NEC

User's Manual

μPD789074 Subseries

8-Bit Single-Chip Microcontrollers

μPD789071 μPD789071(A)
μPD789072 μPD789072(A)
μPD789074 μPD789074(A)
μPD78F9076

Document No. U14801EJ3V0UD00 (3rd edition)
Date Published July 2003 N CP(K)

© NEC Electronics Corporation 2000, 2002
Printed in Japan

[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.

EEPROM and FIP are trademarks of NEC Electronics Corporation.
Windows and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
PC/AT is a trademark of International Business Machines Corporation.
HP9000 series 700 and HP-UX are trademarks of Hewlett-Packard Company.
SPARCstation is a trademark of SPARC International, Inc.
Solaris and SunOS are trademarks of Sun Microsystems, Inc.

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 December, 2002. 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
Throughout	• Addition of μPD789071(A), 789072(A), and 789074(A) • Addition of description of expanded-specification products
pp.20, 22, 29	**CHAPTER 1 GENERAL** • Addition of **1.1 Expanded-Specification Products and Conventional Products** • Addition of **1.5 Quality Grades** • Addition of **1.10 Differences Between Standard Quality Grade Products and (A) Products**
pp.89, 90, 91, 95	**CHAPTER 6 16-BIT TIMER 90** • Modification of description of **6.4.1 Operation as timer interrupt** • Modification of **Figure 6-6. Timing of Timer Interrupt Operation** • Modification of description of **6.4.2 Operation as timer output** • Modification of **Figure 6-8. Timer Output Timing** • Addition of **6.5 Notes on Using 16-Bit Timer 90**
p.108	**CHAPTER 7 8-BIT TIMER/EVENT COUNTER 80** • Addition of **7.5 (3) Timer operation after compare register is rewritten during PWM output** • Addition of **7.5 (4) Cautions when STOP mode is set** • Addition of **7.5 (5) Start timing of external event counter**
p.174	**CHAPTER 13 μPD78F9076** • Total revision of description of flash memory programming
p.195	Addition of **CHAPTER 15 ELECTRICAL SPECIFICATIONS (EXPANDED-SPECIFICATION PRODUCTS)**
p.211	**CHAPTER 16 ELECTRICAL SPECIFICATIONS (CONVENTIONAL PRODUCTS)** • Modification of table of recommended oscillator constant
p.223	**CHAPTER 18 RECOMMENDED SOLDERING CONDITIONS** • Change of recommended soldering conditions of μPD78F9076
p.228	**APPENDIX A DEVELOPMENT TOOLS** • Modification of description of **A.5 Debugging Tools (Hardware)**
p.230	Addition of **APPENDIX B NOTES ON TARGET SYSTEM DESIGN**

The mark ★ shows major revised points.

INTRODUCTION

Readers This manual is intended for user engineers who wish to gain an understanding of the functions of the µPD789074 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 µPD789074 Subseries: this manual and the Instruction Manual (common to the 78K/0S Series).

µPD789074 Subseries User's Manual	78K/0S Series User's Manual Instructions
• Pin functions • Internal block functions • Interrupts • Other internal peripheral functions • Electrical specifications	• CPU function • Instruction set • Instruction description

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

◊ For users who use this document as the manual for the µPD789071(A), 789072(A), or 789074(A)

→ The only differences between standard products and (A) products are the quality grades and electrical specifications (refer to **1.10 Differences Between Standard Quality Grade Products and (A) Products**). For the (A) products, read the part numbers as follows.

µPD789071 → µPD789071(A)
µPD789072 → µPD789072(A)
µPD789074 → µPD789074(A)

◊ To understand the overall functions of the µPD789074 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 C REGISTER INDEX**.

◊ To learn details of the instruction functions of the 78K/0S Series

→ Refer to **78K/0S Series Instructions User's Manual (U11047E)** available separately.

◊ To learn the electrical specifications of the µPD789074 Subseries

→ Refer to **CHAPTER 15 ELECTRICAL SPECIFICATIONS (EXPANDED-SPECIFICATION PRODUCTS)** and **CHAPTER 16 ELECTRICAL SPECIFICATIONS (CONVENTIONAL PRODUCTS)**.

Caution The application examples in this manual are created for "Standard" quality grade products for general electric equipment. When using the application examples in this manual for purposes which require "Special" quality grades, thoroughly examine the quality grade of each part and circuit actually used.

Conversions

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 **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.
μPD789074 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
SM78K Series System Simulator Ver. 2.30 or Later	Operation (Windows™ Based)	U15373E
	External Parts User Open Interface Specifications	U15802E
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-789046-NS-EM1 Emulation Board	U14433E

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 for 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 20
★ 1.1 Expanded-Specification Products and Conventional Products 20
1.2 Features 21
1.3 Applications 21
1.4 Ordering Information 21
★ 1.5 Quality Grades 22
1.6 Pin Configuration (Top View) 23
1.7 78K/0S Series Lineup 24
1.8 Block Diagram 27
1.9 Overview of Functions 28
★ 1.10 Differences Between Standard Quality Grade Products and (A) Products 29

CHAPTER 2 PIN FUNCTIONS 30
2.1 Pin Function List 30
2.2 Description of Pin Functions 32
2.2.1 P00 to P07 (Port 0) 32
2.2.2 P10 to P15 (Port 1) 32
2.2.3 P20 to P27 (Port 2) 32
2.2.4 P30, P31 (Port 3) 33
2.2.5 $\overline{\text{RESET}}$ 33
2.2.6 X1, X2 33
2.2.7 V$_{DD}$ 33
2.2.8 V$_{SS}$ 33
2.2.9 V$_{PP}$ (μPD78F9076 only) 34
2.2.10 IC (mask ROM version only) 34
2.3 Pin I/O Circuits and Recommended Connection of Unused Pins 35

CHAPTER 3 CPU ARCHITECTURE 37
3.1 Memory Space 37
3.1.1 Internal program memory space 41
3.1.2 Internal data memory (internal high-speed RAM) space 41
3.1.3 Special function register (SFR) area 41
3.1.4 Data memory addressing 42
3.2 Processor Registers 46
3.2.1 Control registers 46
3.2.2 General-purpose registers 49
3.2.3 Special function registers (SFRs) 50
3.3 Instruction Address Addressing 53
3.3.1 Relative addressing 53
3.3.2 Immediate addressing 54

3.3.3 Table indirect addressing 55
3.3.4 Register addressing 55
3.4 Operand Address Addressing 56
3.4.1 Direct addressing 56
3.4.2 Short direct addressing 57
3.4.3 Special function register (SFR) addressing 58
3.4.4 Register addressing 59
3.4.5 Register indirect addressing 60
3.4.6 Based addressing 61
3.4.7 Stack addressing 61

CHAPTER 4 PORT FUNCTIONS 62

4.1 Port Functions 62
4.2 Port Configuration 64
4.2.1 Port 0 64
4.2.2 Port 1 65
4.2.3 Port 2 66
4.2.4 Port 3 70
4.3 Port Function Control Registers 71
4.4 Operation of Port Functions 74
4.4.1 Writing to I/O port 74
4.4.2 Reading from I/O port 74
4.4.3 Arithmetic operation of I/O port 74

CHAPTER 5 CLOCK GENERATOR 75

5.1 Clock Generator Functions 75
5.2 Clock Generator Configuration 75
5.3 Clock Generator Control Register 76
5.4 System Clock Oscillators 77
5.4.1 System clock oscillator 77
5.4.2 Examples of incorrect resonator connection 78
5.4.3 Frequency divider 79
5.5 Clock Generator Operation 80
5.6 Changing Setting of CPU Clock 81
5.6.1 Time required for switching CPU clock 81
5.6.2 Switching CPU clock 81

CHAPTER 6 16-BIT TIMER 90 82

6.1 Functions of 16-Bit Timer 90 82
6.2 Configuration of 16-Bit Timer 90 82
6.3 Control Registers of 16-Bit Timer 90 85
6.4 Operation of 16-Bit Timer 90 89
6.4.1 Operation as timer interrupt 89
6.4.2 Operation as timer output 91

6.4.3 Capture operation..92
6.4.4 16-bit timer counter 90 readout..93
6.4.5 Buzzer output operation..94
★ **6.5 Notes on Using 16-Bit Timer 90..95**
6.5.1 Restrictions on rewriting16-bit compare register 90..95

CHAPTER 7 8-BIT TIMER/EVENT COUNTER 80..97

7.1 Functions of 8-Bit Timer/Event Counter 80..97
7.2 Configuration of 8-Bit Timer/Event Counter 80..98
7.3 8-Bit Timer/Event Counter 80 Control Registers..99
7.4 Operation of 8-Bit Timer/Event Counter 80..101
7.4.1 Operation as interval timer..101
7.4.2 Operation as external event counter..103
7.4.3 Operation as square-wave output..104
7.4.4 Operation as PWM output..106
7.5 Notes on Using 8-Bit Timer/Event Counter 80..107

CHAPTER 8 WATCHDOG TIMER..109

8.1 Watchdog Timer Functions..109
8.2 Watchdog Timer Configuration..110
8.3 Watchdog Timer Control Registers..111
8.4 Watchdog Timer Operation..113
8.4.1 Operation as watchdog timer..113
8.4.2 Operation as interval timer..114

CHAPTER 9 SERIAL INTERFACE 20..115

9.1 Functions of Serial Interface 20..115
9.2 Configuration of Serial Interface 20..115
9.3 Control Registers of Serial Interface 20..119
9.4 Operation of Serial Interface 20..126
9.4.1 Operation stop mode..126
9.4.2 Asynchronous serial interface (UART) mode..127
9.4.3 3-wire serial I/O mode..140

CHAPTER 10 INTERRUPT FUNCTIONS..150

10.1 Interrupt Function Types..150
10.2 Interrupt Sources and Configuration..150
10.3 Interrupt Function Control Registers..153
10.4 Interrupt Processing Operation..158
10.4.1 Non-maskable interrupt request acknowledgment operation..158
10.4.2 Maskable interrupt request acknowledgment operation..160
10.4.3 Multiple interrupt servicing..162
10.4.4 Interrupt request hold..164

CHAPTER 11 STANDBY FUNCTION **165**

11.1 Standby Function and Configuration **165**

  11.1.1 Standby function 165

  11.1.2 Standby function control register 166

11.2 Operation of Standby Function **167**

  11.2.1 HALT mode 167

  11.2.2 STOP mode 169

CHAPTER 12 RESET FUNCTION **171**

CHAPTER 13 μPD78F9076 .... **174**

★ **13.1 Flash Memory Characteristics** **175**

  13.1.1 Programming environment 175

  13.1.2 Communication mode 176

  13.1.3 On-board pin connections 179

  13.1.4 Connection of adapter for flash writing 182

CHAPTER 14 INSTRUCTION SET OVERVIEW **185**

14.1 Operation **185**

  14.1.1 Operand identifiers and description methods 185

  14.1.2 Description of "Operation" column 186

  14.1.3 Description of "Flag" column 186

14.2 Operation List **187**

14.3 Instructions Listed by Addressing Type **192**

★ **CHAPTER 15 ELECTRICAL SPECIFICATIONS (EXPANDED-SPECIFICATION PRODUCTS)** **195**

CHAPTER 16 ELECTRICAL SPECIFICATIONS (CONVENTIONAL PRODUCTS) **209**

CHAPTER 17 PACKAGE DRAWING **222**

CHAPTER 18 RECOMMENDED SOLDERING CONDITIONS **223**

APPENDIX A DEVELOPMENT TOOLS **224**

A.1 Software Package **226**

A.2 Language Processing Software **226**

A.3 Control Software **227**

A.4 Flash Memory Writing Tools **228**

A.5 Debugging Tools (Hardware) **228**

A.6 Debugging Tools (Software) **229**

★ **APPENDIX B NOTES ON TARGET SYSTEM DESIGN** **230**

APPENDIX C REGISTER INDEX 232

C.1 Register Name Index (Alphabetic Order) 232

C.2 Register Symbol Index (Alphabetic Order)..234

APPENDIX D REVISION HISTORY..236

LIST OF FIGURES (1/3)

Figure No. Title Page

2-1 Pin I/O Circuits....36

3-1 Memory Map (μPD789071)....37
3-2 Memory Map (μPD789072)....38
3-3 Memory Map (μPD789074)....39
3-4 Memory Map (μPD78F9076)....40
3-5 Data Memory Addressing (μPD789071)....42
3-6 Data Memory Addressing (μPD789072)....43
3-7 Data Memory Addressing (μPD789074)....44
3-8 Data Memory Addressing (μPD78F9076)....45
3-9 Program Counter Configuration....46
3-10 Program Status Word Configuration....46
3-11 Stack Pointer Configuration....48
3-12 Data to Be Saved to Stack Memory....48
3-13 Data to Be Restored from Stack Memory....48
3-14 General-Purpose Register Configuration....49

4-1 Port Types....62
4-2 Block Diagram of P00 to P07....64
4-3 Block Diagram of P10 to P15....65
4-4 Block Diagram of P20....66
4-5 Block Diagram of P21....67
4-6 Block Diagram of P22 to P26....68
4-7 Block Diagram of P27....69
4-8 Block Diagram of P30 and P31....70
4-9 Format of Port Mode Register....71
4-10 Format of Pull-up Resistor Option Register 0....72
4-11 Format of Pull-up Resistor Option Register B2....73

5-1 Block Diagram of Clock Generator....75
5-2 Format of Processor Clock Control Register....76
5-3 External Circuit of System Clock Oscillator....77
5-4 Examples of Incorrect Resonator Connection....78
5-5 Switching Between System Clock and CPU Clock....81

6-1 Block Diagram of 16-Bit Timer 90....83
6-2 Format of 16-Bit Timer Mode Control Register 90....86
6-3 Format of Buzzer Output Control Register 90....87
6-4 Format of Port Mode Register 3....88

LIST OF FIGURES (2/3)

Figure No.	Title	Page
6-5	Settings of 16-Bit Timer Mode Control Register 90 for Timer Interrupt Operation	89
6-6	Timing of Timer Interrupt Operation	90
6-7	Settings of 16-Bit Timer Mode Control Register 90 for Timer Output Operation	91
6-8	Timer Output Timing	91
6-9	Settings of 16-Bit Timer Mode Control Register 90 for Capture Operation	92
6-10	Capture Operation Timing (with Both Edges of CPT90 Pin Specified)	92
6-11	16-Bit Timer Counter 90 Readout Timing	93
6-12	Settings of Buzzer Output Control Register 90 for Buzzer Output Operation	94
7-1	Block Diagram of 8-Bit Timer/Event Counter 80	98
7-2	Format of 8-Bit Timer Mode Control Register 80	99
7-3	Format of Port Mode Register 2	100
7-4	Interval Timer Operation Timing	102
7-5	External Event Counter Operation Timing (with Rising Edge Specified)	103
7-6	Square-Wave Output Timing	105
7-7	PWM Output Timing	106
7-8	Start Timing of 8-Bit Timer Counter 80	107
7-9	External Event Counter Operation Timing	107
7-10	Operation Timing After Compare Register Is Rewritten During PWM Output	108
8-1	Block Diagram of Watchdog Timer	110
8-2	Format of Watchdog Timer Clock Selection Register	111
8-3	Format of Watchdog Timer Mode Register	112
9-1	Block Diagram of Serial Interface 20	116
9-2	Block Diagram of Baud Rate Generator 20	117
9-3	Format of Serial Operation Mode Register 20	119
9-4	Format of Asynchronous Serial Interface Mode Register 20	120
9-5	Format of Asynchronous Serial Interface Status Register 20	122
9-6	Format of Baud Rate Generator Control Register 20	123
9-7	Format of Asynchronous Serial Interface Transmit/Receive Data	133
9-8	Asynchronous Serial Interface Transmission Completion Interrupt Timing	135
9-9	Asynchronous Serial Interface Reception Completion Interrupt Timing	136
9-10	Receive Error Timing	137
9-11	3-Wire Serial I/O Mode Timing	143
10-1	Basic Configuration of Interrupt Function	152
10-2	Format of Interrupt Request Flag Register	154
10-3	Format of Interrupt Mask Flag Register	155

LIST OF FIGURES (3/3)

Figure No. Title Page

10-4 Format of External Interrupt Mode Register 0 156
10-5 Program Status Word Configuration 157
10-6 Flowchart from Non-Maskable Interrupt Request Generation to Acknowledgment 159
10-7 Timing of Non-Maskable Interrupt Request Acknowledgment 159
10-8 Acknowledgment of Non-Maskable Interrupt Request 159
10-9 Interrupt Request Acknowledgment Processing Algorithm 161
10-10 Interrupt Request Acknowledgment Timing (Example of MOV A,r) 162
10-11 Interrupt Request Acknowledgment Timing (When Interrupt Request Flag Is Set at Last Clock During Instruction Execution) 162
10-12 Example of Multiple Interrupts 163

11-1 Format of Oscillation Stabilization Time Selection Register 166
11-2 Releasing HALT Mode by Interrupt 167
11-3 Releasing HALT Mode by $\overline{\text{RESET}}$ Input 168
11-4 Releasing STOP Mode by Interrupt 170
11-5 Releasing STOP Mode by $\overline{\text{RESET}}$ Input 170

12-1 Block Diagram of Reset Function 171
12-2 Reset Timing by $\overline{\text{RESET}}$ Input 172
12-3 Reset Timing by Watchdog Timer Overflow 172
12-4 Reset Timing by $\overline{\text{RESET}}$ Input in STOP Mode 172

13-1 Environment for Writing Program to Flash Memory 175
13-2 Communication Mode Selection Format 176
13-3 Example of Connection with Dedicated Flash Programmer 177
13-4 V_{PP} Pin Connection Example 179
13-5 Signal Conflict (Serial Interface Input Pin) 180
13-6 Malfunction of Another Device 180
13-7 Signal Conflict ($\overline{\text{RESET}}$ Pin) 181
13-8 Wiring Example for Flash Writing Adapter Using 3-Wire Serial I/O 182
13-9 Wiring Example for Flash Writing Adapter Using UART 183
13-10 Wiring Example for Flash Writing Adapter Using Pseudo 3-Wire 184

A-1 Development Tools 225

B-1 Distance Between In-Circuit Emulator and Conversion Adapter 230
B-2 Connection Condition of Target System 231

LIST OF TABLES (1/2)

Table No. Title Page

1-1 Differences Between Expanded-Specification Products and Conventional Products 20
1-2 Differences Between Standard Quality Grade Products and (A) Products 29

2-1 Types of Pin I/O Circuits and Recommended Connection of Unused Pins 35

3-1 Internal ROM Capacity 41
3-2 Vector Table 41
3-3 Special Function Registers 51

4-1 Port Functions 63
4-2 Configuration of Port 64
4-3 Port Mode Register and Output Latch Settings for Using Alternate Functions 72

5-1 Configuration of Clock Generator 75
5-2 Maximum Time Required for Switching CPU Clock 81

6-1 Configuration of 16-Bit Timer 90 82
6-2 Interval Time of 16-Bit Timer 90 89
6-3 Settings of Capture Edge 92
6-4 Buzzer Frequency of 16-Bit Timer 90 94

7-1 Interval Time of 8-Bit Timer/Event Counter 80 97
7-2 Square-Wave Output Range of 8-Bit Timer/Event Counter 80 97
7-3 Configuration of 8-Bit Timer/Event Counter 80 98
7-4 Interval Time of 8-Bit Timer/Event Counter 80 101
7-5 Square-Wave Output Range of 8-Bit Timer/Event Counter 104

8-1 Inadvertent Loop Detection Time of Watchdog Timer 109
8-2 Interval Time 109
8-3 Configuration of Watchdog Timer 110
8-4 Inadvertent Loop Detection Time of Watchdog Timer 113
8-5 Interval Generated Using Interval Timer 114

9-1 Configuration of Serial Interface 20 115
9-2 Operating Mode Settings of Serial Interface 20 121
9-3 Example of Relationship Between System Clock and Baud Rate 124
9-4 Relationship Between ASCK20 Pin Input Frequency and Baud Rate (When BRGC20 Is Set to 80H) 125
9-5 Example of Relationship Between System Clock and Baud Rate 132
9-6 Relationship Between ASCK20 Pin Input Frequency and Baud Rate (When BRGC20 Is Set to 80H) 132

LIST OF TABLES (2/2)

Table No. Title Page

9-7 Receive Error Causes....137

10-1 Interrupt Sources....151
10-2 Interrupt Request Signals and Corresponding Flags....153
10-3 Time from Generation of Maskable Interrupt Request to Servicing....160

11-1 Operation Statuses in HALT Mode....167
11-2 Operation After Releasing HALT Mode....168
11-3 Operation Statuses in STOP Mode....169
11-4 Operation After Releasing STOP Mode....170

12-1 Status of Hardware After Reset....173

13-1 Differences Between Flash Memory and Mask ROM Versions....174
13-2 Communication Mode List....176
13-3 Pin Connection List....178

14-1 Operand Identifiers and Description Methods....185

18-1 Surface Mounting Type Soldering Conditions....223

CHAPTER 1 GENERAL

★ 1.1 Expanded-Specification Products and Conventional Products

Expanded-specification products and conventional products refer to the following products.

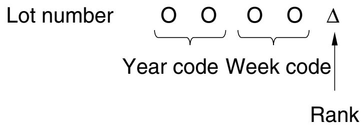
Expanded-specification products: Products with a rank[Note] other than K

- Mask ROM versions for which orders were received after December 1, 2001.
- μPD78F9076 shipped after January 1, 2002.

Conventional products: Products with rank[Note] K

- Products other than the above expanded-specification products.

Note The rank is indicated by the 5th digit from the left in the lot number marked on the package.



Expanded-specification products and conventional products differ in operating frequency ratings. The differences are shown in Table 1-1.

Table 1-1. Differences Between Expanded-Specification Products and Conventional Products

Power Supply Voltage (V_{DD})	Guaranteed Operating Speed (Operating Frequency)	
	Conventional Products	Expanded-Specification Products
4.5 to 5.5 V	5 MHz (0.4 μs)	10 MHz (0.2 μs)
3.0 to 5.5 V	5 MHz (0.4 μs)	6 MHz (0.33 μs)
2.7 to 5.5 V	5 MHz (0.4 μs)	5 MHz (0.4 μs)
1.8 to 5.5 V	1.25 MHz (1.6 μs)	1.25 MHz (1.6 μs)

Remark The parenthesized values indicate the minimum instruction execution time.

1.2 Features

- ROM and RAM capacity

Item Product Name	Program Memory		Data Memory (Internal High-Speed RAM)
μPD789071, 789071(A)	Mask ROM	2 KB	256 KB
μPD789072, 789072(A)		4 KB	
μPD789074, 789074(A)		8 KB	
μPD78F9076	Flash memory	16 KB	

★
- Minimum instruction execution time can be changed from high-speed (0.2 μs) to low speed (0.8 μs) (at 10.0 MHz, V_{DD} = 4.5 to 5.5 V operation with system clock)
- I/O ports: 24
- Serial interface: 1 channel
  3-wire serial I/O mode/UART mode: 1 channel
- Timer: 3 channels
  - 16-bit timer: 1 channel
  - 8-bit timer/event counter: 1 channel
  - Watchdog timer: 1 channel
- Vectored interrupt sources: 9
- Supply voltage: V_{DD} = 1.8 to 5.5 V
- Operating ambient temperature: T_A = –40 to +85°C

1.3 Applications

Small, general home electrical appliances, telephones, etc.

1.4 Ordering Information

	Part Number	Package	Quality Grade
	μPD789071MC-×××-5A4	30-pin plastic SSOP (7.62 mm (300))	Mask ROM
	μPD789072MC-×××-5A4	30-pin plastic SSOP (7.62 mm (300))	Mask ROM
	μPD789074MC-×××-5A4	30-pin plastic SSOP (7.62 mm (300))	Mask ROM
★	μPD789071MC(A)-×××-5A4	30-pin plastic SSOP (7.62 mm (300))	Mask ROM
★	μPD789072MC(A)-×××-5A4	30-pin plastic SSOP (7.62 mm (300))	Mask ROM
★	μPD789074MC(A)-×××-5A4	30-pin plastic SSOP (7.62 mm (300))	Mask ROM
	μPD78F9076MC-5A4	30-pin plastic SSOP (7.62 mm (300))	Flash memory

Remark ××× indicates ROM code suffix.

