)**

This register sets port 2 input/output in 1-bit units.

When using the P20/BSFO0 and P21/BSFO1 pins for data output of the bit sequential buffer, set PM20 and PM21 and the output latches of P20 and P21 to 0.

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

$\overline{\text{RESET}}$ input sets PM2 value to FFH.

Figure 13-3. Format of Port Mode Register 2 (PM2)

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

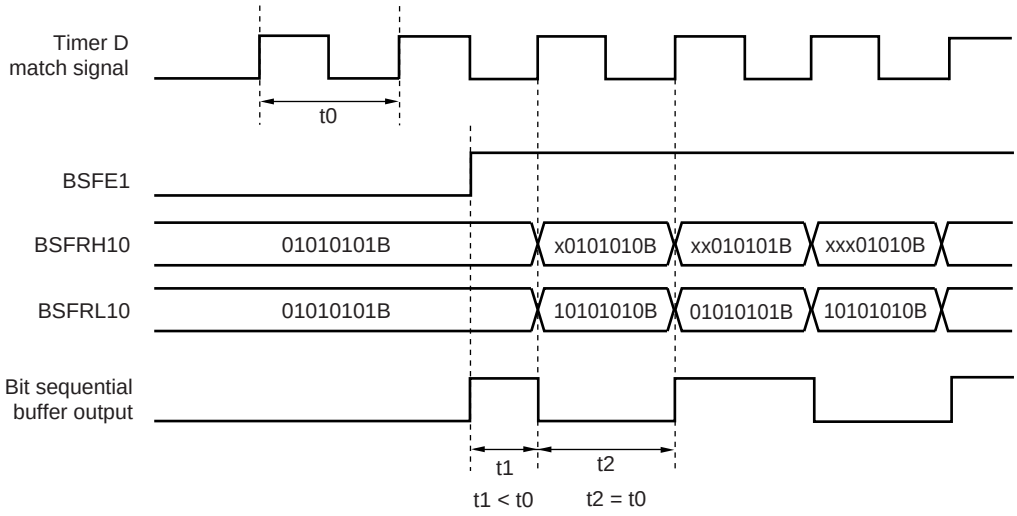
PM2n	P2n pin I/O mode selection (n = 0 to 4)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

13.4 Bit Sequential Buffer Operation

Set as follows to operate the bit sequential buffer.

<1> Set values to bit sequential buffer 1 data registers L and H (BSFRL10, BSFRH10)
<2> Set the bit sequential buffer to operation enabled (BSFE1 = 1)
If the LSB of BSFRL10 is being output at P20/BSFO0, set P20 to output mode (PM20 = 0) and the output latch of P20 to 0
If the LSB of BSFRL10 is being output at P21/BSFO1, set P21 to output mode (PM21 = 0) and the output latch of P21 to 0
<3> Start the clock operation

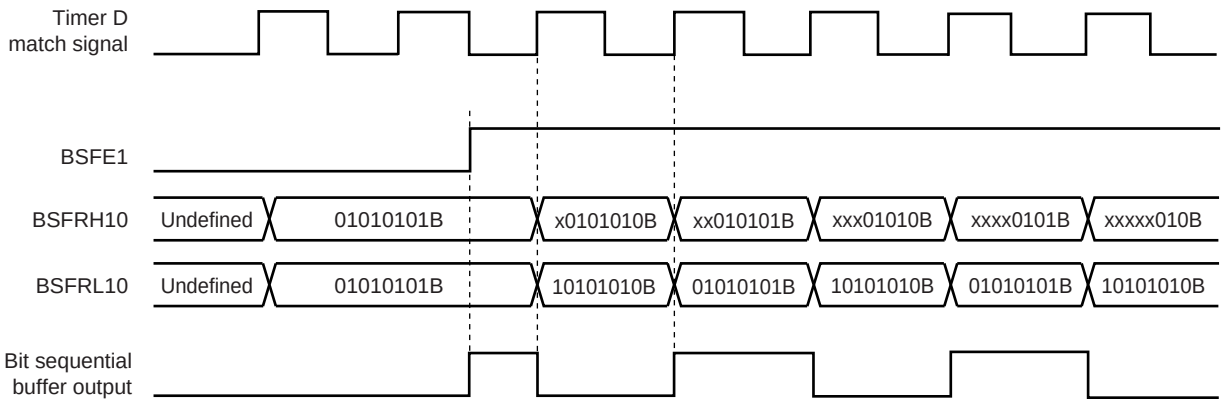
If the clock is input before the bit sequential buffer starts operation, the output time of the start bit may be shorter than one cycle of the clock when output commences, as shown in the figure below.



Remark x: Undefined

Figure 13-4 shows the operation timing of the bit sequential buffer.

Figure 13-4. Operation Timing of Bit Sequential Buffer



Cautions 1. The higher 8 bits (BSFRH10) of the data register can be overwritten while the bit sequential buffer is operating. Even if data is written to the data register, the shift clock will not stop. Write data to the data register when the shift clock (match signal of 16-bit timer D) is low level (refer to 13.5 Notes).

2. The value of the data register is undefined after a shift.

Remark x: Undefined

13.5 Notes

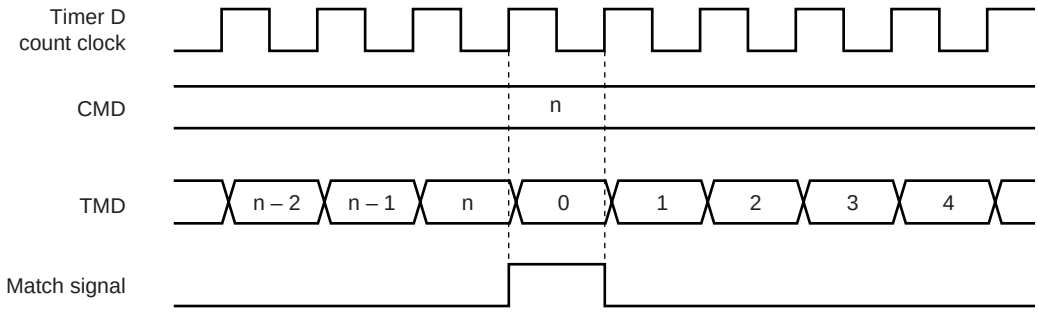
When overwriting the higher 8 bits (BSFRH10) of the data register while the bit sequential buffer is operating, data must be written when the shift clock (match signal of 16-bit timer D) is low level.

The match signal of 16-bit timer D is high level during one cycle of the count clock selected by setting bits 6 to 4 (CS2 to CS0) of 16-bit timer mode control register D (TMCD).

Table 13-2. Count Clock and Match Signal High Level Width of 16-Bit Timer D

Bits 6 to 4 of TMCD			Count Clock	Match Signal High Level Width
CS2	CS1	CS0		
0	0	0	$f_X/2$	$f_X/2$
0	0	1	$f_X/2^2$	$f_X/2^2$
0	1	0	$f_X/2^3$	$f_X/2^3$
0	1	1	$f_X/2^4$	$f_X/2^4$
1	0	0	$f_X/2^5$	$f_X/2^5$
1	0	1	$f_X/2^6$	$f_X/2^6$
1	1	0	$f_X/2^7$	$f_X/2^7$
1	1	1	$f_X/2^8$	$f_X/2^8$

Figure 13-5. Match Signal of 16-Bit Timer D



CHAPTER 14 KEY RETURN CIRCUIT

14.1 Key Return Circuit Function

Either of the following two operations can be selected.

<1> When KRREN pin = High level

This circuit generates an internal non-maskable interrupt based on KR0 to KR5 falling edge input.

- KR0: The falling edge can be independently detected regardless of the status of KR1 to KR5.
- KR1 to KR4: The falling edge is detected by ORing the input of four pins.
- KR5: The rising/falling edge can be selected by setting the EDG register.
  The edge can be independently detected regardless of the status of KR0 to KR4.

<2> When KRREN pin = Low level

In STOP mode, this circuit generates an internal reset signal based on KR0 to KR5 falling edge input.

- KR0: The falling edge can be independently detected regardless of the status of KR1 to KR5.
- KR1 to KR4: The falling edge is detected by ORing the input of four pins.
- KR5: The rising/falling edge can be selected by setting the EDG register.
  The edge can be independently detected regardless of the status of KR0 to KR4.

Caution If the KRREN pin is used at low level, an internal reset signal is generated when a key return occurs in the STOP mode. If a key return occurs in other than STOP mode, a non-maskable interrupt (NMI) is generated and the program branches to the vector address. If an NMI is not used, execute the following routine in the NMI vector routine.

```
(INTNMI: NMI vector processing of key return)
INTNMI: RETI
```

14.2 Key Return Circuit Configuration and Operation

Figures 14-1 and 14-2 show the block diagram of the key return circuit and the format of the key return edge detection register, respectively.

Figure 14-1. Block Diagram of Key Return Circuit

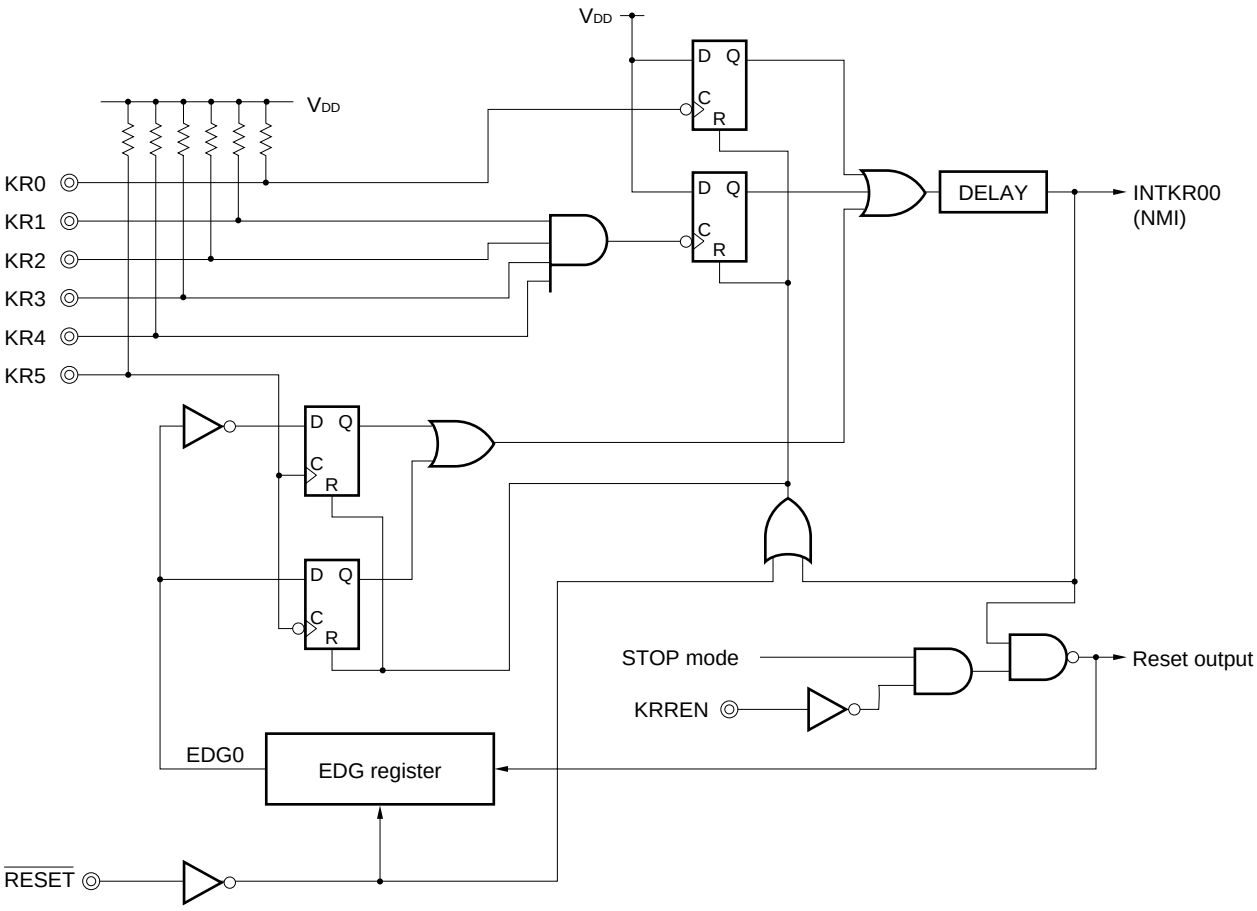


Figure 14-2. Format of Key Return Edge Detection Register

Symbol	<7>	6	5	4	3	2	1	<0>	Address	After reset	R/W
EDG	KRRES	0	0	0	0	0	0	EDG0	FF90H	**Note**	R/W

EDG0	Switching edge detection of KR5 pin
0	Falling edge detected
1	Rising edge detected

KRRES	Reset output type
0	Reset output by source other than key return circuit
1	Reset output by key return circuit

Note The value after the key return reset is 1000000×B (bit 0 retains the value before the key return reset is input), and in other cases, the value is 00H.

Caution When switching the edge detection, change the input waveform after manipulating the EDG0 bit.

CHAPTER 15 INTERRUPT FUNCTIONS

15.1 Interrupt Function Types

The following two types of interrupt functions are used.

(1) Non-maskable interrupt

This interrupt is acknowledged unconditionally. It does not undergo interrupt priority control and is given top priority over all other interrupt requests.

A standby release signal is generated.

The non-maskable interrupt has one external interrupt source and one internal interrupt source.

(2) Maskable interrupt

These interrupts undergo mask control. If two or more interrupts with the same priority are generated, each interrupt has a predetermined priority as shown in Table 15-1.

A standby release signal is generated.

The maskable interrupt has two external interrupt sources and nine internal interrupt sources.

15.2 Interrupt Sources and Configuration

There are total of 13 non-maskable and maskable interrupt sources (see **Table 15-1**).

Table 15-1. Interrupt Source List

Interrupt Type	Priority[Note 1]	Interrupt Source		Internal /External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Non-maskable	–	INTKR00	Key return input falling edge detection (KR5 can detect both the rising and falling edges)	External	0002H	(A)
		INTWDT	Watchdog timer overflow (with watchdog timer mode 1 selected)	Internal	0004H	
Maskable	0	INTWDT	Watchdog timer overflow (with interval timer mode selected)			(B)
	1	INTP0	Pin input edge detection	External	0006H	(C)
	2	INTP1			0008H	
	3	INTCMD	Match of TMD and CMD	Internal	000AH	(B)
	4	INTTM00	Match of TM0 and CR00 (when CR00 is specified as compare register) TI01 pin valid edge detection (when CR00 is specified as capture register)		000CH	
	5	INTTM01	Match of TM0 and CR01 (when CR01 is specified as compare register) TI00 pin valid edge detection (when CR01 is specified as capture register)		000EH	
	6	INTTM50	Match of TM5 and CR5		0010H	
	7	INTSR20	End of serial interface 2 UART reception		0012H	
		INTCSI20	End of serial interface 2 3-wire transfer			
	8	INTST20	End of serial interface 2 UART transmission		0014H	
	9	INTLVI1	LVI interrupt detection		0016H	
	10	INTEE0	EEPROM write completion signal		0018H	

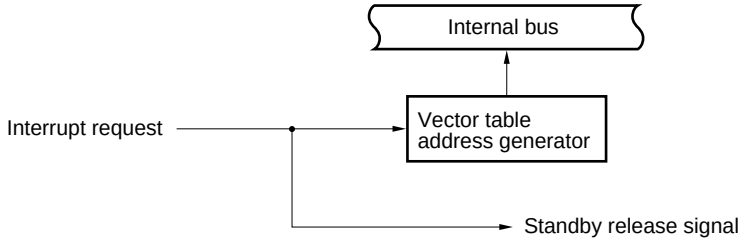
Notes 1. Priority is the priority applicable when two or more maskable interrupts are simultaneously generated. 0 is the highest priority and 10 is the lowest priority.

2. Basic configuration types A to C correspond to A to C in Figure 15-1.

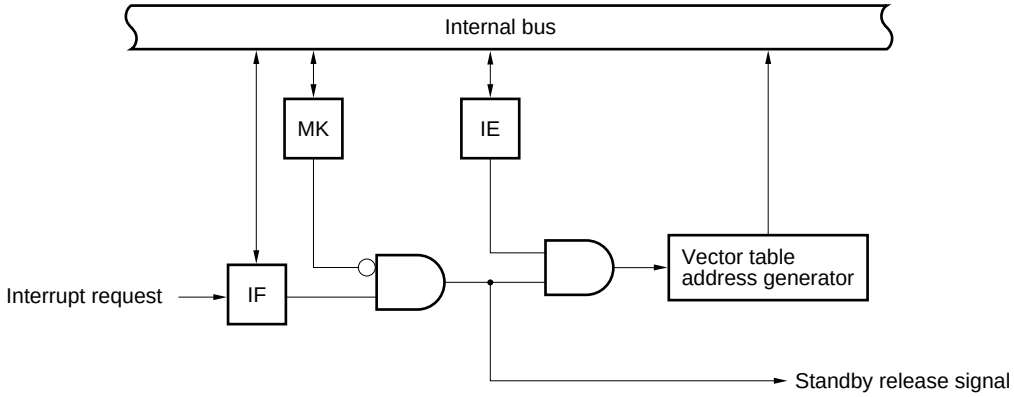
Remark Either a non-maskable interrupt or a maskable interrupt (internal) can be selected as the interrupt source of the watchdog timer (INTWDT).

Figure 15-1. Basic Configuration of Interrupt Function

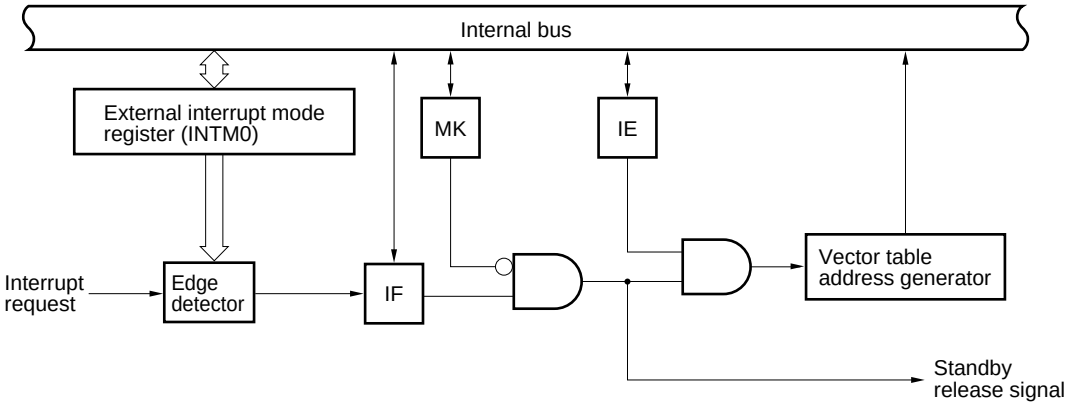
(A) External/internal non-maskable interrupt



(B) Internal maskable interrupt



(C) External maskable interrupt



IF: Interrupt request flag
IE: Interrupt enable flag
MK: Interrupt mask flag

15.3 Interrupt Function Control Registers

The following four registers are used to control the interrupt functions.

- Interrupt request flag registers (IF0, IF1)
- Interrupt mask flag registers (MK0, MK1)
- External interrupt mode register (INTM0)
- Program status word (PSW)

Table 15-2 gives a listing of interrupt request flag and interrupt mask flag names corresponding to interrupt requests.

Table 15-2. Flags Corresponding to Interrupt Request Signal

Interrupt Request Signal Name	Interrupt Request Flag	Interrupt Mask Flag
INTWDT	WDTIF	WDTMK
INTP0	PIF0	PMK0
INTP1	PIF1	PMK1
INTCMD	TMIFD	TMMKD
INTTM00	TMIF00	TMMK00
INTTM01	TMIF01	TMMK01
INTTM50	TMIF50	TMMK50
INTSR20/INTCSI20	SRIF20	SRMK20
INTST20	STIF20	STMK20
INTLVI1	LVIIF1	LVIMK1
INTEE0	EEIF0	EEMK0

(1) Interrupt request flag registers (IF0, IF1)

The interrupt request flag is set to 1 when the corresponding interrupt request is generated or an instruction is executed. It is cleared to 0 when an instruction is executed upon acknowledgement of an interrupt request or upon $\overline{\text{RESET}}$ input.

IF0 and IF1 are set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears these registers to 00H.

Figure 15-2. Format of Interrupt Request Flag Register

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
IF0	SRIF20	TMIF50	TMIF01	TMIF00	TMIFD	PIF1	PIF0	WDTIF	FFE0H	00H	R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>	Address	After reset	R/W
IF1	0	0	0	0	0	EEIF0	LVIIF1	STIF20	FFE1H	00H	R/W

××IF×	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request signal is generated; Interrupt request state

Cautions 1. WDTIF flag is R/W enabled only when the watchdog timer is used as an interval timer. If the watchdog timer mode 1 and 2 are used, set the WDTIF flag to 0.

2. Because ports 2 and 3 have an alternate function as the external interrupt input, when the output level is changed by specifying the output mode of the port function, an interrupt request flag is set. Therefore, the interrupt mask flag should be set to 1 before using the output mode.

3. When an interrupt is acknowledged, the interrupt request flag is automatically cleared and then the interrupt routine is entered.

(2) Interrupt mask flag registers (MK0, MK1)

The interrupt mask flag is used to enable/disable the corresponding maskable interrupt service.

MK0 and MK1 are set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets these registers to FFH.

Figure 15-3. Format of Interrupt Mask Flag Register

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
MK0	SRMK20	TMMK50	TMMK01	TMMK00	TMMKD	PMK1	PMK0	WDTMK	FFE4H	FFH	R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>	Address	After reset	R/W
MK1	1	1	1	1	1	EEMK0	LVIMK1	STMK20	FFE5H	FFH	R/W

××MK×	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Cautions 1. If the WDTMK flag is read when the watchdog timer is used in watchdog timer mode 1 and 2, its value becomes undefined.

2. Because ports 2 and 3 have an alternate function as the external interrupt input, when the output level is changed by specifying the output mode of the port function, an interrupt request flag is set. Therefore, the interrupt mask flag should be set to 1 before using the output mode.

(3) External interrupt mode register 0 (INTM0)

This register is used to set the valid edge of INTP0 and INTP1.

INTM0 is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears INTM0 to 00H.

Figure 15-4. Format of External Interrupt Mode Register 0

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
INTM0	0	0	ES11	ES10	ES01	ES00	0	0	FFECH	00H	R/W

ES11	ES10	INTP1 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ES01	ES00	INTP0 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

Cautions 1. Be sure to set bits 0, 1, 6, and 7 to 0.

2. Before setting the INTM0 register, be sure to set the corresponding interrupt mask flag (××MK× = 1) to disable interrupts. After setting the INTM0 register, clear the interrupt request flag (××IF× = 0), then clear the interrupt mask flag (××MK× = 0), which will enable interrupts.

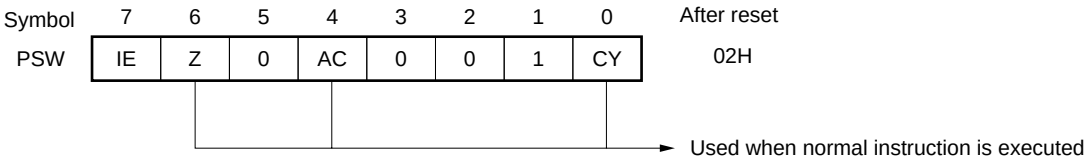
(4) Program status word (PSW)

The program status word is a register used to hold the instruction execution result and the current status for interrupt requests. The IE flag to set maskable interrupt enable/disable is mapped.

This register can be read/written in 8-bit units and can carry out operations using a bit manipulation and dedicated instructions (EI, DI). When a vectored interrupt request is acknowledged, PSW is automatically saved into a stack, and the IE flag is reset to 0. It is reset from the stack by the RETI and POP PSW instructions.

$\overline{\text{RESET}}$ input sets PSW to 02H.

Figure 15-5. Program Status Word Configuration



IE	Interrupt acknowledge enable/disable
0	Disable
1	Enable

15.4 Interrupt Servicing Operation

15.4.1 Non-maskable interrupt request acknowledgement operation

A non-maskable interrupt request is unconditionally acknowledged even when interrupts are disabled. It is not subject to interrupt priority control and takes precedence over all other interrupts.

When a non-maskable interrupt request is acknowledged, the PSW and PC are saved to the stack in that order, the IE flag is reset to 0, the contents of the vector table are loaded to the PC, and then program execution branches.