★ 1.5 Quality Grades

Part Number	Package	Quality Grade
μPD789071MC-×××-5A4	30-pin plastic SSOP (7.62 mm (300))	Standard
μPD789072MC-×××-5A4	30-pin plastic SSOP (7.62 mm (300))	Standard
μPD789074MC-×××-5A4	30-pin plastic SSOP (7.62 mm (300))	Standard
μPD78F9076MC-5A4	30-pin plastic SSOP (7.62 mm (300))	Standard
μPD789071MC(A)-×××-5A4	30-pin plastic SSOP (7.62 mm (300))	Special
μPD789072MC(A)-×××-5A4	30-pin plastic SSOP (7.62 mm (300))	Special
μPD789074MC(A)-×××-5A4	30-pin plastic SSOP (7.62 mm (300))	Special

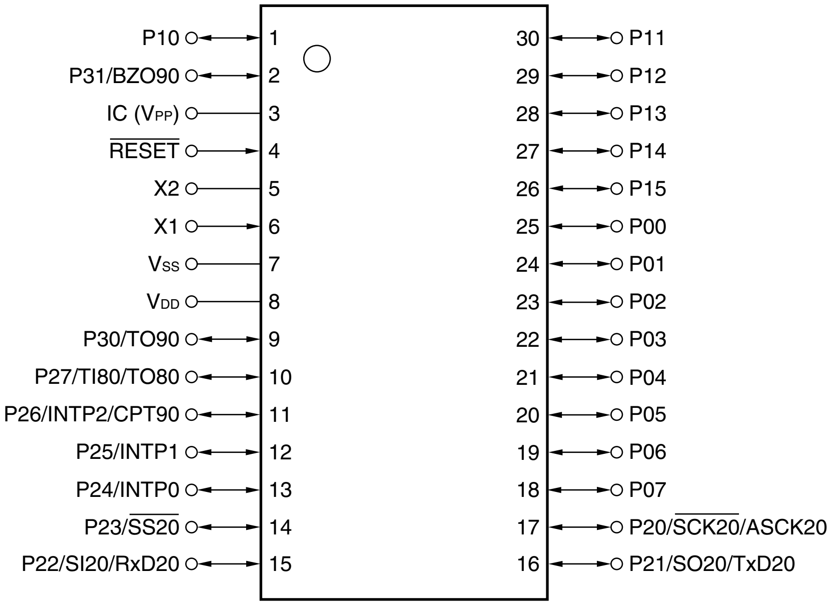
Remark ××× indicates ROM code suffix.

Please refer to "Quality Grades on NEC Semiconductor Devices" (Document No. C11531E) published by NEC Electronics Corporation to know the specification of the quality grade on the device and its recommended applications.

1.6 Pin Configuration (Top View)

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

μPD789071MC-×××-5A4
μPD789072MC-×××-5A4
μPD789074MC-×××-5A4
★ μPD789071MC(A)-×××-5A4
★ μPD789072MC(A)-×××-5A4
★ μPD789074MC(A)-×××-5A4
μPD78F9076MC-5A4


P10 1
P31/BZO90 2
IC (V_{PP}) 3
$\overline{\text{RESET}}$ 4
X2 5
X1 6
V_{SS} 7
V_{DD} 8
P30/TO90 9
P27/TI80/TO80 10
P26/INTP2/CPT90 11
P25/INTP1 12
P24/INTP0 13
P23/$\overline{\text{SS20}}$ 14
P22/SI20/RxD20 15
30 P11
29 P12
28 P13
27 P14
26 P15
25 P00
24 P01
23 P02
22 P03
21 P04
20 P05
19 P06
18 P07
17 P20/$\overline{\text{SCK20}}$/ASCK20
16 P21/SO20/TxD20

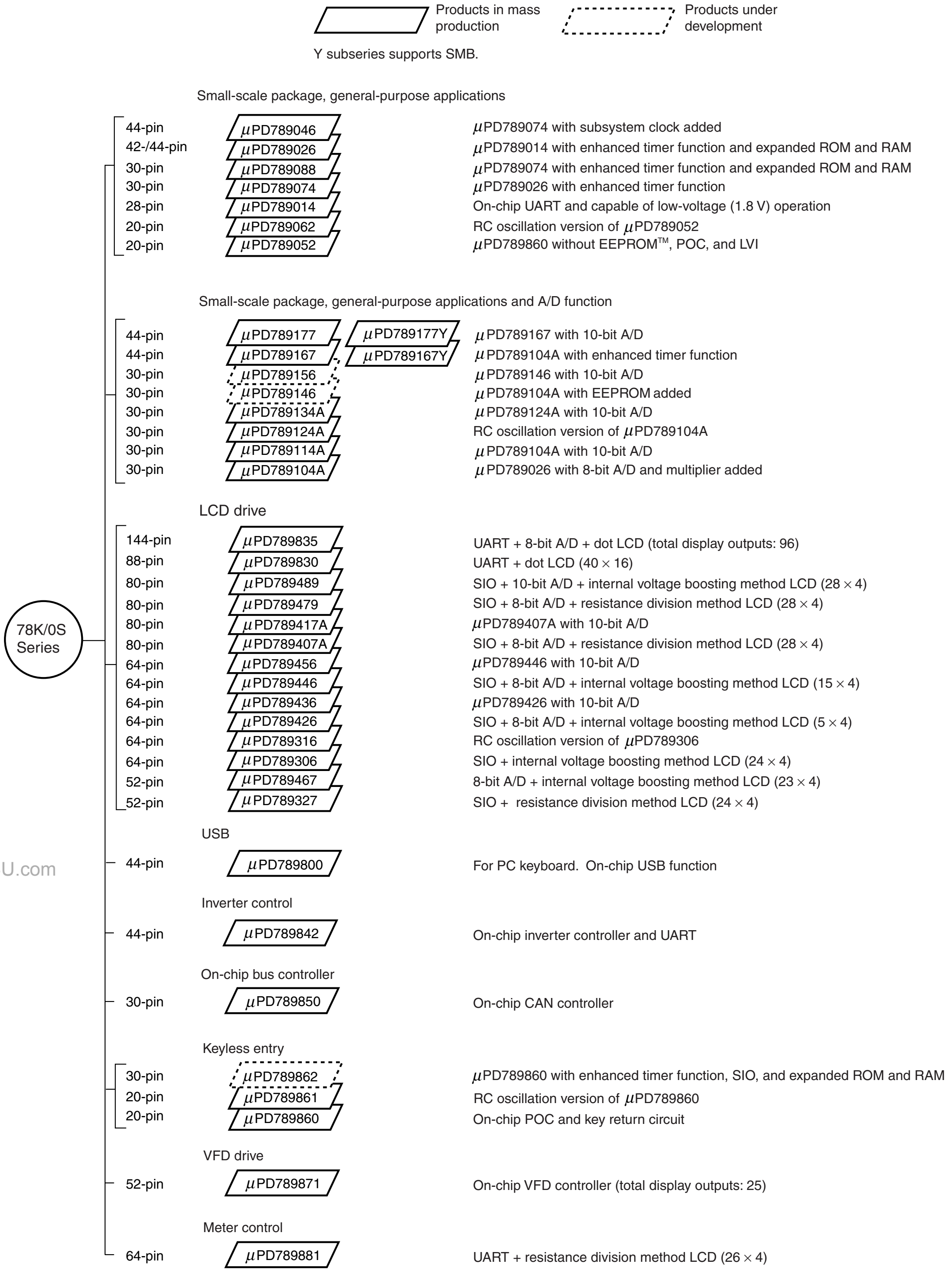
Caution **Connect the IC (Internally Connected) pin directly to V_{SS}.**

Remark Pin connections in parentheses are intended for the μPD78F9076.

ASCK20:	Asynchronous serial input	$\overline{\text{SCK20}}$:	Serial clock
BZO90:	Buzzer output	SI20:	Serial input
CPT90:	Capture trigger input	SO20:	Serial output
IC:	Internally connected	$\overline{\text{SS20}}$:	Chip select input
INTP0 to INTP2:	External interrupt input	TI80:	Timer input
P00 to P07:	Port 0	TO80, TO90:	Timer output
P10 to P15:	Port 1	TxD20:	Transmit data
P20 to P27:	Port 2	V_{DD}:	Power supply
P30, P31:	Port 3	V_{PP}:	Programming power supply
$\overline{\text{RESET}}$:	Reset	V_{SS}:	Ground
RxD20:	Receive data	X1, X2:	Crystal/ceramic oscillator

★ 1.7 78K/0S Series Lineup

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


Products in mass production
Products under development
Y subseries supports SMB.
78K/0S Series

Small-scale package, general-purpose applications

44-pin	μPD789046	μPD789074 with subsystem clock added
42-/44-pin	μPD789026	μPD789014 with enhanced timer function and expanded ROM and RAM
30-pin	μPD789088	μPD789074 with enhanced timer function and expanded ROM and RAM
30-pin	μPD789074	μPD789026 with enhanced timer function
28-pin	μPD789014	On-chip UART and capable of low-voltage (1.8 V) operation
20-pin	μPD789062	RC oscillation version of μPD789052
20-pin	μPD789052	μPD789860 without EEPROM™, POC, and LVI

Small-scale package, general-purpose applications and A/D function

44-pin	μPD789177	μPD789177Y	μPD789167 with 10-bit A/D
44-pin	μPD789167	μPD789167Y	μPD789104A with enhanced timer function
30-pin	μPD789156		μPD789146 with 10-bit A/D
30-pin	μPD789146		μPD789104A with EEPROM added
30-pin	μPD789134A		μPD789124A with 10-bit A/D
30-pin	μPD789124A		RC oscillation version of μPD789104A
30-pin	μPD789114A		μPD789104A with 10-bit A/D
30-pin	μPD789104A		μPD789026 with 8-bit A/D and multiplier added

LCD drive

144-pin	μPD789835	UART + 8-bit A/D + dot LCD (total display outputs: 96)
88-pin	μPD789830	UART + dot LCD (40 × 16)
80-pin	μPD789489	SIO + 10-bit A/D + internal voltage boosting method LCD (28 × 4)
80-pin	μPD789479	SIO + 8-bit A/D + resistance division method LCD (28 × 4)
80-pin	μPD789417A	μPD789407A with 10-bit A/D
80-pin	μPD789407A	SIO + 8-bit A/D + resistance division method LCD (28 × 4)
64-pin	μPD789456	μPD789446 with 10-bit A/D
64-pin	μPD789446	SIO + 8-bit A/D + internal voltage boosting method LCD (15 × 4)
64-pin	μPD789436	μPD789426 with 10-bit A/D
64-pin	μPD789426	SIO + 8-bit A/D + internal voltage boosting method LCD (5 × 4)
64-pin	μPD789316	RC oscillation version of μPD789306
64-pin	μPD789306	SIO + internal voltage boosting method LCD (24 × 4)
52-pin	μPD789467	8-bit A/D + internal voltage boosting method LCD (23 × 4)
52-pin	μPD789327	SIO + resistance division method LCD (24 × 4)

USB

44-pin	μPD789800	For PC keyboard. On-chip USB function

Inverter control

44-pin	μPD789842	On-chip inverter controller and UART

On-chip bus controller

30-pin	μPD789850	On-chip CAN controller

Keyless entry

30-pin	μPD789862	μPD789860 with enhanced timer function, SIO, and expanded ROM and RAM
20-pin	μPD789861	RC oscillation version of μPD789860
20-pin	μPD789860	On-chip POC and key return circuit

VFD drive

52-pin	μPD789871	On-chip VFD controller (total display outputs: 25)

Meter control

64-pin	μPD789881	UART + resistance division method LCD (26 × 4)

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

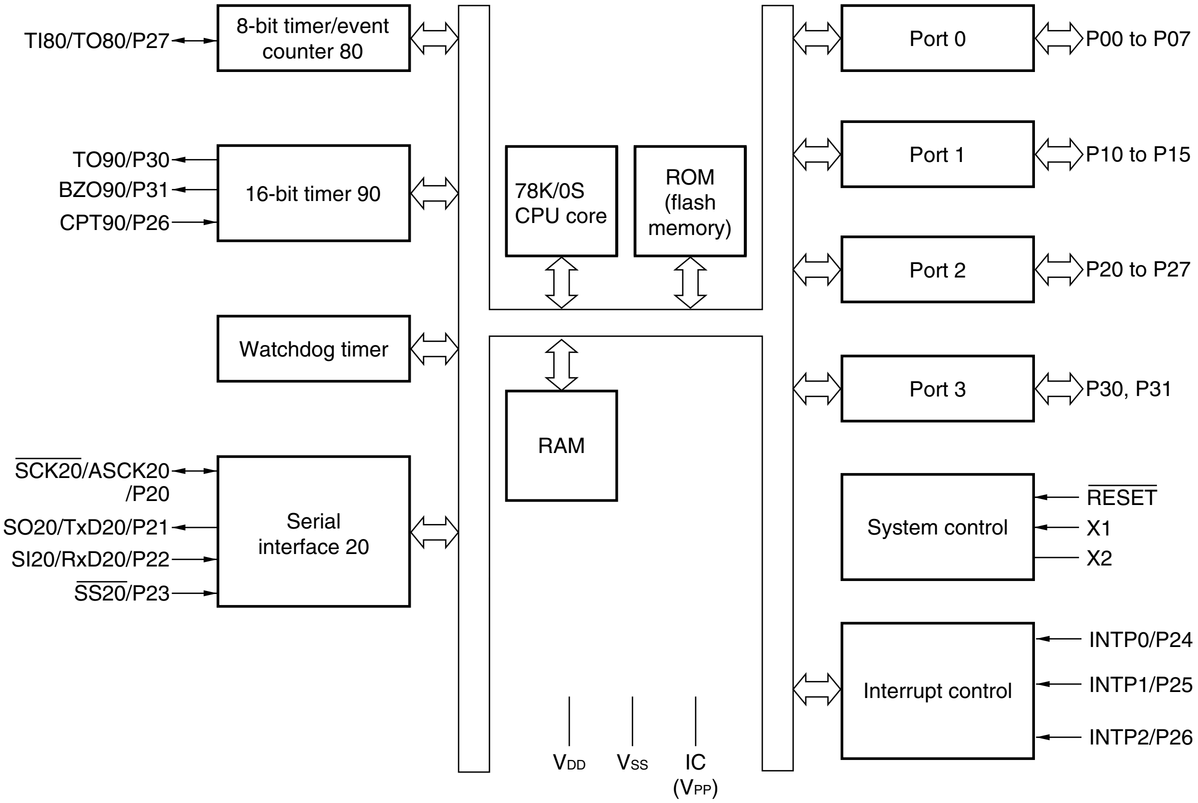
Note Flash memory version: 3.0 V

Series for ASSP

Subseries	Function	ROM Capacity (Bytes)	Timer				8-Bit A/D	10-Bit A/D	Serial Interface	I/O	V_{DD}	Remarks
			8-Bit	16-Bit	Watch	WDT					MIN.Value	
USB	µPD789800	8 K	2 ch	–	–	1 ch	–	–	2 ch (USB: 1 ch)	31	4.0 V	–
Inverter control	µPD789842	8 K to 16 K	3 ch	**Note 1**	1 ch	1 ch	8 ch	–	1 ch (UART: 1 ch)	30	4.0 V	–
On-chip bus controller	µPD789850	16 K	1 ch	1 ch	–	1 ch	4 ch	–	2 ch (UART: 1 ch)	18	4.0 V	–
Keyless entry	µPD789861	4 K	2 ch	–	–	1 ch	–	–	–	14	1.8 V	RC-oscillation version, on-chip EEPROM
	µPD789860											On-chip EEPROM
	µPD789862	16 K	1 ch	2 ch					1 ch (UART: 1 ch)	22		
VFD drive	µPD789871	4 K to 8 K	3 ch	–	1 ch	1 ch	–	–	1 ch	33	2.7 V	–
Meter control	µPD789881	16 K	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

1.8 Block Diagram


TI80/TO80/P27
8-bit timer/event counter 80
TO90/P30
BZO90/P31
CPT90/P26
16-bit timer 90
Watchdog timer
SCK20/ASCK20/P20
SO20/TxD20/P21
SI20/RxD20/P22
SS20/P23
Serial interface 20
78K/0S CPU core
ROM (flash memory)
RAM
VDD
VSS
IC (VPP)
Port 0
P00 to P07
Port 1
P10 to P15
Port 2
P20 to P27
Port 3
P30, P31
System control
RESET
X1
X2
Interrupt control
INTP0/P24
INTP1/P25
INTP2/P26

Remarks 1. The internal ROM capacity varies depending on the product.

2. Pin connections in parentheses are intended for the μPD78F9076.

1.9 Overview of Functions

Item		Part Number: μPD789071 μPD789071(A)	μPD789072 μPD789072(A)	μPD789074 μPD789074(A)	μPD78F9076
Internal memory	ROM	Mask ROM			Flash memory
		2 KB	4 KB	8 KB	16 KB
	High-speed RAM	256 bytes			
★ Minimum instruction execution time		0.2/0.8 μs (@10.0 MHz, V_{DD} = 4.5 to 5.5 V 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		CMOS I/O: 24			
Serial interface		Switchable between 3-wire serial I/O and UART modes: 1 channel			
Timers		• 16-bit timer: 1 channel • 8-bit timer/event counter: 1 channel • Watchdog timer: 1 channel			
Timer outputs		2			
Vectored interrupt sources	Maskable	Internal: 5, external: 3			
	Non-maskable	Internal: 1			
Power supply voltage		V_{DD} = 1.8 to 5.5 V			
Operating ambient temperature		T_A = –40 to +85°C			
Package		30-pin plastic SSOP (7.62 mm (300))			

The outline of the timers are as follows.

		16-Bit Timer 90	8-Bit Timer/Event Counter 80	Watchdog Timer
Operating mode	Interval timer	–	1 channel	1 channel[Note]
	External event counter	–	1 channel	–
Function	Timer outputs	1	1	–
	PWM outputs	–	1	–
	Square-wave outputs	–	1	–
	Buzzer outputs	1	–	–
	Capture	1 input	–	–
	Interrupt sources	1	1	2

Note The watchdog timer provides a watchdog timer function and interval timer function. Use either of the functions.

★ 1.10 Differences Between Standard Quality Grade Products and (A) Products

The differences between standard grade products (μPD789071, 789072, 789074) and (A) products (μPD789071(A), 789072(A), 789074(A)) are shown in Table 1-2.

Table 1-2. Differences Between Standard Quality Grade Products and (A) Products

Part Number / Item	Standard Products	(A) Products
Quality grade	Standard	Special
Electrical specifications	Refer to **CHAPTER 15 ELECRIDAL SPECIFICATIONS (EXPANDED-SPECIFICATION PRODUCTS)** and **CHAPTER 16 ELECTRICAL SPECIFICATIONS (CONVENTIONAL PRODUCTS)**	

CHAPTER 2 PIN FUNCTIONS

2.1 Pin Function List

(1) Port pins

Pin Name	I/O	Function	After Reset	Alternate Function
P00 to P07	I/O	Port 0 8-bit I/O port Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be specified by setting pull-up resistor option register 0 (PU0).	Input	–
P10 to P15	I/O	Port 1 6-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 specified by setting pull-up resistor option register 0 (PU0).	Input	–
P20	I/O	Port 2 8-bit I/O port Input/output can be specified in 1-bit units. An on-chip pull-up resistor can be specified by setting pull-up resistor option register B2 (PUB2).	Input	$\overline{\text{SCK20}}$/ASCK20
P21				SO20/TxD20
P22				SI20/RxD20
P23				$\overline{\text{SS20}}$
P24				INTP0
P25				INTP1
P26				INTP2/CPT90
P27				TI80/TO80
P30	I/O	Port 3 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 specified by setting pull-up resistor option register 0 (PU0).	Input	TO90
P31				BZO90

(2) Non-port pins

Pin Name	I/O	Function	After Reset	Alternate Function
INTP0	Input	External interrupt input for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified	Input	P24
INTP1				P25
INTP2				P26/CPT90
$\overline{\text{SCK20}}$	I/O	Serial interface (SIO10) serial clock input	Input	P20/ASCK20
SI20	Input	Serial interface (SIO20) serial data input	Input	P22/RxD20
SO20	Output	Serial interface (SIO20) serial data output	Input	P21/TxD20
$\overline{\text{SS20}}$	Input	Serial interface chip select input	Input	P23
ASCK20	Input	Serial clock input for asynchronous serial interface	Input	P20/$\overline{\text{SCK20}}$
RxD20	Input	Serial data input for asynchronous serial interface	Input	P22/SI20
TxD20	Output	Serial data output for asynchronous serial interface	Input	P21/SO20
TO90	Output	16-bit timer (TM90) output	Input	P30
BZO90	Output	Buzzer output	Input	P31
CPT90	Input	Capture edge input	Input	P26/INTP2
TO80	Output	8-bit timer (TM80) output	Input	P27/TI80
TI80	Input	External count clock input to 8-bit timer (TM80)	Input	P27/TO80
X1	Input	Connecting crystal resonator for system clock oscillation	–	–
X2	–		–	–
$\overline{\text{RESET}}$	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 flash memory programming mode and applies a high voltage when a program is written or verified.	–	–

w w w . D a t a

2.2 Description of Pin Functions

2.2.1 P00 to P07 (Port 0)

These pins constitute an 8-bit I/O port and can be set to input or output port mode in 1-bit units by using port mode register 0 (PM0). When these pins are used as an input port, an on-chip pull-up resistor can be used by setting pull-up resistor option register 0 (PU0).

2.2.2 P10 to P15 (Port 1)

These pins constitute a 6-bit I/O port and can be set to input or output port mode in 1-bit units by using port mode register 1 (PM1). When these pins are used as an input port, an on-chip pull-up resistor can be used by setting pull-up resistor option register 0 (PU0).

2.2.3 P20 to P27 (Port 2)

These pins constitute an 8-bit I/O port. In addition, these pins provide a function to perform input/output to/from the timer, to input/output the data and clock of the serial interface, and to input the external interrupt.

Port 2 can be set to the following operation modes in 1-bit units.

(1) Port mode

In port mode, P20 to P27 function as an 8-bit I/O port. Port 2 can be set to input or output mode in 1-bit units by using port mode register 2 (PM2). For P20 to P27, whether to use on-chip pull-up resistors can be specified in 1-bit units by using pull-up resistor option register B2 (PUB2), regardless of the setting of port mode register 2 (PM2).

(2) Control mode

In this mode, P20 to P27 function as the timer input/output and the serial interface data and clock input/output.

(a) TI80

This is the external clock input pin for the 8-bit timer/event counter 80.

(b) TO80

This is the timer output pin of the 8-bit timer/event counter 80.

(c) INTP0 to INTP2

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

(d) CPT90

This is the capture edge input pin of the 16-bit timer counter 90.

(e) SI20, SO20

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

(f) $\overline{\text{SCK20}}$

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

(g) $\overline{\text{SS20}}$

This is the chip select input pin of the serial interface.

(h) RxD20, TxD20

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

(i) ASCK20

This is the serial clock input pin of the asynchronous serial interface.

Caution When using P20 to P27 as serial interface pins, the input/output mode and output latch must be set according to the functions to be used. For details of the setting, see Table 9-2 Serial Interface 20 Operation Mode Settings.

2.2.4 P30, P31 (Port 3)

These pins constitute a 2-bit I/O port. In addition, these pins function as the timer output and the buzzer output. Port 3 can be set to the following operation modes in 1-bit units.

(1) Port mode

When this port is used as an input port, an on-chip pull-up resistor can be used by setting pull-up resistor option register 0 (PU0).

(2) Control mode

In this mode, P30 and P31 function as the timer output and the buzzer output.

(a) TO90

This is the output pin of the 16-bit timer 90.

(b) BZO90

This is the buzzer output pin of the 16-bit timer 90.

2.2.5 $\overline{\text{RESET}}$

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

2.2.6 X1, X2

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

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

2.2.7 V_{DD}

This pin supplies positive power.

2.2.8 V_{SS}

This pin is the ground potential pin.

2.2.9 V_{PP} (μPD78F9076 only)

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

Handle this pin in either of the following ways.

- Connect a 10 kΩ pull-down resistor to the pin.
- Provide a jumper on the board so that the pin is connected to a dedicated flash programmer in programming mode and to V_{SS} during normal operation.

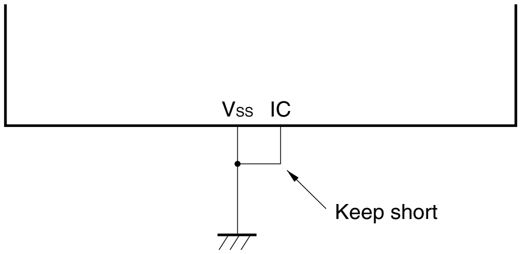
★ If there is a long wiring length between the V_{PP} pin and the V_{SS} pin or external noise superimposed on the V_{PP} pin, the user program may not run correctly.

2.2.10 IC (mask ROM version only)

The IC (Internally Connected) pin is used to set the μPD789071, 789072, and 789074 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 external noise superimposed on the IC pin, the user program may not run correctly.