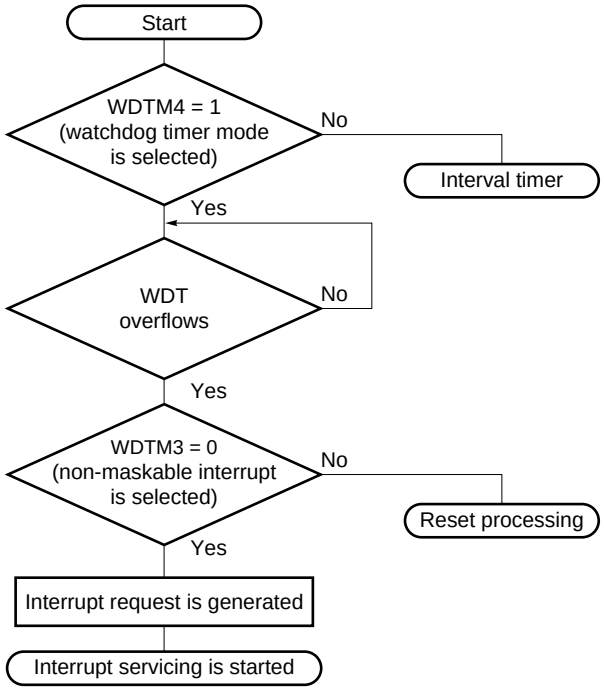
Figure 15-6 shows the flowchart from non-maskable interrupt request generation to acknowledgement. Figure 15-7 shows the timing of non-maskable interrupt request acknowledgement. Figure 15-8 shows the acknowledgement operation if multiple non-maskable interrupts are generated.

Caution The μPD789862 Subseries has two non-maskable interrupt sources. Therefore, during execution of a non-maskable interrupt servicing program, a new non-maskable interrupt request is not acknowledged until the RETI instruction is executed. Be sure to execute the RETI instruction after the interrupt servicing program has been executed.
When using the watchdog timer as a non-maskable interrupt, push the address of the restore destination before executing the RETI instruction. If the RETI instruction is executed without pushing the restore destination, the program will jump to an illegal address. A program example is shown below.

<Example> Program example in which watchdog timer is used as non-maskable interrupt and program branches to reset vector when interrupt occurs

```
XVECT           CSEG    AT  0000H
DW      IRESET          ;(00)       RESET
DW      IKR             ;(02)       KeyReturn
DW      IWDT            ;(04)       INTWDT
:
XRST            CSEG    AT  0080H
IRESET: DI
        MOVW    AX, #0FEFFH
        MOVW    SP, AX
:
:
IWDT:
            (Interrupt servicing)
    :
        MOVW    AX, #0080H
        PUSH    AX
        RETI
```

Figure 15-6. Flowchart from Non-Maskable Interrupt Request Generation to Acknowledgement (INTWDT)



WDTM: Watchdog timer mode register
WDT: Watchdog timer

Figure 15-7. Timing of Non-Maskable Interrupt Request Acknowledgement

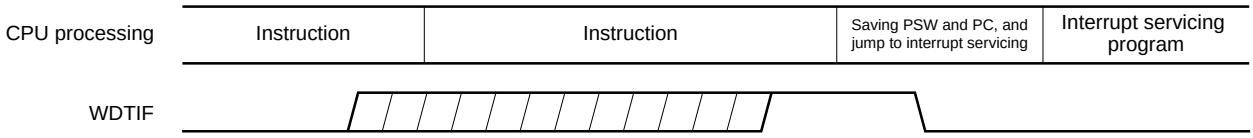
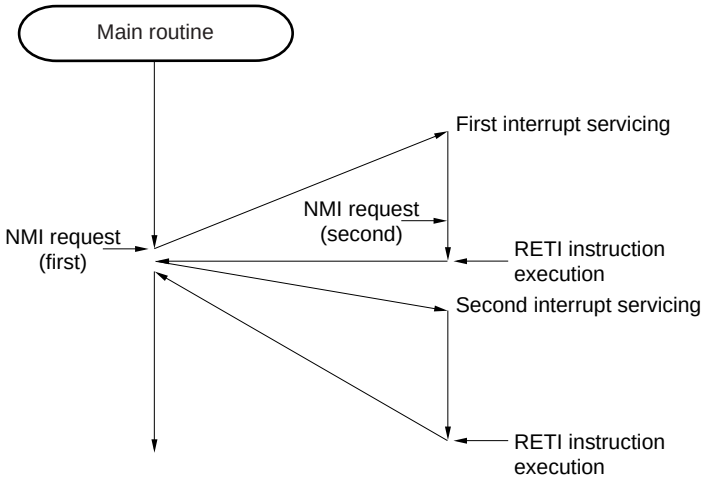


Figure 15-8. Acknowledgement of Non-Maskable Interrupt Request



15.4.2 Maskable interrupt request acknowledgement operation

A maskable interrupt request can be acknowledged when the interrupt request flag is set to 1 and the corresponding interrupt mask flag is cleared to 0. A vectored interrupt request is acknowledged in the interrupt enabled status (when the IE flag is set to 1).

The time required to start the interrupt servicing after a maskable interrupt request has been generated is shown in Table 15-3.

See **Figures 15-10** and **15-11** for the interrupt request acknowledgement timing.

Table 15-3. Time from Generation of Maskable Interrupt Request to Servicing

Minimum Time	Maximum Time[Note]
9 clocks	19 clocks

Note The wait time is maximum when an interrupt request is generated immediately before the BT and BF instructions.

Remark 1 clock: $\frac{1}{f_{CPU}}$ (f_{CPU}: CPU clock)

When two or more maskable interrupt requests are generated at the same time, they are acknowledged starting from the interrupt request assigned the highest priority.

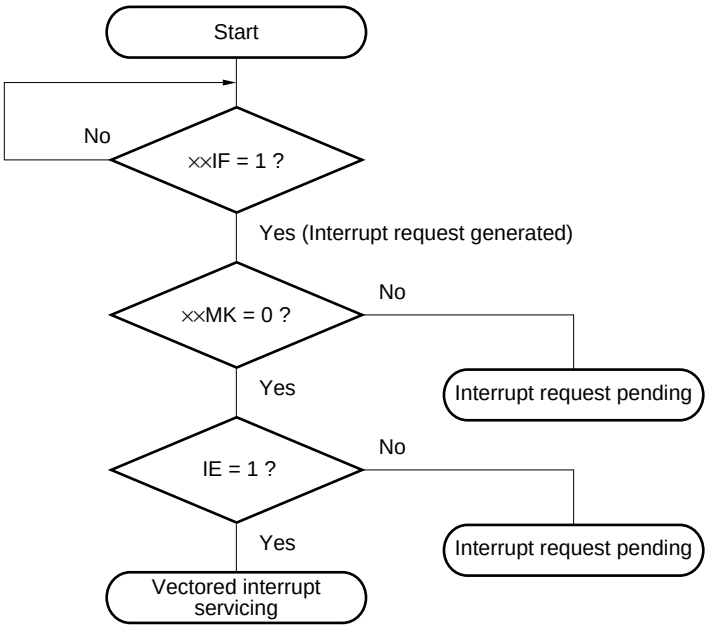
A pending interrupt is acknowledged when a status in which it can be acknowledged is set.

Figure 15-9 shows the algorithm of interrupt request acknowledgement.

When a maskable interrupt request is acknowledged, the contents of the PSW and PC are saved to the stack in that order, the IE flag is reset to 0, and the data in the vector table determined for each interrupt request is loaded to the PC, and execution branches.

To return from interrupt servicing, use the RETI instruction.

Figure 15-9. Interrupt Request Acknowledgement Processing Algorithm

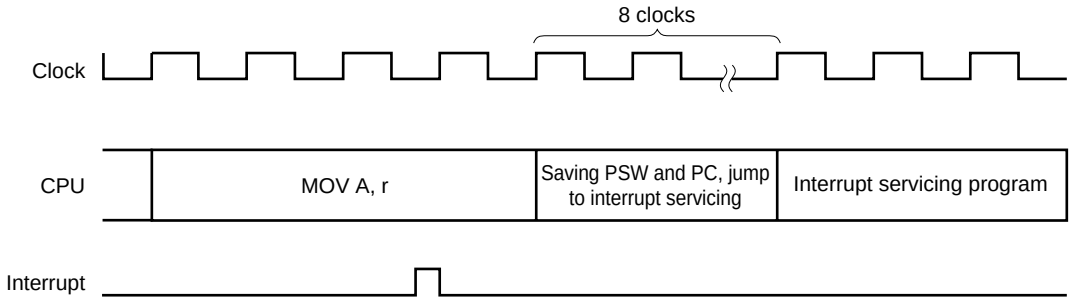


××IF: Interrupt request flag

××MK: Interrupt mask flag

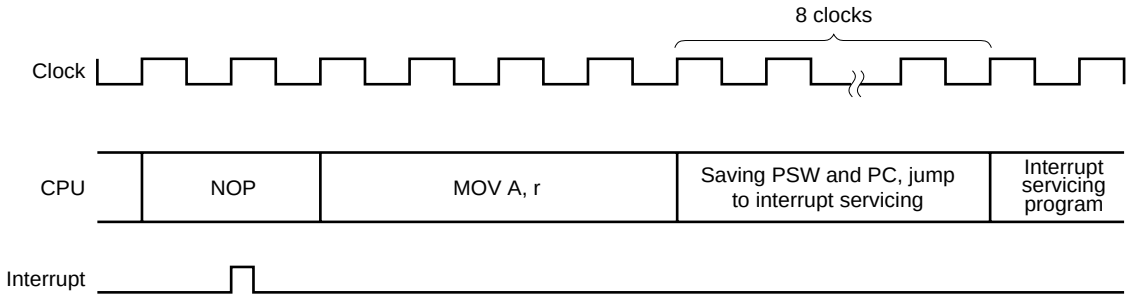
IE: Flag to control maskable interrupt request acknowledgement (1 = enable, 0 = disable)

Figure 15-10. Interrupt Request Acknowledgement Timing (Example of MOV A, r)



If an interrupt request flag (××IF) is set before instruction clock n (n = 4 to 10) under execution becomes n – 1, the interrupt is acknowledged after the instruction under execution is complete. Figure 15-10 shows an example of the interrupt request acknowledgement timing for an 8-bit data transfer instruction MOV A, r. Since this instruction is executed for 4 clocks, if an interrupt occurs for 3 clocks after the execution starts, the interrupt acknowledgement processing is performed after the MOV A, r instruction is executed.

Figure 15-11. Interrupt Request Acknowledgement Timing (When Interrupt Request Flag Is Set at Last Clock During Instruction Execution)



If an interrupt request flag (××IF) is set at the last clock of the instruction, the interrupt acknowledgement processing starts after the next instruction is executed.

Figure 15-11 shows an example of the interrupt acknowledgement timing for an interrupt request flag that is set at the second clock of NOP (2-clock instruction). In this case, the MOV A, r instruction after the NOP instruction is executed, and then the interrupt acknowledgement processing is performed.

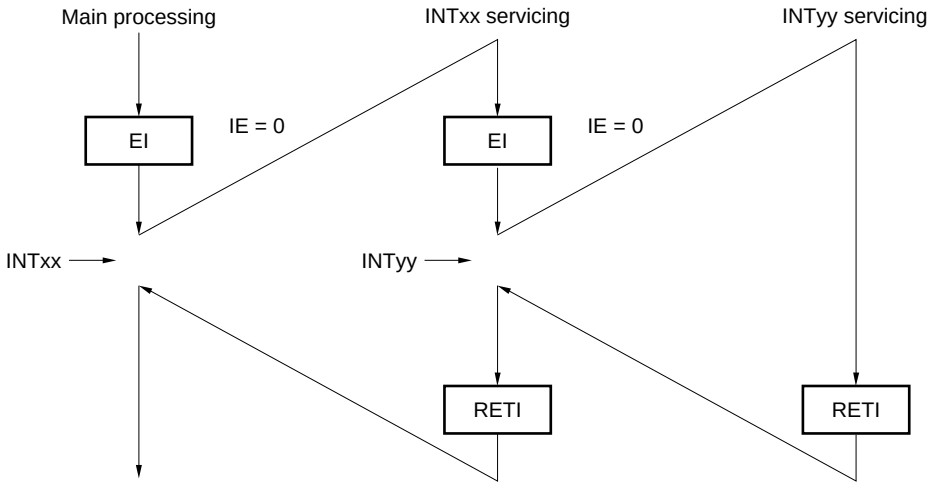
Caution Interrupt requests will be held pending while interrupt request flag registers (IF0, IF1) or interrupt mask flag registers (MK0, MK1) are being accessed.

15.4.3 Multiple interrupt servicing

Multiple interrupt servicing in which another interrupt is acknowledged while an interrupt is being serviced can be performed using a priority order system. When two or more interrupts are generated at once, interrupt servicing is performed according to the priority assigned to each interrupt request in advance (see **Table 15-1**).

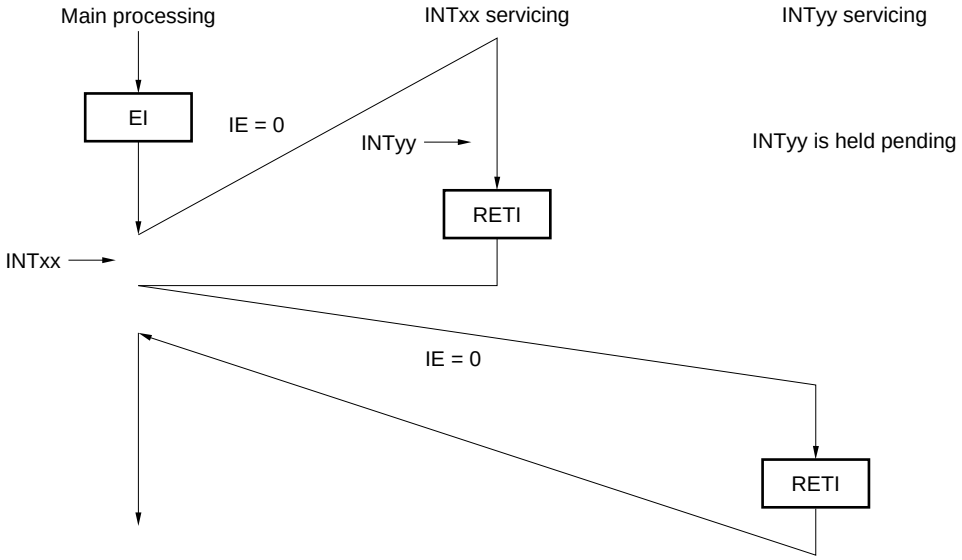
Figure 15-12. Example of Multiple Interrupts

Example 1. Multiple interrupts are acknowledged



During interrupt INTxx servicing, interrupt request INTyy is acknowledged, and multiple interrupts are generated. The EI instruction is issued before each interrupt request acknowledgement, and the interrupt request acknowledgement enable state is set.

Example 2. Multiple interrupts are not generated because interrupts are not enabled



Because interrupts are not enabled in interrupt INTxx servicing (the EI instruction is not issued), interrupt request INTyy is not acknowledged, and multiple interrupts are not generated. The INTyy request is held pending and acknowledged after the INTxx servicing is performed.

IE = 0: Interrupt request acknowledgement disabled

15.4.4 Interrupt request pending

Some instructions may hold pending the acknowledgement of an interrupt request until the completion of the execution of the next instruction even if the interrupt request (maskable interrupt, non-maskable interrupt, and external interrupt) is generated during the execution. The following shows such instructions (interrupt request pending instruction).

- Manipulation instruction for interrupt request flag registers (IF0, IF1)
- Manipulation instruction for interrupt mask flag registers (MK0, MK1)

CHAPTER 16 STANDBY FUNCTION

16.1 Standby Function and Configuration

16.1.1 Standby function

The standby function is used to reduce the power consumption of the system and can be effected in the following two modes.

(1) HALT mode

This mode is set when the HALT instruction is executed. HALT mode stops the operation clock of the CPU. The system clock oscillator continues oscillating. This mode does not reduce the current consumption as much as STOP mode, but is useful for resuming processing immediately when an interrupt request is generated, or for intermittent operations.

(2) STOP mode

This mode is set when the STOP instruction is executed. STOP mode stops the main system clock oscillator and stops the entire system. The current consumption of the CPU can be substantially reduced in this mode.

The data memory can be retained at a low voltage (V_{DD} = 1.8 V max.). Therefore, this mode is useful for retaining the contents of the data memory at an extremely low current consumption.

STOP mode can be released by an interrupt request, so that this mode can be used for intermittent operations. However, some time is required until the system clock oscillator stabilizes after STOP mode has been released. If processing must be resumed immediately by using an interrupt request, therefore, use the HALT mode.

In both modes, the previous contents of the registers, flags, and data memory before setting standby mode are all retained. In addition, the statuses of the output latches of the I/O ports and output buffers are also retained.

Caution To set STOP mode, be sure to stop the operations of the peripheral hardware, and then execute the STOP instruction.

16.1.2 Standby function control register

The wait time after STOP mode is released upon interrupt request until the oscillation stabilizes is controlled with the oscillation stabilization time selection register (OSTS).

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

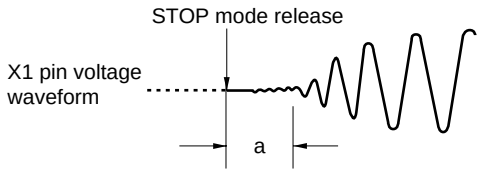
$\overline{\text{RESET}}$ input sets OSTS to 04H. However, the oscillation stabilization time after $\overline{\text{RESET}}$ release is $2^{12}/f_X$, independent of OSTS.

Figure 16-1. Format of Oscillation Stabilization Time Selection Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0	FFFAH	04H	R/W

OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection
0	0	0	$2^{12}/f_X$ (819 μs)
0	1	0	$2^{15}/f_X$ (6.55 ms)
1	0	0	$2^{17}/f_X$ (26.2 ms)
Other than above			Setting prohibited

Caution The wait time after STOP mode is released does not include the time from STOP mode release to clock oscillation start ("a" in the figure below), regardless of release by $\overline{\text{RESET}}$ input or by interrupt generation.



Remarks 1. f_X: System clock oscillation frequency (ceramic/crystal oscillation)

2. Figures in parentheses are for operation with f_X = 5.0 MHz.

16.2 Standby Function Operation

16.2.1 HALT mode

(1) HALT mode

HALT mode is set by executing the HALT instruction.

The operation statuses in HALT mode are shown in the following table.

Table 16-1. Operation Statuses in HALT Mode

Item		HALT Mode Operation Status
System clock		System clock oscillation enabled Clock supply to CPU stopped
CPU		Operation stopped
EEPROM		Operation enabled[Note 1]
Port (output latch)		Remains in the state existing before HALT mode has been set
16-bit timer/event counter 0		Operation enabled
16-bit timer D		Operation enabled
8-bit timer/event counter 5		Operation enabled
8-bit timer 80		Operation enabled
Watchdog timer		Operation enabled
Serial interface 2		Operation enabled
Power-on-clear circuit	POC	Operation enabled[Note 2]
	LVI	Operation enabled
Bit sequential buffer		Operation enabled
Key return circuit		Operation enabled
External interrupt		Operation enabled[Note 3]

★ **Notes 1.** HALT mode can be set after executing a write instruction. During writing, wait is possible in the HALT mode. To release standby by an interrupt request signal (INTEE0) on detection of write completion, set EEMK0 to 0.

2. If a POC switching circuit is selected by a mask option and the POC circuit is set to operation enabled by software or if POC circuit normally operating is selected by a mask option (refer to **CHAPTER 19 MASK OPTIONS** regarding mask options).

3. A maskable interrupt that is not masked

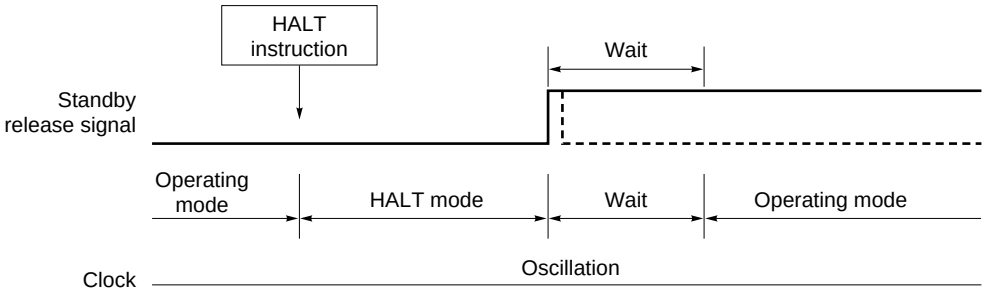
(2) Releasing HALT mode

HALT mode can be released by the following three sources.

(a) Releasing by unmasked interrupt request

HALT mode is released by an unmasked interrupt request. In this case, if interrupt request acknowledgement is enabled, vectored interrupt servicing is performed. If interrupt acknowledgement is disabled, the instruction at the next address is executed.

Figure 16-2. Releasing HALT Mode by Interrupt



Remarks 1. The broken lines indicate the case where the interrupt request that has released standby mode is acknowledged.

2. The wait time is as follows:

- When vectored interrupt servicing is performed: 9 to 10 clocks
- When vectored interrupt servicing is not performed: 1 to 2 clocks

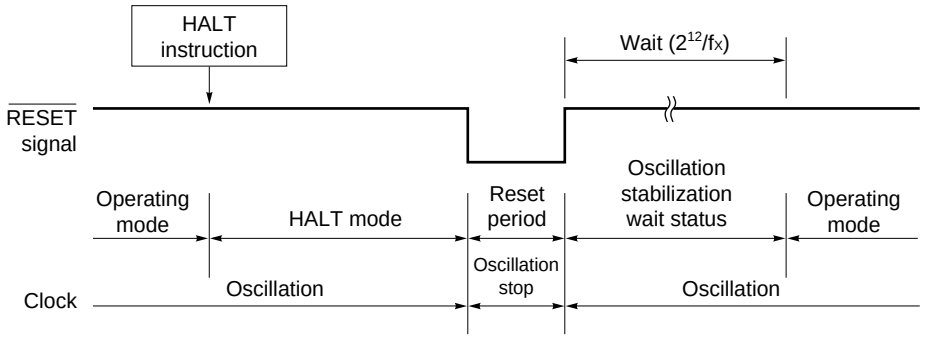
(b) Releasing by non-maskable interrupt request

HALT mode is released regardless of whether interrupts are enabled or disabled, and vectored interrupt servicing is performed.

(c) Releasing by $\overline{\text{RESET}}$ input

When HALT mode is released by the $\overline{\text{RESET}}$ signal, execution branches to the reset vector address in the same manner as the ordinary reset operation, and program execution starts.

Figure 16-3. Releasing HALT Mode by $\overline{\text{RESET}}$ Input



Remark fx: System clock oscillation frequency

Table 16-2. Operation After Releasing HALT Mode

Releasing Source	MK××	IE	Operation
Maskable interrupt request	0	0	Executes next address instruction.
	0	1	Executes interrupt servicing.
	1	×	Retains HALT mode.
Non-maskable interrupt request	–	×	Executes interrupt servicing.
$\overline{\text{RESET}}$ input	–	–	Reset processing

×: Don't care

16.2.2 STOP mode

(1) Setting and operation status of STOP mode

STOP mode is set by executing the STOP instruction.

Caution Because standby mode can be released by an interrupt request signal, standby mode is released as soon as it is set if there is an interrupt source whose interrupt request flag is set and interrupt mask flag is reset. Thus, the STOP mode is reset to the HALT mode immediately after execution of the STOP instruction and the operation mode is set after the wait for the oscillation stabilization time elapses.

The operation statuses in STOP mode are shown in the following table.

Table 16-3. Operation Statuses in STOP Mode

Item		STOP Mode Operation Status
System clock		System clock oscillation stopped Clock supply to CPU stopped
CPU		Operation stopped
EEPROM		Operation stopped
Port (output latch)		Remains in the state existing before STOP mode has been set
16-bit timer/event counter 0		Operation stopped
16-bit timer D		Operation stopped
8-bit timer/event counter 5		Operation enabled[Note 1]
8-bit timer 80		Operation stopped
Watchdog timer		Operation stopped
Serial interface 2		Operation enabled[Note 2]
Power-on-clear circuit	POC	Operation enabled[Note 3]
	LVI	Operation stopped
Bit sequential buffer		Operation stopped
Key return circuit		Operation enabled
External interrupt		Operation enabled[Note 4]

Notes 1. Operation enabled only when external clock is selected for count clock

2. Operation enabled in both 3-wire serial I/O mode and UART mode when an external clock is selected.

3. If a POC switching circuit is selected by the mask option and the POC circuit is set to operation enabled by software or if POC circuit normally operating is selected by the mask option (refer to **CHAPTER 19 MASK OPTIONS** regarding mask options).

4. A maskable interrupt that is not masked

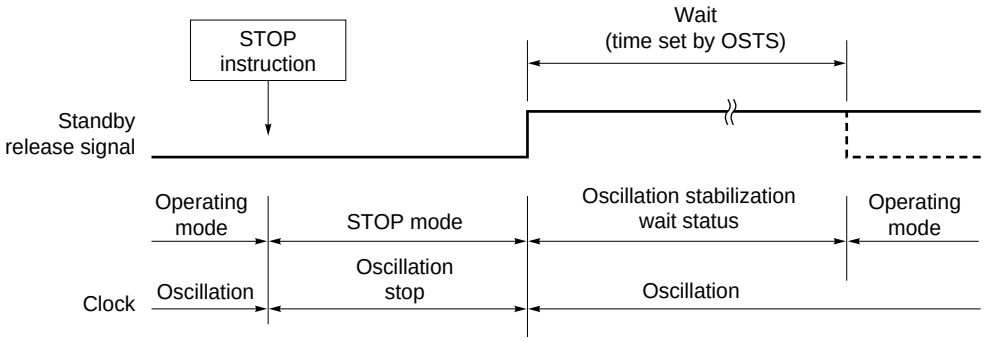
(2) Releasing STOP mode

STOP mode can be released by the following two sources.

(a) Releasing by unmasked interrupt request

STOP mode is released by an unmasked interrupt request. In this case, vectored interrupt servicing is performed if interrupt acknowledgement is enabled after the oscillation stabilization time has elapsed. If interrupt acknowledgement is disabled, the instruction at the next address is executed.

Figure 16-4. Releasing STOP Mode by Interrupt



Remark The broken lines indicate the case where the interrupt request that has released standby mode is acknowledged.

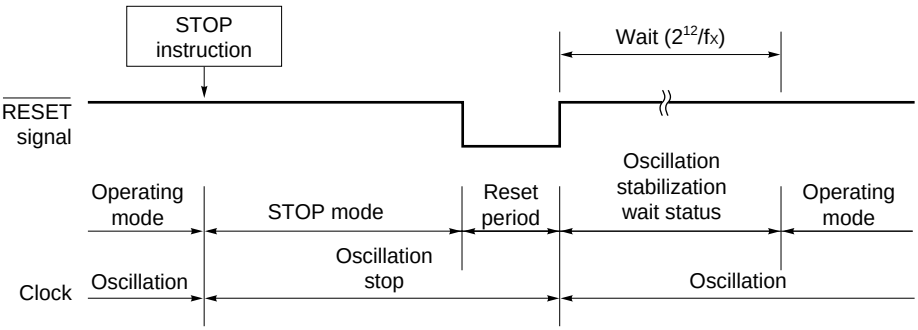
★ **(b) Releasing by non-maskable interrupt request**

STOP mode is released regardless of whether interrupts are enabled or disabled, and vectored interrupt servicing is performed.

(c) Releasing by $\overline{\text{RESET}}$ input

When STOP mode is released by the $\overline{\text{RESET}}$ signal, the reset operation is performed after the oscillation stabilization time has elapsed.

Figure 16-5. Releasing STOP Mode by $\overline{\text{RESET}}$ Input



Remark fx: System clock oscillation frequency

Table 16-4. Operation After Releasing STOP Mode

Releasing Source	MK××	IE	Operation
Maskable interrupt request	0	0	Executes next address instruction.
	0	1	Executes interrupt servicing.
	1	×	Retains STOP mode.
★ Non-maskable interrupt request	–	×	Executes interrupt servicing.
$\overline{\text{RESET}}$ input	–	–	Reset processing

×: Don't care

CHAPTER 17 RESET FUNCTION

The following four operations are available to generate reset signals.

(1) External reset input by $\overline{\text{RESET}}$ signal input
(2) Internal reset by watchdog timer inadvertent program loop time detection
(3) Internal reset by comparison of POC circuit power supply voltage and detection voltage
(4) Internal reset by key input

External reset and internal reset have no functional differences. In both cases, program execution starts at the address at 0000H and 0001H by reset signal input.

When a low level is input to the $\overline{\text{RESET}}$ pin, the watchdog timer overflows, the POC circuit voltage is detected, or the KRREN pin is low level and a valid edge of any key input is detected in STOP mode, a reset is applied and each hardware item is set to the status shown in Table 17-1. Each pin is high impedance during reset input or during the oscillation stabilization time just after reset release.

When a high level is input to the $\overline{\text{RESET}}$ pin, the reset is released and program execution is started after the oscillation stabilization time has elapsed. The reset applied by the watchdog timer overflow is automatically released after reset, and program execution is started after the oscillation stabilization time has elapsed (see **Figures 17-2** through **17-4**).

Cautions 1. For an external reset, input a low level of 10 μs or more to the $\overline{\text{RESET}}$ pin.
2. When STOP mode is released by reset, the STOP mode contents are held during reset input. However, the port pins become high impedance.

Figure 17-1. Block Diagram of Reset Function

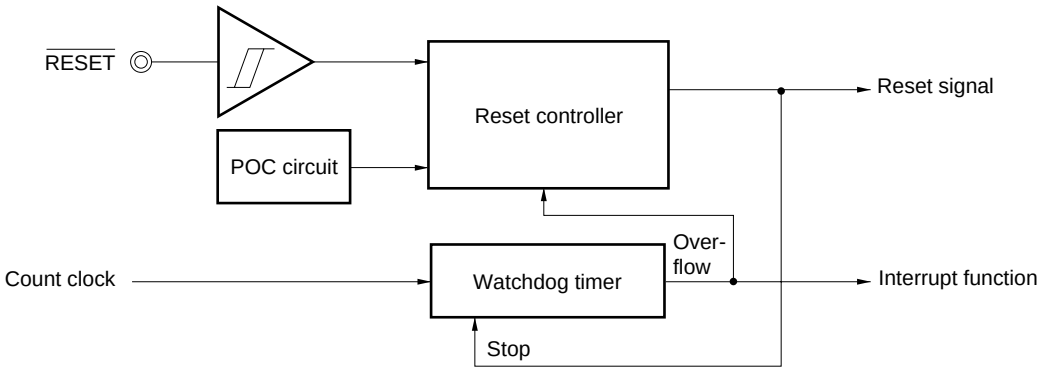


Figure 17-2. Reset Timing by $\overline{\text{RESET}}$ Input

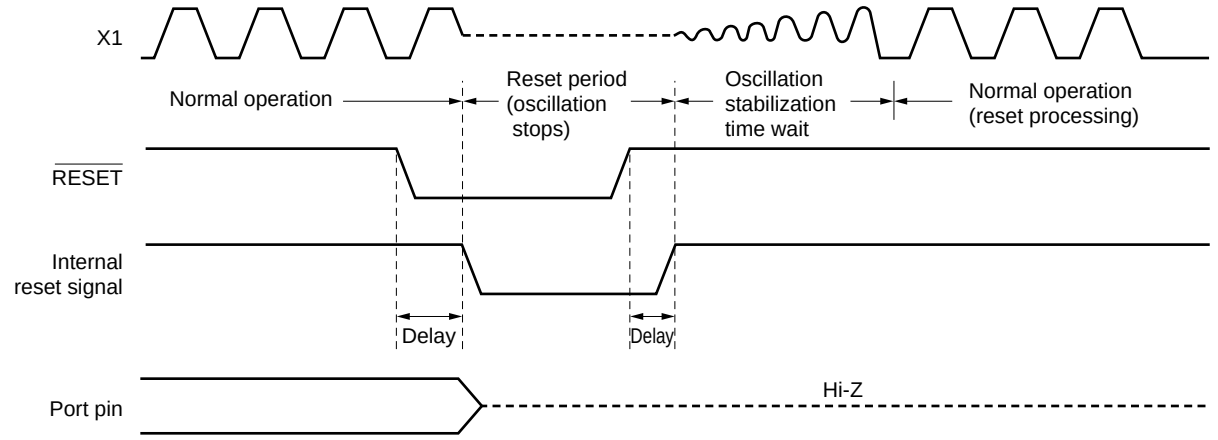


Figure 17-3. Reset Timing by Watchdog Timer Overflow

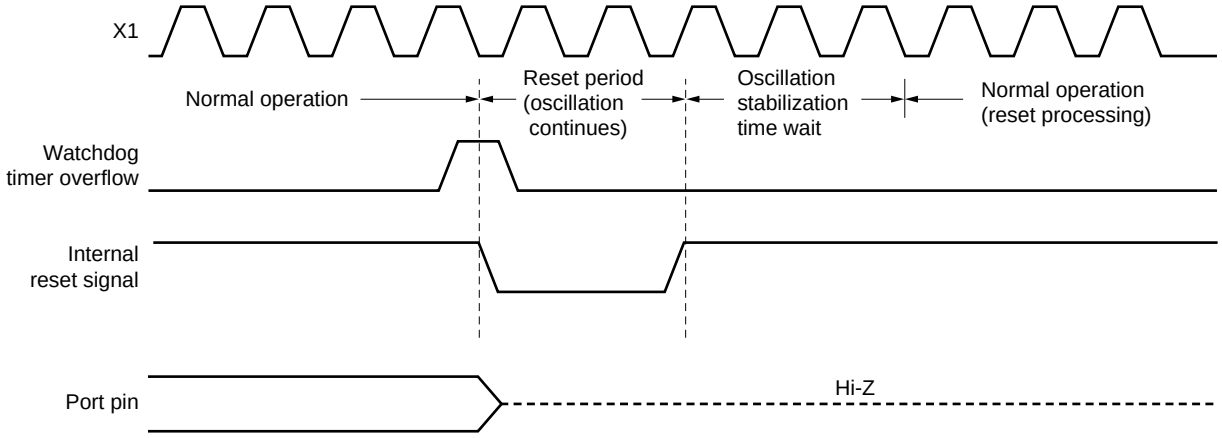


Figure 17-4. Reset Timing by $\overline{\text{RESET}}$ Input in STOP Mode

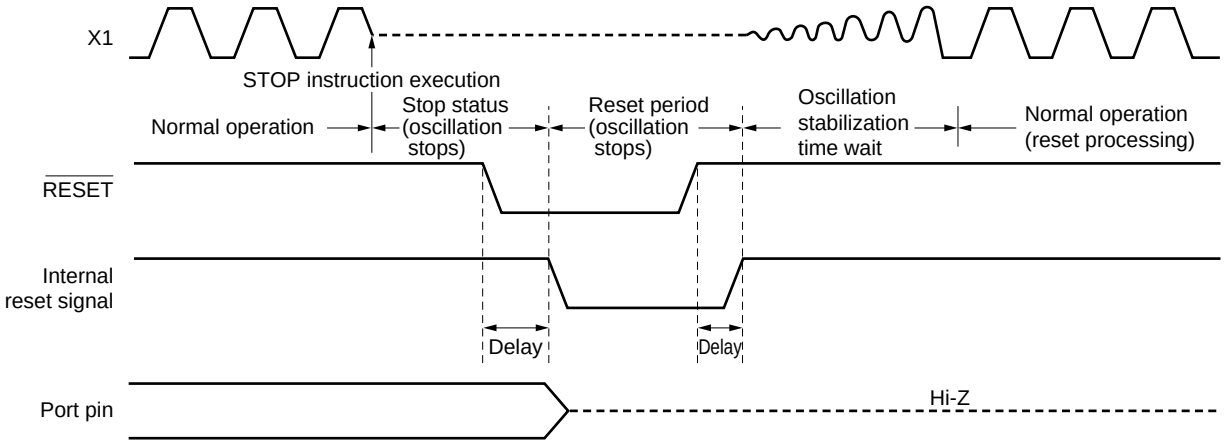


Table 17-1. Status of Hardware After Reset (1/2)

Hardware		Status After Reset
Program counter (PC)[Note 1]		The contents of the reset vector table (0000H, 0001H) are set
Stack pointer (SP)		Undefined
Program status word (PSW)		02H
EEPROM	Write control register (EEWC10)	08H[Note 2]
RAM	Data memory	Undefined[Note 3]
	General-purpose registers	Undefined[Note 3]
Ports (P0 to P3, P7) (output latch)		00H
Port mode registers (PM0 to PM3)		FFH
Pull-up resistor option registers (PUB0 to PUB3)		00H
Processor clock control register (PCC)		02H
Oscillation stabilization time selection register (OSTS)		04H
16-bit timer/event counter 0	Timer counter (TM0)	0000H
	Capture/compare registers (CR00, CR01)	0000H
	Prescaler mode register (PRM0)	00H
	Mode control register (TMC0)	00H
	Output control register (TOC0)	00H
16-bit timer D	Timer counter (TMD)	0000H
	Compare register (CMD)	0000H
	Mode control register (TMCD)	00H
8-bit timer/event counter 5	Timer counter (TM5)	00H
	Compare register (CR5)	00H
	Timer clock selection register (TCL5)	00H
	Mode control register (TMC5)	00H
8-bit timer 80	Timer counter (TM80)	00H
	Compare register (CR80)	Undefined
	Mode control register (TMC80)	00H
Watchdog timer	Timer clock selection register (WDCS)	00H
	Mode register (WDTM)	00H
Serial interface 2	Mode register (CSIM2)	00H
	Asynchronous serial interface mode register (ASIM2)	00H
	Asynchronous serial interface status register (ASIS2)	00H
	Baud rate generator control register (BRGC2)	00H
	Transmit shift register (TXS2)	FFH
	Receive buffer register (RXB2)	Undefined

Notes 1. While a reset signal is being input, and during the oscillation stabilization time wait, the contents of the PC will be undefined, while the remainder of the hardware will be the same as after the reset.

2. The initial value of bit 7 (EEWEM10) varies depending on the status of the EEWE pin.

3. In standby mode, the RAM enters the hold state after a reset.

Table 17-1. Status of Hardware After Reset (2/2)

Hardware		Status After Reset
Power-on-clear circuit	Power-on-clear register (POCF)	00H[Note 1]
	Low-voltage detection register (LVIF)	00H
	Low-voltage detection level selection register (LVIS)	00H
Bit sequential buffer	Data registers (BSFRL10, BSFRH10)	Undefined
	Output control register (BSF1C)	00H
Key return circuit	Key return edge detection register (EDG)	**Note 2**
Interrupt	Request flag registers (IF0, IF1)	00H
	Mask flag registers (MK0, MK1)	FFH
	External interrupt mode register (INTM0)	00H

Notes 1. This value is 04H only after a power-on-clear reset.

2. The value after the key return reset is 1000000×B (bit 0 retains the value before the key return reset is input), and in other cases, the value is 00H.

CHAPTER 18 μPD78E9862

The μPD78E9862 is the EEPROM version in the μPD789862 Subseries.

The μPD78E9862 replaces the internal ROM of the μPD789862 with EEPROM. The differences between the μPD78E9862 and the mask ROM version are shown in Table 18-1.

Table 18-1. Differences Between μPD78E9862 and Mask ROM Version

Item \ Part Number			EEPROM Version	Mask ROM Version
			μPD78E9862[Note 1]	μPD789862
Internal memory	Program memory	ROM structure	EEPROM	Mask ROM
		ROM capacity	16 KB	
	Data memory	High-speed RAM	512 bytes	
		EEPROM	256 bytes	
System clock			Ceramic/crystal oscillation	
IC pin			Not provided	Provided
V_{PP} pin			Provided	Not provided
POC circuit selection by mask option			Not provided[Note 2]	Provided
Electrical specifications			Varies depending on EEPROM or mask ROM version.	

★ **Notes 1.** The μPD78E9862 is provided exclusively for program development; the product life and reliability are not guaranteed. Use the μPD789862 for reliability testing and mass production.

2. The POC circuit can be switched (operation of the POC circuit can be controlled by software).

Caution There are differences in noise immunity and noise radiation between the EEPROM and mask ROM versions. When pre-producing an application set with the EEPROM version and then mass-producing it with the mask ROM version, be sure to conduct sufficient evaluations for the commercial samples (not engineering samples) of the mask ROM version.

18.1 Features of EEPROM (Program Memory)

The on-chip program memory in the μPD78E9862 is EEPROM.

This chapter describes the functions of the EEPROM incorporated in the program memory area. For the EEPROM incorporated in data memory, refer to **CHAPTER 4 EEPROM (DATA MEMORY)**.

EEPROM can be written with the μPD78E9862 mounted on the target system (on-board write). Connect the dedicated flash programmer (Flashpro III (part no. FL-PR3, PG-FP3), Flashpro IV (part no. FL-PR4, PG-FP4)) to the host machine and target system to write to EEPROM.

Remark FL-PR3 and FL-PR4 are products of Naito Densei Machida Mfg. Co., Ltd (TEL +81-45-475-4191).

Programming using EEPROM has the following advantages.

- Software can be modified after the microcontroller is solder-mounted on the target system.
- Distinguishing software facilities small-quantity, varied model production
- Easy data adjustment when starting mass production

18.1.1 Programming environment

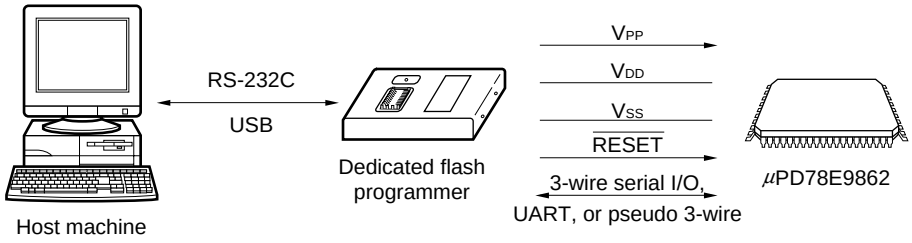
The following shows the environment required for μPD78E9862 EEPROM programming.

When Flashpro III (part no. FL-PR3, PG-FP3) or Flashpro IV (part no. FL-PR4, PG-FP4) is used as a dedicated flash programmer, a host machine is required to control the dedicated flash programmer. Communication between the host machine and flash programmer is performed via RS-232C/USB (Rev. 1.1).

For details, refer to the manuals for Flashpro III/Flashpro IV.

Remark USB is supported by Flashpro IV only.

Figure 18-1. Environment for Writing Program to EEPROM (Program Memory)



18.1.2 Communication mode

Use the communication mode shown in Table 18-2 to perform communication between the dedicated flash programmer and μPD78E9862.

★ **Table 18-2. Communication Mode List**

Communication Mode	TYPE Setting[Note 1]					Pins Used	Number of V_{PP} Pulses
	COMM PORT	SIO Clock	CPU CLOCK		Multiple Rate		
			In Flashpro	On Target Board			
3-wire serial I/O	SIO ch-0 (3-wire, sync.)	100 Hz to 1.25 MHz	4 MHz and 5 MHz[Note 2]	3 to 5 MHz	1.0	SI2/RxD2/P32 SO2/TxD2/P31 $\overline{SCK2}$/ASCK2/P30	0
UART	UART ch-0 (Async.)	4800 to 76800 bps	5 MHz[Note 3]	4.91 and 5 MHz	1.0	RxD2/SI2/P32 TxD2/SO2/P31	8
Pseudo 3-wire	Port A (Pseudo-3 wire)	100 Hz to 1 kHz	4 MHz and 5 MHz[Note 2]	3 to 5 MHz	1.0	P02 (serial data input) P01 (serial data output) P00 (serial clock input)	12

Notes 1. Selection items for TYPE settings on the dedicated flash programmer (Flashpro III (part no. FL-PR3, PG-FP3)/Flashpro IV (part no. FL-PR4, PG-FP4)).

2. Only 4 MHz for Flashpro III.

3. Selectable only for Flashpro IV.

Figure 18-2. Communication Mode Selection Format

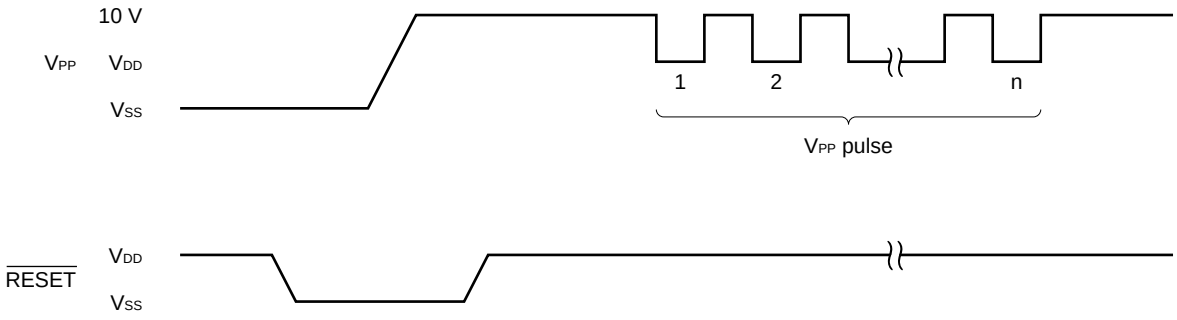
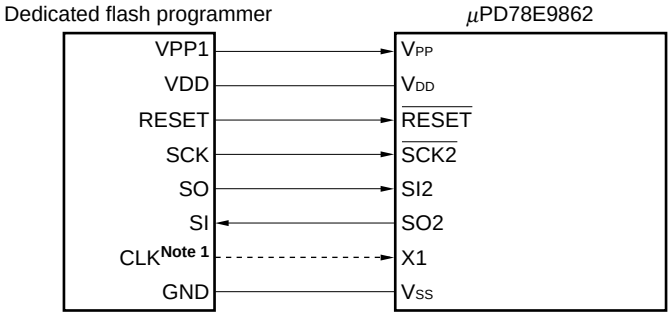
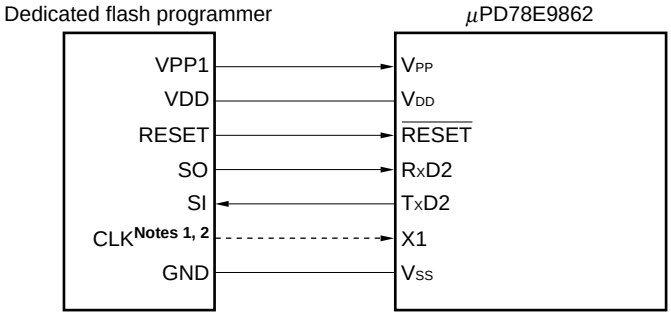


Figure 18-3. Example of Connection with Dedicated Flash Programmer

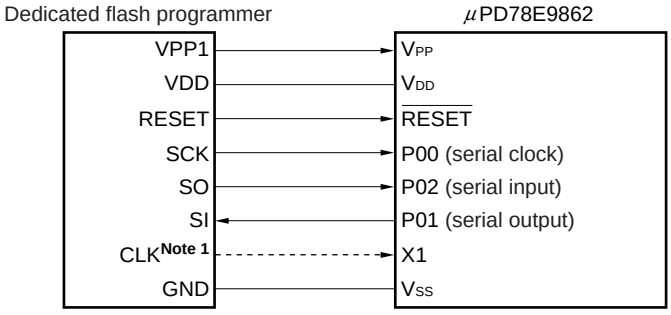
(a) 3-wire serial I/O



(b) UART



(c) Pseudo 3-wire



Notes 1. Connect this pin when the system clock is supplied from the dedicated flash programmer. If a resonator is already connected to the X1 pin, the CLK pin does not need to be connected.

2. When using UART with Flashpro III, the clock of the resonator connected to the X1 pin must be used, so connection to the CLK pin is not necessary.

Caution The V_{DD} pin, if already connected to the power supply, must be connected to the VDD pin of the dedicated flash programmer. When using the power supply connected to the VDD pin, supply voltage before starting programming.

If Flashpro III (part no. FL-PR3, PG-FP3)/Flashpro IV (part no. FL-PR4, PG-FP4) is used as a dedicated flash programmer, the following signals are generated for the μPD78E9862. For details, refer to the manual of Flashpro III/Flashpro IV.

Table 18-3. Pin Connection List

Signal Name	I/O	Pin Function	Pin Name	3-Wire Serial I/O	UART	Pseudo 3-Wire
VPP1	Output	Write voltage	V_{PP}	◎	◎	◎
VPP2	–	–	–	×	×	×
VDD	I/O	V_{DD} voltage generation/voltage monitoring	V_{DD}	◎Note	◎Note	◎Note
GND	–	Ground	V_{SS}	◎	◎	◎
CLK	Output	Clock output	X1	○	○	○
RESET	Output	Reset signal	$\overline{\text{RESET}}$	◎	◎	◎
SI	Input	Receive signal	SO2/TxD2 (P01)	◎	◎	◎
SO	Output	Transmit signal	SI2/RxD2 (P02)	◎	◎	◎
SCK	Output	Transfer clock	$\overline{\text{SCK2}}$ (P00)	◎	×	◎
HS	Input	Handshake signal	–	×	×	×

Note V_{DD} voltage must be supplied before programming is started.