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



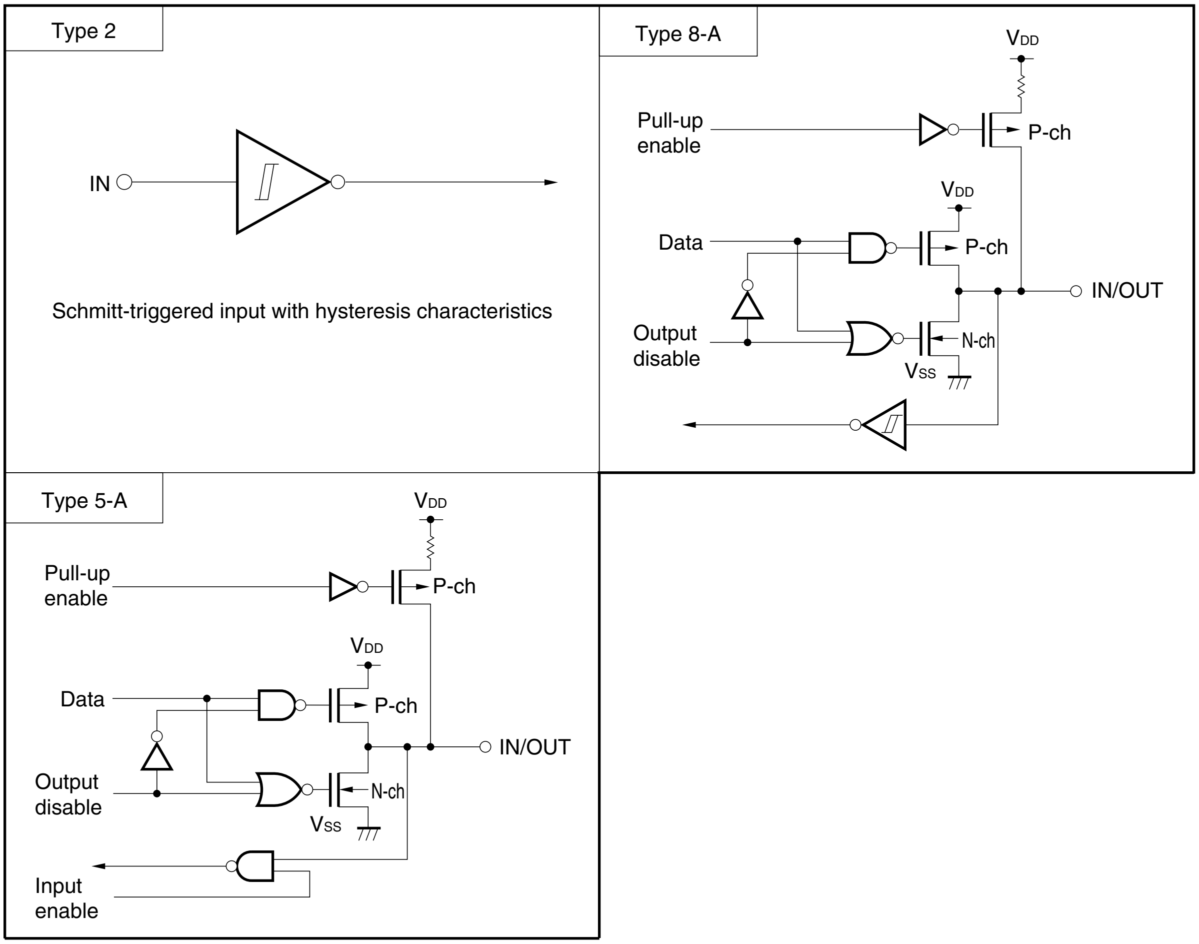
2.3 Pin I/O Circuits and Recommended Connection of Unused Pins

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 3-1.

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 P07	5-A	I/O	Input: Connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P10 to P15			
P20/$\overline{\text{SCK20}}$/ASCK20	8-A		
P21/SO20/TxD20			
P22/SI20/RxD20			
P23/$\overline{\text{SS20}}$			
P24/INTP0			Input: Connect to V_{SS} via a resistor. Output: Leave open.
P25/INTP1			
P26/INTP2/CPT90			
P27/TI80/TO80			Input: Connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P30/TO90	5-A		
P31/BZO90			
$\overline{\text{RESET}}$	2	Input	–
IC	–	–	Connect directly to V_{SS}.
V_{PP}			Connect to a 10 kΩ pull-down resistor or directly to V_{SS}.

Figure 2-1. Pin I/O Circuits


Type 2
IN
Type 8-A
VDD
Pull-up enable
P-ch
VDD
Data
P-ch
IN/OUT
Output disable
N-ch
VSS
Type 5-A
VDD
Pull-up enable
P-ch
VDD
Data
P-ch
IN/OUT
Output disable
N-ch
VSS
Input enable

Schmitt-triggered input with hysteresis characteristics

CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

Products in the μPD789074 Subseries can each access up to 64 KB of memory space. Figures 3-1 through 3-4 show the memory maps.

Figure 3-1. Memory Map (μPD789071)

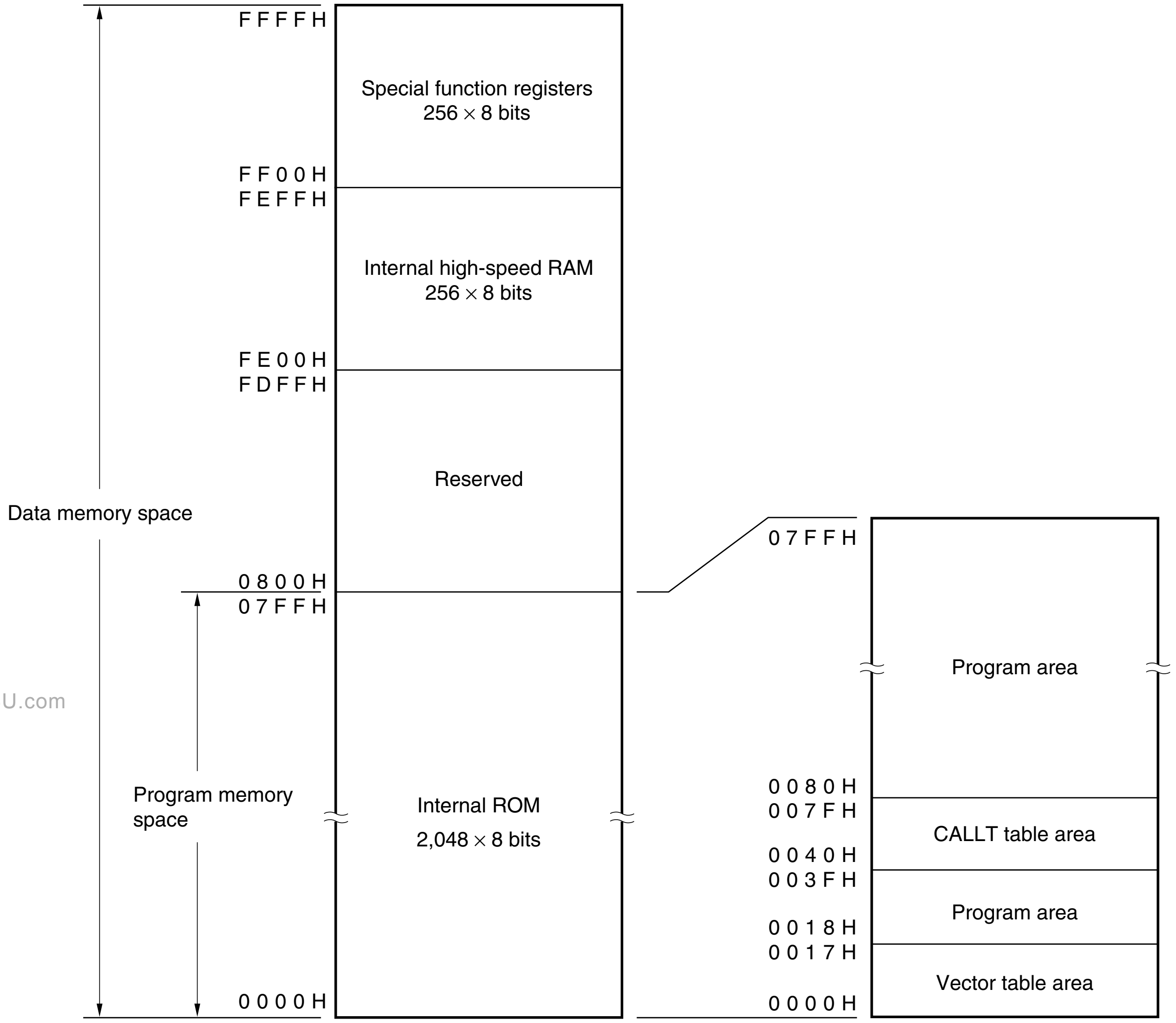

FFFFH
Special function registers
256 × 8 bits
FF00H
FEFFH
Internal high-speed RAM
256 × 8 bits
FE00H
FDFFH
Reserved
Data memory space
0800H
07FFH
Program memory space
Internal ROM
2,048 × 8 bits
0000H
07FFH
Program area
0080H
007FH
CALLT table area
0040H
003FH
Program area
0018H
0017H
Vector table area
0000H

Figure 3-2. Memory Map (μPD789072)

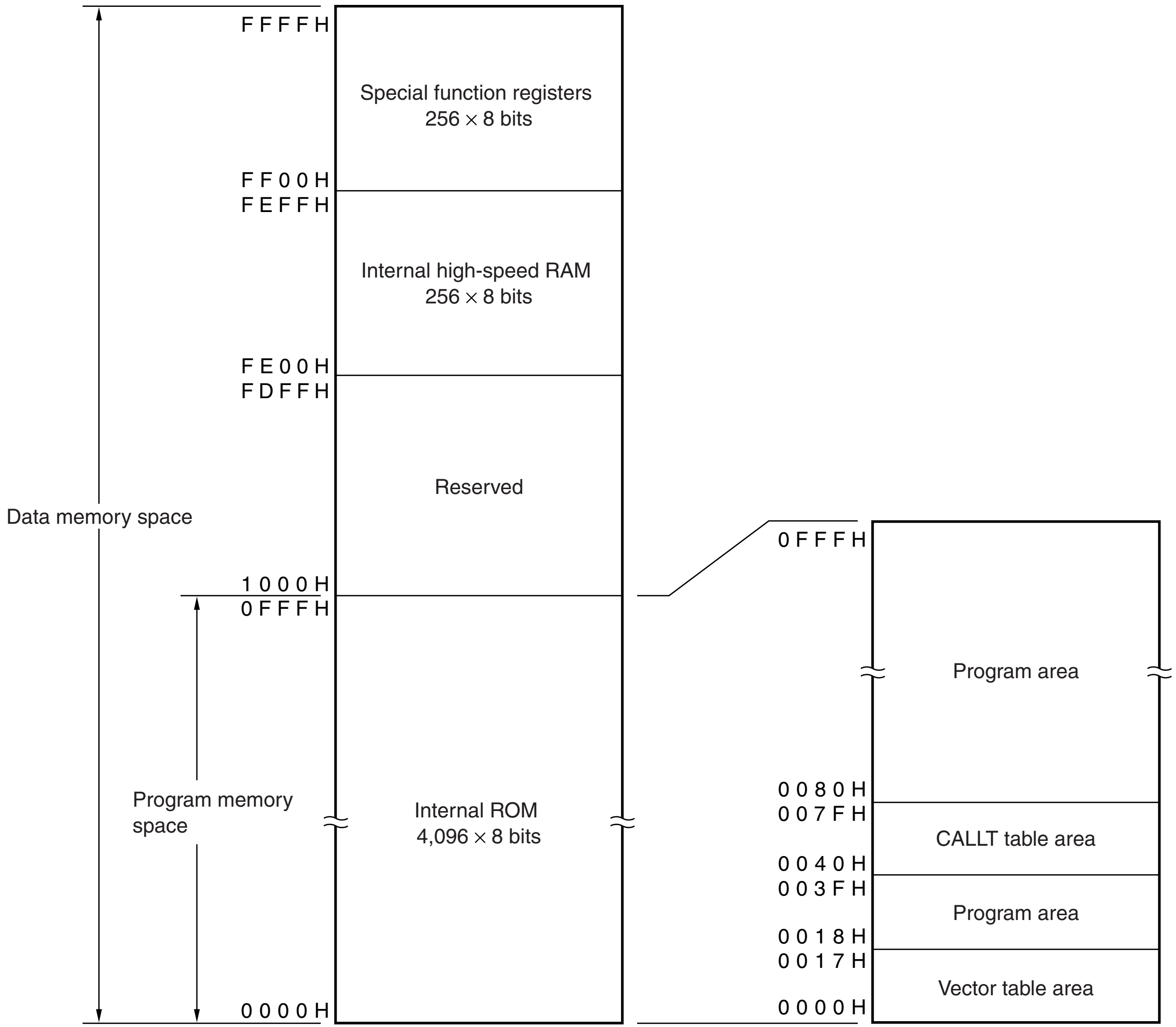

FFFFH
Special function registers
256 × 8 bits
FF00H
FEFFH
Internal high-speed RAM
256 × 8 bits
FE00H
FDFFH
Reserved
Data memory space
1000H
0FFFH
Program memory space
Internal ROM
4,096 × 8 bits
0000H
0FFFH
Program area
0080H
007FH
CALLT table area
0040H
003FH
Program area
0018H
0017H
Vector table area
0000H

Figure 3-3. Memory Map (μPD789074)

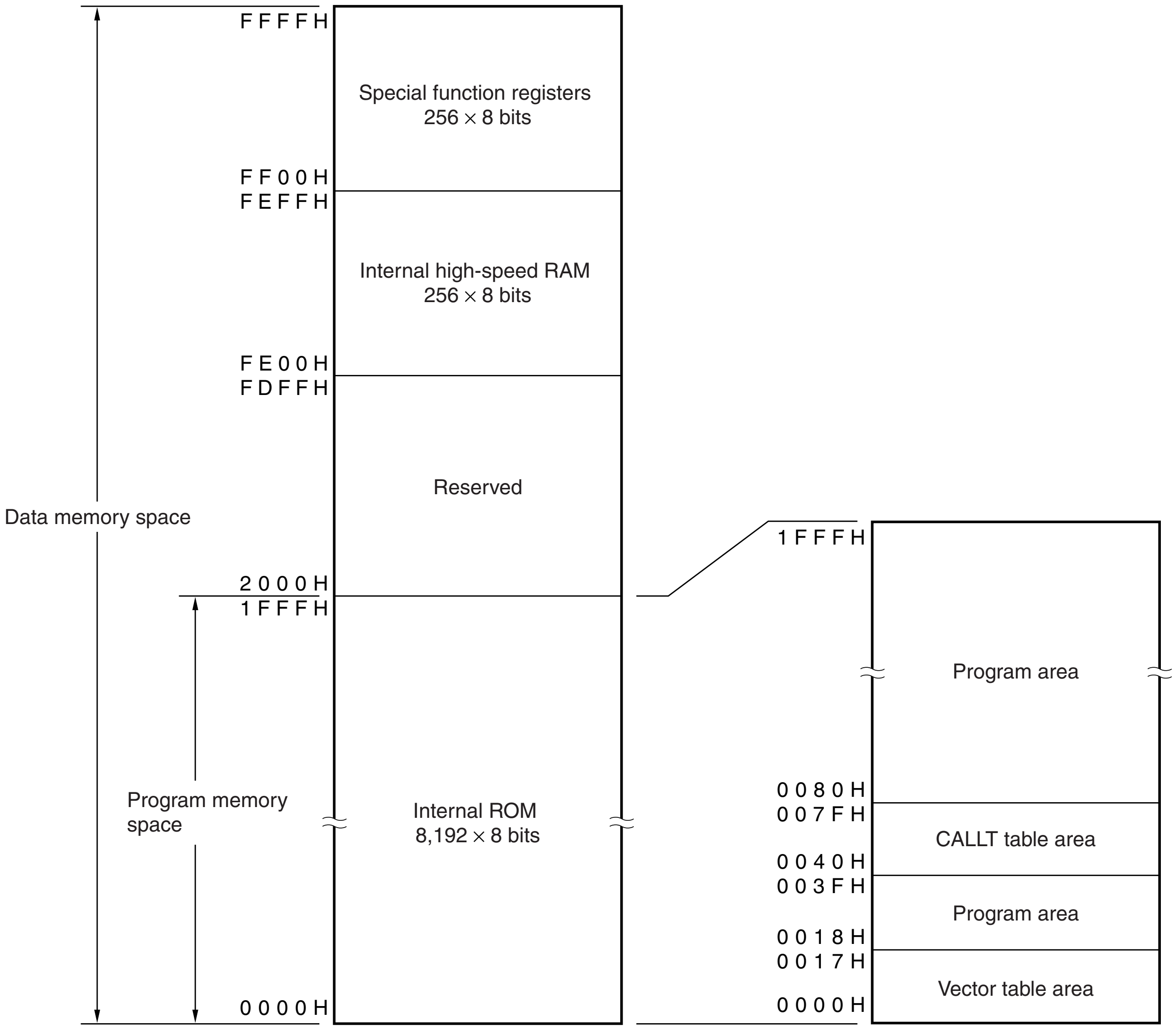
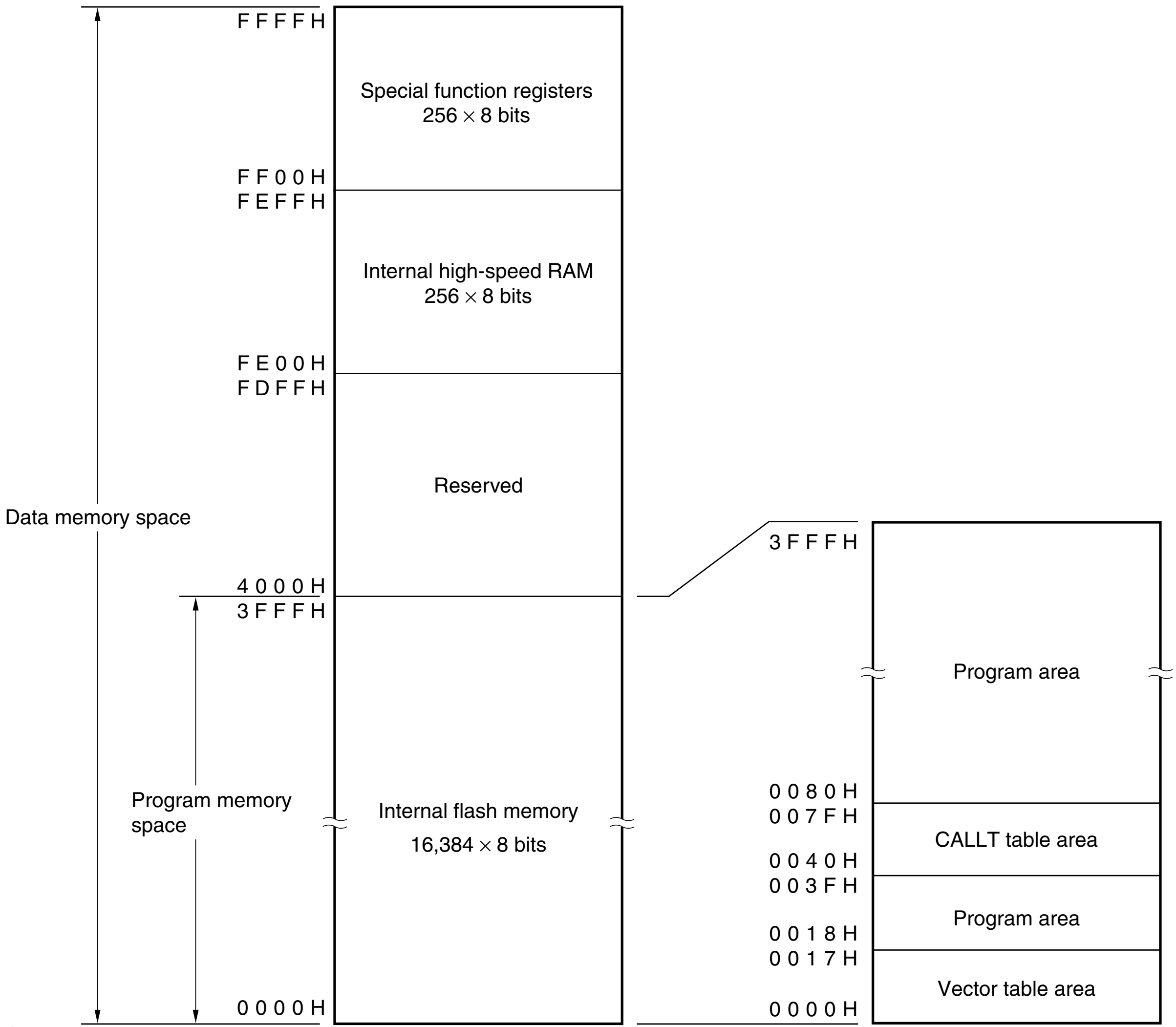

FFFFH
Special function registers
256 × 8 bits
FF00H
FEFFH
Internal high-speed RAM
256 × 8 bits
FE00H
FDFFH
Reserved
Data memory space
2000H
1FFFH
Program memory space
Internal ROM
8,192 × 8 bits
0000H
1FFFH
Program area
0080H
007FH
CALLT table area
0040H
003FH
Program area
0018H
0017H
Vector table area
0000H

Figure 3-4. Memory Map (μPD78F9076)


FFFFH
Special function registers
256 × 8 bits
FF00H
FEFFH
Internal high-speed RAM
256 × 8 bits
FE00H
FDFFH
Reserved
Data memory space
4000H
3FFFH
Program memory space
Internal flash memory
16,384 × 8 bits
0000H
3FFFH
Program area
0080H
007FH
CALLT table area
0040H
003FH
Program area
0018H
0017H
Vector table area
0000H

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).

Products in the μPD789074 Subseries provide the following internal ROM (or flash memory) containing the following capacities.

Table 3-1. Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
μPD789071	Mask ROM	2,048 × 8 bits
μPD789072		4,096 × 8 bits
μPD789074		8,192 × 8 bits
μPD78F9076	Flash memory	16,384 × 8 bits

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

(1) Vector table area

The 24-byte area of addresses 0000H to 0017H 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 program 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	000CH	INTSR20/INTCSI20
0004H	INTWDT	000EH	INTST20
0006H	INTP0	0014H	INTTM80
0008H	INTP1	0016H	INTTM90
000AH	INTP2		

(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 (internal high-speed RAM) space

The μPD789074 Subseries provides 256-byte internal high-speed RAM.

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

Each product of the μPD789074 Subseries is provided with a wide range of addressing modes to make memory manipulation as efficient as possible. The data memory area (FE00H to FFFFH) can be accessed using a unique addressing mode according to its use, such as a special function register (SFR). Figures 3-5 through 3-8 illustrate the data memory addressing.

Figure 3-5. Data Memory Addressing (μPD789071)

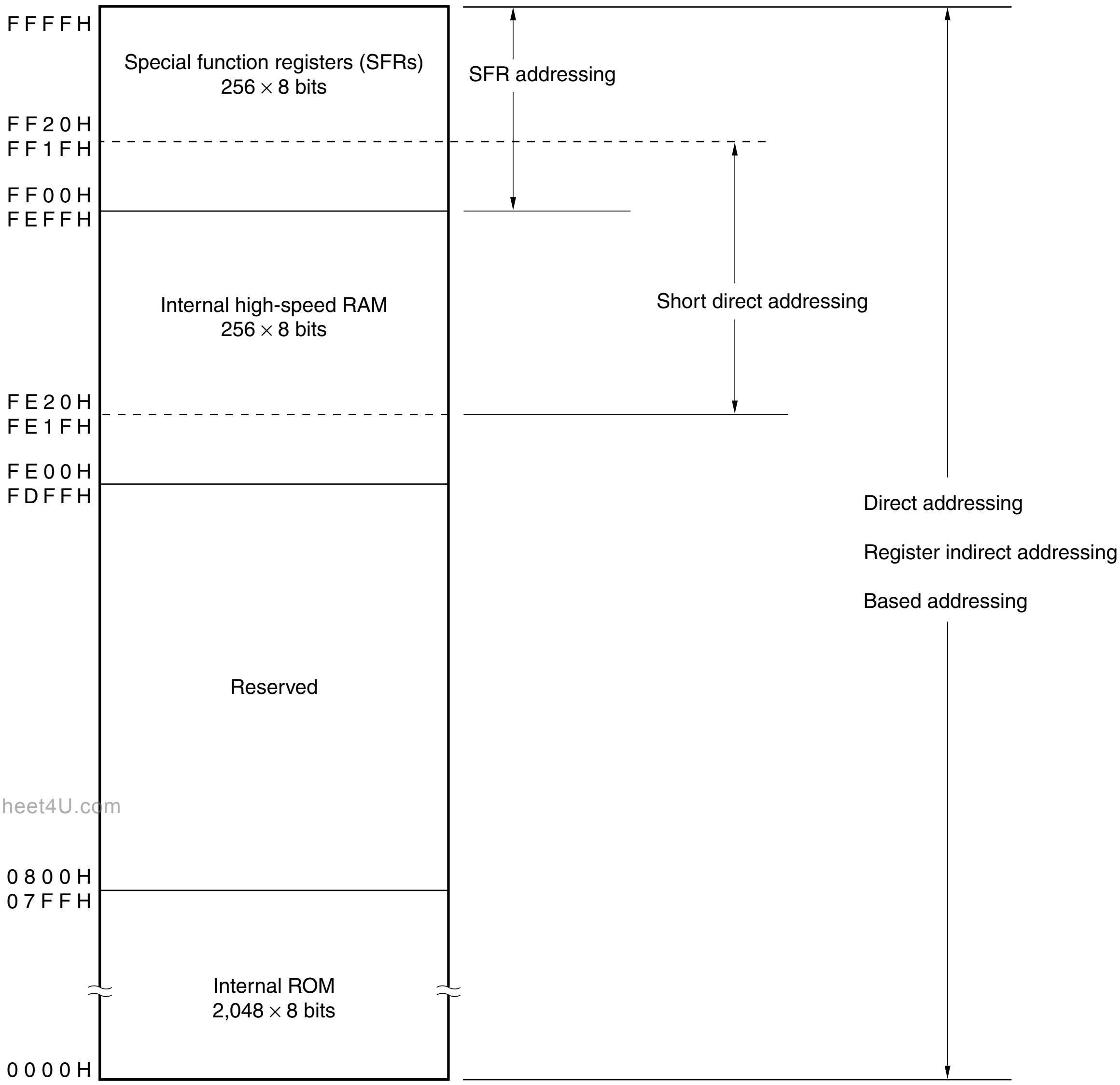
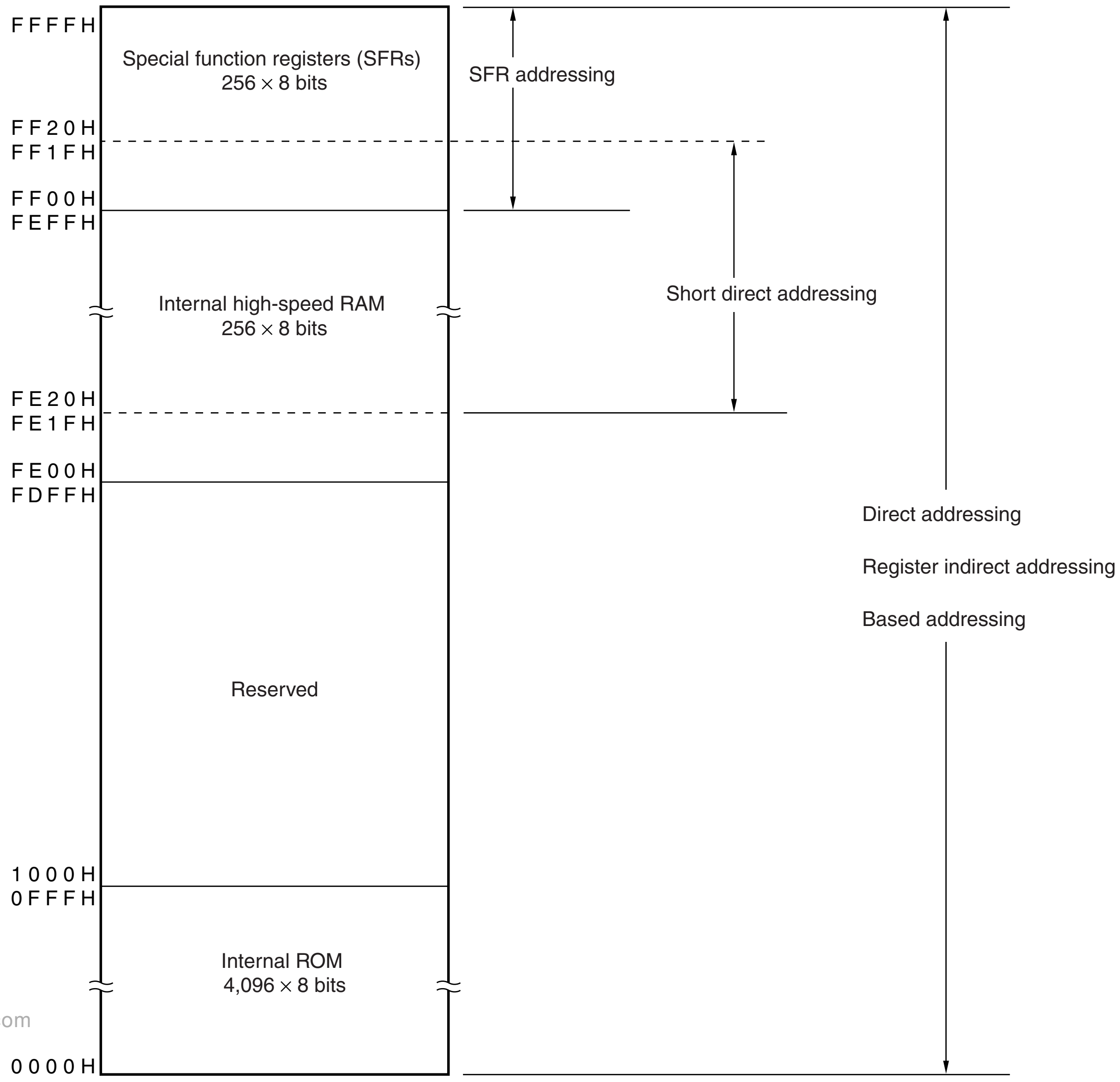