Remark ◎: Pin must be connected.
○: If the signal is supplied on the target board, pin does not need to be connected.
×: Pin does not need to be connected.
Pin names in parentheses are for pseudo 3-wire.

18.1.3 On-board pin processing

When performing programming on the target system, provide a connector on the target system to connect the dedicated flash programmer.

An on-board function that allows switching between normal operation mode and EEPROM programming mode may be required in some cases.

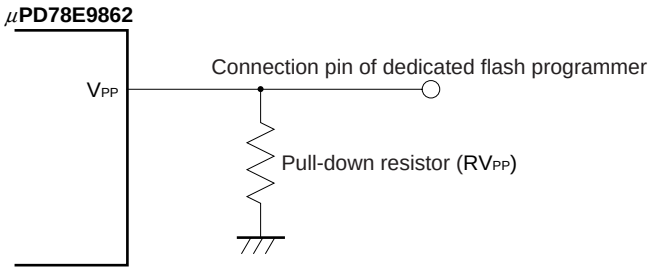
<V_{PP} pin>

In normal operation mode, input 0 V to the V_{PP} pin. In EEPROM programming mode, a write voltage of 10.0 V (TYP.) is supplied to the V_{PP} pin, so perform either of the following.

(1) Connect a pull-down resistor RV_{PP} = 10 kΩ to the V_{PP} pin.
(2) Use the jumper on the board to switch the V_{PP} pin input to either the programmer or directly to GND.

A V_{PP} pin connection example is shown below.

Figure 18-4. V_{PP} Pin Connection Example



<Serial interface pins>

The following shows the pins used by the serial interface.

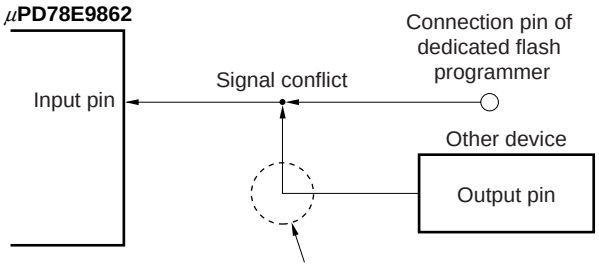
Serial Interface	Pins Used
3-wire serial I/O	SI2, SO2, $\overline{SCK2}$
UART	RxD2, TxD2
Pseudo 3-wire	P02, P01, P00

When connecting the dedicated flash programmer to a serial interface pin that is connected to another device on-board, signal conflict or abnormal operation of the other device may occur. Care must therefore be taken with such connections.

(1) Signal conflict

If the dedicated flash programmer (output) is connected to a serial interface pin (input) that is connected to another device (output), a signal conflict occurs. To prevent this, isolate the connection with the other device or set the other device to the output high impedance status.

Figure 18-5. Signal Conflict (Input Pin of Serial Interface)

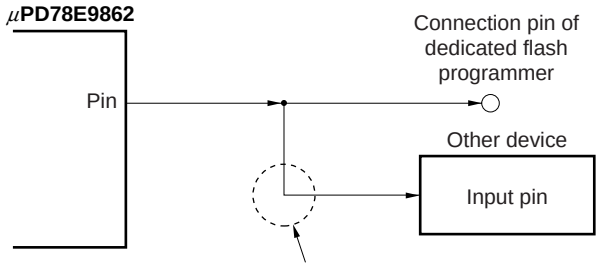


In the EEPROM programming mode, the signal output by another device and the signal sent by the dedicated flash programmer conflict; therefore, isolate the signal of the other device.

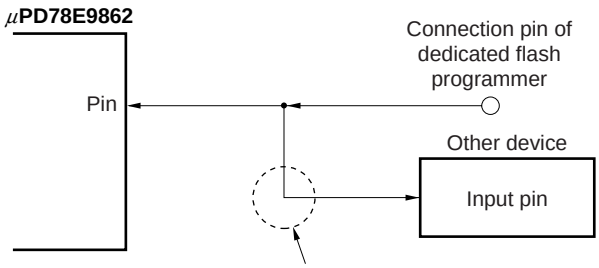
(2) Abnormal operation of other device

If the dedicated flash programmer (output or input) is connected to a serial interface pin (input or output) that is connected to another device (input), a signal is output to the device, and this may cause an abnormal operation. To prevent this abnormal operation, isolate the connection with the other device or set so that the signals input to the other device are ignored.

Figure 18-6. Abnormal Operation of Other Device



If the signal output by the μPD78E9862 affects another device in the EEPROM programming mode, isolate the signals of the other device.



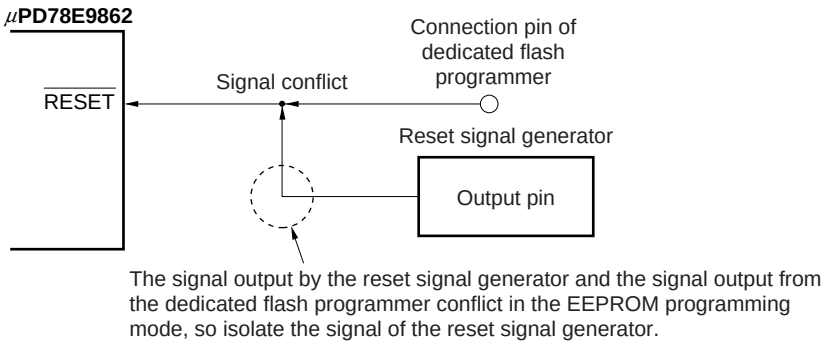
If the signal output by the dedicated flash programmer affects another device in the EEPROM programming mode, isolate the signals of the other device.

<$\overline{\text{RESET}}$ pin>

If the reset signal of the dedicated flash programmer is connected to the $\overline{\text{RESET}}$ pin connected to the reset signal generator on-board, a signal conflict occurs. To prevent this, isolate the connection with the reset signal generator.

If the reset signal is input from the user system in the EEPROM programming mode, a normal programming operation cannot be performed. Therefore, do not input other than reset signals from the dedicated flash programmer.

Figure 18-7. Signal Conflict ($\overline{\text{RESET}}$ Pin)



<Port pins>

When the μPD78E9862 enters the EEPROM programming mode, all the pins other than those that communicate with the flash programmer are in the same status as immediately after reset.

If the external device does not recognize initial statuses such as the output high impedance status, therefore, connect the external device to V_{DD} or V_{SS} via a resistor.

<Oscillation pins>

When using the on-board clock, connect X1 and X2 as required in the normal operation mode.

When using the clock output of the flash programmer, connect it directly to X1, disconnecting the resonator on-board, and leave the X2 pin open.

<Power supply>

To use the power output from the flash programmer, connect the V_{DD} pin to VDD of the flash programmer, and the V_{SS} pin to GND of the flash programmer.

To use the on-board power supply, make connections that accord with the normal operation mode. However, because the voltage is monitored by the flash programmer, be sure to connect VDD of the flash programmer.

18.1.4 Connection on EEPROM writing adapter

The following shows examples of the recommended connection when using the EEPROM writing adapter.

Figure 18-8. Wiring Example for EEPROM Writing Adapter in 3-Wire Serial I/O Mode

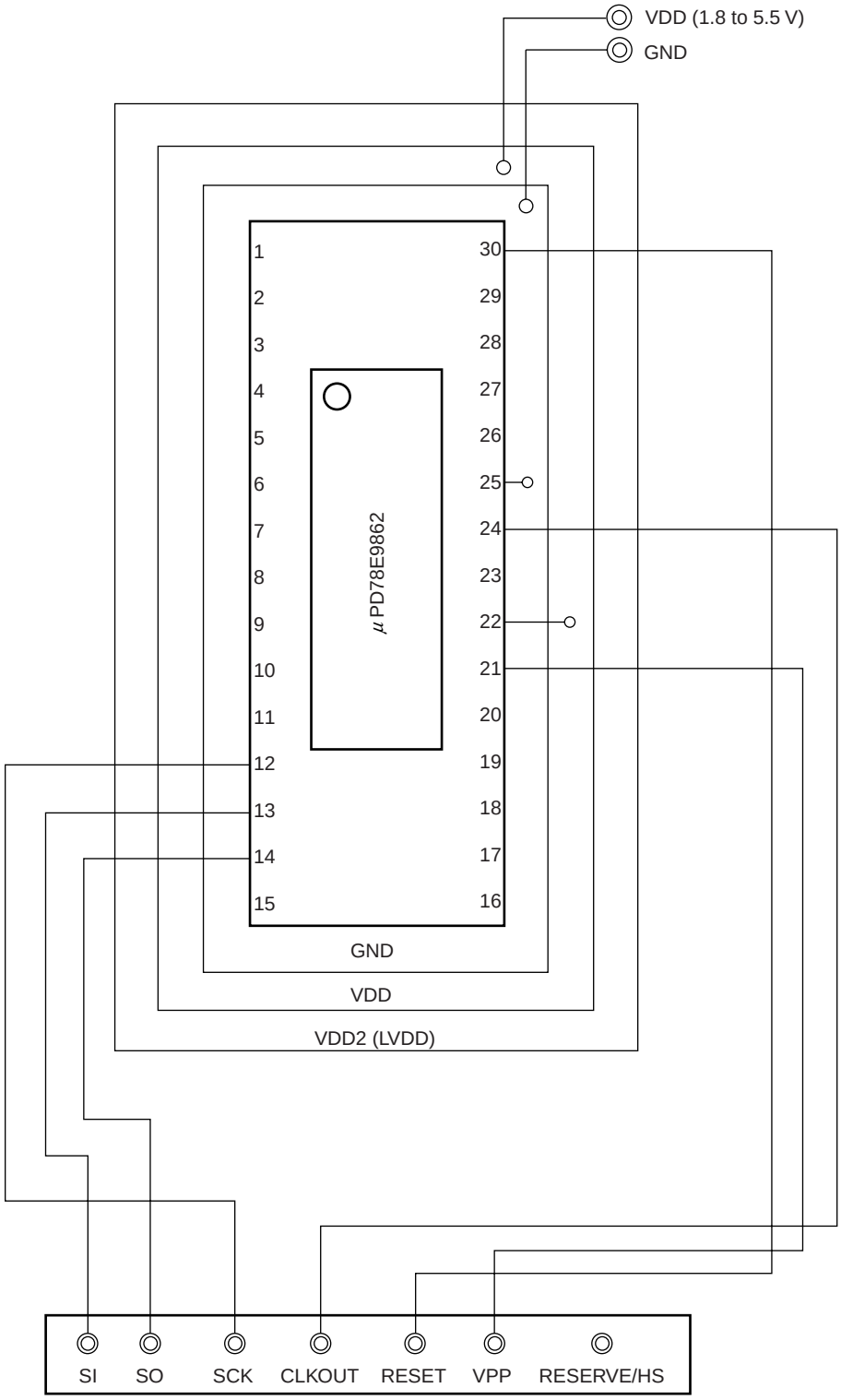


Figure 18-9. Wiring Example for EEPROM Writing Adapter in UART Mode

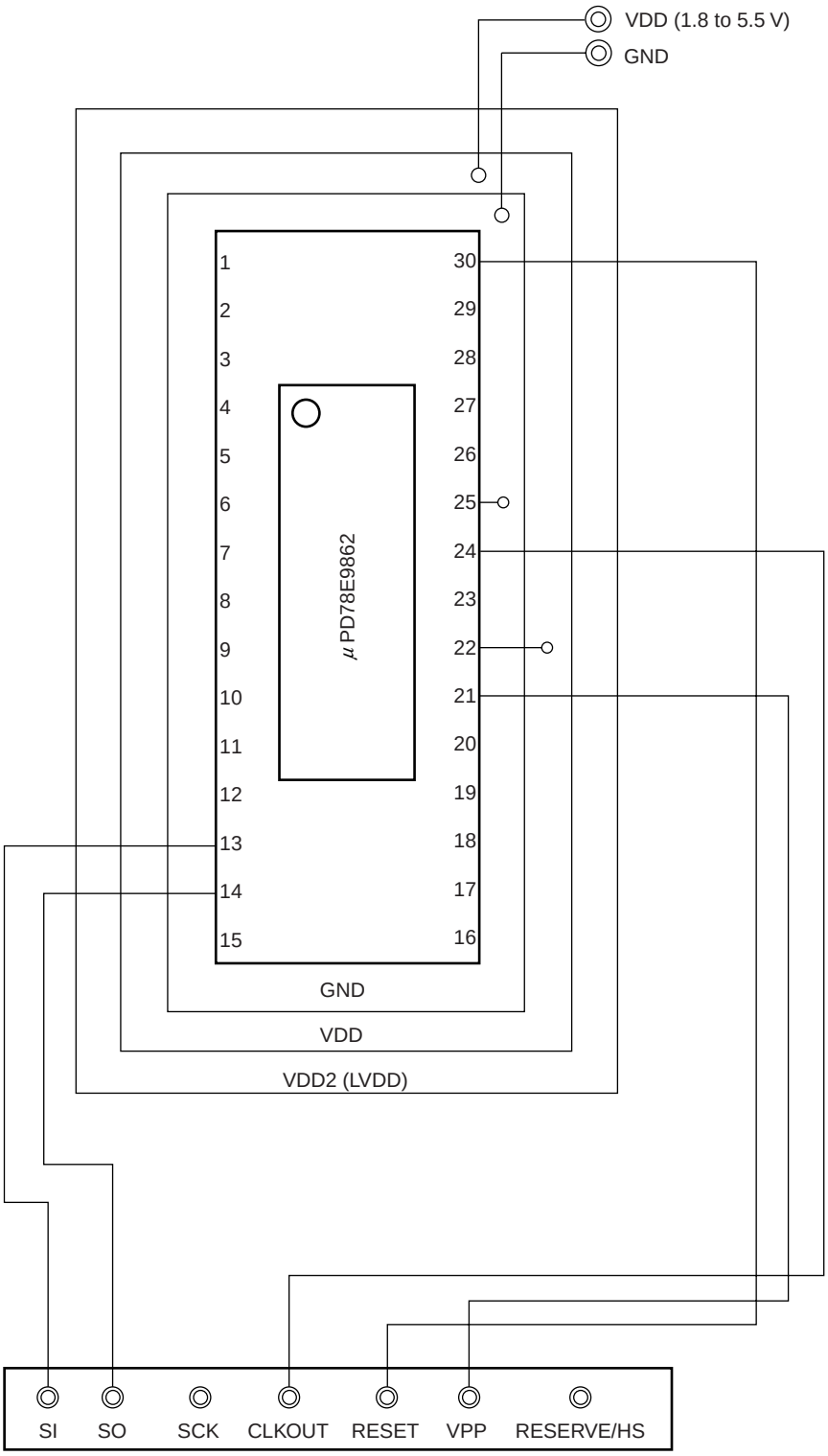
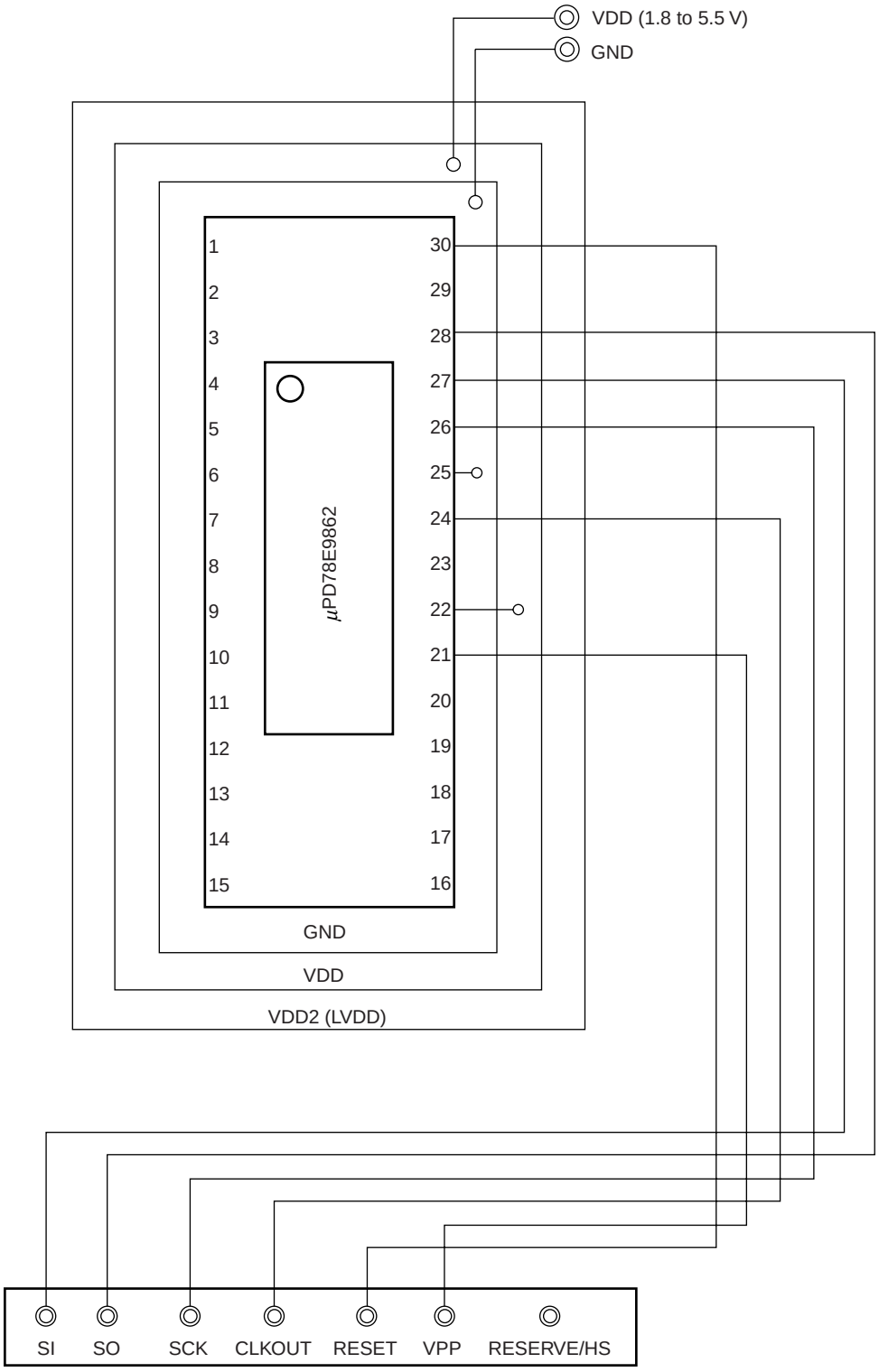


Figure 18-10. Wiring Example for EEPROM Writing Adapter in Pseudo 3-Wire Mode



CHAPTER 19 MASK OPTIONS

The μPD789862 has the following mask options.

- POC circuit mask options
  The POC circuit can be selected.
  <1> Select POC switching circuit (POC circuit operation control by software is possible)
  <2> Select POC circuit normally operating
  <3> Select POC circuit normally halted

CHAPTER 20 INSTRUCTION SET OVERVIEW

This chapter lists the instruction set of the μPD789862 Subseries. For details of the operation and machine language (instruction code) of each instruction, refer to **78K/0S Series Instructions User's Manual (U11047E)**.

20.1 Operation

20.1.1 Operand identifiers and description methods

Operands are described in "Operand" column of each instruction in accordance with the description method of the instruction operand identifier (refer to the assembler specifications for details). When there are two or more description methods, select one of them. Uppercase letters and the symbols #, !, $, and [ ] are key words and are described as they are. Each symbol has the following meaning.

- #: Immediate data specification
- !: Absolute address specification
- $: Relative address specification
- [ ]: Indirect address specification

In the case of immediate data, describe an appropriate numeric value or a label. When using a label, be sure to describe 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 description.

Table 20-1. Operand Identifiers and Description Methods

Identifier	Description Method
r rp sfr	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
saddr saddrp	FE20H to FF1FH Immediate data or labels FE20H to FF1FH Immediate data or labels (even addresses only)
addr16 addr5	0000H to FFFFH Immediate data or labels (only even addresses for 16-bit data transfer instructions) 0040H to 007FH Immediate data or labels (even addresses only)
word byte bit	16-bit immediate data or label 8-bit immediate data or label 3-bit immediate data or label

Remark For symbols of special function registers, see **Table 3-3 Special Function Registers**.

20.1.2 Description of “Operation” column

A:	A register; 8-bit accumulator
X:	X register
B:	B register
C:	C register
D:	D register
E:	E register
H:	H register
L:	L register
AX:	AX register pair; 16-bit accumulator
BC:	BC register pair
DE:	DE register pair
HL:	HL register pair
PC:	Program counter
SP:	Stack pointer
PSW:	Program status word
CY:	Carry flag
AC:	Auxiliary carry flag
Z:	Zero flag
IE:	Interrupt request enable flag
NMIS:	Flag indicating non-maskable interrupt servicing in progress
( ):	Memory contents indicated by address or register contents in parentheses
$\times_H$, $\times_L$:	Higher 8 bits and lower 8 bits of 16-bit register
$\wedge$:	Logical product (AND)
$\vee$:	Logical sum (OR)
$\veebar$:	Exclusive logical sum (exclusive OR)
$\overline{\ }$:	Inverted data
addr16:	16-bit immediate data or label
jdisp8:	Signed 8-bit data (displacement value)

20.1.3 Description of “Flag” column

(Blank):	Unchanged
0:	Cleared to 0
1:	Set to 1
$\times$:	Set/cleared according to the result
R:	Previously saved value is stored

20.2 Operation List

Mnemonic	Operand	Bytes	Clocks	Operation	Flag
					Z AC CY
MOV	r, #byte	3	6	r ← byte	
	saddr, #byte	3	6	(saddr) ← byte	
	sfr, #byte	3	6	sfr ← byte	
	A, r Note 1	2	4	A ← r	
	r, A Note 1	2	4	r ← A	
	A, saddr	2	4	A ← (saddr)	
	saddr, A	2	4	(saddr) ← A	
	A, sfr	2	4	A ← sfr	
	sfr, A	2	4	sfr ← A	
	A, !addr16	3	8	A ← (addr16)	
	!addr16, A	3	8	(addr16) ← A	
	PSW, #byte	3	6	PSW ← byte	× × ×
	A, PSW	2	4	A ← PSW	
	PSW, A	2	4	PSW ← A	× × ×
	A, [DE]	1	6	A ← (DE)	
	[DE], A	1	6	(DE) ← A	
	A, [HL]	1	6	A ← (HL)	
	[HL], A	1	6	(HL) ← A	
	A, [HL + byte]	2	6	A ← (HL + byte)	
	[HL + byte], A	2	6	(HL + byte) ← A	
XCH	A, X	1	4	A ↔ X	
	A, r Note 2	2	6	A ↔ r	
	A, saddr	2	6	A ↔ (saddr)	
	A, sfr	2	6	A ↔ sfr	
	A, [DE]	1	8	A ↔ (DE)	
	A, [HL]	1	8	A ↔ (HL)	
	A, [HL, byte]	2	8	A ↔ (HL + byte)	

Notes 1. Except r = A.

2. Except r = A, X.

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

Mnemonic	Operand	Bytes	Clocks	Operation	Flag		
					Z	AC	CY
MOVW	rp, #word	3	6	rp ← word			
	AX, saddrp	2	6	AX ← (saddrp)			
	saddrp, AX	2	8	(saddrp) ← AX			
	AX, rp Note	1	4	AX ← rp			
	rp, AX Note	1	4	rp ← AX			
XCHW	AX, rp Note	1	8	AX ↔ rp			
ADD	A, #byte	2	4	A, CY ← A + byte	×	×	×
	saddr, #byte	3	6	(saddr), CY ← (saddr) + byte	×	×	×
	A, r	2	4	A, CY ← A + r	×	×	×
	A, saddr	2	4	A, CY ← A + (saddr)	×	×	×
	A, !addr16	3	8	A, CY ← A + (addr16)	×	×	×
	A, [HL]	1	6	A, CY ← A + (HL)	×	×	×
	A, [HL + byte]	2	6	A, CY ← A + (HL + byte)	×	×	×
ADDC	A, #byte	2	4	A, CY ← A + byte + CY	×	×	×
	saddr, #byte	3	6	(saddr), CY ← (saddr) + byte + CY	×	×	×
	A, r	2	4	A, CY ← A + r + CY	×	×	×
	A, saddr	2	4	A, CY ← A + (saddr) + CY	×	×	×
	A, !addr16	3	8	A, CY ← A + (addr16) + CY	×	×	×
	A, [HL]	1	6	A, CY ← A + (HL) + CY	×	×	×
	A, [HL + byte]	2	6	A, CY ← A + (HL + byte) + CY	×	×	×
SUB	A, #byte	2	4	A, CY ← A – byte	×	×	×
	saddr, #byte	3	6	(saddr), CY ← (saddr) – byte	×	×	×
	A, r	2	4	A, CY ← A – r	×	×	×
	A, saddr	2	4	A, CY ← A – (saddr)	×	×	×
	A, !addr16	3	8	A, CY ← A – (addr16)	×	×	×
	A, [HL]	1	6	A, CY ← A – (HL)	×	×	×
	A, [HL + byte]	2	6	A, CY ← A – (HL + byte)	×	×	×

Note Only when rp = BC, DE, or HL.

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

Mnemonic	Operand	Bytes	Clocks	Operation	Flag		
					Z	AC	CY
SUBC	A, #byte	2	4	A, CY ← A – byte – CY	×	×	×
	saddr, #byte	3	6	(saddr), CY ← (saddr) – byte – CY	×	×	×
	A, r	2	4	A, CY ← A – r – CY	×	×	×
	A, saddr	2	4	A, CY ← A – (saddr) – CY	×	×	×
	A, !addr16	3	8	A, CY ← A – (addr16) – CY	×	×	×
	A, [HL]	1	6	A, CY ← A – (HL) – CY	×	×	×
	A, [HL + byte]	2	6	A, CY ← A – (HL + byte) – CY	×	×	×
AND	A, #byte	2	4	A ← A ∧ byte	×		
	saddr, #byte	3	6	(saddr) ← (saddr) ∧ byte	×		
	A, r	2	4	A ← A ∧ r	×		
	A, saddr	2	4	A ← A ∧ (saddr)	×		
	A, !addr16	3	8	A ← A ∧ (addr16)	×		
	A, [HL]	1	6	A ← A ∧ (HL)	×		
	A, [HL + byte]	2	6	A ← A ∧ (HL + byte)	×		
OR	A, #byte	2	4	A ← A ∨ byte	×		
	saddr, #byte	3	6	(saddr) ← (saddr) ∨ byte	×		
	A, r	2	4	A ← A ∨ r	×		
	A, saddr	2	4	A ← A ∨ (saddr)	×		
	A, !addr16	3	8	A ← A ∨ (addr16)	×		
	A, [HL]	1	6	A ← A ∨ (HL)	×		
	A, [HL + byte]	2	6	A ← A ∨ (HL + byte)	×		
XOR	A, #byte	2	4	A ← A ⊻ byte	×		
	saddr, #byte	3	6	(saddr) ← (saddr) ⊻ byte	×		
	A, r	2	4	A ← A ⊻ r	×		
	A, saddr	2	4	A ← A ⊻ (saddr)	×		
	A, !addr16	3	8	A ← A ⊻ (addr16)	×		
	A, [HL]	1	6	A ← A ⊻ (HL)	×		
	A, [HL + byte]	2	6	A ← A ⊻ (HL + byte)	×		

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

Mnemonic	Operand	Bytes	Clocks	Operation	Flag		
					Z	AC	CY
CMP	A, #byte	2	4	A – byte	×	×	×
	saddr, #byte	3	6	(saddr) – byte	×	×	×
	A, r	2	4	A – r	×	×	×
	A, saddr	2	4	A – (saddr)	×	×	×
	A, !addr16	3	8	A – (addr16)	×	×	×
	A, [HL]	1	6	A – (HL)	×	×	×
	A, [HL + byte]	2	6	A – (HL + byte)	×	×	×
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	×	×	×
INC	r	2	4	r ← r + 1	×	×	
	saddr	2	4	(saddr) ← (saddr) + 1	×	×	
DEC	r	2	4	r ← r – 1	×	×	
	saddr	2	4	(saddr) ← (saddr) – 1	×	×	
INCW	rp	1	4	rp ← rp + 1			
DECW	rp	1	4	rp ← rp – 1			
ROR	A, 1	1	2	(CY, A_7 ← A_0, A_{m-1} ← A_m) × 1			×
ROL	A, 1	1	2	(CY, A_0 ← A_7, A_{m+1} ← A_m) × 1			×
RORC	A, 1	1	2	(CY ← A_0, A_7 ← CY, A_{m-1} ← A_m) × 1			×
ROLC	A, 1	1	2	(CY ← A_7, A_0 ← CY, A_{m+1} ← A_m) × 1			×
SET1	saddr.bit	3	6	(saddr.bit) ← 1			
	sfr.bit	3	6	sfr.bit ← 1			
	A.bit	2	4	A.bit ← 1			
	PSW.bit	3	6	PSW.bit ← 1	×	×	×
	[HL].bit	2	10	(HL).bit ← 1			
CLR1	saddr.bit	3	6	(saddr.bit) ← 0			
	sfr.bit	3	6	sfr.bit ← 0			
	A.bit	2	4	A.bit ← 0			
	PSW.bit	3	6	PSW.bit ← 0	×	×	×
	[HL].bit	2	10	(HL).bit ← 0			
SET1	CY	1	2	CY ← 1			1
CLR1	CY	1	2	CY ← 0			0
NOT1	CY	1	2	CY ← $\overline{CY}$			×

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

Mnemonic	Operand	Bytes	Clocks	Operation	Flag		
					Z	AC	CY
CALL	!addr16	3	6	$(SP - 1) \leftarrow (PC + 3)_H$, $(SP - 2) \leftarrow (PC + 3)_L$, $PC \leftarrow$ addr16, $SP \leftarrow SP - 2$			
CALLT	[addr5]	1	8	$(SP - 1) \leftarrow (PC + 1)_H$, $(SP - 2) \leftarrow (PC + 1)_L$, $PC_H \leftarrow$ (00000000, addr5 + 1), $PC_L \leftarrow$ (00000000, addr5), $SP \leftarrow SP - 2$			
RET		1	6	$PC_H \leftarrow (SP + 1)$, $PC_L \leftarrow (SP)$, $SP \leftarrow SP + 2$			
RETI		1	8	$PC_H \leftarrow (SP + 1)$, $PC_L \leftarrow (SP)$, PSW $\leftarrow (SP + 2)$, $SP \leftarrow SP + 3$, NMIS $\leftarrow 0$	R	R	R
PUSH	PSW	1	2	$(SP - 1) \leftarrow$ PSW, $SP \leftarrow SP - 1$			
	rp	1	4	$(SP - 1) \leftarrow rp_H$, $(SP - 2) \leftarrow rp_L$, $SP \leftarrow SP - 2$			
POP	PSW	1	4	PSW $\leftarrow (SP)$, $SP \leftarrow SP + 1$	R	R	R
	rp	1	6	$rp_H \leftarrow (SP + 1)$, $rp_L \leftarrow (SP)$, $SP \leftarrow SP + 2$			
MOVW	SP, AX	2	8	$SP \leftarrow AX$			
	AX, SP	2	6	$AX \leftarrow SP$			
BR	!addr16	3	6	$PC \leftarrow$ addr16			
	$addr16	2	6	$PC \leftarrow PC + 2 +$ jdisp8			
	AX	1	6	$PC_H \leftarrow A$, $PC_L \leftarrow X$			
BC	$saddr16	2	6	$PC \leftarrow PC + 2 +$ jdisp8 if CY = 1			
BNC	$saddr16	2	6	$PC \leftarrow PC + 2 +$ jdisp8 if CY = 0			
BZ	$saddr16	2	6	$PC \leftarrow PC + 2 +$ jdisp8 if Z = 1			
BNZ	$saddr16	2	6	$PC \leftarrow PC + 2 +$ jdisp8 if Z = 0			
BT	saddr.bit, $addr16	4	10	$PC \leftarrow PC + 4 +$ jdisp8 if (saddr.bit) = 1			
	sfr.bit, $addr16	4	10	$PC \leftarrow PC + 4 +$ jdisp8 if sfr.bit = 1			
	A.bit, $addr16	3	8	$PC \leftarrow PC + 3 +$ jdisp8 if A.bit = 1			
	PSW.bit, $addr16	4	10	$PC \leftarrow PC + 4 +$ jdisp8 if PSW.bit = 1			
BF	saddr.bit, $addr16	4	10	$PC \leftarrow PC + 4 +$ jdisp8 if (saddr.bit) = 0			
	sfr.bit, $addr16	4	10	$PC \leftarrow PC + 4 +$ jdisp8 if sfr.bit = 0			
	A.bit, $addr16	3	8	$PC \leftarrow PC + 3 +$ jdisp8 if A.bit = 0			
	PSW.bit, $addr16	4	10	$PC \leftarrow PC + 4 +$ jdisp8 if PSW.bit = 0			
DBNZ	B, $addr16	2	6	$B \leftarrow B - 1$, then $PC \leftarrow PC + 2 +$ jdisp8 if $B \neq 0$			
	C, $addr16	2	6	$C \leftarrow C - 1$, then $PC \leftarrow PC + 2 +$ jdisp8 if $C \neq 0$			
	saddr, $addr16	3	8	$(saddr) \leftarrow (saddr) - 1$, then $PC \leftarrow PC + 3 +$ jdisp8 if $(saddr) \neq 0$			
NOP		1	2	No Operation			
EI		3	6	IE $\leftarrow 1$ (Enable Interrupt)			
DI		3	6	IE $\leftarrow 0$ (Disable Interrupt)			
HALT		1	2	Set HALT Mode			
STOP		1	2	Set STOP Mode			

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

20.3 Instructions Listed by Addressing Type

(1) 8-bit instructions

MOV, XCH, ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP, INC, DEC, ROR, ROL, RORC, ROLC, PUSH, POP, DBNZ

1st Operand \ 2nd Operand	#byte	A	r	sfr	saddr	!addr16	PSW	[DE]	[HL]	[HL + byte]	$addr16	1	None
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV[Note] XCH[Note] ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV 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											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											
[HL + byte]		MOV											

Note Except r = A.

(2) 16-bit instructions

MOVW, XCHW, ADDW, SUBW, CMPW, PUSH, POP, INCW, DECW

1st Operand \ 2nd Operand	#word	AX	rp[Note]	saddrp	SP	None
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	
rp	MOVW	MOVW[Note]				INCW DECW PUSH POP
saddrp		MOVW				
sp		MOVW				

Note Only when rp = BC, DE, or HL.

(3) Bit manipulation instructions

SET1, CLR1, NOT1, BT, BF

1st Operand \ 2nd Operand	$addr16	None
A.bit	BT BF	SET1 CLR1
sfr.bit	BT BF	SET1 CLR1
saddr.bit	BT BF	SET1 CLR1
PSW.bit	BT BF	SET1 CLR1
[HL].bit		SET1 CLR1
CY		SET1 CLR1 NOT1

(4) Call instructions/branch instructions

CALL, CALLT, BR, BC, BNC, BZ, BNZ, DBNZ

1st Operand \ 2nd Operand	AX	!addr16	[addr5]	$addr16
Basic instructions	BR	CALL BR	CALLT	BR BC BNC BZ BNZ
Compound instructions				DBNZ

(5) Other instructions

RET, RETI, NOP, EI, DI, HALT, STOP

Absolute Maximum Ratings (T_A = 25°C)

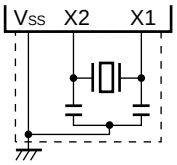
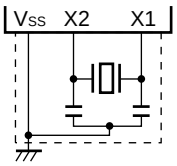
Parameter	Symbol	Conditions	Ratings	Unit
Supply voltage	V_{DD}		–0.3 to +6.5	V
Input voltage	V_I		–0.3 to V_{DD} + 0.3[Note]	V
Output voltage	V_O		–0.3 to V_{DD} + 0.3[Note]	V
Output current, high	I_{OH}	Per pin	–10	mA
		Total for all pins	–30	mA
Output current, low	I_{OL}	Per pin	30	mA
		Total for all pins	80	mA
Operating ambient temperature	T_A		–40 to +85	°C
Storage temperature	T_{stg}		–40 to +125	°C

Note 6.5 V or less

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

System Clock Oscillator Characteristics (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic resonator		Oscillation frequency (f_X)Note 1	V_{DD} = oscillation voltage range	3		5	MHz
		Oscillation stabilization timeNote 2	After V_{DD} reaches oscillation voltage range MIN.		80		μs
Crystal resonator		Oscillation frequency (f_X)Note 1	V_{DD} = oscillation voltage range	3		5	MHz
		Oscillation stabilization timeNote 2	After V_{DD} reaches oscillation voltage range MIN.		8		ms

Notes 1. Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

2. Time required to stabilize oscillation after reset or STOP mode release. Use the resonator that stabilizes oscillation within the oscillation wait time.

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

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

Recommended Oscillator Constant

Ceramic resonator (T_A = –40 to +85°C)

Manufacturer	Part Number	Frequency (MHz)	Recommended Circuit Constant (pF)		Oscillation Voltage Range (V_{DD})		Remarks
			C1	C2	MIN.	MAX.	
Murata Mfg. Co., Ltd.	CSTCR4M00G53-R0	4.000	–	–	1.8	5.5	On-chip capacitor
	CSTLS4M00G53-B0						
	CSTCR4M19G53-R0	4.195					
	CSTLS4M19G53-B0						
	CSTCR5M00G53-R0	5.000					
	CSTLS5M00G53-B0						

Caution The oscillator constant is a reference value based on evaluation in specific environments by the resonator manufacturer. If the oscillator characteristics need to be optimized in the actual application, request the resonator manufacturer for evaluation on the implementation circuit. Note that the oscillation voltage and oscillation frequency merely indicate the characteristics of the oscillator. Use the internal operation conditions of the μPD789862 within the specifications of the DC and AC characteristics.

DC Characteristics (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V) (1/2)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, low	I_{OL}	Per pin				2.5	mA
		Total for all pins				80.0	mA
Output current, high	I_{OH}	Per pin				–0.5	mA
		Total for all pins				–15.0	mA
Input voltage, high	V_{IH1}	P22 to P24, P30, P32, P33, P40 to P45, $\overline{\text{RESET}}$, KRREN	2.7 V ≤ V_{DD} ≤ 5.5 V	0.8V_{DD}		V_{DD}	V
			1.8 V ≤ V_{DD} < 2.7 V	0.9V_{DD}		V_{DD}	V
	V_{IH2}	P00 to P03, P10, P11, P20, P21, P31	2.7 V ≤ V_{DD} ≤ 5.5 V	0.7V_{DD}		V_{DD}	V
			1.8 V ≤ V_{DD} < 2.7 V	0.9V_{DD}		V_{DD}	V
	V_{IH3}	X1, X2		V_{DD} – 0.1		V_{DD}	V
	V_{IH4}	EEWE		0.9V_{DD}		V_{DD}	V
Input voltage, low	V_{IL1}	P22 to P24, P30, P32, P33, P40 to P45, $\overline{\text{RESET}}$, KRREN, EEWE	2.7 V ≤ V_{DD} ≤ 5.5 V	0		0.2V_{DD}	V
			1.8 V ≤ V_{DD} < 2.7 V	0		0.1V_{DD}	V
	V_{IL2}	P00 to P03, P10, P11, P20, P21, P31	2.7 V ≤ V_{DD} ≤ 5.5 V	0		0.3V_{DD}	V
			1.8 V ≤ V_{DD} < 2.7 V	0		0.1V_{DD}	V
	V_{IL3}	X1, X2		0		0.1	V
Output voltage, high	V_{OH1}	I_{OH} = –100 μA		V_{DD} – 0.5			V
	V_{OH2}	I_{OH} = –500 μA		V_{DD} – 0.7			V
Output voltage, low	V_{OL1}	P00 to P03, P10, P11, P22 to P24, P30 to P33, P70	I_{OL} = 400 μA			0.5	V
	V_{OL2}	P00 to P03, P10, P11, P22 to P24, P30 to P33, P70	I_{OL} = 2 mA			0.7	V
	V_{OL3}	P20, P21	I_{OL} = 2.5 mA			0.5	V
Input leakage current, high	I_{LIH1}	P00 to P03, P10, P11, P20 to P24, P30 to P33, P40 to P45, $\overline{\text{RESET}}$, KRREN	V_I = V_{DD}			3	μA
	I_{LIH2}	X1, X2	V_I = V_{DD}			20	μA
Input leakage current, low	I_{LIL1}	P00 to P03, P10, P11, P20 to P24, P30 to P33, KRREN, EEWE	V_I = 0 V			–3	μA
	I_{LIL2}	X1, X2	V_I = 0 V			–20	μA
Pull-up resistance	R1	Other than P20, P21	V_I = 0 V	50	100	200	kΩ
	R3	P20, P21	V_I = 0 V	10	30	60	kΩ
Pull-down resistance	R2	EEWE	V_I = V_{DD}	2	5	10	kΩ

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V) (2/2)

Parameter		Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Power supply current[Note 1]		I_{DD1}	5.0 MHz oscillation operating mode[Note 2] (C1 = C2 = 22 pF)	EEPROM operating	V_{DD} = 5.0 V ±10%		2.7	5.4	mA
					V_{DD} = 3.0 V ±10%		1.2	2.4	mA
		I_{DD2}		EEPROM stopped	V_{DD} = 5.0 V ±10%		2.3	4.6	mA
					V_{DD} = 3.0 V ±10%		1.1	2.2	mA
		I_{DD3}	5.0 MHz oscillation HALT mode[Note 2] (C1 = C2 = 22 pF)	EEPROM stopped	V_{DD} = 5.0 V ±10%		1.0	2.0	mA
					V_{DD} = 3.0 V ±10%		0.5	1.0	mA
		I_{DD4}	STOP mode	EEPROM stopped	V_{DD} = 5.0 V ±10%		1.0	14.0	μA
				POC operating	V_{DD} = 3.0 V ±10%		0.9	7.0	μA
		I_{DD5}		EEPROM stopped	V_{DD} = 5.0 V ±10%		0.1	10.0	μA
				POC stopped	V_{DD} = 3.0 V ±10%		0.05	5.0	μA
POC	Detection voltage	V_{POC}	Response time 2 ms			1.8	1.9	2.0	V
LVI	LVI7 detection voltage	V_{LVI7}	Response time 2 ms			2.4	2.6	2.8	V
	LVI6 detection voltage	V_{LVI6}	Response time 2 ms				2.5		V
	LVI5 detection voltage	V_{LVI5}	Response time 2 ms				2.4		V
	LVI4 detection voltage	V_{LVI4}	Response time 2 ms				2.3		V
	LVI3 detection voltage	V_{LVI3}	Response time 2 ms				2.2		V
	LVI2 detection voltage	V_{LVI2}	Response time 2 ms				2.1		V
	LVI1 detection voltage	V_{LVI1}	Response time 2 ms				2.0		V
	LVI0 detection voltage	V_{LVI0}	Response time 2 ms			1.7	1.9	2.1	V

Notes 1. Port current (including the current flowing through the internal pull-up resistors) is not included.

2. During high-speed mode operation (when the processor clock control register (PCC) is set to 00H)

Remark $V_{LVI7} > V_{LVI6} > V_{LVI5} > V_{LVI4} > V_{LVI3} > V_{LVI2} > V_{LVI1} > V_{LVI0}$

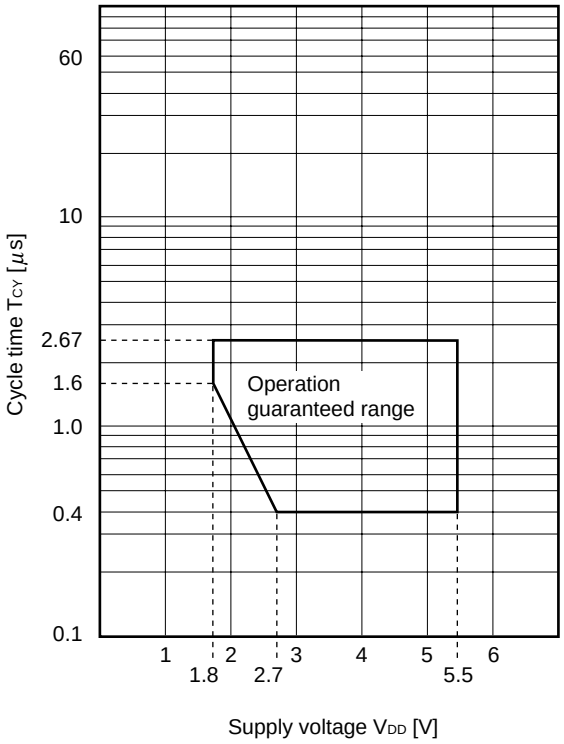
AC Characteristics

(1) Basic operation (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Instruction cycle (Minimum instruction execution time)	T_{CY}	2.7 V ≤ V_{DD} ≤ 5.5 V	0.4		2.67	μs
		1.8 V ≤ V_{DD} < 2.7 V	1.6		2.67	μs
TI00, TI01 input high-/low-level width	t_{TI0H}, t_{TI0L}	2.7 V ≤ V_{DD} ≤ 5.5 V	$2/f_{sam}$ + 0.2			μs
		1.8 V ≤ V_{DD} < 2.7 V	$2/f_{sam}$ + 0.4			μs
TI50 input high-/low-level width	t_{TIH}, t_{TIL}	2.7 V ≤ V_{DD} ≤ 5.5 V	100			ns
		1.8 V ≤ V_{DD} < 2.7 V	400			ns
Key input pin high-level width	t_{KRIH}	KR5	10			μs
Key input pin low-level width	t_{KRIL}	KR0 to KR5	10			μs
Interrupt input high-/low-level width	t_{INTH}, t_{INTL}	INTP0, INTP1	10			μs
$\overline{\text{RESET}}$ input low-level width	t_{RST}		10			μs

Remark f_{sam}: Timer count clock (Note that when selecting the TI00 and TI01 valid edges as the timer count clock, $f_{sam} = f_X/8$ (refer to **Figure 7-1 Block Diagram of 16-Bit Timer/Event Counter 0**))

T_{CY} vs. V_{DD} (main system clock)



(2) Serial interface 2 (V_{DD} = 1.8 to 5.5 V, T_A = –40 to +85°C)

(a) 3-wire serial I/O mode ($\overline{SCK2}$...Internal clock)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{SCK2}$ cycle time	t_{KCY1}	2.7 V ≤ V_{DD} ≤ 5.5 V		800			ns
		1.8 V ≤ V_{DD} < 2.7 V		3200			ns
$\overline{SCK2}$ high-/low-level width	t_{KH1}, t_{KL1}	2.7 V ≤ V_{DD} ≤ 5.5 V		t_{KCY1}/2 – 50			ns
		1.8 V ≤ V_{DD} < 2.7 V		t_{KCY1}/2 – 150			ns
SI2 setup time (to $\overline{SCK2}$↑)	t_{SIK1}	2.7 V ≤ V_{DD} ≤ 5.5 V		150			ns
		1.8 V ≤ V_{DD} < 2.7 V		500			ns
SI2 hold time (from $\overline{SCK2}$↑)	t_{KSI1}	2.7 V ≤ V_{DD} ≤ 5.5 V		400			ns
		1.8 V ≤ V_{DD} < 2.7 V		600			ns
Delay time from $\overline{SCK2}$↓ to SO2 output	t_{KSO1}	R = 1 kΩ, C = 100 pF[Note]	2.7 V ≤ V_{DD} ≤ 5.5 V	0		250	ns
			1.8 V ≤ V_{DD} < 2.7 V	0		1000	ns

Note R and C are the load resistance and load capacitance of the SO2 output line.

(b) 3-wire serial I/O mode ($\overline{SCK2}$...External clock)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{SCK2}$ cycle time	t_{KCY2}	2.7 V ≤ V_{DD} ≤ 5.5 V		800			ns
		1.8 V ≤ V_{DD} < 2.7 V		3200			ns
$\overline{SCK2}$ high-/low-level width	t_{KH2}, t_{KL2}	2.7 V ≤ V_{DD} ≤ 5.5 V		400			ns
		1.8 V ≤ V_{DD} < 2.7 V		1600			ns
SI2 setup time (to $\overline{SCK2}$↑)	t_{SIK2}	2.7 V ≤ V_{DD} ≤ 5.5 V		100			ns
		1.8 V ≤ V_{DD} < 2.7 V		150			ns
SI2 hold time (from $\overline{SCK2}$↑)	t_{KSI2}	2.7 V ≤ V_{DD} ≤ 5.5 V		400			ns
		1.8 V ≤ V_{DD} < 2.7 V		600			ns
Delay time from $\overline{SCK2}$↓ to SO2 output	t_{KSO2}	R = 1 kΩ, C = 100 pF[Note]	2.7 V ≤ V_{DD} ≤ 5.5 V	0		300	ns
			1.8 V ≤ V_{DD} < 2.7 V	0		1000	ns

Note R and C are the load resistance and load capacitance of the SO2 output line.