FFFFH
FF20H
FF1FH
FF00H
FEFFH
FE20H
FE1FH
FE00H
FDFFH
0800H
07FFH
0000H
Special function registers (SFRs)
256 × 8 bits
Internal high-speed RAM
256 × 8 bits
Reserved
Internal ROM
2,048 × 8 bits
SFR addressing
Short direct addressing
Direct addressing
Register indirect addressing
Based addressing

Figure 3-6. Data Memory Addressing (μPD789072)



FFFFH
Special function registers (SFRs)
256 × 8 bits
SFR addressing
FF20H
FF1FH
FF00H
FEFFH
Internal high-speed RAM
256 × 8 bits
Short direct addressing
FE20H
FE1FH
FE00H
FDFFH
Direct addressing
Register indirect addressing
Based addressing
Reserved
1000H
0FFFH
Internal ROM
4,096 × 8 bits
0000H

Figure 3-7. Data Memory Addressing (μPD789074)

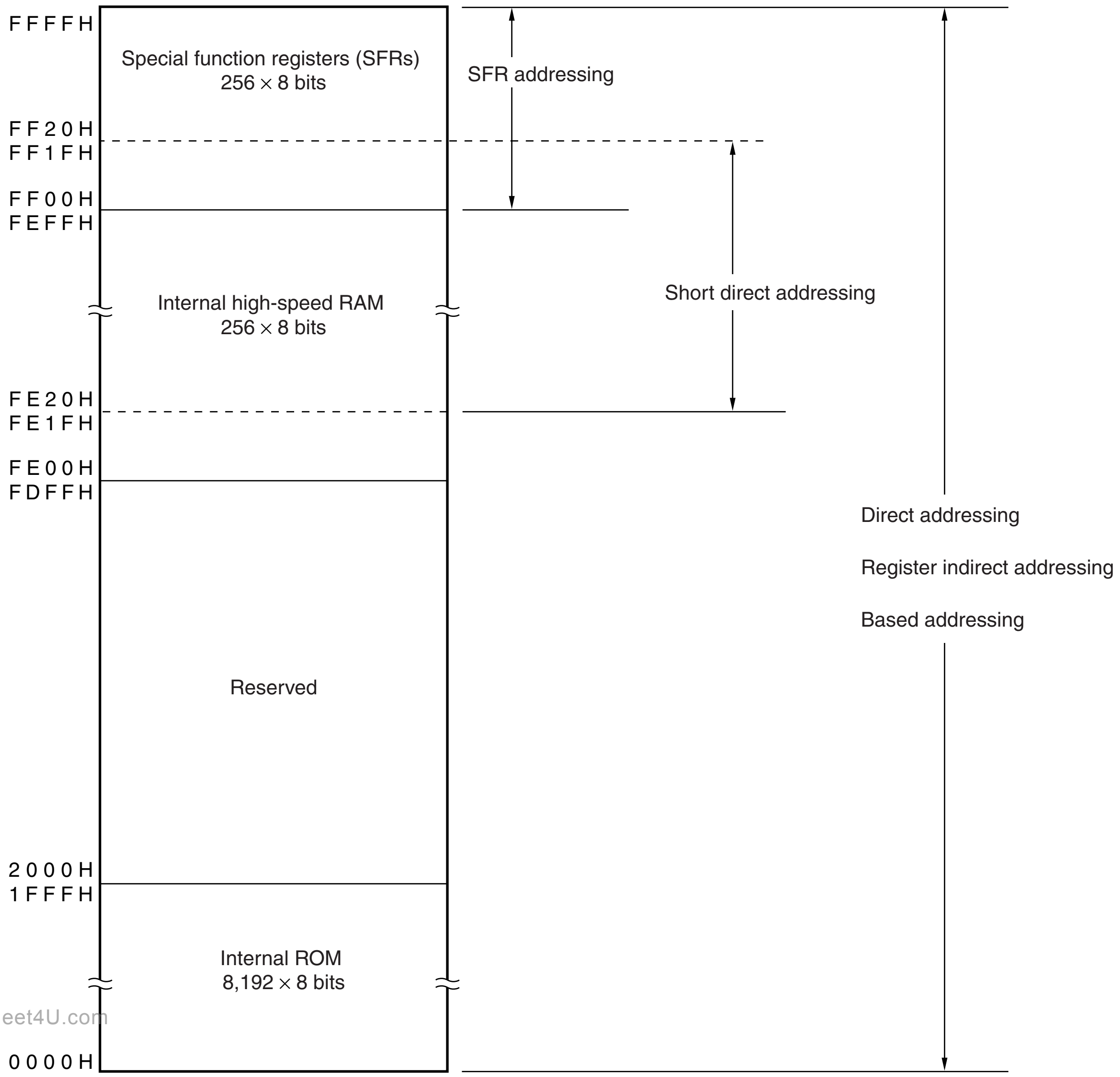
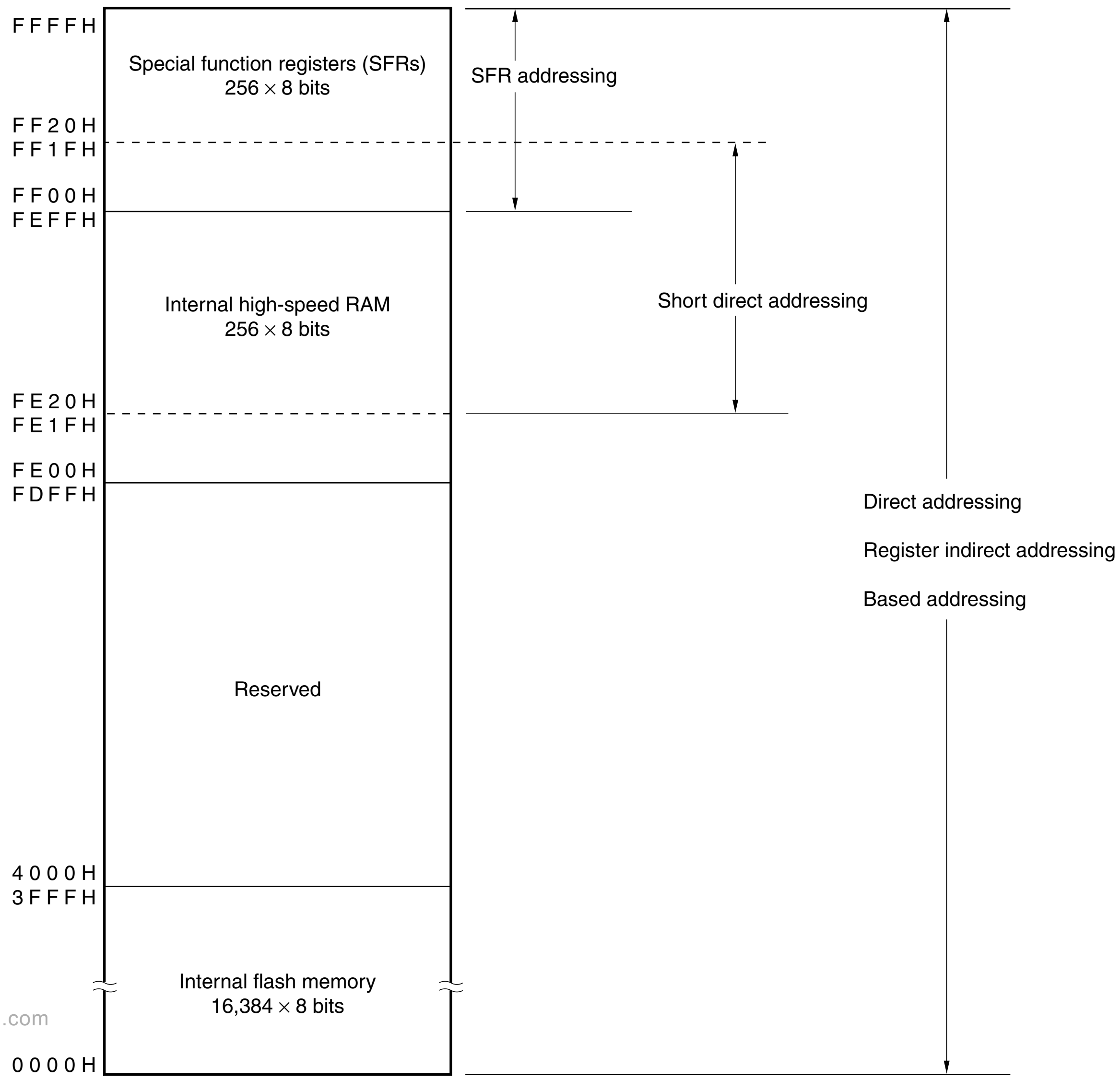

FFFFH
Special function registers (SFRs)
256 × 8 bits
SFR addressing
FF20H
FF1FH
FF00H
FEFFH
Internal high-speed RAM
256 × 8 bits
Short direct addressing
FE20H
FE1FH
FE00H
FDFFH
Direct addressing
Register indirect addressing
Based addressing
Reserved
2000H
1FFFH
Internal ROM
8,192 × 8 bits
0000H

Figure 3-8. Data Memory Addressing (μPD78F9076)


FFFFH
Special function registers (SFRs)
256 × 8 bits
SFR addressing
FF20H
FF1FH
FF00H
FEFFH
Internal high-speed RAM
256 × 8 bits
Short direct addressing
FE20H
FE1FH
FE00H
FDFFH
Direct addressing
Register indirect addressing
Based addressing
Reserved
4000H
3FFFH
Internal flash memory
16,384 × 8 bits
0000H

3.2 Processor Registers

The μPD789074 Subseries provides 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 3-9. 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 the PSW to 02H.

Figure 3-10. 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 interrupts are disabled.

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

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 3-11. 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 3-12 and 3-13.

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

Figure 3-12. Data to Be Saved to Stack Memory

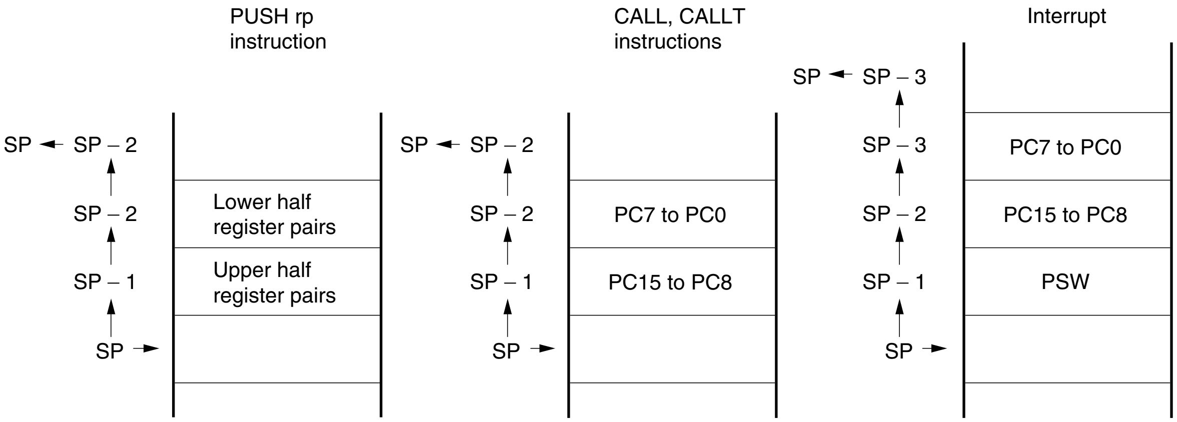
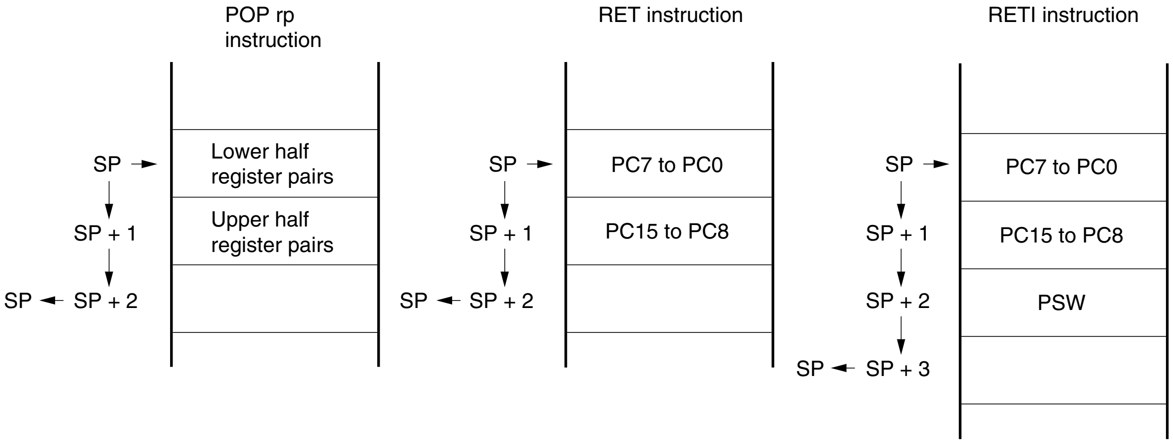

PUSH rp instruction
SP ← SP – 2
SP – 2
SP – 1
SP
Lower half register pairs
Upper half register pairs
CALL, CALLT instructions
SP ← SP – 2
SP – 2
SP – 1
SP
PC7 to PC0
PC15 to PC8
Interrupt
SP ← SP – 3
SP – 3
SP – 2
SP – 1
SP
PC7 to PC0
PC15 to PC8
PSW

Figure 3-13. Data to Be Restored from Stack Memory


POP rp instruction
SP
SP + 1
SP ← SP + 2
Lower half register pairs
Upper half register pairs
RET instruction
SP
SP + 1
SP ← SP + 2
PC7 to PC0
PC15 to PC8
RETI instruction
SP
SP + 1
SP + 2
SP ← SP + 3
PC7 to PC0
PC15 to PC8
PSW

3.2.2 General-purpose registers

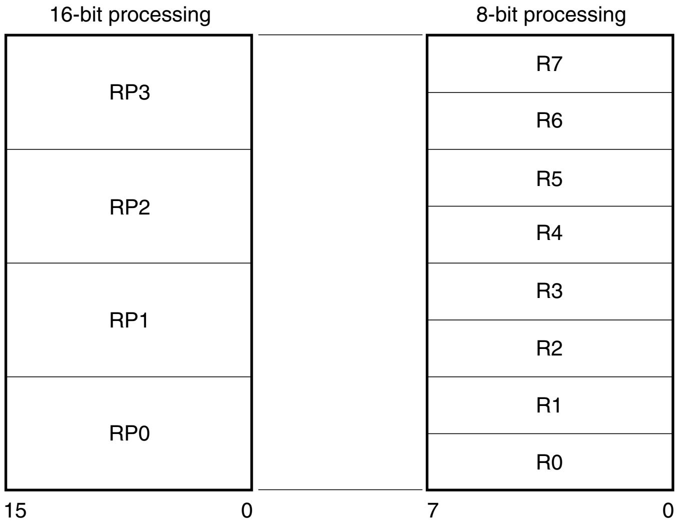
The general-purpose registers consist of eight 8-bit registers (X, A, C, B, E, D, L, and H).

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

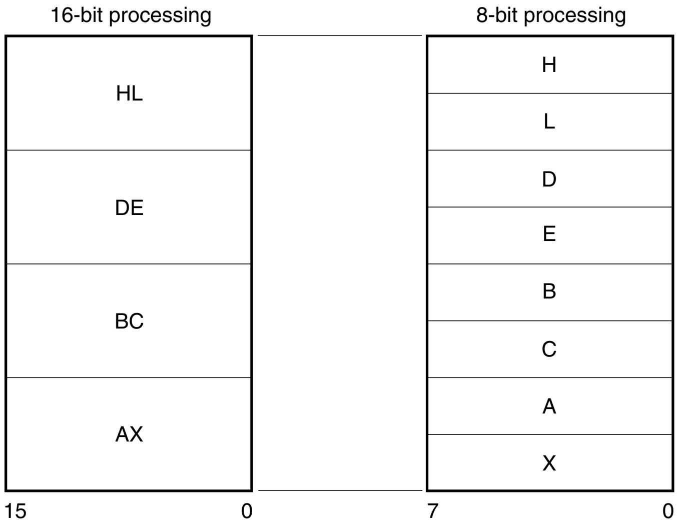
Registers can be described in terms of function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL) and absolute names (R0 to R7 and RP0 to RP3).

Figure 3-14. General-Purpose Register Configuration

(a) Absolute names


16-bit processing
8-bit processing
RP3
RP2
RP1
RP0
R7
R6
R5
R4
R3
R2
R1
R0
15
0
7
0

(b) Function names


16-bit processing
8-bit processing
HL
DE
BC
AX
H
L
D
E
B
C
A
X
15
0
7
0

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
  Describe 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
  Describe 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
  Describe 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		√	√	–	
FF16H	16-bit compare register 90	CR90[Note 1]	W	–	√[Note 2]	√[Note 3]	FFFFH
FF17H							
FF18H	16-bit timer counter 90	TM90[Note 1]	R	–	√[Note 2]	√[Note 3]	0000H
FF19H							
FF1AH	16-bit capture register 90	TCP90[Note 1]		–	√[Note 2]	√[Note 3]	Undefined
FF1BH							
FF20H	Port mode register 0	PM0	R/W	√	√	–	FFH
FF21H	Port mode register 1	PM1		√	√	–	
FF22H	Port mode register 2	PM2		√	√	–	
FF23H	Port mode register 3	PM3		√	√	–	
FF32H	Pull-up resistor option register B2	PUB2		√	√	–	00H
FF42H	Watchdog timer clock selection register 2	WDCS		–	√	–	
FF48H	16-bit timer mode control register 90	TMC90		√	√	–	
FF49H	Buzzer output control register 90	BZC90		√	√	–	
FF50H	8-bit compare register 80	CR80	W	–	√	–	Undefined
FF51H	8-bit timer counter 80	TM80	R	–	√	–	00H
FF53H	8-bit timer mode control register 80	TMC80	R/W	√	√	–	
FF70H	Asynchronous serial interface mode register 20	ASIM20		√	√	–	
FF71H	Asynchronous serial interface status register 20	ASIS20	R	√	√	–	
FF72H	Serial operation mode register 20	CSIM20	R/W	√	√	–	
FF73H	Baud rate generator control register 20	BRGC20		–	√	–	

Notes 1. These SFRs are for 16-bit access only.

2. CR90, TM90, and TCP90 are designed only for 16-bit access. In direct addressing, however, 8-bit access can also be performed.

3. 16-bit access is allowed only in short direct addressing.

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	
FF74H	Transmission shift register 20	TXS20	SIO20	W	–	√	–	FFH
	Receive buffer register 20	RXB20		R	–	√	–	Undefined
FFE0H	Interrupt request flag register 0	IF0		R/W	√	√	–	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
FFF7H	Pull-up resistor option register 0	PU0			√	√	–	
FFF9H	Watchdog timer mode register	WDTM			√	√	–	
FFFAH	Oscillation stabilization time selection register	OSTS			–	√	–	04H
FFFBH	Processor clock control register	PCC			√	√	–	02H

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 to branch by the following addressing (for details of each instruction, refer to **78K/0S Series Instructions User's Manual (U11047E)**).