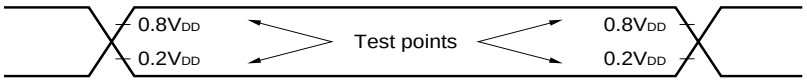
(c) UART mode (dedicated baud rate generator output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		2.7 V ≤ V_{DD} ≤ 5.5 V			78125	bps
		1.8 V ≤ V_{DD} < 2.7 V			19531	bps

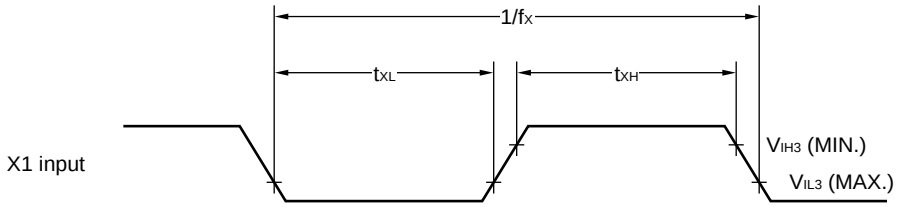
(d) UART mode (external clock input)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
ASCK2 cycle time	t_{KCY3}	2.7 V ≤ V_{DD} ≤ 5.5 V	800			ns
		1.8 V ≤ V_{DD} < 2.7 V	3200			ns
ASCK2 high-/low-level width	t_{KH3}, t_{KL3}	2.7 V ≤ V_{DD} ≤ 5.5 V	400			ns
		1.8 V ≤ V_{DD} < 2.7 V	1600			ns
Transfer rate		2.7 V ≤ V_{DD} ≤ 5.5 V			39063	bps
		1.8 V ≤ V_{DD} < 2.7 V			9766	bps
ASCK2 rise time, fall time	t_R, t_F				1	μs

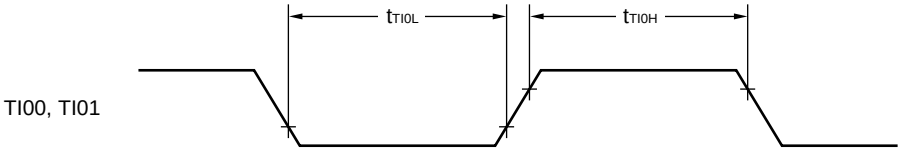
AC Timing Test Points (Excluding X1 Input)

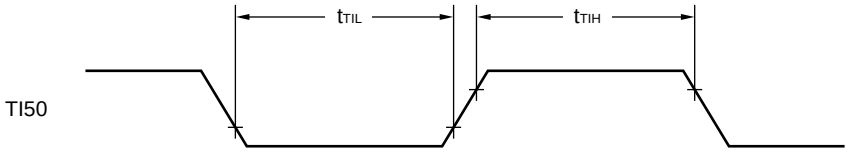


Clock Timing

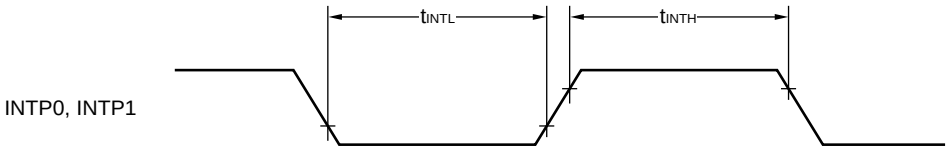


TI Timing

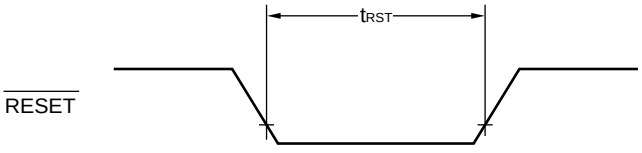




Interrupt Input Timing

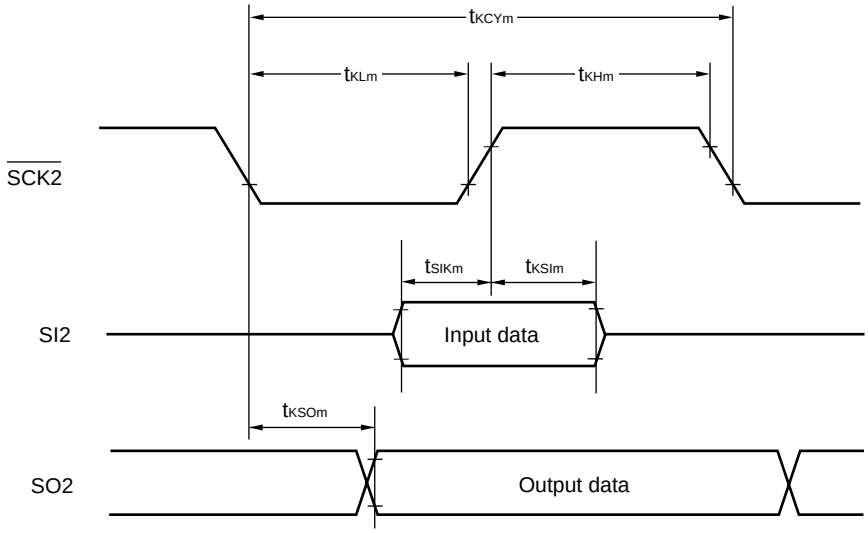


$\overline{\text{RESET}}$ Input Timing



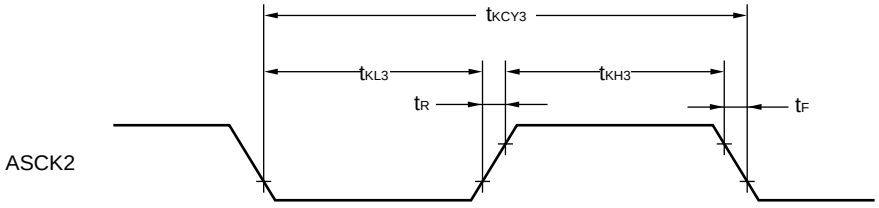
Serial Transfer Timing

3-wire serial I/O mode:



Remark m = 1, 2

UART mode (external clock input):



(3) EEPROM (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Write time[Note]			3.3		6.6	ms
Rewrite count		Per byte			100000	Times

Note Write time = T × 145 (T: Time for the cycle of the clock selected by EWCS10 to EWCS12)

(4) Power supply startup (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Power supply startup time 1	t_{PTH1}	POC switching circuit used (V_{DD}: 0 V → 1.8 V)	0.01		100	ms
Power supply startup time 2	t_{PTH2}	POC always used (V_{DD}: 0 V → 1.8 V)	0.01		100	ms

Memory Storage Data Retention Characteristics (T_A = –40 to +85°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention power supply voltage	V_{DDDR}		1.8		5.5	V
Release signal set time	t_{SREL}		0			μs

Oscillation Wait Time (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

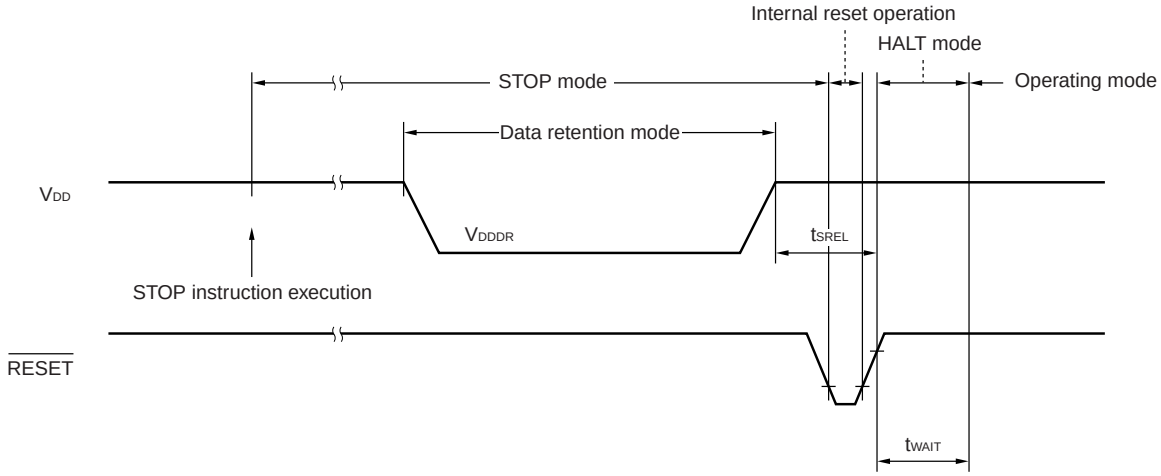
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Oscillation stabilization wait time[Note 1]	t_{WAIT}	Release by $\overline{\text{RESET}}$ pin or POC		$2^{12}/f_X$		s
		Release by interrupt		**Note 2**		s

Notes 1. The oscillation stabilization wait time is the time the CPU operation is stopped to prevent unstable operation when oscillation starts.

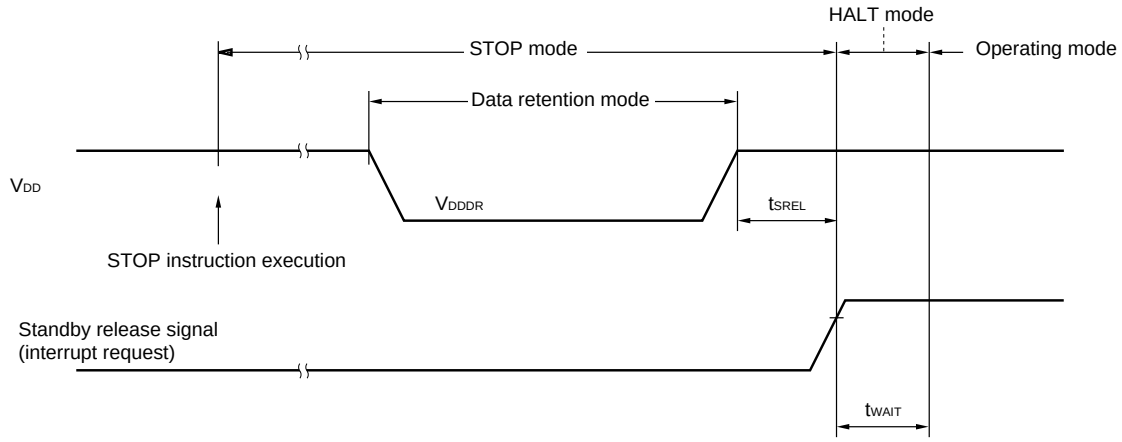
2. $2^{12}/f_X$, $2^{15}/f_X$, or $2^{17}/f_X$ can be selected by using bits 0 to 2 (OSTS0 to OSTS2) of the oscillation stabilization time selection register (OSTS).

Remark f_X: System clock oscillation frequency

Data Retention Timing (STOP Mode Release by $\overline{\text{RESET}}$)



Data Retention Timing (Standby Release Signal: STOP Mode Release by Interrupt Signal)



CHAPTER 22 ELECTRICAL SPECIFICATIONS (μPD78E9862)

★ **Caution The μPD78E9862 is provided exclusively for program development; the product life and reliability are not guaranteed. Use the μPD789862 for reliability testing and mass production.**

Absolute Maximum Ratings (T_A = 25°C)

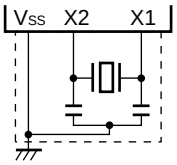
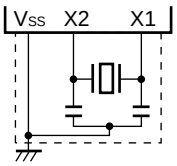
Parameter	Symbol	Conditions	Ratings	Unit
Supply voltage	V_{DD}		–0.3 to +6.5	V
	V_{PP}		–0.3 to +10.5	V
Input voltage	V_I		–0.3 to V_{DD} + 0.3[Note]	V
Output voltage	V_O		–0.3 to V_{DD} + 0.3[Note]	V
Output current, high	I_{OH}	Per pin	–10	mA
		Total for all pins	–30	mA
Output current, low	I_{OL}	Per pin	30	mA
		Total for all pins	80	mA
Operating ambient temperature	T_A		–40 to +70	°C
Storage temperature	T_{stg}		–40 to +125	°C

Note 6.5 V or less

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

System Clock Oscillator Characteristics (T_A = –40 to +70°C, V_{DD} = 1.8 to 5.5 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic resonator		Oscillation frequency (f_X)Note 1	V_{DD} = oscillation voltage range	3		5	MHz
		Oscillation stabilization timeNote 2	After V_{DD} reaches oscillation voltage range MIN.		80		μs
Crystal resonator		Oscillation frequency (f_X)Note 1	V_{DD} = oscillation voltage range	3		5	MHz
		Oscillation stabilization timeNote 2	After V_{DD} reaches oscillation voltage range MIN.		8		ms

Notes 1. Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

2. Time required to stabilize oscillation after reset or STOP mode release. Use the resonator that stabilizes oscillation within the oscillation wait time.

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

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

Recommended Oscillator Constant

Ceramic resonator (T_A = –40 to +70°C)

Manufacturer	Part Number	Frequency (MHz)	Recommended Circuit Constant (pF)		Oscillation Voltage Range (V_{DD})		Remarks
			C1	C2	MIN.	MAX.	
Murata Mfg. Co., Ltd.	CSTCR4M00G53-R0	4.000	–	–	1.8	5.5	On-chip capacitor
	CSTLS4M00G53-B0						
	CSTCR4M19G53-R0	4.195					
	CSTLS4M19G53-B0						
	CSTCR5M00G53-R0	5.000					
	CSTLS5M00G53-B0						

Caution The oscillator constant is a reference value based on evaluation in specific environments by the resonator manufacturer. If the oscillator characteristics need to be optimized in the actual application, request the resonator manufacturer for evaluation on the implementation circuit. Note that the oscillation voltage and oscillation frequency merely indicate the characteristics of the oscillator. Use the internal operation conditions of the μPD78E9862 within the specifications of the DC and AC characteristics.

DC Characteristics (T_A = –40 to +70°C, V_{DD} = 1.8 to 5.5 V) (1/2)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, low	I_{OL}	Per pin				2.5	mA
		Total for all pins				80.0	mA
Output current, high	I_{OH}	Per pin				–0.5	mA
		Total for all pins				–15.0	mA
Input voltage, high	V_{IH1}	P22 to P24, P30, P32, P33, P40 to P45, $\overline{\text{RESET}}$, KRREN	2.7 V ≤ V_{DD} ≤ 5.5 V	0.8V_{DD}		V_{DD}	V
			1.8 V ≤ V_{DD} < 2.7 V	0.9V_{DD}		V_{DD}	V
	V_{IH2}	P00 to P03, P10, P11, P20, P21, P31	2.7 V ≤ V_{DD} ≤ 5.5 V	0.7V_{DD}		V_{DD}	V
			1.8 V ≤ V_{DD} < 2.7 V	0.9V_{DD}		V_{DD}	V
	V_{IH3}	X1, X2		V_{DD} – 0.1		V_{DD}	V
	V_{IH4}	EEWE		0.9V_{DD}		V_{DD}	V
Input voltage, low	V_{IL1}	P22 to P24, P30, P32, P33, P40 to P45, $\overline{\text{RESET}}$, KRREN, EEWE	2.7 V ≤ V_{DD} ≤ 5.5 V	0		0.2V_{DD}	V
			1.8 V ≤ V_{DD} < 2.7 V	0		0.1V_{DD}	V
	V_{IL2}	P00 to P03, P10, P11, P20, P21, P31	2.7 V ≤ V_{DD} ≤ 5.5 V	0		0.3V_{DD}	V
			1.8 V ≤ V_{DD} < 2.7 V	0		0.1V_{DD}	V
	V_{IL3}	X1, X2		0		0.1	V
Output voltage, high	V_{OH1}	I_{OH} = –100 μA		V_{DD} – 0.5			V
	V_{OH2}	I_{OH} = –500 μA		V_{DD} – 0.7			V
Output voltage, low	V_{OL1}	P00 to P03, P10, P11, P22 to P24, P30 to P33, P70	I_{OL} = 400 μA			0.5	V
	V_{OL2}	P00 to P03, P10, P11, P22 to P24, P30 to P33, P70	I_{OL} = 2 mA			0.7	V
	V_{OL3}	P20, P21	I_{OL} = 2.5 mA			0.5	V
Input leakage current, high	I_{LIH1}	P00 to P03, P10, P11, P20 to P24, P30 to P33, P40 to P45, $\overline{\text{RESET}}$, KRREN	V_I = V_{DD}			3	μA
	I_{LIH2}	X1, X2	V_I = V_{DD}			20	μA
Input leakage current, low	I_{LIL1}	P00 to P03, P10, P11, P20 to P24, P30 to P33, KRREN, EEWE	V_I = 0 V			–3	μA
	I_{LIL2}	X1, X2	V_I = 0 V			–20	μA
Pull-up resistance	R1	Other than P20, P21	V_I = 0 V	50	100	200	kΩ
	R3	P20, P21	V_I = 0 V	10	30	60	kΩ
Pull-down resistance	R2	EEWE	V_I = V_{DD}	2	5	10	kΩ

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (T_A = –40 to +70°C, V_{DD} = 1.8 to 5.5 V) (2/2)

Parameter		Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Power supply current[Note 1]		I_{DD1}	5.0 MHz oscillation operating mode[Note 2] (C1 = C2 = 22 pF)	EEPROM operating	V_{DD} = 5.0 V ±10%		3.1	6.2	mA
					V_{DD} = 3.0 V ±10%		1.5	3.0	mA
		I_{DD2}		EEPROM stopped	V_{DD} = 5.0 V ±10%		2.9	5.8	mA
					V_{DD} = 3.0 V ±10%		1.4	2.8	mA
		I_{DD3}	5.0 MHz oscillation HALT mode[Note 2] (C1 = C2 = 22 pF)	EEPROM stopped	V_{DD} = 5.0 V ±10%		1.6	3.2	mA
					V_{DD} = 3.0 V ±10%		0.7	1.4	mA
		I_{DD4}	STOP mode	EEPROM stopped	V_{DD} = 5.0 V ±10%		1.0	26.0	μA
				POC operating	V_{DD} = 3.0 V ±10%		0.9	13.0	μA
		I_{DD5}		EEPROM stopped	V_{DD} = 5.0 V ±10%		0.1	22.0	μA
				POC stopped	V_{DD} = 3.0 V ±10%		0.05	11.0	μA
POC	Detection voltage	V_{POC}	Response time 2 ms			1.8	1.9	2.0	V
LVI	LVI7 detection voltage	V_{LVI7}	Response time 2 ms			2.4	2.6	2.8	V
	LVI6 detection voltage	V_{LVI6}	Response time 2 ms				2.5		V
	LVI5 detection voltage	V_{LVI5}	Response time 2 ms				2.4		V
	LVI4 detection voltage	V_{LVI4}	Response time 2 ms				2.3		V
	LVI3 detection voltage	V_{LVI3}	Response time 2 ms				2.2		V
	LVI2 detection voltage	V_{LVI2}	Response time 2 ms				2.1		V
	LVI1 detection voltage	V_{LVI1}	Response time 2 ms				2.0		V
	LVI0 detection voltage	V_{LVI0}	Response time 2 ms			1.7	1.9	2.1	V

Notes 1. Port current (including the current flowing through the internal pull-up resistors) is not included.

2. During high-speed mode operation (when the processor clock control register (PCC) is set to 00H)

Remark $V_{LVI7} > V_{LVI6} > V_{LVI5} > V_{LVI4} > V_{LVI3} > V_{LVI2} > V_{LVI1} > V_{LVI0}$

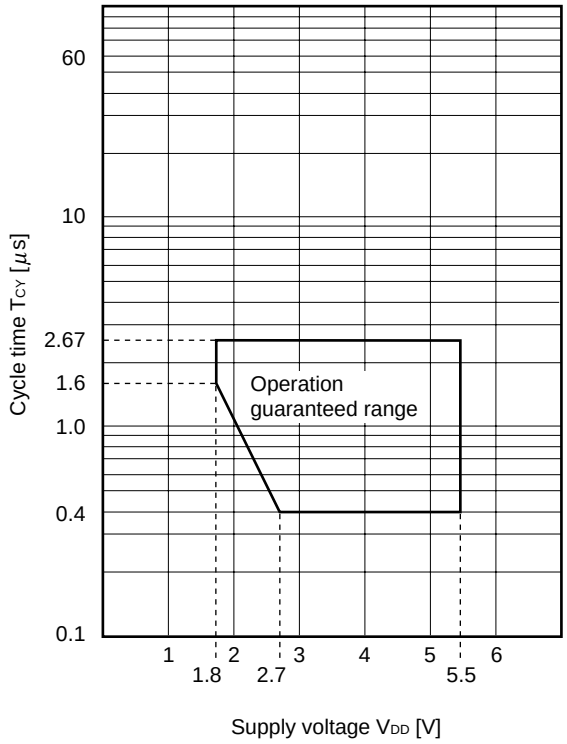
AC Characteristics

(1) Basic operation (T_A = –40 to +70°C, V_{DD} = 1.8 to 5.5 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Instruction cycle (Minimum instruction execution time)	T_{CY}	2.7 V ≤ V_{DD} ≤ 5.5 V	0.4		2.67	μs
		1.8 V ≤ V_{DD} < 2.7 V	1.6		2.67	μs
TI00, TI01 input high-/low-level width	t_{TI0H}, t_{TI0L}	2.7 V ≤ V_{DD} ≤ 5.5 V	$2/f_{sam}$ + 0.2			μs
		1.8 V ≤ V_{DD} < 2.7 V	$2/f_{sam}$ + 0.4			μs
TI50 input high-/low-level width	t_{TIH}, t_{TIL}	2.7 V ≤ V_{DD} ≤ 5.5 V	100			ns
		1.8 V ≤ V_{DD} < 2.7 V	400			ns
Key input pin high-level width	t_{KRIH}	KR5	10			μs
Key input pin low-level width	t_{KRIL}	KR0 to KR5	10			μs
Interrupt input high-/low-level width	t_{INTH}, t_{INTL}	INTP0, INTP1	10			μs
$\overline{\text{RESET}}$ input low-level width	t_{RST}		10			μs

Remark f_{sam}: Timer count clock (Note that when selecting the TI00 and TI01 valid edges as the timer count clock, $f_{sam} = f_X/8$ (refer to **Figure 7-1 Block Diagram of 16-Bit Timer/Event Counter 0**))

T_{CY} vs. V_{DD} (main system clock)



(2) Serial interface 2 (V_{DD} = 1.8 to 5.5 V, T_A = –40 to +70°C)

(a) 3-wire serial I/O mode ($\overline{SCK2}$...Internal clock)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{SCK2}$ cycle time	t_{KCY1}	2.7 V ≤ V_{DD} ≤ 5.5 V		800			ns
		1.8 V ≤ V_{DD} < 2.7 V		3200			ns
$\overline{SCK2}$ high-/low-level width	t_{KH1}, t_{KL1}	2.7 V ≤ V_{DD} ≤ 5.5 V		t_{KCY1}/2 – 50			ns
		1.8 V ≤ V_{DD} < 2.7 V		t_{KCY1}/2 – 150			ns
SI2 setup time (to $\overline{SCK2}$↑)	t_{SIK1}	2.7 V ≤ V_{DD} ≤ 5.5 V		150			ns
		1.8 V ≤ V_{DD} < 2.7 V		500			ns
SI2 hold time (from $\overline{SCK2}$↑)	t_{KSI1}	2.7 V ≤ V_{DD} ≤ 5.5 V		400			ns
		1.8 V ≤ V_{DD} < 2.7 V		600			ns
Delay time from $\overline{SCK2}$↓ to SO2 output	t_{KSO1}	R = 1 kΩ, C = 100 pF[Note]	2.7 V ≤ V_{DD} ≤ 5.5 V	0		250	ns
			1.8 V ≤ V_{DD} < 2.7 V	0		1000	ns

Note R and C are the load resistance and load capacitance of the SO2 output line.

(b) 3-wire serial I/O mode ($\overline{SCK2}$...External clock)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{SCK2}$ cycle time	t_{KCY2}	2.7 V ≤ V_{DD} ≤ 5.5 V		800			ns
		1.8 V ≤ V_{DD} < 2.7 V		3200			ns
$\overline{SCK2}$ high-/low-level width	t_{KH2}, t_{KL2}	2.7 V ≤ V_{DD} ≤ 5.5 V		400			ns
		1.8 V ≤ V_{DD} < 2.7 V		1600			ns
SI2 setup time (to $\overline{SCK2}$↑)	t_{SIK2}	2.7 V ≤ V_{DD} ≤ 5.5 V		100			ns
		1.8 V ≤ V_{DD} < 2.7 V		150			ns
SI2 hold time (from $\overline{SCK2}$↑)	t_{KSI2}	2.7 V ≤ V_{DD} ≤ 5.5 V		400			ns
		1.8 V ≤ V_{DD} < 2.7 V		600			ns
Delay time from $\overline{SCK2}$ ↓ to SO2 output	t_{KSO2}	R = 1 kΩ, C = 100 pF[Note]	2.7 V ≤ V_{DD} ≤ 5.5 V	0		300	ns
			1.8 V ≤ V_{DD} < 2.7 V	0		1000	ns

Note R and C are the load resistance and load capacitance of the SO2 output line.