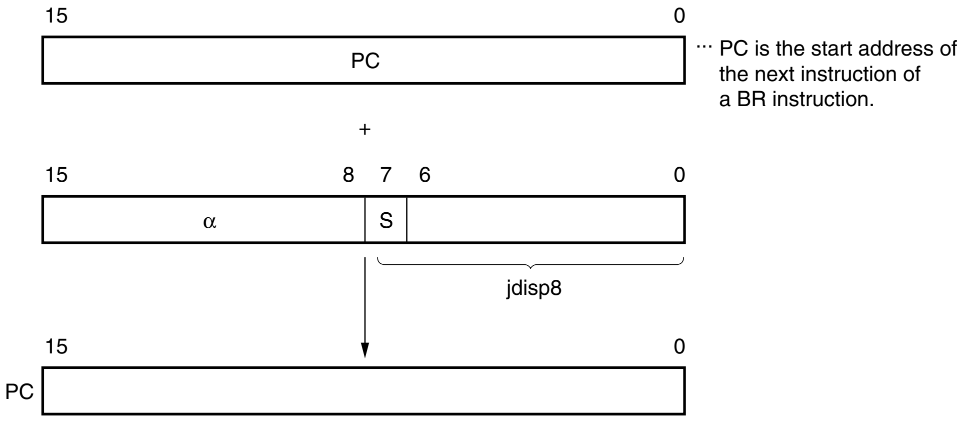
3.3.1 Relative addressing

[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]


15
0
PC
··· PC is the start address of the next instruction of a BR instruction.
+
15
8
7
6
0
α
S
jdisp8
15
0
PC

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

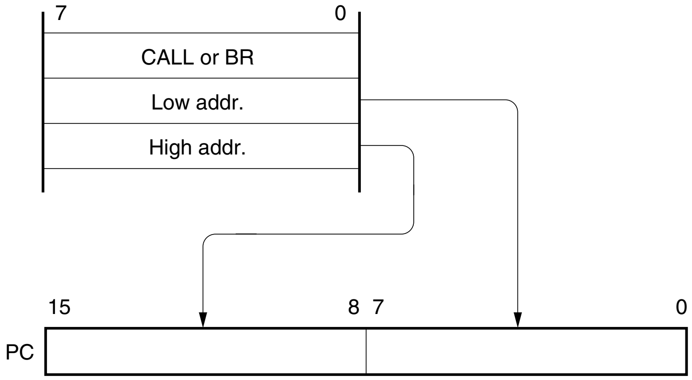
3.3.2 Immediate addressing

[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



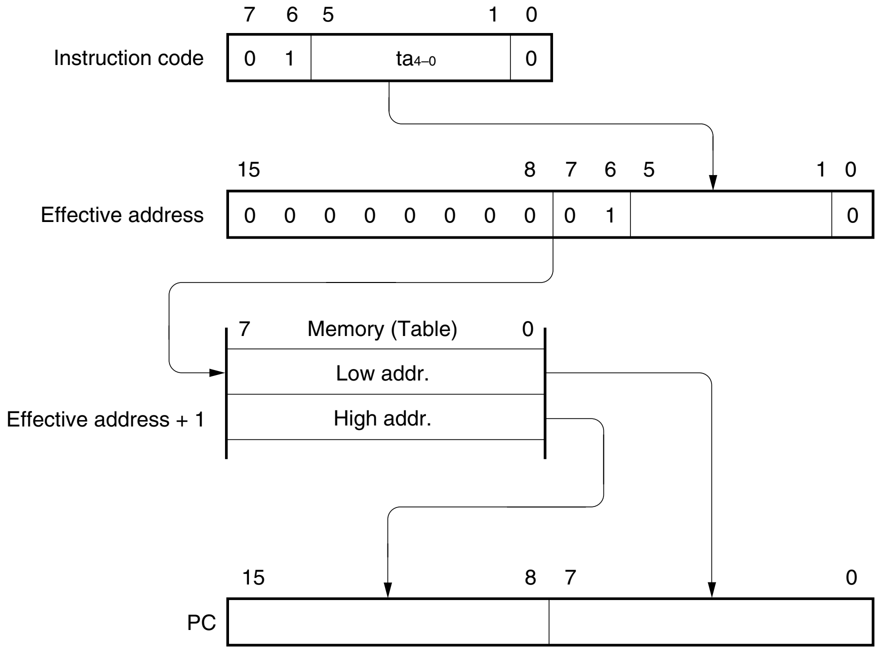
3.3.3 Table indirect addressing

[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]


7 6 5 1 0
Instruction code
0 1 ta_{4-0} 0
15 8 7 6 5 1 0
Effective address
0 0 0 0 0 0 0 0 0 1 0
7 Memory (Table) 0
Low addr.
High addr.
Effective address + 1
15 8 7 0
PC

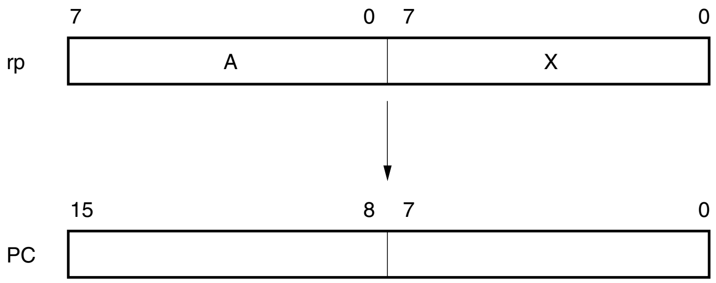
3.3.4 Register addressing

[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]


7 0 7 0
rp
A
X
15 8 7 0
PC

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]

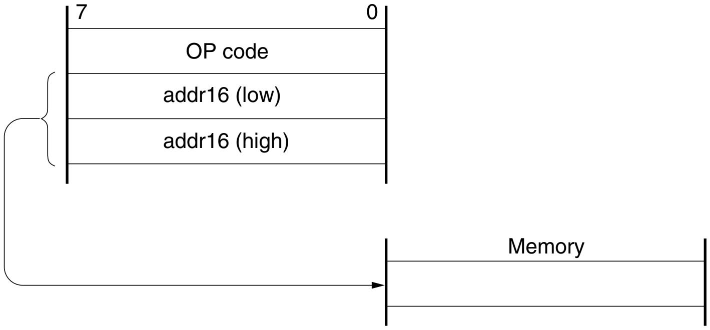
Identifier	Description
addr16	Label or 16-bit immediate data

[Description example]

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

Instruction code	0	0	1	0	1	0	0	1	OP Code
	0	0	0	0	0	0	0	0	00H
	1	1	1	1	1	1	1	0	FEH

[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]

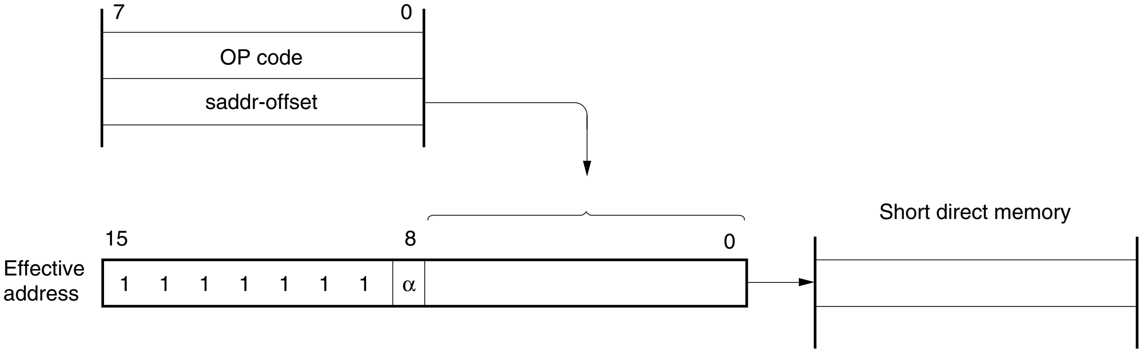
Identifier	Description
saddr	Label or FE20H to FF1FH immediate data
saddrp	Label or FE20H to FF1FH immediate data (even address only)

[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	OP code
	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 spaces FF00H to FFFFH. However, SFRs mapped at FF00H to FF1FH can also be accessed with short direct addressing.

[Operand format]

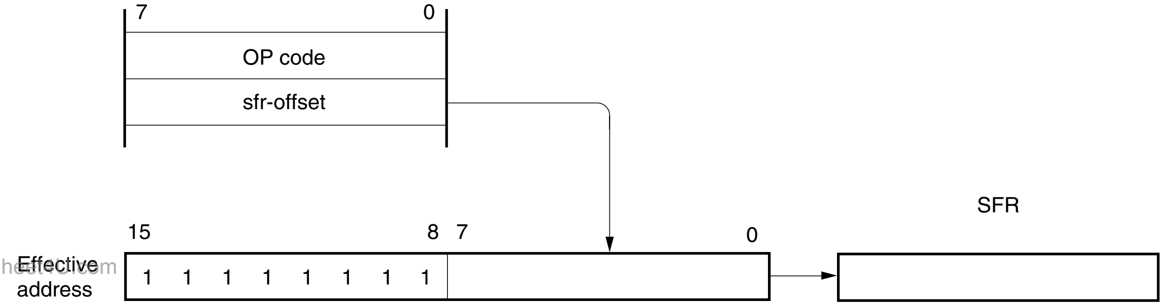
Identifier	Description
sfr	Special function register name

[Description example]

MOV PM0, A; When selecting PM0 for sfr

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

[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.

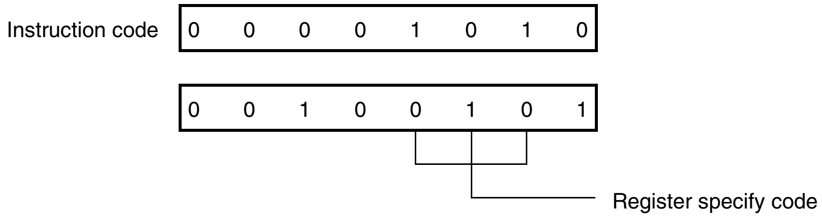
[Operand format]

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

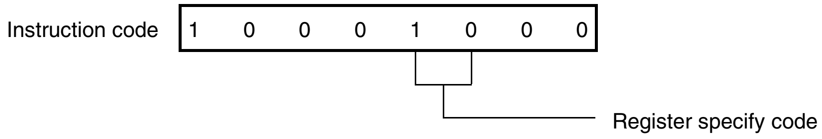
'r' and 'rp' can be described with absolute names (R0 to R7 and RP0 to RP3) as well as function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL).

★ **[Description example]**

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


Instruction code
0 0 0 0 1 0 1 0
0 0 1 0 0 1 0 1
Register specify code

INCW DE; When selecting the DE register pair for rp


Instruction code
1 0 0 0 1 0 0 0
Register specify code

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]

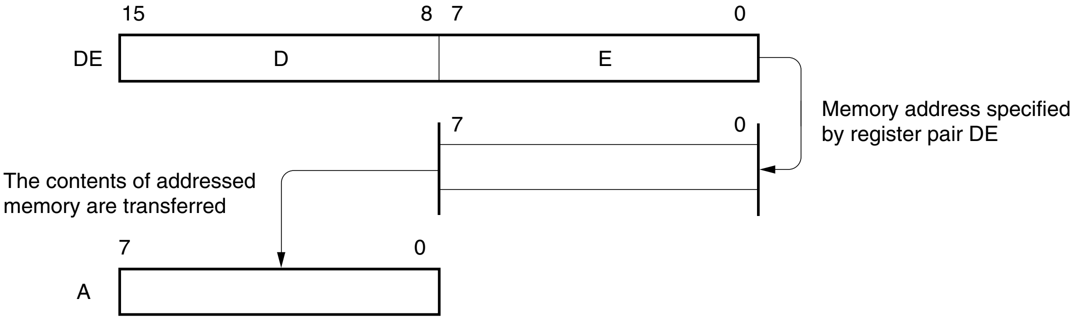
Identifier	Description
–	[DE], [HL]

[Description example]

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

Instruction code	0	0	1	0	1	0	1	1

[Illustration]


15
8 7
0
DE
D
E
Memory address specified
by register pair DE
7
0
The contents of addressed
memory are transferred
7
0
A

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 PORT FUNCTIONS

4.1 Port Functions

The μPD789074 Subseries is provided with the ports shown in Figure 4-1. These ports enable several types of control. Table 4-1 lists the functions of each port.

These ports have digital I/O port functions as well as alternate functions. For the alternate functions, refer to **2.1 Pin Function List**.

Figure 4-1. Port Types

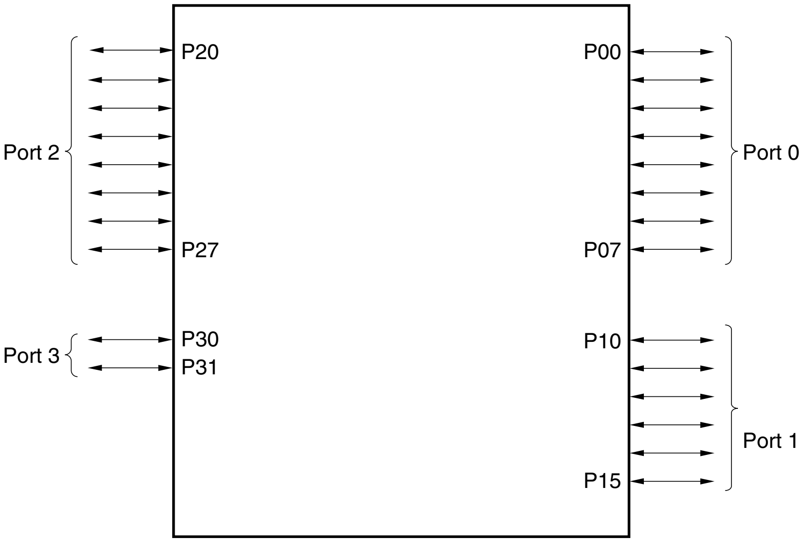

Port 2
P20
P27
Port 3
P30
P31
P00
P07
Port 0
P10
P15
Port 1

Table 4-1. Port Functions

Pin Name	I/O	Function	After Reset	Alternate Function
P00 to P07	I/O	Port 0 8-bit I/O port Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be specified by setting pull-up resistor option register 0 (PU0).	Input	–
P10 to P15	I/O	Port 1 6-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 specified by setting pull-up resistor option register 0 (PU0).	Input	–
P20	I/O	Port 2 8-bit I/O port Input/output can be specified in 1-bit units. An on-chip pull-up resistor can be specified by setting pull-up resistor option register B2 (PUB2).	Input	$\overline{\text{SCK20}}$/ASCK20
P21				SO20/TxD20
P22				SI20/RxD20
P23				$\overline{\text{SS20}}$
P24				INTP0
P25				INTP1
P26				INTP2/CPT90
P27				TI80/TO80
P30	I/O	Port 3 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 specified by setting pull-up resistor option register 0 (PU0).	Input	TO90
P31				BZO90

4.2 Port Configuration

Ports include the following hardware.

Table 4-2. Configuration of Port

Parameter	Configuration
Control registers	Port mode registers (PMm: m = 0 to 3) Pull-up resistor option register 0 (PU0) Pull-up resistor option register B2 (PUB2)
Ports	CMOS I/O: 24
Pull-up resistors	Software control: 24

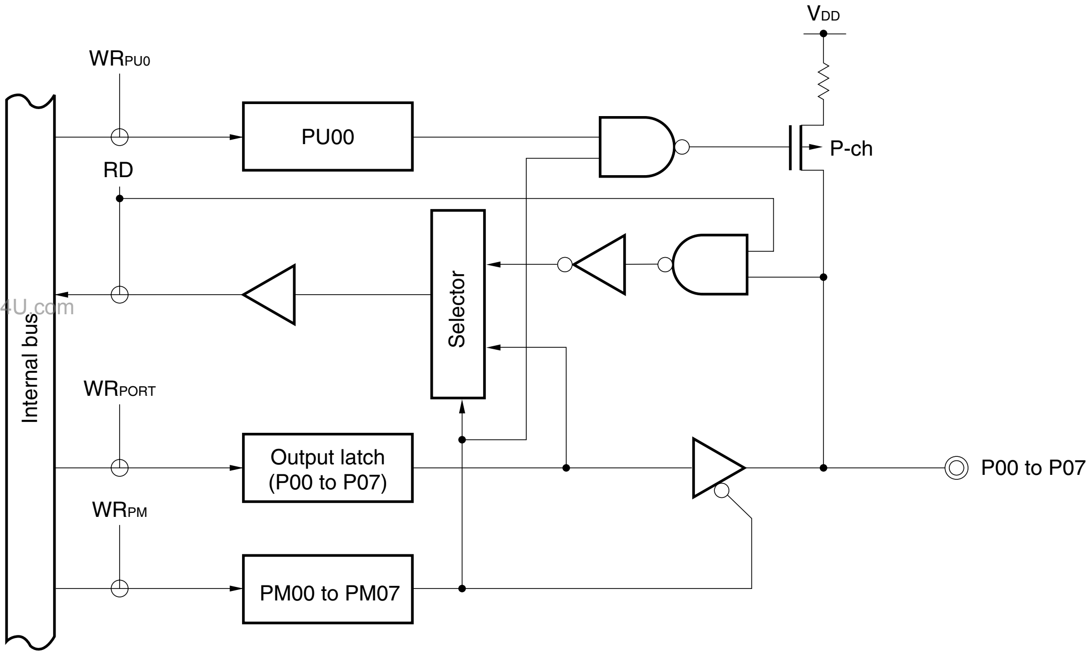
4.2.1 Port 0

This is an 8-bit I/O port with an output latch. Port 0 can be set to input or output mode in 1-bit units by using port mode register 0 (PM0). When pins P00 to P07 are used as input port pins, on-chip pull-up resistors can be connected in 8-bit units by setting pull-up resistor option register 0 (PU0).

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

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

Figure 4-2. Block Diagram of P00 to P07


VDD
WRPU0
PU00
P-ch
RD
Selector
Internal bus
WRPORT
Output latch
(P00 to P07)
P00 to P07
WRPM
PM00 to PM07

PU0: Pull-up resistor option register 0
PM: Port mode register
RD: Port 0 read signal
WR: Port 0 write signal

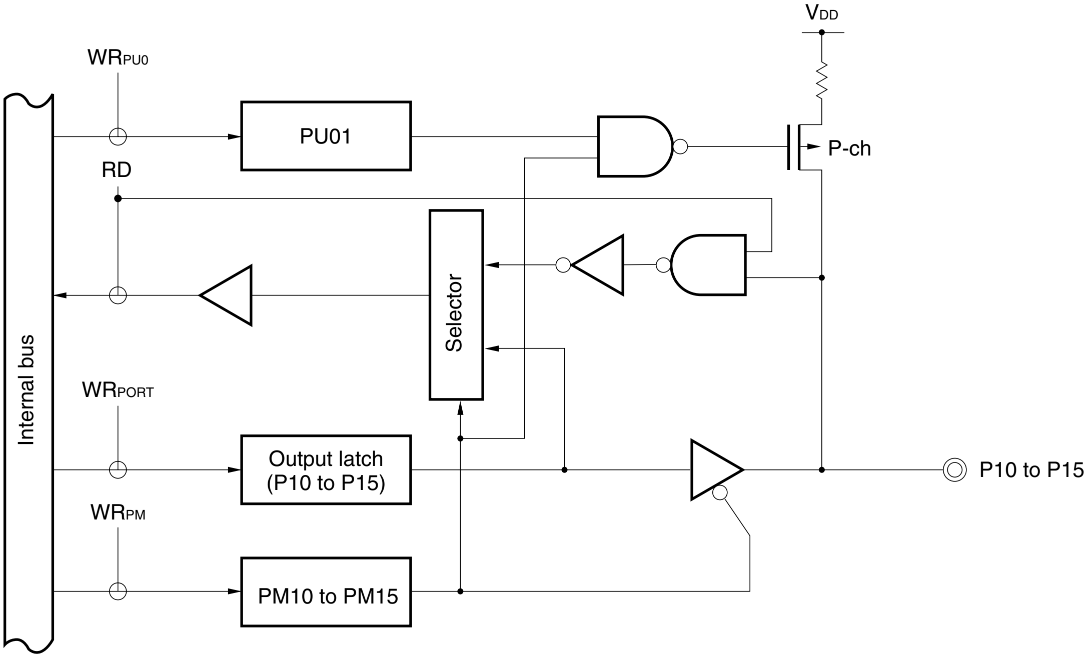
4.2.2 Port 1

This is a 6-bit I/O port with an output latch. Port 1 can be set to input or output mode in 1-bit units by using port mode register 1 (PM1). When pins P10 to P15 are used as input port pins, on-chip pull-up resistors can be connected in 6-bit units by setting pull-up resistor option register 0 (PU0).

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

Figure 4-3 shows a block diagram of port 1.

Figure 4-3. Block Diagram of P10 to P15


VDD
WRPU0
PU01
P-ch
RD
Selector
Internal bus
WRPORT
Output latch (P10 to P15)
P10 to P15
WRPM
PM10 to PM15

PU0:	Pull-up resistor option register 0
PM:	Port mode register
RD:	Port 1 read signal
WR:	Port 1 write signal

4.2.3 Port 2

This is an 8-bit I/O port with an output latch. Port 2 can be set to input or output mode in 1-bit units by using port mode register 2 (PM2). For pins P20 to P27, on-chip pull-up resistors can be connected in 1-bit units by setting pull-up resistor option register B2 (PUB2).

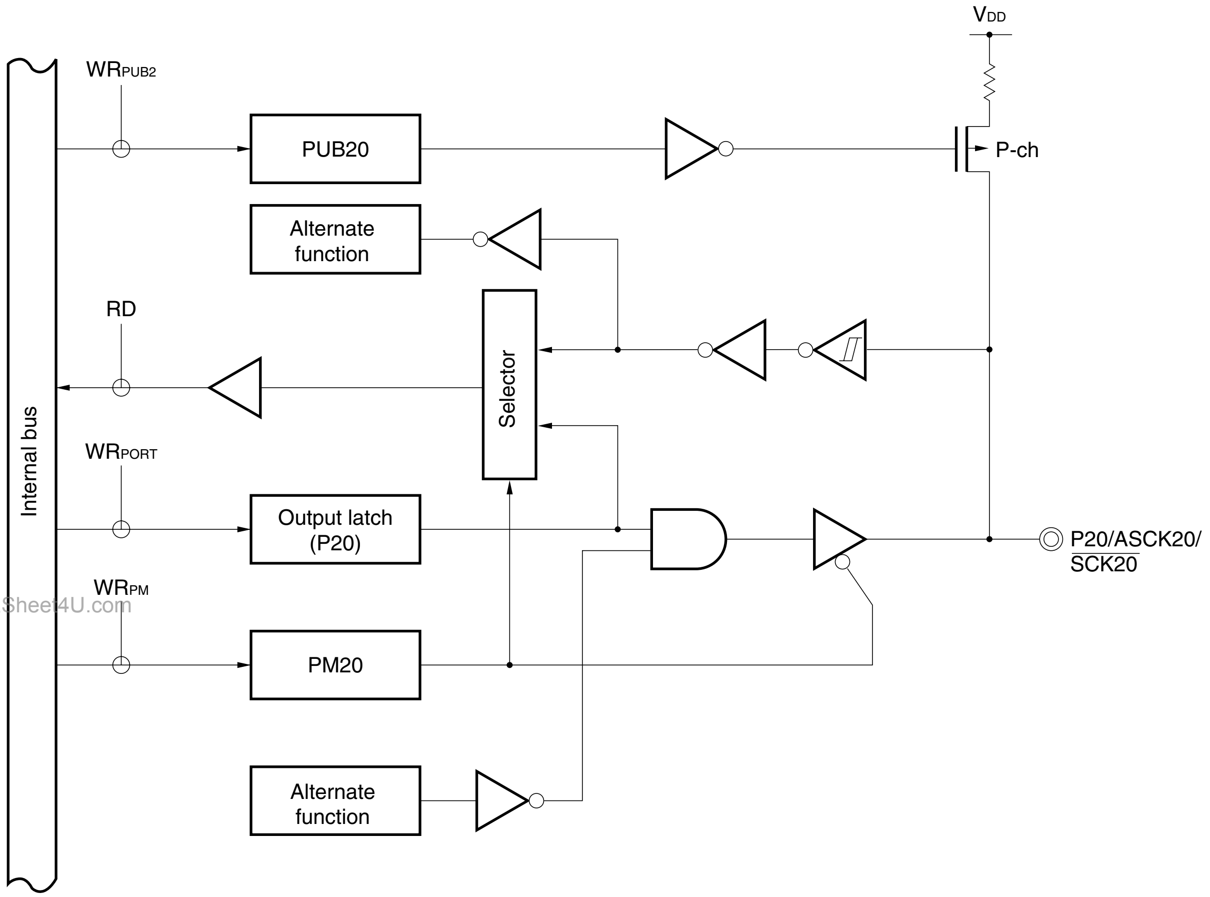
Port 2 is also used for external interrupt input, serial interface I/O, and timer I/O.

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

Figures 4-4 through 4-7 show block diagrams of port 2.

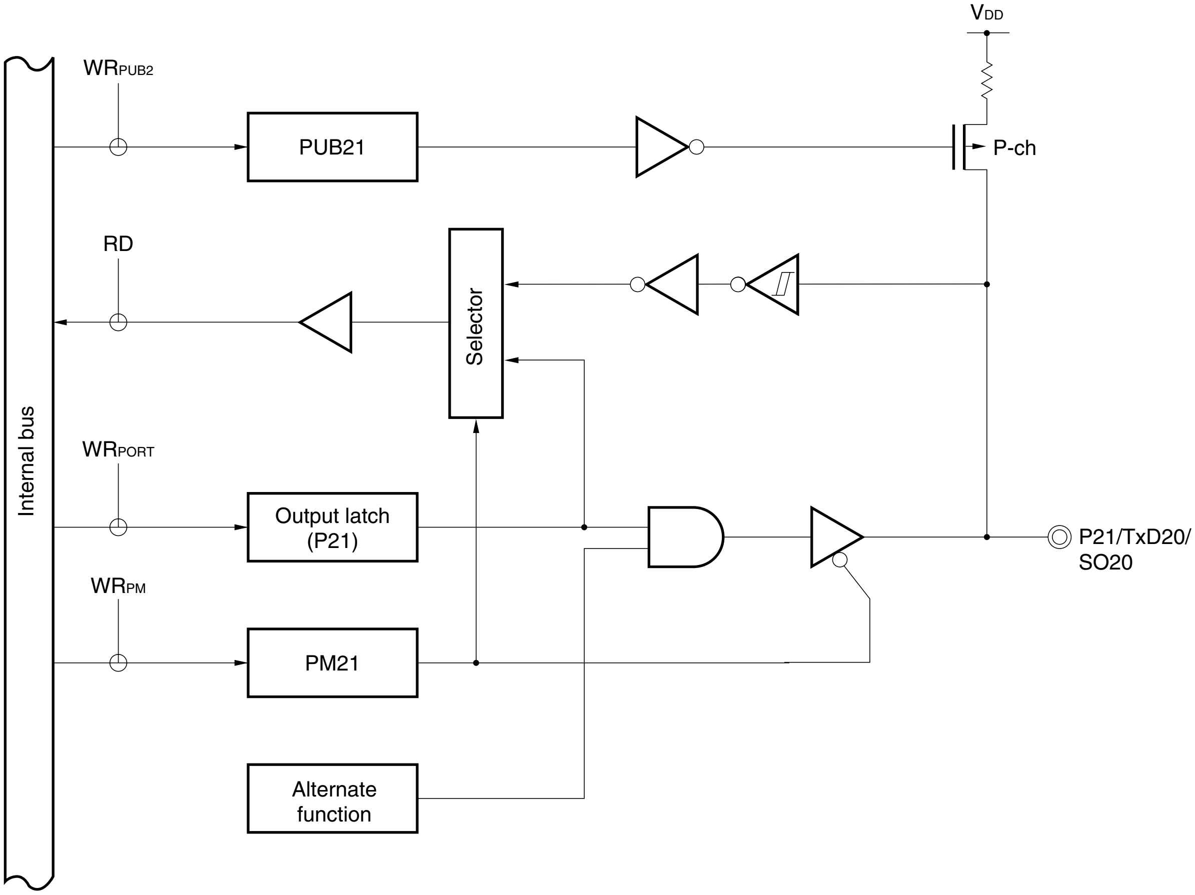
Caution When using the pins of port 2 for the serial interface, the I/O and output latches must be set according to the function to be used. For details of the settings, see Table 9-2 Operation Mode Settings of Serial Interface 20.

Figure 4-4. Block Diagram of P20


VDD
WRPUB2
PUB20
P-ch
Alternate function
RD
Selector
Internal bus
WRPORT
Output latch (P20)
P20/ASCK20/$\overline{\text{SCK20}}$
WRPM
PM20
Alternate function

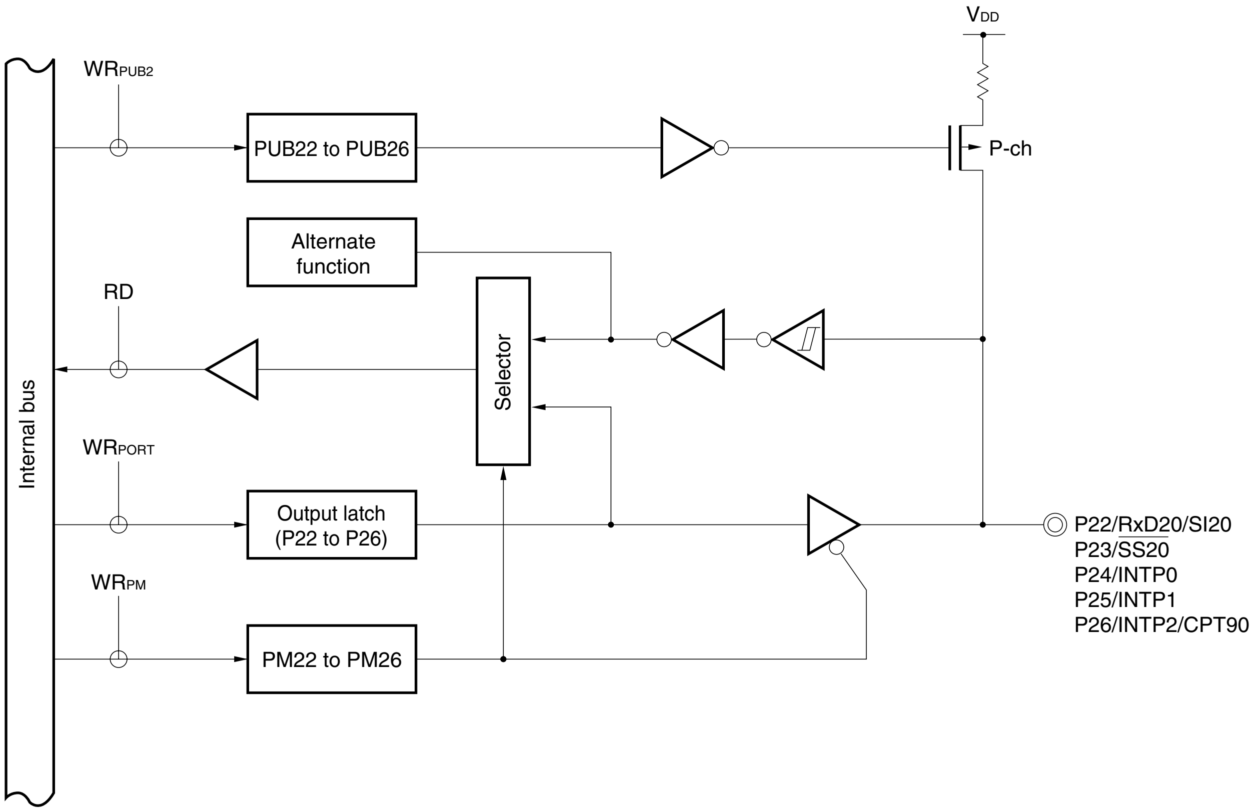
PUB2: Pull-up resistor option register B2
PM: Port mode register
RD: Port 2 read signal
WR: Port 2 write signal

Figure 4-5. Block Diagram of P21


VDD
WRPUB2
PUB21
P-ch
RD
Selector
Internal bus
WRPORT
Output latch (P21)
P21/TxD20/ SO20
WRPM
PM21
Alternate function

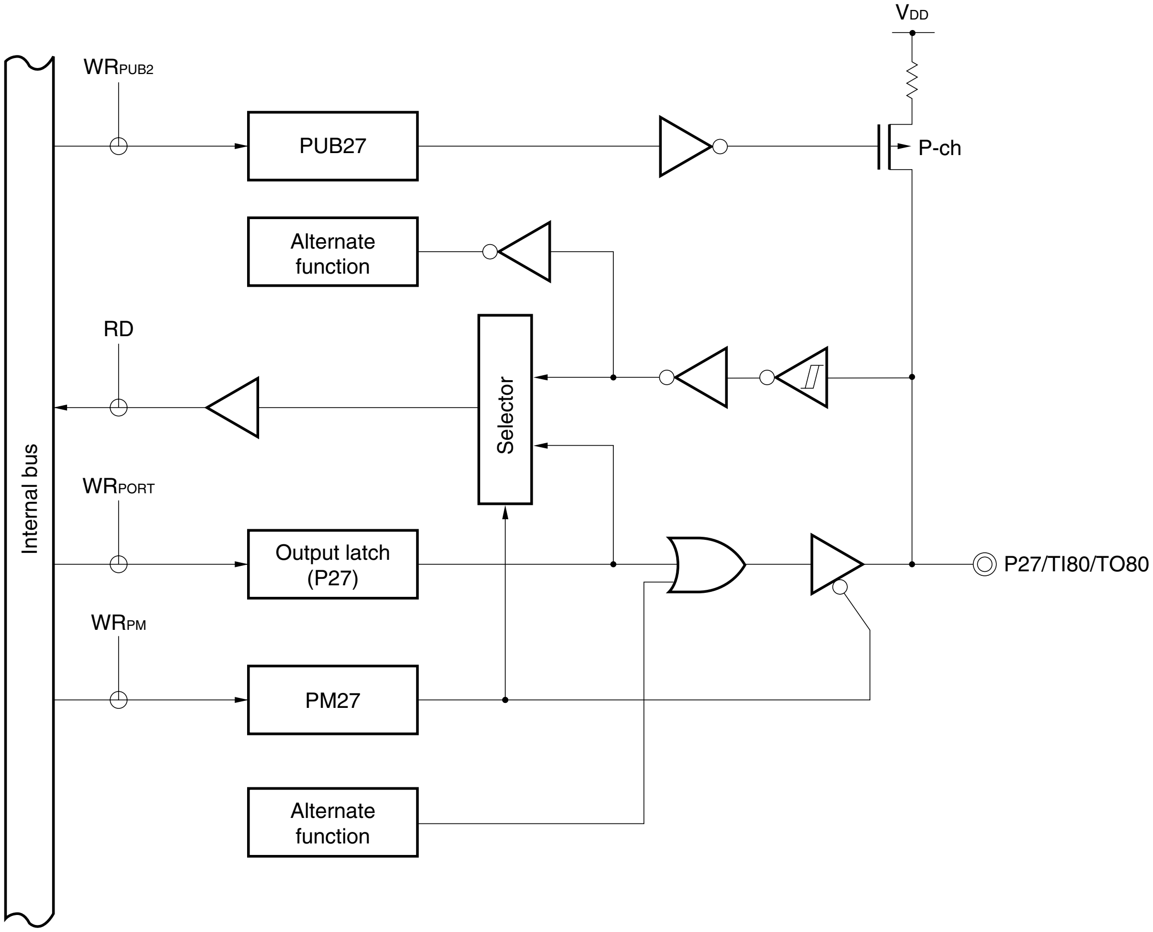
PUB2: Pull-up resistor option register B2
PM: Port mode register
RD: Port 2 read signal
WR: Port 2 write signal

Figure 4-6. Block Diagram of P22 to P26


VDD
WRPUB2
PUB22 to PUB26
P-ch
Alternate function
RD
Selector
Internal bus
WRPORT
Output latch (P22 to P26)
WRPM
PM22 to PM26
P22/RxD20/SI20
P23/SS20
P24/INTP0
P25/INTP1
P26/INTP2/CPT90

PUB2:	Pull-up resistor option register B2
PM:	Port mode register
RD:	Port 2 read signal
WR:	Port 2 write signal

Figure 4-7. Block Diagram of P27


VDD
WRPUB2
PUB27
P-ch
Alternate function
RD
Selector
Internal bus
WRPORT
Output latch (P27)
P27/TI80/TO80
WRPM
PM27
Alternate function

PUB2:	Pull-up resistor option register B2
PM:	Port mode register
RD:	Port 2 read signal
WR:	Port 2 write signal

4.2.4 Port 3

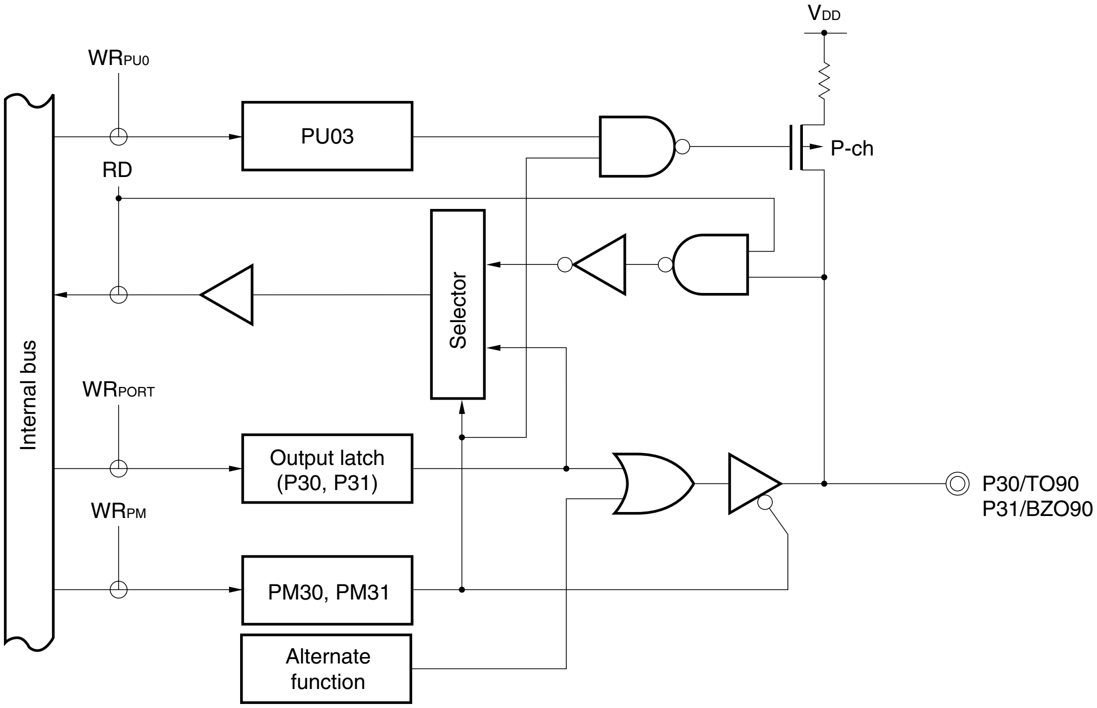
This is a 2-bit I/O port with an output latch. Port 3 can be set to input or output mode in 1-bit units by using port mode register 3 (PM3). When P30 and P31 are used as input port pins, on-chip pull-up resistors can be connected in 2-bit units by setting pull-up resistor option register 0 (PU0).

Port 3 is also used for timer output and buzzer output.

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

Figure 4-9 shows a block diagram of port 3.

Figure 4-8. Block Diagram of P30 and P31


VDD
WRPU0
PU03
P-ch
RD
Selector
Internal bus
WRPORT
Output latch
(P30, P31)
P30/TO90
P31/BZO90
WRPM
PM30, PM31
Alternate
function

PU0: Pull-up resistor option register 0
PM: Port mode register
RD: Port 3 read signal
WR: Port 3 write signal

4.3 Port Function Control Registers

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

- Port mode registers (PM0 to PM3)
- Pull-up resistor option registers (PU0 and PUB2)

(1) Port mode registers (PM0 to PM3)

The port mode registers separately set each port bit to either input or output.

Each port mode register is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the port mode registers to FFH.

When port pins are used for alternate functions, the corresponding port mode register and output latch must be set or reset as described in Table 4-3.

Caution When port 2 is acting as an output port, and its output level is changed, an interrupt request flag is set, because this port is also used as the input for an external interrupt. To use port 2 in output mode, therefore, the interrupt mask flag must be set to 1 in advance.

Figure 4-9. Format of Port Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00	FF20H	FFH	R/W
PM1	1	1	PM15	PM14	PM13	PM12	PM11	PM10	FF21H	FFH	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FF22H	FFH	R/W
PM3	1	1	1	1	1	1	PM31	PM30	FF23H	FFH	R/W

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

Table 4-3. Port Mode Register and Output Latch Settings for Using Alternate Functions

Pin Name	Alternate Function		PM××	P××
	Name	Input/Output		
P24	INTP0	Input	1	×
P25	INTP1	Input	1	×
P26	INTP2	Input	1	×
	CPT90	Input	1	×
P27	TI80	Input	1	×
	TO80	Output	0	0
P30	TO90	Output	0	0
P31	BZO90	Output	0	0

Caution When using the pins of port 2 for the serial interface, the I/O or output latch must be set according to the function to be used. For details of the settings, see Table 9-2 Operation Mode Settings of Serial Interface 20.

Remark ×: Don't care

PM××: Port mode register

P××: Port output latch

(2) Pull-up resistor option register 0 (PU0)

Pull-up resistor option register 0 (PU0) sets whether an on-chip pull-up resistor is used for ports 0, 1, and 3. For ports specified by PU0 to use on-chip pull-up resistors, 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 regardless of the setting of PU0. On-chip pull-up resistors also cannot be used when the pins are used as the alternate-function output pins.

PU0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears PU0 to 00H.

Figure 4-10. Format of Pull-up Resistor Option Register 0

Symbol	7	6	5	4	<3>	2	<1>	<0>	Address	After reset	R/W
PU0	0	0	0	0	PU03	0	PU01	PU00	FFF7H	00H	R/W

PU0m	Pm on-chip pull-up resistor selection (m = 0, 1, 3)
0	On-chip pull-up resistor is not used.
1	On-chip pull-up resistor is used.

Caution Bits 2 and 4 to 7 must all be set to 0.

(3) Pull-up resistor option register B2 (PUB2)

This register specifies whether the on-chip pull-up resistor connected to each pin of port 2 is used. The pins for which use of an on-chip pull-up resistor is specified by PUB2 can use a pull-up register internally, regardless of the setting of the port mode register.

PUB2 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears PUB2 to 00H.

Figure 4-11. Format of Pull-up Resistor Option Register B2

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
PUB2	PUB27	PUB26	PUB25	PUB24	PUB23	PUB22	PUB21	PUB20	FF32H	00H	R/W

PUB2n	P2n on-chip pull-up resistor selection (n = 0 to 7)
0	On-chip pull-up resistor is not used.
1	On-chip pull-up resistor is used.

4.4 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.

4.4.1 Writing to I/O port

(1) In output mode

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

Once data is written to the output latch, it 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.

Once data is written to the output latch, it is retained until new data is written to the output latch.

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

4.4.2 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.

4.4.3 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.

Once data is written to the output latch, it 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 one bit of a port. However, this instruction accesses the port in 8-bit units. When this instruction is executed to manipulate a bit of a port consisting both of inputs and outputs, therefore, the contents of the output latch of the pin that is set to input mode and not subject to manipulation become undefined.

CHAPTER 5 CLOCK GENERATOR

★ 5.1 Clock Generator Functions

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

- **Expanded-specification products**
  The system oscillator oscillates a frequency of 1.0 to 10.0 MHz. Oscillation can be stopped by executing the STOP instruction.

- **Conventional products**
  The system oscillator oscillates a frequency of 1.0 to 5.0 MHz. Oscillation can be stopped by executing the STOP instruction.

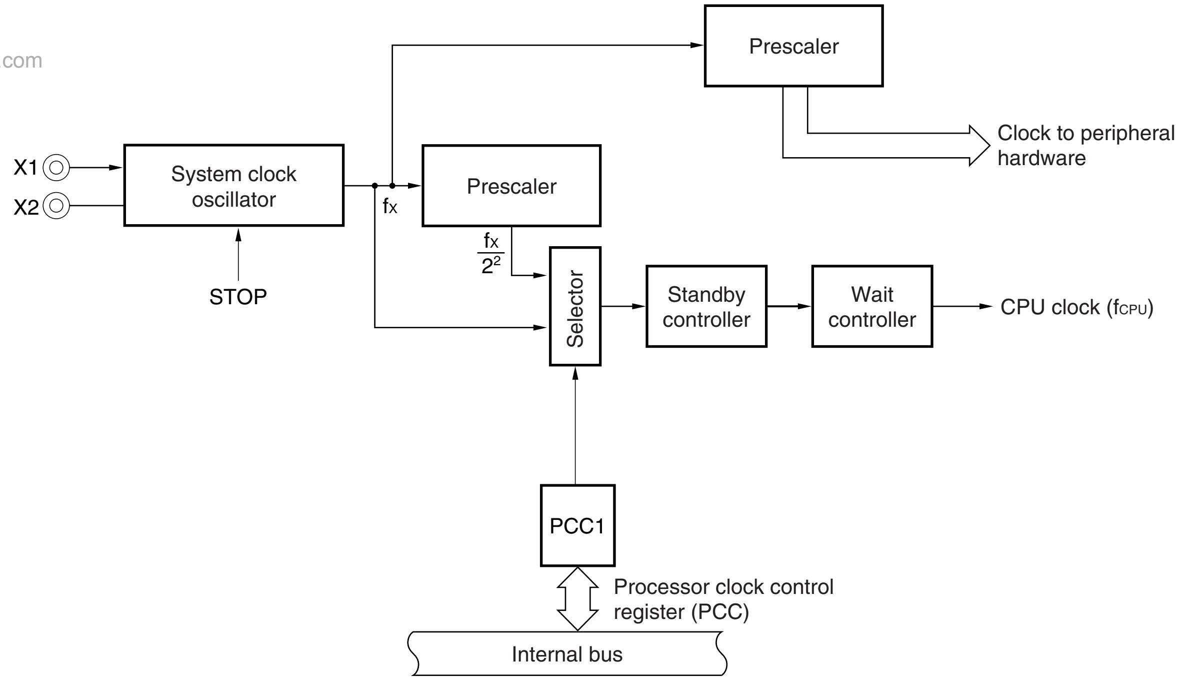
5.2 Clock Generator Configuration

The clock generator includes the following hardware.

Table 5-1. Configuration of Clock Generator

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

Figure 5-1. Block Diagram of Clock Generator


Prescaler
Clock to peripheral hardware
X1
X2
System clock oscillator
fx
Prescaler
$\frac{f_X}{2^2}$
STOP
Selector
Standby controller
Wait controller
CPU clock (f_{CPU})
PCC1
Processor clock control register (PCC)
Internal bus

5.3 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 with a 1-bit or 8-bit memory manipulation instruction.

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

★ **Figure 5-2. 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	0	FFFBH	02H	R/W

PCC1	CPU clock (f_{CPU}) selection	Minimum instruction execution time: $2/f_{CPU}$	
		At f_X = 10.0 MHz[Note]	At f_X = 5.0 MHz
0	f_X	0.2 μs	0.4 μs
1	$f_X/2^2$	0.8 μs	1.6 μs

Note Expanded-specification products only.

Caution Bits 0 and 2 to 7 must all be set to 0.

Remark f_X: System clock oscillation frequency

5.4 System Clock Oscillators

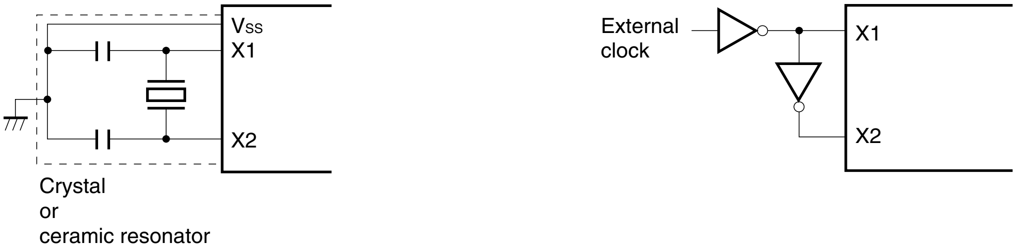
5.4.1 System clock oscillator

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

An external clock can also be input to the circuit. In this case, input the clock signal to the X1 pin, and input the inverted signal to the X2 pin.

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

Figure 5-3. External Circuit of System Clock Oscillator



(a) Crystal or ceramic oscillation

(b) External clock

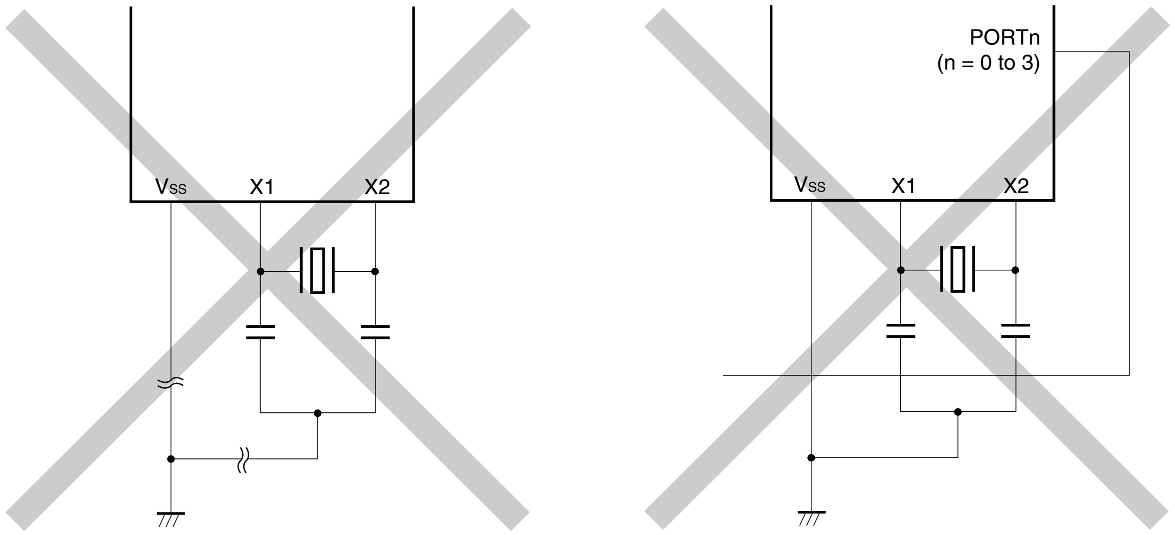
Caution When using the system clock oscillator, wire as follows in the area enclosed by the broken lines in Figure 5-3 to avoid an adverse effect from wiring capacitance.

- **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.**

5.4.2 Examples of incorrect resonator connection

Figure 5-4 shows an example of incorrect resonator connections.

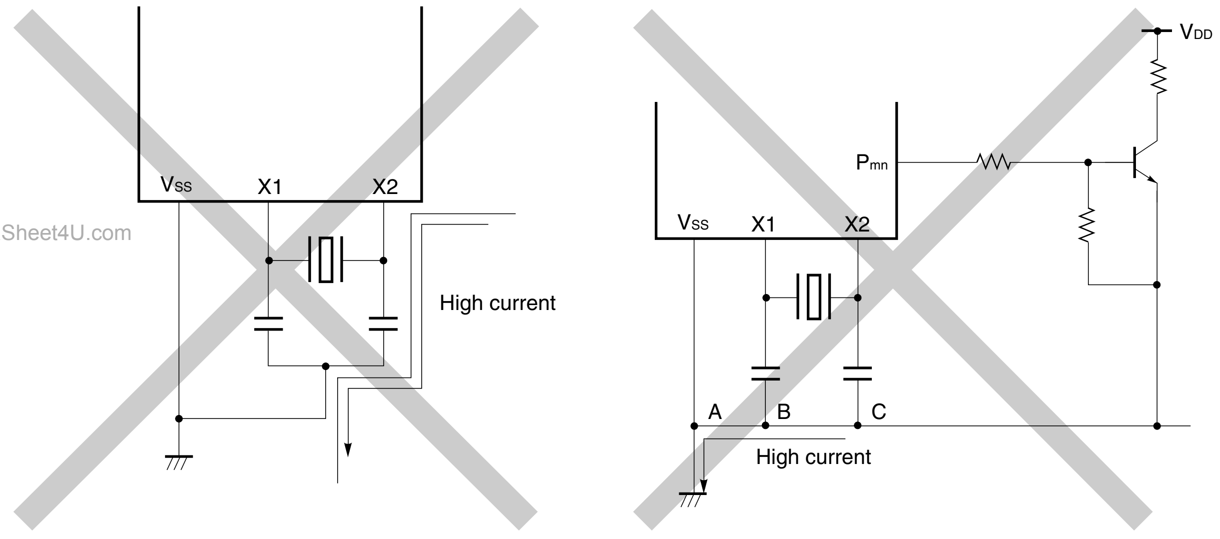
Figure 5-4. Examples of Incorrect Resonator Connection (1/2)



(a) Wiring too long

(b) Crossed signal line

VSS
X1
X2
VSS
X1
X2
PORTn
(n = 0 to 3)



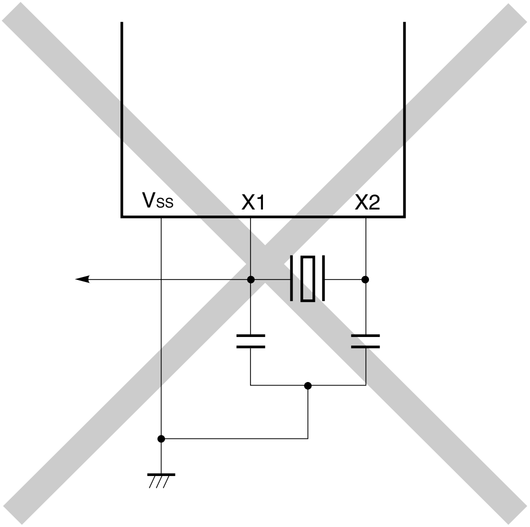
(c) Wiring near high fluctuating current

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

VSS
X1
X2
High current
VDD
Pmn
VSS
X1
X2
A
B
C
High current

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

(e) Signal is fetched



5.4.3 Frequency divider

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

5.5 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 (0.8 μs: at 10.0 MHz operation/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 input to the $\overline{\text{RESET}}$ pin, oscillation of the system clock is stopped.