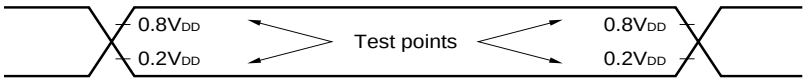
(c) UART mode (dedicated baud rate generator output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		2.7 V ≤ V_{DD} ≤ 5.5 V			78125	bps
		1.8 V ≤ V_{DD} < 2.7 V			19531	bps

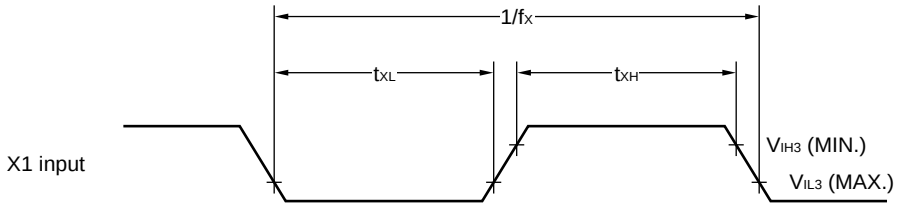
(d) UART mode (external clock input)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
ASCK2 cycle time	t_{KCY3}	$2.7\ V \le V_{DD} \le 5.5\ V$	800			ns
		$1.8\ V \le V_{DD} < 2.7\ V$	3200			ns
ASCK2 high-/low-level width	t_{KH3}, t_{KL3}	$2.7\ V \le V_{DD} \le 5.5\ V$	400			ns
		$1.8\ V \le V_{DD} < 2.7\ V$	1600			ns
Transfer rate		$2.7\ V \le V_{DD} \le 5.5\ V$			39063	bps
		$1.8\ V \le V_{DD} < 2.7\ V$			9766	bps
ASCK2 rise time, fall time	t_R, t_F				1	μs

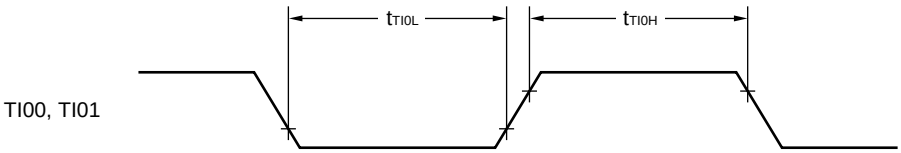
AC Timing Test Points (Excluding X1 Input)

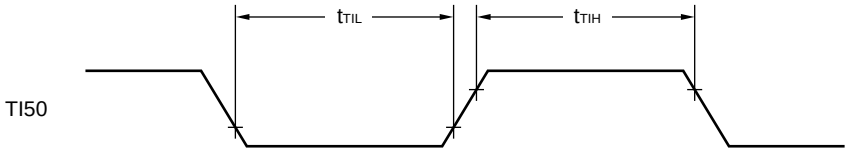


Clock Timing

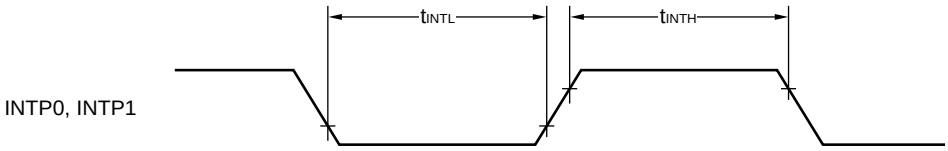


TI Timing

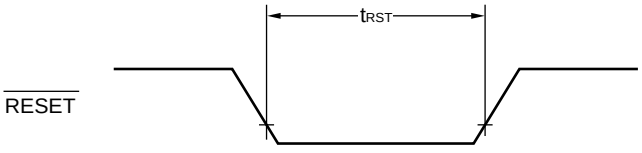




Interrupt Input Timing

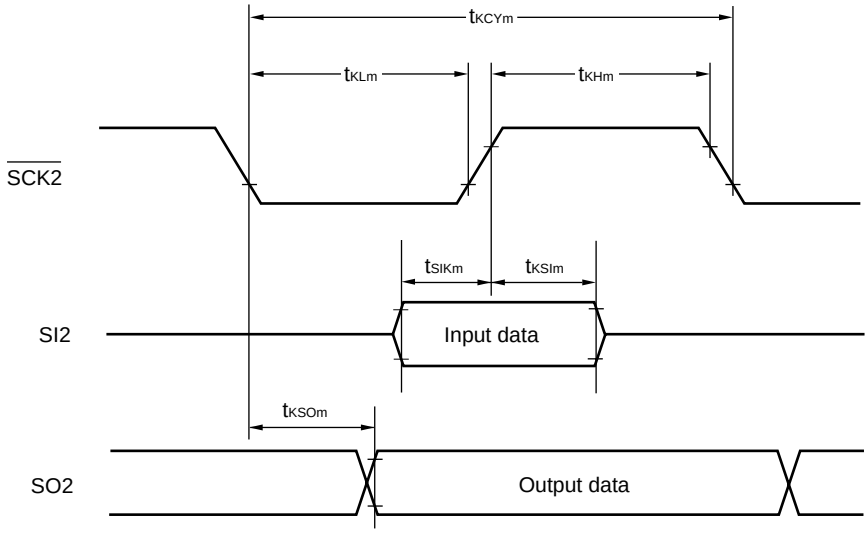


$\overline{\text{RESET}}$ Input Timing



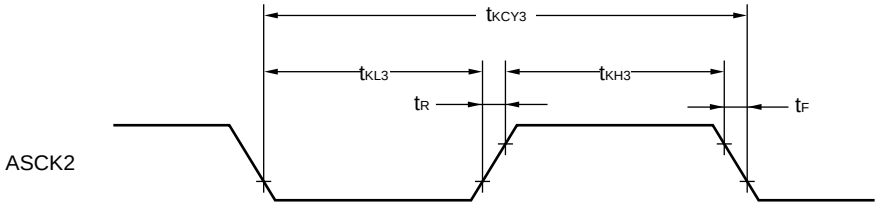
Serial Transfer Timing

3-wire serial I/O mode:



Remark m = 1, 2

UART mode (external clock input):



(3) EEPROM (T_A = –40 to +70°C, V_{DD} = 1.8 to 5.5 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Write time[Note 1]			3.3		6.6	ms
Rewrite count		Per byte			**Note 2**	Times

Notes 1. Write time = T × 145 (T: Time for the cycle of the clock selected by EWCS10 to EWCS12)

★ **2.** The rewrite count of EEPROM is not guaranteed because the μPD78E9862 is used exclusively for program development.

(4) Power supply startup (T_A = –40 to +70°C, V_{DD} = 1.8 to 5.5 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Power supply startup time 1	t_{PTH1}	POC switching circuit used (V_{DD}: 0 V → 1.8 V)	0.01		100	ms
Power supply startup time 2	t_{PTH2}	POC always used (V_{DD}: 0 V → 1.8 V)	0.01		100	ms

Memory Storage Data Retention Characteristics (T_A = –40 to +70°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention power supply voltage	V_{DDDR}		1.8		5.5	V
Release signal set time	t_{SREL}		0			μs

Oscillation Wait Time (T_A = –40 to +70°C, V_{DD} = 1.8 to 5.5 V)

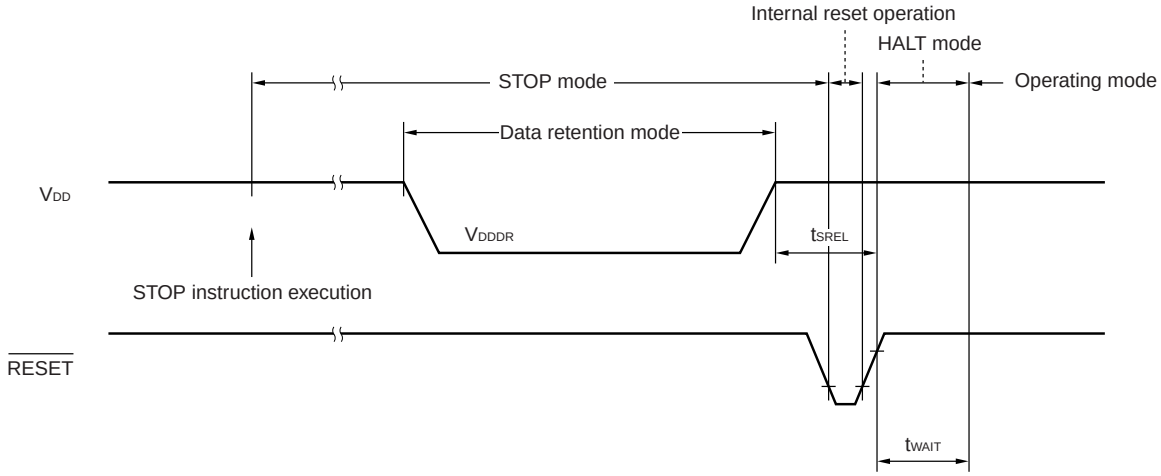
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Oscillation stabilization wait time[Note 1]	t_{WAIT}	Release by $\overline{\text{RESET}}$ pin or POC		$2^{12}/f_X$		s
		Release by interrupt		**Note 2**		s

Notes 1. The oscillation stabilization wait time is the time the CPU operation is stopped to prevent unstable operation when oscillation starts.

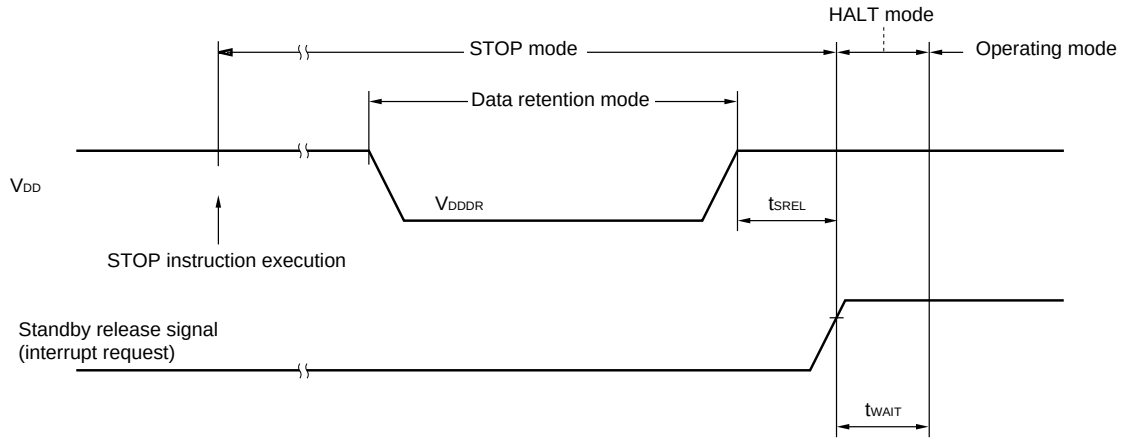
2. $2^{12}/f_X$, $2^{15}/f_X$, or $2^{17}/f_X$ can be selected by using bits 0 to 2 (OSTS0 to OSTS2) of the oscillation stabilization time selection register (OSTS).

Remark f_X: System clock oscillation frequency

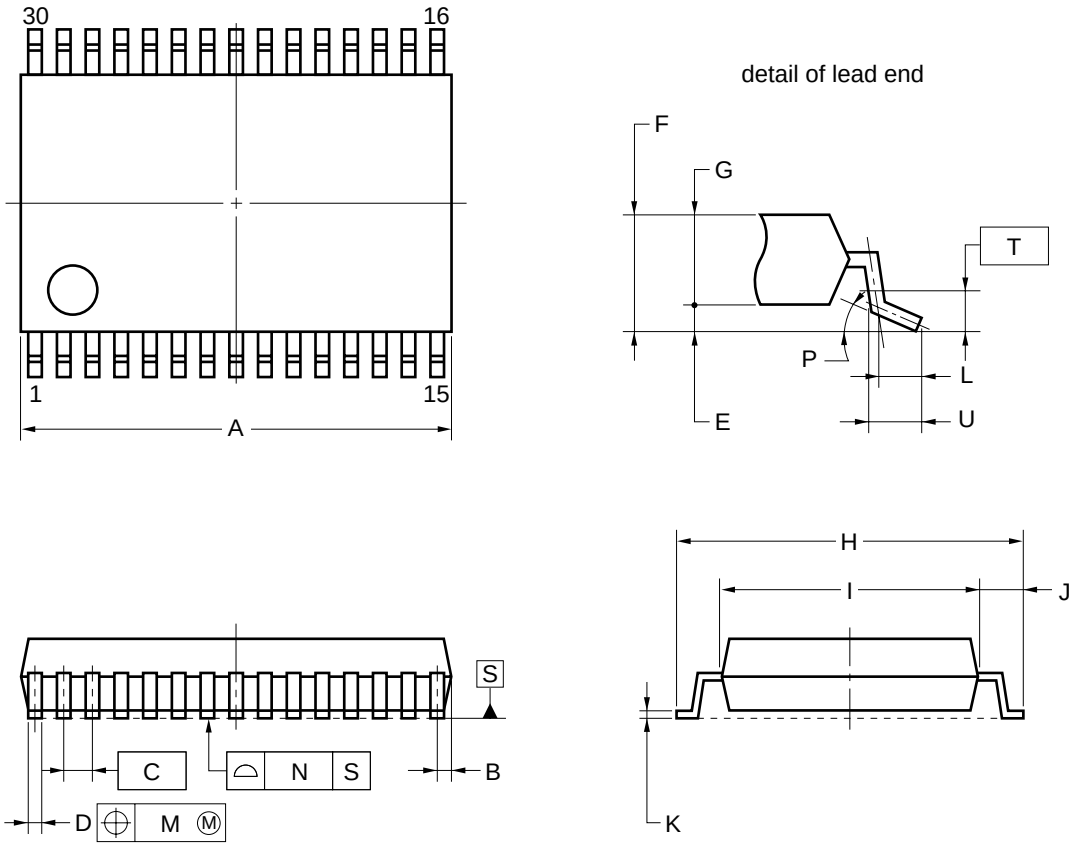
Data Retention Timing (STOP Mode Release by $\overline{\text{RESET}}$)



Data Retention Timing (Standby Release Signal: STOP Mode Release by Interrupt Signal)



30-PIN PLASTIC SSOP (7.62 mm (300))



NOTE

Each lead centerline is located within 0.13 mm of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS
A	9.85±0.15
B	0.45 MAX.
C	0.65 (T.P.)
D	$0.24^{+0.08}_{-0.07}$
E	0.1±0.05
F	1.3±0.1
G	1.2
H	8.1±0.2
I	6.1±0.2
J	1.0±0.2
K	0.17±0.03
L	0.5
M	0.13
N	0.10
P	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
T	0.25
U	0.6±0.15

S30MC-65-5A4-2

★ CHAPTER 24 RECOMMENDED SOLDERING CONDITIONS

The μPD789862 Subseries should be soldered and mounted under the following recommended conditions.

For soldering methods and conditions other than those recommended below, contact an NEC Electronics sales representative.

For technical information, see the following website.

Semiconductor Device Mount Manual (http://www.necel.com/pkg/en/mount/index.html)

Table 24-1. Surface Mounting Type Soldering Conditions (1/2)

(1) μPD789862MC-×××-5A4: 30-pin plastic SSOP (7.62 mm (300))

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 235°C, Time: 30 seconds max. (at 210°C or higher), Count: Twice or less, Exposure limit: 7 days[Note] (after that, prebake at 125°C for 20 hours)	IR35-207-2
VPS	Package peak temperature: 215°C, Time: 40 seconds max. (at 200°C or higher), Count: Twice or less, Exposure limit: 7 days[Note] (after that, prebake at 125°C for 20 hours)	VP15-207-2
Wave soldering	Solder bath temperature: 260°C max., Time: 10 seconds max., Count: Once, Preheating temperature: 120°C max. (package surface temperature), Exposure limit: 7 days[Note] (after that, prebake at 125°C for 20 hours)	WS60-207-1
Partial heating	Pin temperature: 300°C max., Time: 3 seconds max. (per pin row)	–

Note After opening the dry pack, store it at 25°C or less and 65% RH or less for the allowable storage period.

Caution Do not use different soldering methods together (except for partial heating).

Table 24-1. Surface Mounting Type Soldering Conditions (2/2)

(2) μPD78E9862MC-5A4: 30-pin plastic SSOP (7.62 mm (300))

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 235°C, Time: 30 seconds max. (at 210°C or higher), Count: Three times or less, Exposure limit: 7 days[Note] (after that, prebake at 125°C for 10 hours)	IR35-107-3
VPS	Package peak temperature: 215°C, Time: 40 seconds max. (at 200°C or higher), Count: Three times or less, Exposure limit: 7 days[Note] (after that, prebake at 125°C for 10 hours)	VP15-107-3
Wave soldering	Solder bath temperature: 260°C max., Time: 10 seconds max., Count: Once, Preheating temperature: 120°C max. (package surface temperature), Exposure limit: 7 days[Note] (after that, prebake at 125°C for 10 hours)	WS60-107-1
Partial heating	Pin temperature: 300°C max., Time: 3 seconds max. (per pin row)	–

Note After opening the dry pack, store it at 25°C or less and 65% RH or less for the allowable storage period.

Caution Do not use different soldering methods together (except for partial heating).

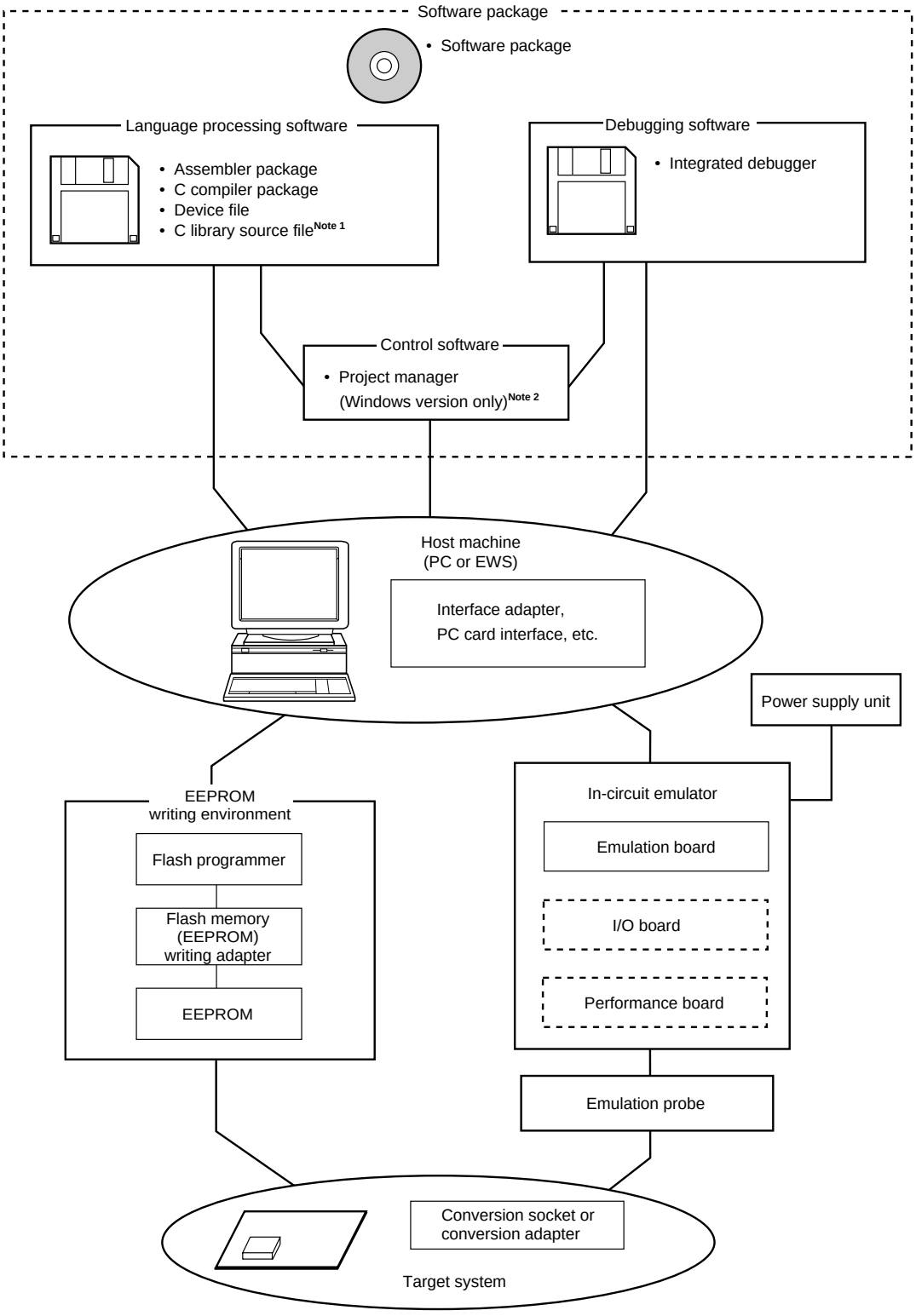
APPENDIX A DEVELOPMENT TOOLS

The following development tools are available for development of systems using the μPD789862 Subseries. Figure A-1 shows development tools.

- Compatibility with PC98-NX Series
  Unless stated otherwise, products which are supported for IBM PC/AT™ and compatibles can also be used with the PC98-NX Series. When using the PC98-NX Series, therefore, refer to the explanations for IBM PC/AT and compatibles.

- Windows
  Unless stated otherwise, “Windows” refers to the following operating systems.
  - Windows 3.1
  - Windows 95
  - Windows 98
  - Windows 2000
  - Windows NT™ Ver. 4.0

Figure A-1. Development Tools



Notes 1. The C library source file is not included in the software package.

2. The project manager is included in the assembler package.
The project manager is not used in environments other than Windows.

A.1 Software Package

SP78K0S Software package	Various software tools for 78K/0S Series development are integrated in one package. The following tools are included. RA78K0S, CC78K0S, ID78K0S-NS, various device files
	Part number: μS××××SP78K0S

Remark ×××× in the part number differs depending on the operating system to be used.

μS×××× SP78K0S

××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT and compatibles	Japanese Windows	CD-ROM
BB17		English Windows	

A.2 Language Processing Software

RA78K0S Assembler package	Program that converts program written in mnemonic into object code that can be executed by microcontroller. In addition, automatic functions to generate symbol table and optimize branch instructions are also provided. Used in combination with device file (DF789862) (sold separately). **<Caution when used in PC environment>** The assembler package is a DOS-based application but may be used in the Windows environment by using Project Manager of Windows (included in the assembler package).
	Part number: μS××××RA78K0S
CC78K0S C compiler package	Program that converts program written in C language into object codes that can be executed by microcontroller. Used in combination with assembler package (RA78K0S) and device file (DF789862) (both sold separately). **<Caution when used in PC environment>** The C compiler package is a DOS-based application but may be used in the Windows environment by using Project Manager of Windows (included in the assembler package).
	Part number: μS××××CC78K0S
DF789862[Note 1] Device file	File containing the information inherent to the device. Used in combination with other tools (RA78K0S, CC78K0S, ID78K0S-NS) (all sold separately).
	Part number: μS××××DF789862
CC78K0S-L[Note 2] C library source file	Source file of functions constituting object library included in C compiler package. Necessary for changing object library included in C compiler package according to customer's specifications. Since this is the source file, its working environment does not depend on any particular operating system.
	Part number: μS××××CC78K0S-L

Notes 1. DF789862 is a common file that can be used with RA78K0S and CC78K0S.

2. CC78K0S-L is not included in the software package (SP78K0S).

Remark ××××in the part number differs depending on the host machines and operating systems to be used.

μS××××RA78K0S
μS××××CC78K0S

××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT and compatibles	Japanese Windows	3.5" 2HD FD
BB13		English Windows	
AB17		Japanese Windows	CD-ROM
BB17		English Windows	
3P17	HP9000 series 700™	HP-UX™ (Rel.10.10)	
3K17	SPARCstation™	SunOS™ (Rel.4.1.4), Solaris™ (Rel.2.5.1)	

μS××××DF789862
μS××××CC78K0S-L

××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT and compatibles	Japanese Windows	3.5" 2HD FD
BB13		English Windows	
3P16	HP9000 series 700	HP-UX (Rel.10.10)	DAT
3K13	SPARCstation	SunOS (Rel.4.1.4), Solaris (Rel.2.5.1)	3.5" 2HD FD
3K15			1/4" CGMT

A.3 Control Software

Project manager	Control software provided for an efficient user program development in the Windows environment. The project manager allows a series of tasks required for user program development to be performed, including starting the editor, building, and starting the debugger. **<Caution>** The project manager is included in the assembler package (RA78K0S). It cannot be used in an environment other than Windows.