★ (b) Two types of minimum instruction execution time (0.2 μs, 0.8 μs: at 10.0 MHz operation/0.4 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 external input clock).

5.6 Changing Setting of CPU Clock

5.6.1 Time required for switching CPU clock

The CPU clock can be switched by using bit 1 (PCC1) of the processor clock control register (PCC).

Actually, the specified clock is not switched immediately after the setting of PCC has been changed; old clock is used for the duration of several instructions after that (see **Table 5-2**).

Table 5-2. Maximum Time Required for Switching CPU Clock

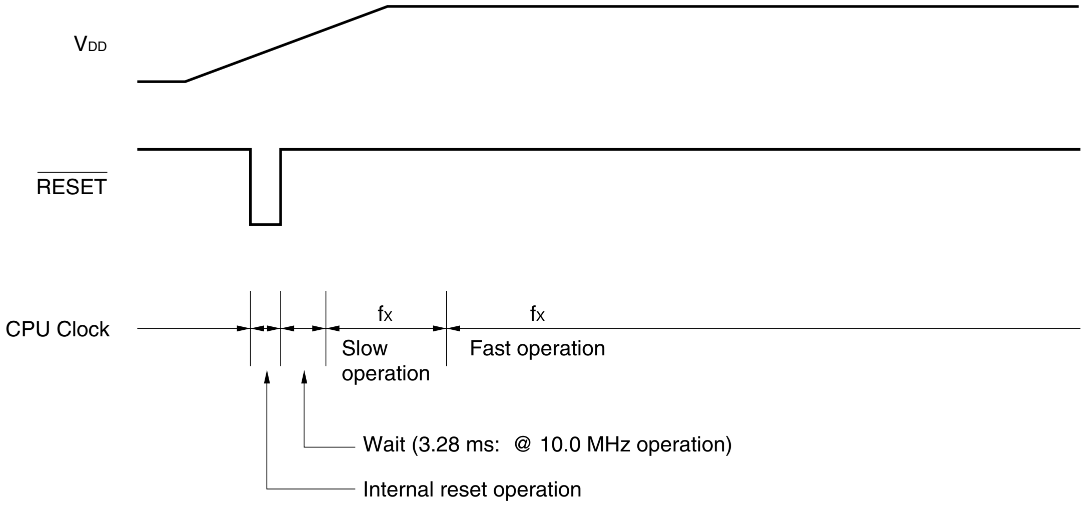
Set Value Before Switching	Set Value After Switching	
PCC1	PCC1	PCC1
	0	1
0		4 clocks
1	2 clocks	

Remark Two clocks is the minimum instruction execution time of the CPU clock before switching.

5.6.2 Switching CPU clock

The following figure illustrates how the CPU clock is switched.

Figure 5-5. 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 ($2^{15}/f_X$) is automatically secured.

★ After that, the CPU starts instruction execution at the slow speed of the system clock (0.8 µs: @10.0 MHz operation/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 6 16-BIT TIMER 90

6.1 Functions of 16-Bit Timer 90

16-bit timer 90 has the following functions.

- Timer interrupt
- Timer output
- Buzzer output
- Count value capture

(1) Timer interrupt
An interrupt is generated when the count value and compare value match.

(2) Timer output
Timer output can be controlled when the count value and compare value match.

(3) Buzzer output
Buzzer output can be controlled by software.

(4) Count value capture
The count value of 16-bit timer counter 90 (TM90) is latched into a capture register in synchronization with the capture trigger and retained.

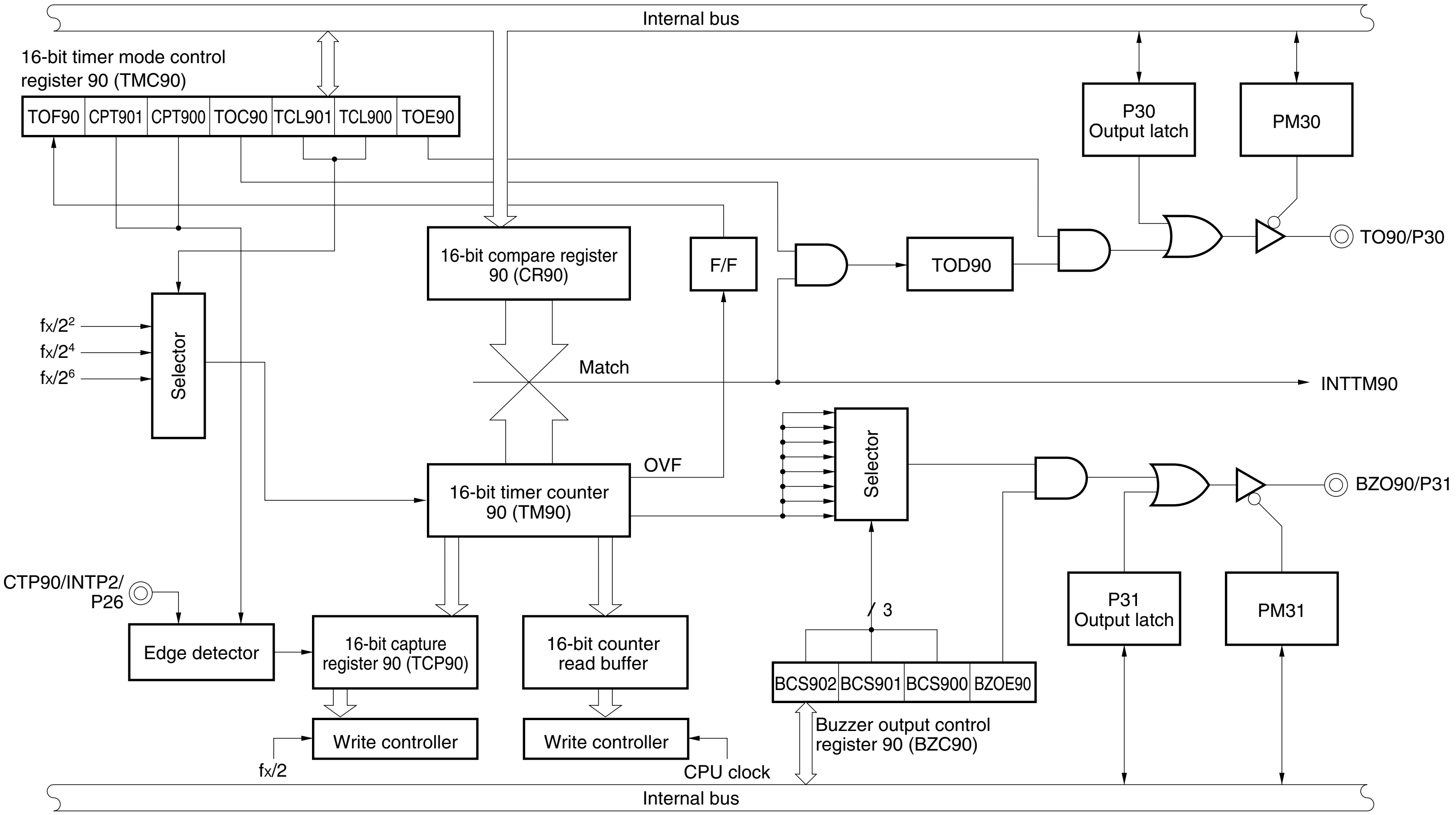
6.2 Configuration of 16-Bit Timer 90

16-bit timer 90 includes the following hardware.

Table 6-1. Configuration of 16-Bit Timer 90

Item	Configuration
Timer counter	16 bits × 1 (TM90)
Registers	Compare register: 16 bits × 1 (CR90) Capture register: 16 bits × 1 (TCP90)
Timer outputs	1 (TO90)
Control registers	16-bit timer mode control register 90 (TMC90) Buzzer output control register 90 (BZC90) Port mode register 3 (PM3) Port 3 (P3)

Figure 6-1. Block Diagram of 16-Bit Timer 90


Internal bus
16-bit timer mode control register 90 (TMC90)
TOF90
CPT901
CPT900
TOC90
TCL901
TCL900
TOE90
P30 Output latch
PM30
TO90/P30
16-bit compare register 90 (CR90)
F/F
TOD90
fx/2²
fx/2⁴
fx/2⁶
Selector
Match
INTTM90
OVF
16-bit timer counter 90 (TM90)
Selector
BZO90/P31
/ 3
P31 Output latch
PM31
CTP90/INTP2/ P26
Edge detector
16-bit capture register 90 (TCP90)
16-bit counter read buffer
BCS902
BCS901
BCS900
BZOE90
Buzzer output control register 90 (BZC90)
Write controller
Write controller
fx/2
CPU clock
Internal bus

(1) 16-bit compare register 90 (CR90)

The value specified in CR90 is compared with the count in 16-bit timer counter 90 (TM90). If they match, an interrupt request (INTTM90) is issued by CR90.

CR90 is set with an 8-bit or 16-bit memory manipulation instruction. Any value from 0000H to FFFFH can be set.

$\overline{\text{RESET}}$ input sets CR90 to FFFFH.

Cautions 1. CR90 is designed to be manipulated with a 16-bit memory manipulation instruction. It can also be manipulated with 8-bit memory manipulation instructions, however. When an 8-bit memory manipulation instruction is used to set CR90, it must be accessed by direct addressing.

2. To re-set CR90 during a count operation, it is necessary to disable interrupts in advance, using interrupt mask flag register 1 (MK1). It is also necessary to disable inversion of the timer output data, using 16-bit timer mode control register 90 (TMC90). If CR90 is rewritten with interrupts enabled, an interrupt request may be issued immediately at the point of rewrite.

(2) 16-bit timer counter 90 (TM90)

TM90 is used to count the number of pulses.

The contents of TM90 are read with an 8-bit or 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears TM90 to 0000H.

Cautions 1. The count becomes undefined when STOP mode is released, because the count operation is performed before oscillation stabilizes.

2. TM90 is designed to be manipulated with a 16-bit memory manipulation instruction. It can also be manipulated with 8-bit memory manipulation instructions, however. When an 8-bit memory instruction is used to manipulate TM90, it must be accessed by direct addressing.

3. When an 8-bit memory manipulation instruction is used to manipulate TM90, the lower and higher bytes must be read as a pair, in that order.

(3) 16-bit capture register 90 (TCP90)

TCP90 captures the contents of 16-bit timer counter 90 (TM90).

This register is set with an 8-bit or 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes TCP90 undefined.

Caution TCP90 is designed to be manipulated with a 16-bit memory manipulation instruction. It can also be manipulated with 8-bit memory manipulation instructions, however. When an 8-bit memory manipulation instruction is used to manipulate TCP90, it must be accessed by direct addressing.

(4) 16-bit counter read buffer 90

This buffer is used to latch and hold the count for 16-bit timer counter 90 (TM90).

6.3 Control Registers of 16-Bit Timer 90

The following four registers are used to control 16-bit timer 90.

- 16-bit timer mode control register 90 (TMC90)
- Buzzer output control register 90 (BZC90)
- Port mode register 3 (PM3)
- Port 3 (P3)

(1) 16-bit timer mode control register 90 (TMC90)

16-bit timer mode control register 90 (TMC90) controls the setting of the count clock, capture edge, etc.
TMC90 is set with a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears TMC90 to 00H.

Figure 6-2. Format of 16-Bit Timer Mode Control Register 90

Symbol	7	<6>	5	4	3	2	1	<0>	Address	After reset	R/W
TMC90	TOD90	TOF90	CPT901	CPT900	TOC90	TCL901	TCL900	TOE90	FF48H	00H	R/W[Note 1]

TOD90	Timer output data
0	Timer output of 0
1	Timer output of 1

TOF90	Overflow flag setting
0	Reset or cleared by software
1	Set when the 16-bit timer overflows

CPT901	CPT900	Capture edge selection
0	0	Capture operation disabled
0	1	Captured at the rising edge at the CPT90 pin
1	0	Captured at the falling edge at the CPT90 pin
1	1	Captured at both the rising and falling edges at the CPT90 pin

TOC90	Timer output data inversion control
0	Inversion disabled
1	Inversion enabled

★

TCL901	TCL900	16-bit timer counter 90 count clock (fcl) selection		
			At f_X = 10.0 MHz[Note 2]	At f_X = 5.0 MHz
0	0	$f_X/2^2$	2.5 MHz	1.25 MHz
0	1	$f_X/2^6$	156 kHz	78.1 kHz
1	0	$f_X/2^4$	625 kHz	313 kHz
1	1	Setting prohibited		

TOE90	16-bit timer counter output control
0	Output disabled (port mode)
1	Output enabled

Notes 1. Bit 7 is read-only.

2. Expanded-specification products only.

Caution Disable interrupts in advance using interrupt mask flag register 1 (MK1) when changing the data of TCL901 and TCL900. Also, prevent the timer output data from being inverted by setting TOC90 to 1.

Remark f_X: System clock oscillation frequency

(2) Buzzer output control register 90 (BZC90)

This register selects a buzzer frequency based on fcl selected with the count clock select bits (TCL901 and TCL900), and controls the output of a square wave.

BZC90 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears BZC90 to 00H.

Figure 6-3. Format of Buzzer Output Control Register 90

Symbol	7	6	5	4	3	2	1	<0>	Address	After reset	R/W
BZC90	0	0	0	0	BCS902	BCS901	BCS900	BZOE90	FF49H	00H	R/W

★

BCS902	BCS901	BCS900	Buzzer frequency						
				fx = 10.0 MHz operation$^{\text{Note}}$			fx = 5.0 MHz operation		
				fcl = fx/2^2	fcl = fx/2^6	fcl = fx/2^4	fcl = fx/2^2	fcl = fx/2^6	fcl = fx/2^4
0	0	0	fcl/2^4	156 kHz	9.77 kHz	39.1 kHz	78.1 kHz	4.88 kHz	19.5 kHz
0	0	1	fcl/2^5	78.1 kHz	4.88 kHz	19.5 kHz	39.1 kHz	2.44 kHz	9.77 kHz
0	1	0	fcl/2^8	9.77 kHz	610 Hz	2.44 kHz	4.88 kHz	305 Hz	1.22 kHz
0	1	1	fcl/2^9	4.88 kHz	305 Hz	1.22 kHz	2.44 kHz	153 Hz	610 Hz
1	0	0	fcl/2^{10}	2.44 kHz	153 Hz	610 Hz	1.22 kHz	76.3 Hz	305 Hz
1	0	1	fcl/2^{11}	1.22 kHz	76.3 Hz	305 Hz	610 Hz	38.1 Hz	153 Hz
1	1	0	fcl/2^{12}	610 Hz	38.1 Hz	153 Hz	305 Hz	19.1 Hz	76.3 Hz
1	1	1	fcl/2^{13}	305 Hz	19.1 Hz	76.3 Hz	153 Hz	9.54 Hz	38.1 Hz

BZOE90	Buzzer port output control
0	Disables buzzer port output.
1	Enables buzzer port output.

Note Expanded-specification products only.

Caution Bits 4 to 7 must all be set to 0.

Remarks 1. fx: System clock oscillation frequency

2. fcl: Count clock frequency of 16-bit timer 90.

(3) Port mode register 3 (PM3)

PM3 is used to set each bit of port 3 to input or output.

When pin P30/TO90 is used for timer output, reset the output latch of P30 and PM30 to 0; when pin P31/BZO90 is used for buzzer output, reset the output latch of P31 and PM31 to 0.

PM3 is set with a 1-bit or 8-bit memory manipulation instruction.

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

Figure 6-4. Format of Port Mode Register 3

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM3	1	1	1	1	1	1	PM31	PM30	FF23H	FFH	R/W

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

6.4 Operation of 16-Bit Timer 90

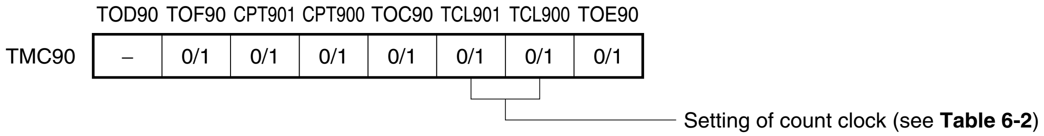
6.4.1 Operation as timer interrupt

★ 16-bit timer 90 can generate interrupts repeatedly each time the free-running counter value reaches the value set to CR90. Since this counter is not cleared and holds the count even after an interrupt is generated, the interval time is equal to one cycle of the count clock set in TCL901 and TCL900.

To operate 16-bit timer 90 as a timer interrupt, the following settings are required.

- Set the count value in CR90
- Set 16-bit timer mode control register 90 (TMC90) as shown in Figure 6-5.

Figure 6-5. Settings of 16-Bit Timer Mode Control Register 90 for Timer Interrupt Operation



Caution If both the CPT901 and CPT900 flags are set to 0, the capture operation is disabled.

When the count value of 16-bit timer counter 90 (TM90) matches the value set in CR90, counting of TM90 continues and an interrupt request signal (INTTM90) is generated.

Table 6-2 shows the interval time, and Figure 6-6 shows the timing of the timer interrupt operation.

Caution Perform the following processing when rewriting CR90 during a count operation.

<1> Disable interrupts (TMMK90 (bit 1 of interrupt mask flag register 1 (MK1)) = 1).
<2> Disable inversion control of timer output data (TOC90 = 0).

If CR90 is rewritten with interrupts enabled, an interrupt request may be issued immediately at the point of rewrite.

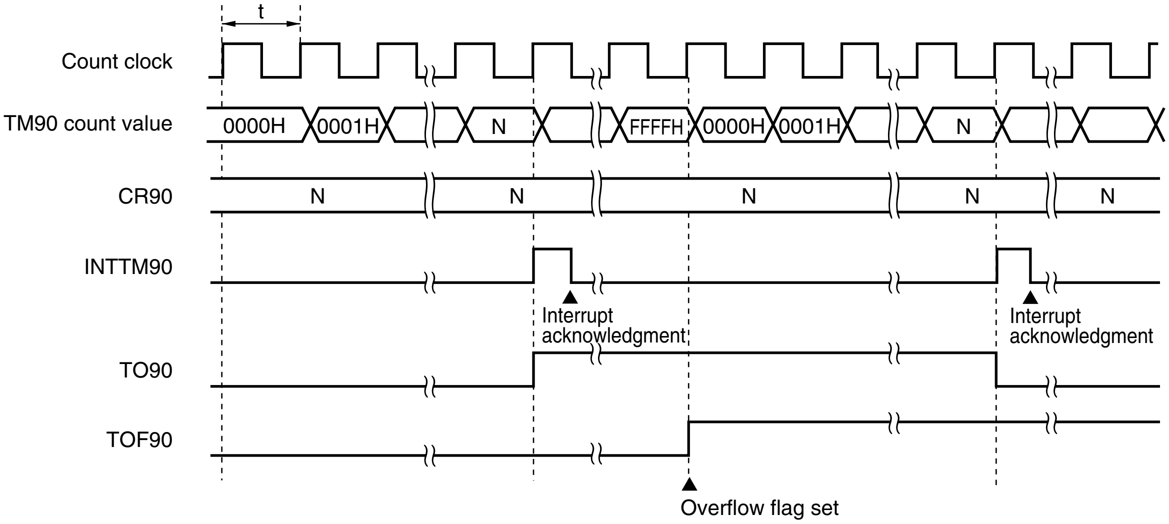
★ **Table 6-2. Interval Time of 16-Bit Timer 90**

TCL901	TCL900	Count Clock			Interval Time		
			At f_X = 10.0 MHz Operation[Note]	At f_X = 5.0 MHz Operation		At f_X = 10.0 MHz Operation[Note]	At f_X = 5.0 MHz Operation
0	0	$2^2/f_X$	0.4 μs	0.8 μs	$2^{18}/f_X$	26.2 ms	52.4 ms
0	1	$2^6/f_X$	6.4 μs	12.8 μs	$2^{22}/f_X$	419 ms	839 ms
1	0	$2^4/f_X$	1.6 μs	3.2 μs	$2^{20}/f_X$	105 ms	210 ms
1	1	Setting prohibited					

Note Expanded-specification products only.

Remark f_X: System clock oscillation frequency

★ Figure 6-6. Timing of Timer Interrupt Operation


t
Count clock
TM90 count value
0000H
0001H
N
FFFFH
0000H
0001H
N
CR90
N
N
N
N
N
INTTM90
Interrupt acknowledgment
Interrupt acknowledgment
TO90
TOF90
Overflow flag set

Remark N = 0000H to FFFFH

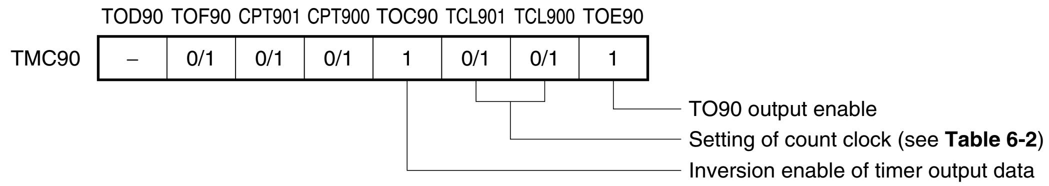
6.4.2 Operation as timer output

★ 16-bit timer 90 can invert the timer output repeatedly each time the free-running counter value reaches the value set to CR90. Since this counter is not cleared and holds the count even after the timer output is inverted, the interval time is equal to one cycle of the count clock set in TCL901 and TCL900.

To operate 16-bit timer 90 as a timer output, the following settings are required.

- Set P30 to output mode (PM30 = 0).
- Reset the output latch of P30 to 0.
- Set the count value in CR90.
- Set 16-bit timer mode control register 90 (TMC90) as shown in Figure 6-7.

Figure 6-7. Settings of 16-Bit Timer Mode Control Register 90 for Timer Output Operation

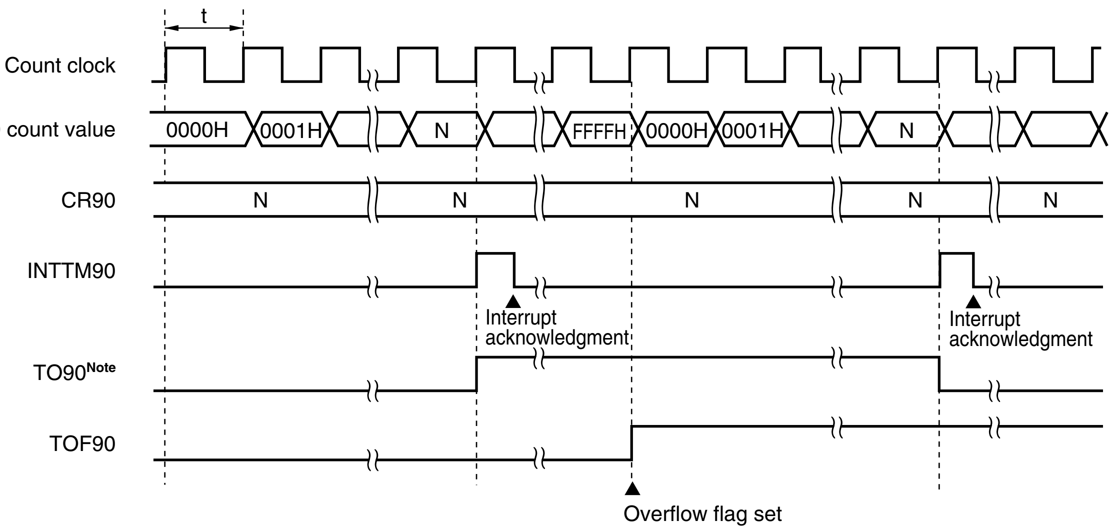

TOD90
TOF90
CPT901
CPT900
TOC90
TCL901
TCL900
TOE90
TMC90
–
0/1
0/1
0/1
1
0/1
0/1
1
TO90 output enable
Setting of count clock (see **Table 6-2**)
Inversion enable of timer output data

Caution If both the CPT901 and CPT900 flags are set to 0, the capture operation is disabled.