A.4 EEPROM (Program Memory) Writing Tools

Flashpro III (part number: FL-PR3, PG-FP3) Flashpro IV (part number: FL-PR4, PG-FP4) Flash programmer	Flash programmer dedicated to microcontrollers incorporating flash memory (EEPROM)
FA-30MC Flash memory (EEPROM) writing adapter	Flash memory (EEPROM) writing adapter. Used connected to Flashpro III/Flashpro IV.

Remark FL-PR3, FL-PR4, and FA-30MC are products of Naito Densei Machida Mfg. Co., Ltd.
For further information, contact: Naito Densei Machida Mfg. Co., Ltd. (+81-45-475-4191)

A.5 Debugging Tools

A.5.1 Hardware

IE-78K0S-NS In-circuit emulator	In-circuit emulator for debugging hardware and software of application system using 78K/0S Series. Supports integrated debugger (ID78K0S-NS). Used in combination with power supply unit, emulation probe, and interface adapter for connecting the host machine.
IE-78K0S-NS-A In-circuit emulator	In-circuit emulator with expanded functions of IE-78K0S-NS. The debugging function is enhanced by the addition of a coverage function and the tracer function and timer function are also enhanced.
IE-70000-MC-PS-B Power supply unit	Adapter for supplying power from 100 to 240 VAC outlet.
IE-70000-98-IF-C Interface adapter	Adapter required when using a PC-9800 series (except notebook type) as the host machine of IE-78K0S-NS (C bus supported).
IE-70000-CD-IF-A PC card interface	PC card and interface cable required when using a notebook type PC as the host machine of IE-78K0S-NS (PCMCIA socket supported).
IE-70000-PC-IF-C Interface adapter	Adapter required when using IBM PC/AT and compatibles as the host machine of IE-78K0S-NS (ISA bus supported).
IE-70000-PCI-IF-A Interface adapter	Adapter required when using a personal computer incorporating the PCI bus as the host machine of IE-78K0S-NS.
IE-789862-NS-EM1 Emulation board	Emulation board for emulating the peripheral hardware inherent to the device. Used in combination with in-circuit emulator.
NP-30MC Emulation probe	Probe for connecting in-circuit emulator and target system. Used in combination with NSPACK30BK and YSPACK30BK.
NSPACK30BK YSPACK30BK Conversion adapter	Conversion adapter for connecting target system board for mounting 30-pin plastic SSOP and NP-30MC.

Remarks 1. NP-30MC is a product of Naito Densei Machida Mfg. Co., Ltd.
For further information, contact: Naito Densei Machida Mfg. Co., Ltd. (+81-45-475-4191)

2. NSPACK30BK and YSPACK30BK are products of TOKYO ELETECH CORPORATION.
For further information, contact: Daimaru Kogyo, Ltd.
Tokyo Electronics Department (+81-3-3820-7112)
Osaka Electronics Department (+81-6-6244-6672)

A.5.2 Software

ID78K0S-NS Integrated debugger (Supports in-circuit emulator IE-78K0S-NS, IE-78K0S-NS-A)	Control program for debugging the 78K/0S Series. This program provides a graphical user interface. It runs on Windows, and has visual designs and operability that comply with these operating systems. In addition, it has a powerful debug function that supports C language. Therefore, trace results can be displayed at a C language level by the window integration function that links source program, disassembled display, and memory display, to the trace result. This software also allows users to add other function extension modules such as task debugger and system performance analyzer to improve the debug efficiency for programs using a real-time operating system. Used in combination with a device file (DF789862) (sold separately).
	Part number: μS××××ID78K0S-NS

Remark ×××× in the part number differs depending on the host machines and operating systems to be used.

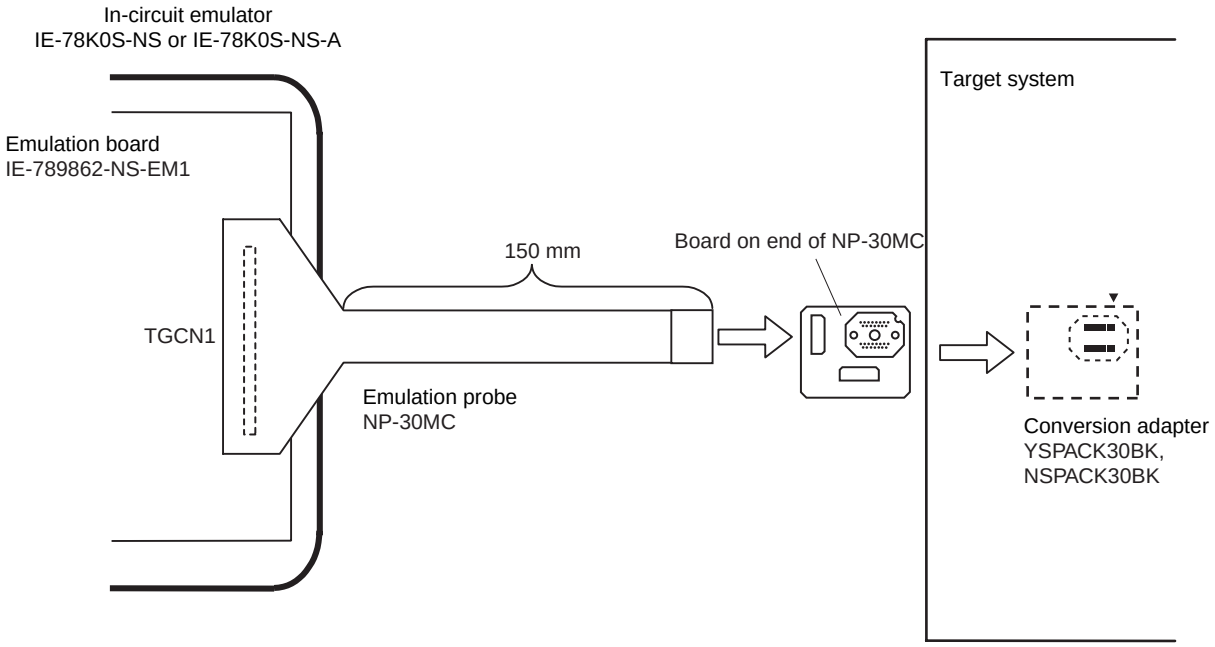
μS××××ID78K0S-NS

××××	Host Machine	OS	Supply Medium
AB13	IBM PC/AT and compatibles	Japanese Windows	3.5" 2HD FD
BB13		English Windows	
AB17		Japanese Windows	CD-ROM
BB17		English Windows	

A.6 Notes on Target System Design

The following show the conditions when connecting the emulation probe to the conversion adapter. Follow the configuration below and consider the shape of parts to be mounted on the target system when designing a system.

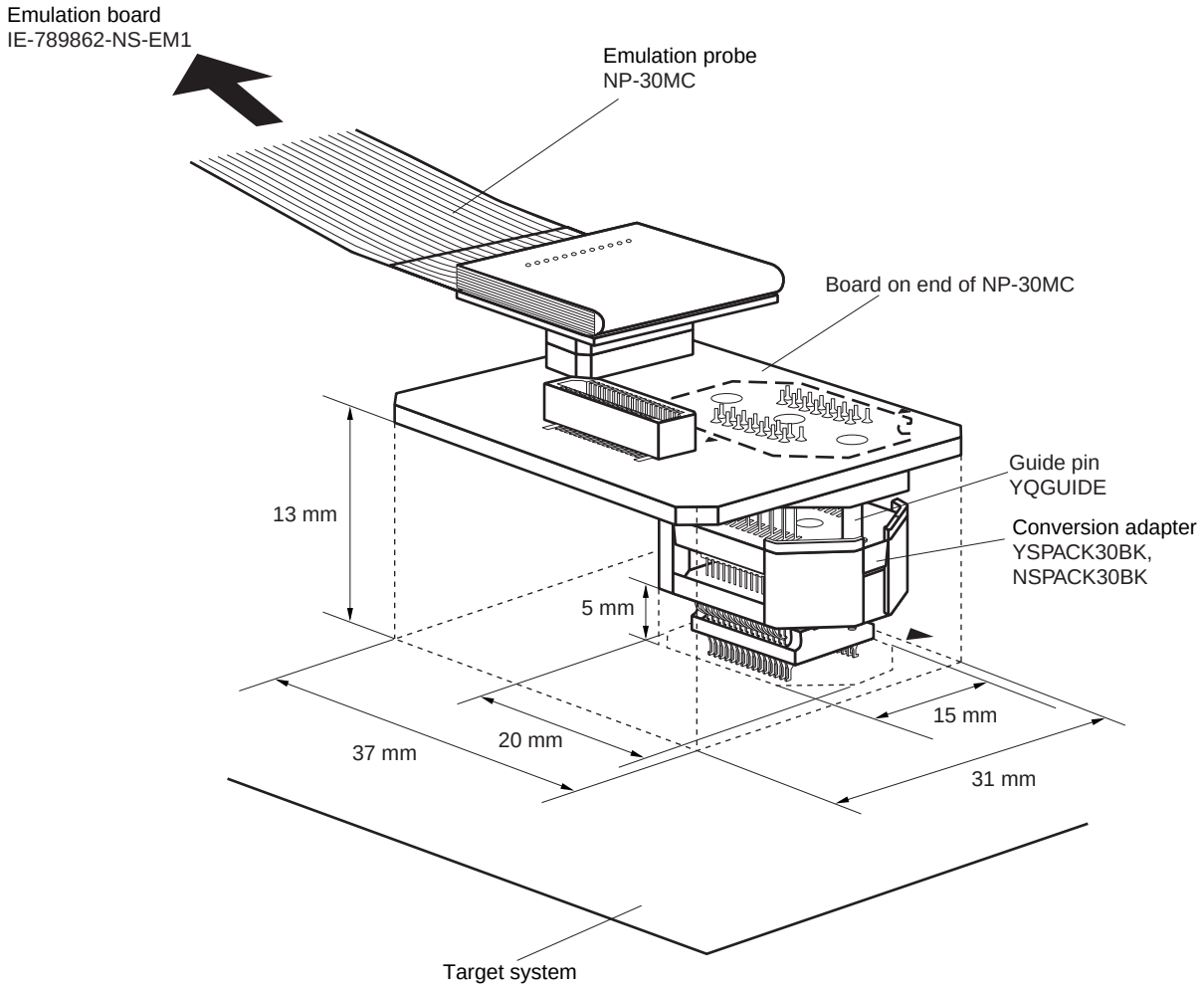
Figure A-2. Distance Between In-Circuit Emulator and Conversion Adapter



Remarks 1. The NP-30MC is a product of Naito Densei Machida Mfg. Co., Ltd.

2. The YSPACK30BK and NSPACK30BK are products of TOKYO ELETECH CORPORATION.

Figure A-3. Connection Condition of Target System



Remarks 1. NP-30MC is a product of Naito Densei Machida Mfg. Co., Ltd.

2. YSPACK30BK, NSPACK30BK, and YQGUIDE are products of TOKYO ELETECH CORPORATION.

APPENDIX B REGISTER INDEX

B.1 Register Name Index (in Alphabetical Order)

8-bit timer compare register 5 (CR5) 127
8-bit timer compare register 80 (CR80) 63
8-bit timer counter 5 (TM5) 127
8-bit timer counter 80 (TM80) 63
8-bit timer mode control register 5 (TMC5) 129
8-bit timer mode control register 80 (TMC80) 64
16-bit compare register D (CMD) 120
16-bit timer capture/compare register 00 (CR00) 92
16-bit timer capture/compare register 01 (CR01) 93
16-bit timer counter 0 (TM0) 92
16-bit timer counter D (TMD) 119
16-bit timer mode control register 0 (TMC0) 94
16-bit timer mode control register D (TMCD) 122
16-bit timer output control register 0 (TOC0) 97

[A]

Asynchronous serial interface mode register 2 (ASIM2) 150, 156, 158, 172
Asynchronous serial interface status register 2 (ASIS2) 152, 159

[B]

Baud rate generator control register 2 (BRGC2) 153, 160, 173
Bit sequential buffer 1 data registers L, H (BSFRL10, BSFRH10) 186
Bit sequential buffer 1 output control register (BSF1C) 187

[C]

Capture/compare control register 0 (CRC0) 96

[E]

EEPROM write control register 10 (EEWC10) 61
External interrupt mode register 0 (INTM0) 200

[I]

Interrupt mask flag registers 0, 1 (MK0, MK1) 199
Interrupt request flag registers 0, 1 (IF0, IF1) 198

[K]

Key return edge detection register (EDG) 193

[L]

Low-voltage detection level selection register (LVIS) 180
Low-voltage detection register (LVIF) 180

[O]

Oscillation stabilization time selection register (OSTS).....210

[P]

Port 0 (P0).....70
Port 1 (P1).....71
Port 2 (P2).....72
Port 3 (P3).....75
Port 4 (P4).....78
Port 7 (P7).....79
Port mode register 0 (PM0).....80
Port mode register 1 (PM1).....80
Port mode register 2 (PM2).....80, 99, 130, 187
Port mode register 3 (PM3).....80, 163, 174
Power-on-clear register (POCF).....179
Prescaler mode register 0 (PRM0).....98
Processor clock control register (PCC).....84
Pull-up resistor option registers 0 to 3 (PUB0 to PUB3).....81

[R]

Receive buffer register 2 (RXB2).....148
Receive shift register 2 (RXS2).....148

[S]

Serial operating mode register 2 (CSIM2).....149, 156, 157, 171

[T]

Timer clock select register 5 (TCL5).....128
Transmit shift register 2 (TXS2).....148

[W]

Watchdog timer clock select register (WDCS).....140
Watchdog timer mode register (WDTM).....141

B.2 Register Symbol Index (in Alphabetical Order)

[A]

ASIM2: Asynchronous serial interface mode register 2 ..... 150, 156, 158, 172
ASIS2: Asynchronous serial interface status register 2 ..... 152, 159

[B]

BRGC2: Baud rate generator control register 2 ..... 153, 160, 173
BSF1C: Bit sequential buffer 1 output control register 187
BSFRL10, BSFRH10: Bit sequential buffer 1 data registers L, H 186

[C]

CMD: 16-bit compare register D 120
CR00: 16-bit timer capture/compare register 00 92
CR01: 16-bit timer capture/compare register 01 93
CR5: 8-bit timer compare register 5 127
CR80: 8-bit timer compare register 80 63
CRC0: Capture/compare control register 0 96
CSIM2: Serial operating mode register 2 ..... 149, 156, 157, 171

[E]

EDG: Key return edge detection register 193
EEWC10: EEPROM write control register 10 61

[I]

IF0, IF1: Interrupt request flag registers 0, 1 198
INTM0: External interrupt mode register 0 200

[L]

LVIF: Low-voltage detection register 180
LVIS: Low-voltage detection level selection register 180

[M]

MK0, MK1: Interrupt mask flag registers 0, 1 199

[O]

OSTS: Oscillation stabilization time selection register 210

[P]

P0: Port 0 70
P1: Port 1 71
P2: Port 2 72
P3: Port 3 75
P4: Port 4 78
P7: Port 7 79
PCC: Processor clock control register 84
PM0: Port mode register 0 80
PM1: Port mode register 1 80

PM2: Port mode register 2 ..... 80, 99, 130, 187
PM3: Port mode register 3 ..... 80, 163, 174
POCF: Power-on-clear register 179
PRM0: Prescaler mode register 0 98
PUB0 to PUB3: Pull-up resistor option registers 0 to 3 81

[R]

RXB2: Receive buffer register 2 148
RXS2: Receive shift register 2 148

[T]

TCL5: Timer clock select register 5 128
TM0: 16-bit timer counter 0 92
TM5: 8-bit timer counter 5 127
TM80: 8-bit timer counter 80 63
TMC0: 16-bit timer mode control register 0 94
TMC5: 8-bit timer mode control register 5 129
TMC80: 8-bit timer mode control register 80 64
TMCD: 16-bit timer mode control register D 122
TMD: 16-bit timer counter D 119
TOC0: 16-bit timer output control register 0 97
TXS2: Transmit shift register 2 148

[W]

WDCS: Watchdog timer clock select register 140
WDTM: Watchdog timer mode register 141

APPENDIX C REVISION HISTORY

A history of the revisions up to this edition is shown below. “Applied to:” indicates the chapters to which the revision was applied.

(1/3)

Edition	Contents	Applied to:
2nd	Update of **1.5 78K/0S Series Lineup**	**CHAPTER 1 GENERAL**
	Modification of description in **Table 2-1 Types of Pin I/O Circuits and Recommended Connection of Unused Pins**	**CHAPTER 2 PIN FUNCTIONS**
	Modification of **Figure 2-1 Pin I/O Circuits**	
	Addition of description to **4.5 Notes for EEPROM Writing**	**CHAPTER 4 EEPROM (DATA MEMORY)**
	Modification of description in **Figure 9-2 Format of Timer Clock Select Register 5 (TCL5)**	**CHAPTER 9 8-BIT TIMER/EVENT COUNTER 5**
	Correction of description in **Figure 12-5 Format of Low-Voltage Detection Level Selection Register**	**CHAPTER 12 POWER-ON-CLEAR CIRCUITS**
	Addition of Caution to **Figure 15-2 Format of Interrupt Request Flag Register**	**CHAPTER 15 INTERRUPT FUNCTIONS**
	Modification and addition of description in **18.1 EEPROM (Program Memory)**	**CHAPTER 18 μPD78E9862**
	Modification and addition of description	**CHAPTER 21 ELECTRICAL SPECIFICATIONS (TARGET)**
	Addition of description to **A.4 EEPROM (Program Memory) Writing Tools**	**APPENDIX A DEVELOPMENT TOOLS**
	Addition of chapter	**APPENDIX C REVISION HISTORY**
3rd	• μPD789862 Under development → Developed • μPD78E9862 has ES version (for development evaluation) only	Throughout
	• Update of **1.5 78K/0S Series Lineup** • Modification of description of operating ambient temperature in **1.7 Overview of Functions**	**CHAPTER 1 GENERAL**
	• Modification of description of V_{PP} pin in **2.1 (2) Non-port pins** • Modification of description in **2.2.13 V_{PP} (μPD78E9862 only)** • Modification of description of V_{PP} pin in **Table 2-1 Types of Pin I/O Circuits and Recommended Connection of Unused Pins**	**CHAPTER 2 PIN FUNCTIONS**
	Correction of **[Description example]** in **3.4.4 Register addressing**	**CHAPTER 3 CPU ARCHITECTURE**

(2/3)

Edition	Contents	Applied to:
3rd	• Addition of description in **Table 7-1 Configuration of 16-Bit Timer/Event Counter 0** • Correction of **Figure 7-1 Block Diagram of 16-Bit Timer/Event Counter 0** • Correction of Caution in **7.2 (2) 16-bit timer capture/compare register 00 (CR00)** • Correction of Caution in **7.2 (3) 16-bit timer capture/compare register 01 (CR01)** • Addition of description in **7.3 Registers to Control 16-Bit Timer/Event Counter 0** • Correction of Caution in **Figure 7-2 Format of 16-Bit Timer Mode Control Register 0 (TMC0)** • Addition of Caution in **Figure 7-5 Format of Prescaler Mode Register 0 (PRM0)** • Addition of description in **7.3 (5) Port mode register 2 (PM2)** • Addition of Note in **Figure 7-8 Interval Timer Configuration Diagram** • Correction of Caution in **Figure 7-10 Control Register Settings for PPG Output Operation** • Addition of **Figure 7-11 PPG Output Configuration Diagram** • Addition of **Figure 7-12 PPG Output Operation Timing** • Addition of Note in **Figure 7-15 Timing of Pulse Width Measurement Operation by Free-Running Counter and One Capture Register (with Both Edges Specified)** • Addition of Note in **Figure 7-18 Timing of Pulse Width Measurement Operation by Free-Running Counter (with Both Edges Specified)** • Addition of Note in **Figure 7-20 Timing of Pulse Width Measurement Operation by Free-Running Counter and Two Capture Registers (with Rising Edge Specified)** • Correction of **Figure 7-24 External Event Counter Configuration Diagram** • Correction of description in **7.5 (4) Capture register data retention timing** • Correction of **Figure 7-30 Capture Register Data Retention Timing**	**CHAPTER 7 16-BIT TIMER/EVENT COUNTER 0**
	• Correction of **Figure 9-1 Block Diagram of 8-Bit Timer/Event Counter 5** • Addition of description in **Table 9-1 Configuration of 8-Bit Timer/Event Counter 5** • Addition of description in **9.2 (2) 8-bit timer compare register 5 (CR5)** • Addition of description in **9.3 Registers to Control 8-Bit Timer/Event Counter 5** • Modification of **Figure 9-2 Format of Timer Clock Select Register 5 (TCL5)** • Addition of description in **9.3 (3) Port mode register 2 (PM2)** • Correction of **Figure 9-5 Interval Timer Operation Timing** • Addition of Remark in **Figure 9-8 PWM Output Operation Timing**	**CHAPTER 9 8-BIT TIMER/EVENT COUNTER 5**

(3/3)

Edition	Contents	Applied to:
3rd	• Addition of description in **Table 11-1 Configuration of Serial Interface 2** • Addition of description in **11.3 Control Registers of Serial Interface 2** • Addition of **11.4.2 (1) (e) Port mode register 3 (PM3)** • Addition of **11.4.3 (1) (d) Port mode register 3 (PM3)**	**CHAPTER 11 SERIAL INTERFACE 2**
	• Correction of **Figure 12-1 Block Diagram of Power-on-Clear Circuit** • Correction of **Figure 12-2 Block Diagram of Low-Voltage Detection Circuit**	**CHAPTER 12 POWER-ON-CLEAR CIRCUITS**
	Modification of description	**CHAPTER 18 μPD78E9862**
	• Addition of **Recommended Oscillator Constant** • Modification of high-level input voltage and power supply current in **DC Characteristics** • Modification of **(4) Power supply startup** in **AC Characteristics**	**CHAPTER 21 ELECTRICAL SPECIFICATIONS (μPD789862)**
	• Modification of operating ambient temperature in **Absolute Maximum Ratings** • Addition of **Recommended Oscillator Constant** • Modification of high-level input voltage in **DC Characteristics** • Modification of **(4) Power supply startup** in **AC Characteristics**	**CHAPTER 22 ELECTRICAL SPECIFICATIONS (PRELIMINARY) (μPD78E9862)**
	Addition of chapter	**CHAPTER 24 RECOMMENDED SOLDERING CONDITIONS**
	• Modification of description in **A.5.1 Hardware** • Addition of **A.6 Notes on Target System Design**	**APPENDIX A DEVELOPMENT TOOLS**
4th	• Modification of **Note** in **1.3 Ordering Information** • Revision of **1.5 78K/0S Series Lineup**	**CHAPTER 1 GENERAL**
	• Modification of **Table 4-1 EEPROM Write Time** • Modification of program example of **(3)** in **4.5 Notes for EEPROM Writing**	**CHAPTER 4 EEPROM (DATA MEMORY)**
	• Addition of **7.5 (2) Prohibition of compare register change during timer count operation**	**CHAPTER 7 16-BIT TIMER/EVENT COUNTER 0**
	• Addition of **11.3 (4) (c) Generation of serial clock from system clock in 3-wire serial I/O mode**	**CHAPTER 11 SERIAL INTERFACE 2**
	• Modification of description and addition of **Note** in **12.4.1 Power-on-clear (POC) circuit operation** • Addition of **Caution** to **12.4.2 Operation of low-voltage detector (LVI)** • Modification of **Figure 12-9 LVI Circuit Operation Timing**	**CHAPTER 12 POWER-ON-CLEAR CIRCUITS**
	• Addition of **13.3 (2) Port mode register 2 (PM2)**	**CHAPTER 13 BIT SEQUENTIAL BUFFER**
	• Modification of **Note 1** in **Table 18-1 Difference Between μPD78E9862 and Mask ROM Version** • Modification of **Table 18-2 Communication Mode List**	**CHAPTER 18 μPD78E9862**
	• Addition of **Caution** • Modification of **Note 2** in **AC Characteristics (3) EEPROM**	**CHAPTER 22 ELECTRICAL SPECIFICATIONS (μPD78E9862)**
	• Change of **Table 24-1 Surface Mounting Type Soldering Conditions**	**CHAPTER 24 RECOMMENDED SOLERING CONDITIONS**