When the count value of 16-bit timer counter 90 (TM90) matches the value set in CR90, the output status of the TO90/P30 pin is inverted. This enables timer output. At that time, the TM90 count continues and an interrupt request signal (INTTM90) is generated.

Figure 6-8 shows the timing of timer output (see Table 6-2 for the interval time of 16-bit timer 90).

★ **Figure 6-8. Timer Output Timing**


t
Count clock
TM90 count value
0000H
0001H
N
FFFFH
0000H
0001H
N
CR90
N
N
N
N
N
INTTM90
Interrupt acknowledgment
Interrupt acknowledgment
TO90Note
TOF90
Overflow flag set

Note The TO90 initial value becomes low level during output enable (TOE90 = 1).

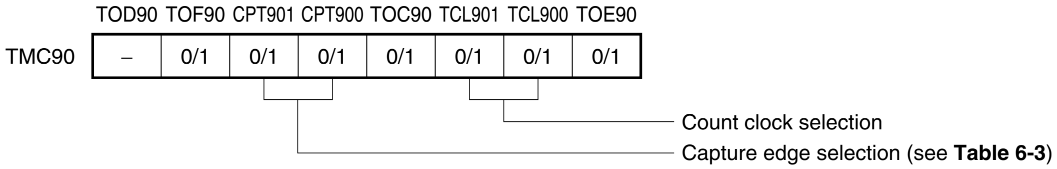
Remark N = 0000H to FFFFH

6.4.3 Capture operation

The capture operation consists of latching the count value of 16-bit timer counter 90 (TM90) into a capture register in synchronization with a capture trigger, and retaining the count value.

Set TMC90 as shown in Figure 6-9 to allow 16-bit timer 90 to start the capture operation.

Figure 6-9. Settings of 16-Bit Timer Mode Control Register 90 for Capture Operation


TOD90
TOF90
CPT901
CPT900
TOC90
TCL901
TCL900
TOE90
TMC90
–
0/1
0/1
0/1
0/1
0/1
0/1
0/1
Count clock selection
Capture edge selection (see **Table 6-3**)

16-bit capture register 90 (TCP90) starts a capture operation after the CPT90 capture trigger edge is detected, and latches and retains the count value of 16-bit timer counter 90. TCP90 fetches the count value within 2 clocks and retains the count value until the next capture edge detection.

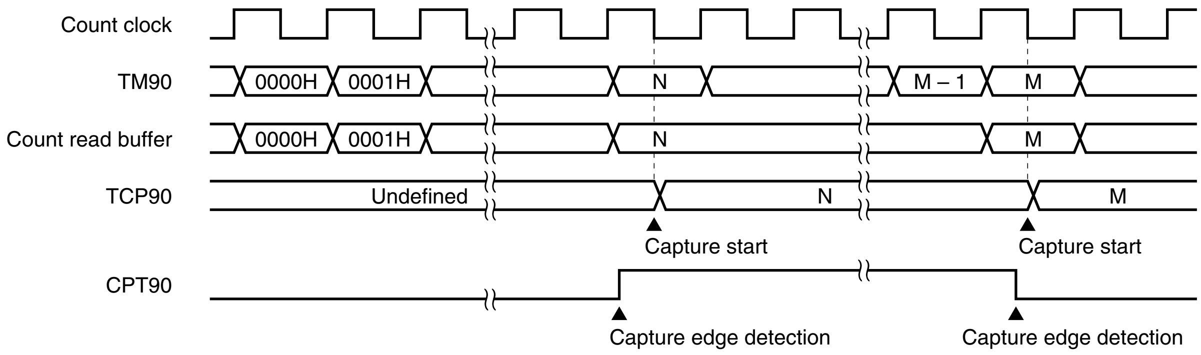
Table 6-3 and Figure 6-10 show the settings of the capture edge and capture operation timing, respectively.

Table 6-3. Settings of Capture Edge

CPT901	CPT900	Capture Edge Selection
0	0	Capture operation disabled
0	1	CPT90 pin rising edge
1	0	CPT90 pin falling edge
1	1	CPT90 pin both edges

Caution Because TCP90 is rewritten when a capture trigger edge is detected during a TCP90 read, disable the capture trigger edge detection during a TCP90 read.

Figure 6-10. Capture Operation Timing (with Both Edges of CPT90 Pin Specified)


Count clock
TM90
0000H
0001H
N
M – 1
M
Count read buffer
0000H
0001H
N
M
TCP90
Undefined
N
M
Capture start
Capture start
CPT90
Capture edge detection
Capture edge detection

Remark N = 0000H to FFFFH
M = 0000H to FFFFH

6.4.4 16-bit timer counter 90 readout

The count value of 16-bit timer counter 90 (TM90) is read out with a 16-bit manipulation instruction.

TM90 readout is performed through a counter read buffer. The counter read buffer latches the TM90 count value. The buffer operation is then held pending at the CPU clock falling edge after the read signal of the TM90 lower byte rises and the count value is retained. The counter read buffer value at the retention state can be read out as the count value.

Cancellation of the pending state is performed at the CPU clock falling edge after the read signal of the TM90 higher byte falls.

$\overline{\text{RESET}}$ input clears TM90 to 0000H and TM90 resumes counting in the free-running mode.

Figure 6-11 shows the timing of 16-bit timer counter 90 readout.

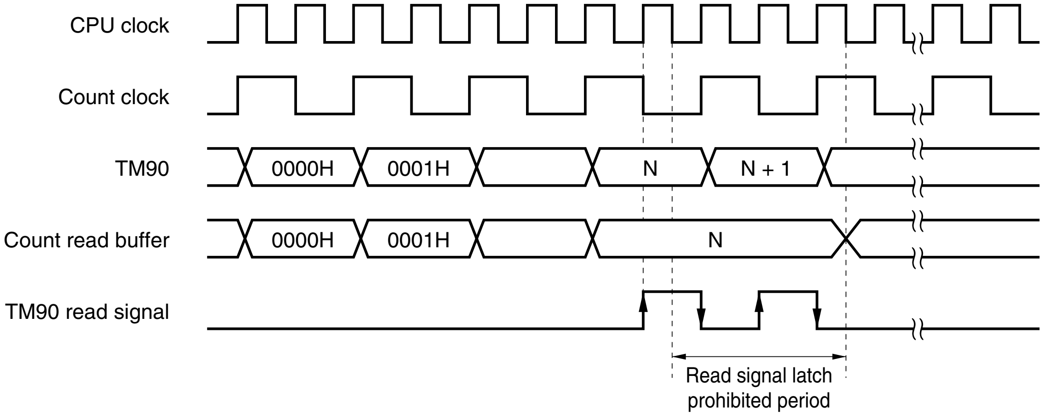
Cautions 1. The count value after releasing the stop mode becomes undefined because the count operation is executed during the oscillation stabilization time.

2. Though TM90 is designed for a 16-bit transfer instruction, an 8-bit transfer instruction can also be used.

When using an 8-bit transfer instruction, execute it by direct addressing.

3. When using an 8-bit transfer instruction, execute in the order from the lower byte to the higher byte in pairs. If only the lower byte is read, the pending state of the counter read buffer is not canceled, and if only the higher byte is read, an undefined count value is read.

Figure 6-11. 16-Bit Timer Counter 90 Readout Timing



Remark N = 0000H to FFFFH

6.4.5 Buzzer output operation

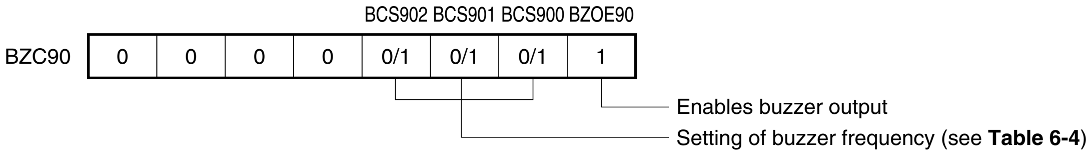
The buzzer frequency is set using buzzer output control register 90 (BZC90) based on the count clock selected with TCL901 and TCL900 of TMC90 (source clock). A square wave of the set buzzer frequency is output.

Table 6-4 shows the buzzer frequency.

To operate 16-bit timer 90 as a buzzer output, the following settings are required.

- Set P31 to output mode (PM31 = 0).
- Reset output latch of P31 to 0.
- Set a count clock by using TCL901 and TCL900.
- Set BZC90 as shown in Figure 6-12.

Figure 6-12. Settings of Buzzer Output Control Register 90 for Buzzer Output Operation



★

Table 6-4. Buzzer Frequency of 16-Bit Timer 90

BCS902	BCS901	BCS900	Buzzer Frequency						
				fx =10.0 MHz Operation[Note]			fx = 5.0 MHz Operation		
				fcl = fx/2^{2}	fcl = fx/2^{6}	fcl = fx/2^{4}	fcl = fx/2^{2}	fcl = fx/2^{6}	fcl = fx/2^{4}
0	0	0	fcl/2^{4}	156 kHz	9.77 kHz	39.1 kHz	78.1 kHz	4.88 kHz	19.5 kHz
0	0	1	fcl/2^{5}	78.1 kHz	4.88 kHz	19.5 kHz	39.1 kHz	2.44 kHz	9.77 kHz
0	1	0	fcl/2^{8}	9.77 kHz	610 Hz	2.44 kHz	4.88 kHz	305 Hz	1.22 kHz
0	1	1	fcl/2^{9}	4.88 kHz	305 Hz	1.22 kHz	2.44 kHz	153 Hz	610 Hz
1	0	0	fcl/2^{10}	2.44 kHz	153 Hz	610 Hz	1.22 kHz	76.3 Hz	305 Hz
1	0	1	fcl/2^{11}	1.22 kHz	76.3 Hz	305 Hz	610 Hz	38.1 Hz	153 Hz
1	1	0	fcl/2^{12}	610 Hz	38.1 Hz	153 Hz	305 Hz	19.1 Hz	76.3 Hz
1	1	1	fcl/2^{13}	305 Hz	19.1 Hz	76.3 Hz	153 Hz	9.54 Hz	38.1 Hz

Note Expanded-specification products only.

Remark fx: System clock oscillation frequency

★ 6.5 Notes on Using 16-Bit Timer 90

6.5.1 Restrictions on rewriting 16-bit compare register 90

(1) When rewriting the compare register (CR90), be sure to disable interrupts (TMMK90 = 1), and disable inversion control of timer output (TOC90 = 0) first.
If CR90 is rewritten with interrupts enabled, an interrupt request may be generated at the point of rewrite.

(2) The interval time may be double the intended time depending on the timing at which the compare register (CR90) is rewritten. Likewise, the timer output waveform may be shorter or double the intended output.
To avoid this, rewrite using one of the following procedures.

<Prevention method A> Rewriting by 8-bit access

<1> Disable interrupts (TMMK90 = 1), and disable inversion control of timer output (TOC90 = 0)
<2> Rewrite the higher byte of CR90 (16 bits) first
<3> Next, rewrite the lower byte of CR90 (16 bits)
<4> Clear the interrupt request flag (TMIF90)
<5> After more than half the cycle of the count clock has passed from the start of the interrupt, enable timer interrupts and timer output inversion

<Program example A> (When count clock = 64/fx, CPU clock = fx)

```
TM90_VCT: SET1   TMMK90      ;Timer interrupt disable (6 clocks)
          CLR1   TMC90.3     ;Timer output inversion disable (6 clocks)
          MOV    A,#xxH      ;Higher byte rewrite value setting (6 clocks)
          MOV    !0FF17H,A   ;CR90 higher byte rewriting (8 clocks)
          MOV    A,#yyH      ;Lower byte rewrite value setting (6 clocks)
          MOV    !0FF16H,A   ;CR90 lower byte rewriting (8 clocks)
          CLR1   TMIF90      ;Interrupt request flag clearing (6 clocks)
          CLR1   TMMK90      ;Timer interrupt enable (6 clocks)
          SET1   TMC90.3     ;Timer output inversion enable
```

More than 32 clocks in total[Note]

Note This is because the INTTM90 signal is set to the high level for a period of half the cycle of the count clock after an interrupt is generated, so the output will be inverted if TOC90 is set to 1 during this period.

<Prevention method B> Rewriting by 16-bit access

<1> Disable interrupts (TMMK90 = 1), and disable inversion control of timer output (TOC90 = 0)
<2> Rewrite CR90 (16 bits)
<3> Wait for more than one cycle of the count clock
<4> Clear the interrupt request flag (TMIF90)
<5> Enable timer interrupts and timer output inversion

<Program example B> (When count clock = 64/fx, CPU clock = fx)

```
TM90_VCT: SET1   TMMK90      ;Timer interrupt disable
          CLR1   TMC90.3     ;Timer output inversion disable
          MOVW   AX,#xxyyH   ;CR90 rewrite value setting
          MOVW   CR90,AX     ;CR90 rewriting
          NOP
          NOP
            :              }  NOP 32 (Wait for 64/fx)Note
          NOP
          NOP
          CLR1   TMIF90      ;Interrupt request flag clearing
          CLR1   TMMK90      ;Timer interrupt enable
          SET1   TMC90.3     ;Timer output inversion enable
```

Note Wait for more than one cycle of the count clock after the instruction rewriting CR90 (MOVW CR90, AX) before clearing the interrupt request flag (TMIF90).

CHAPTER 7 8-BIT TIMER/EVENT COUNTER 80

7.1 Functions of 8-Bit Timer/Event Counter 80

8-bit timer/event counter 80 has the following functions.

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

(1) 8-bit interval timer

When 8-bit timer/event counter 80 is used as an interval timer, it generates an interrupt at a time interval set in advance.

★ **Table 7-1. Interval Time of 8-Bit Timer/Event Counter 80**

Minimum Interval Time			Maximum Interval Time			Resolution		
	At fx = 10.0 MHz Operation[Note]	At fx = 5.0 MHz Operation		At fx = 10.0 MHz Operation[Note]	At fx = 5.0 MHz Operation		At fx = 10.0 MHz Operation[Note]	At fx = 5.0 MHz Operation
1/fx	100 ns	200 ns	2^8/fx	25.6 μs	51.2 μs	1/fx	100 ns	200 ns
2^8/fx	25.6 μs	51.2 μs	2^{16}/fx	6.55 ms	13.1 ms	2^8/fx	25.6 μs	51.2 μs

Note Expanded-specification products only.

Remark fx: System clock oscillation frequency

(2) External event counter

The number of pulses of an externally input signal can be counted.

(3) Square-wave output

A square-wave of arbitrary frequency can be output.

★ **Table 7-2. Square-Wave Output Range of 8-Bit Timer/Event Counter 80**

Minimum Pulse Width			Maximum Pulse Width			Resolution		
	At fx = 10.0 MHz Operation[Note]	At fx = 5.0 MHz Operation		At fx = 10.0 MHz Operation[Note]	At fx = 5.0 MHz Operation		At fx = 10.0 MHz Operation[Note]	At fx = 5.0 MHz Operation
1/fx	100 ns	200 ns	2^8/fx	25.6 μs	51.2 μs	1/fx	100 ns	200 ns
2^8/fx	25.6 μs	51.2 μs	2^{16}/fx	6.55 ms	13.1 ms	2^8/fx	25.6 μs	51.2 μs

Note Expanded-specification products only.

Remark fx: System clock oscillation frequency

(4) PWM output

8-bit resolution PWM output can be produced.

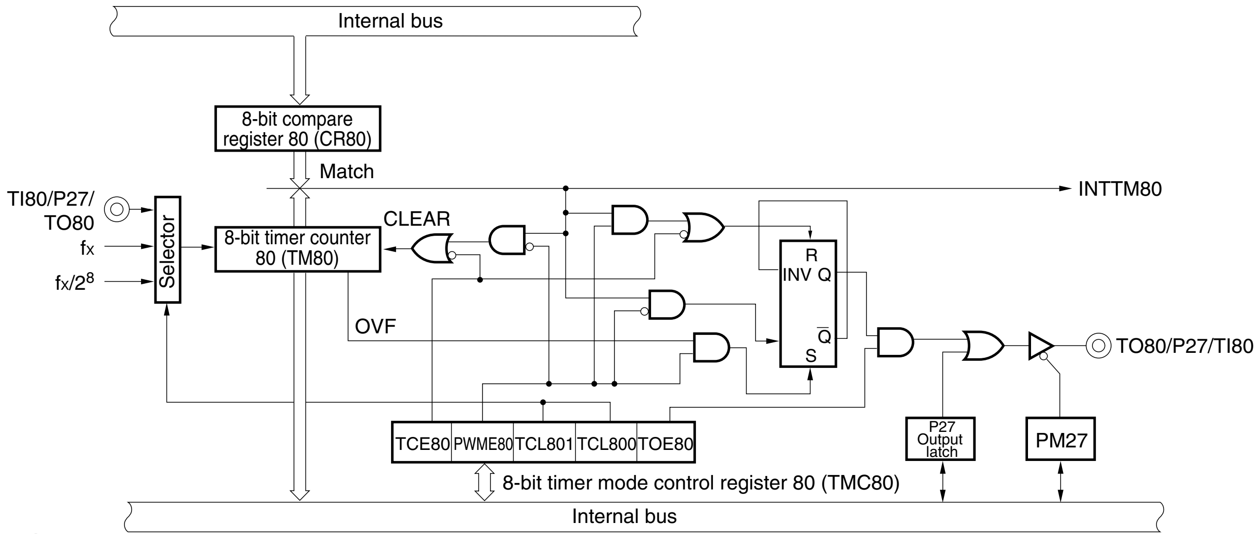
7.2 Configuration of 8-Bit Timer/Event Counter 80

8-bit timer/event counter 80 includes the following hardware.

Table 7-3. Configuration of 8-Bit Timer/Event Counter 80

Item	Configuration
Timer counter	8 bits × 1 (TM80)
Register	Compare register: 8 bits × 1 (CR80)
Timer outputs	1 (TO80)
Control registers	8-bit timer mode control register 80 (TMC80) Port mode register 2 (PM2) Port 2 (P2)

Figure 7-1. Block Diagram of 8-Bit Timer/Event Counter 80


Internal bus
8-bit compare register 80 (CR80)
Match
INTTM80
TI80/P27/TO80
fx
fx/2^8
Selector
8-bit timer counter 80 (TM80)
CLEAR
R
INV Q
OVF
Q̄
S
TO80/P27/TI80
TCE80
PWME80
TCL801
TCL800
TOE80
P27 Output latch
PM27
8-bit timer mode control register 80 (TMC80)
Internal bus

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

The value specified in CR80 is compared with the count in 8-bit timer counter 80 (TM80). If they match, an interrupt request (INTTM80) is issued.

CR80 is set with an 8-bit memory manipulation instruction. Any value from 00H to FFH can be set.

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

Cautions 1. Before rewriting CR80, stop the timer operation. If CR80 is rewritten while the timer operation is enabled, the match interrupt request signal may be generated immediately at the point of rewrite.

2. Do not clear CR80 to 00H in PWM output mode (when PWME80 = 1: bit 6 of 8-bit timer mode control register 80 (TMC80)); otherwise, PWM output may not be produced normally.

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

TM80 is used to count the number of pulses.

Its contents are read with an 8-bit memory manipulation instruction.

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

7.3 8-Bit Timer/Event Counter 80 Control Registers

The following three registers are used to control 8-bit timer/event counter 80.

- 8-bit timer mode control register 80 (TMC80)
- Port mode register 2 (PM2)
- Port 2 (P2)

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

TMC80 determines whether to enable or disable 8-bit timer counter 80 (TM80), specifies the count clock for TM80, and controls the operation of the output controller of 8-bit timer/event counter 80.

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

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

Figure 7-2. 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	PWME80	0	0	0	TCL801	TCL800	TOE80	FF53H	00H	R/W

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

PWME80	Operation mode selection
0	Timer counter operation mode
1	PWM output mode

★

TCL801	TCL800	8-bit timer counter 80 count clock selection		
			At f_X = 10.0 MHz operation[Note 1]	At f_X = 5.0 MHz operation
0	0	f_X	10.0 MHz	5.0 MHz
0	1	$f_X/2^8$	39.1 kHz	19.5 kHz
1	0	Rising edge of TI80[Note 2]		
1	1	Falling edge of TI80[Note 2]		

TOE80	8-bit timer/event counter output control
0	Output disabled (port mode)
1	Output enabled

Notes 1. Expanded-specification products only.

2. When inputting a clock signal externally, timer output cannot be used.

Caution Always stop the timer before setting TMC80.

Remark f_X: System clock oscillation frequency

(2) Port mode register 2 (PM2)

PM2 specifies whether each bit of port 2 is used for input or output.

To use the TO80/P27/TI80 pin for timer output, the PM27 and P27 output latch must be reset to 0.

To use the TO80/P27/TI80 pin for timer input, PM27 must be set to 1.

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

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

Figure 7-3. Format of Port Mode Register 2

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

PM2n	P2n pin input/output mode selection (n = 0 to 5)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

7.4 Operation of 8-Bit Timer/Event Counter 80

7.4.1 Operation as interval timer

The interval timer repeatedly generates an interrupt at a time interval specified by the count value preset in 8-bit compare register 80 (CR80).

To operate 8-bit timer/event counter 80 as an interval timer, settings must be made in the following sequence.

<1> Disable operation of 8-bit timer counter 80 (TM80) (TCE80 (bit 7 of 8-bit timer mode control register 80 (TMC80)) = 0).
<2> Set the count clock of 8-bit timer/event counter 80 (see **Table 7-4**).
<3> Set a count value in CR80.
<4> Enable the operation of TM80 (TCE80 = 1).

When the count value of 8-bit timer counter 80 (TM80) matches the value set in CR80, TM80 is cleared to 0 and continues counting. At the same time, an interrupt request signal (INTTM80) is generated.

Table 7-4 shows the interval time, and Figure 7-4 shows the timing of the interval timer operation.

Cautions 1. Stop the timer operation before rewriting CR80. If CR80 is rewritten while the timer operation is enabled, a match signal may be generated immediately at the point of rewrite (an interrupt request will be generated if interrupts are enabled).

2. If setting the count clock to TMC80 and enabling the operation of TM80 are performed at the same time with an 8-bit memory manipulation instruction, the error one cycle after the timer has been started may exceed one clock. To use 8-bit timer/event counter 80 as an interval timer, therefore, make the settings in the above sequence.

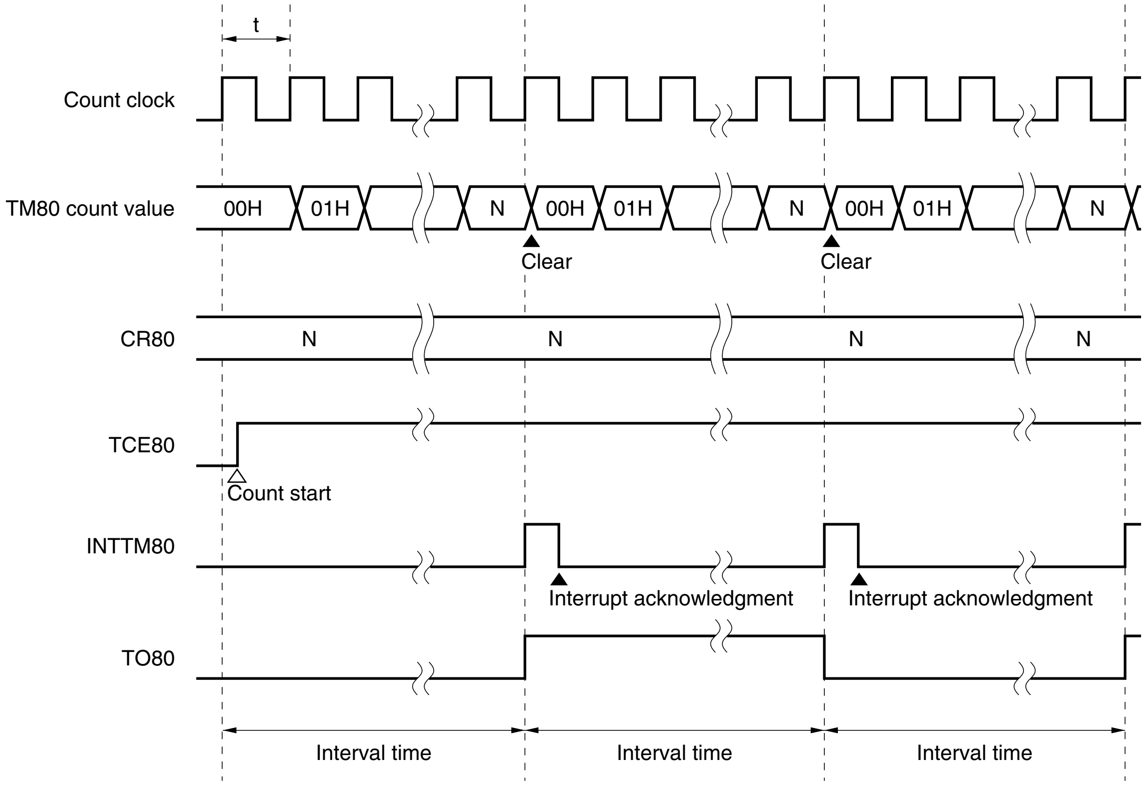
★ **Table 7-4. Interval Time of 8-Bit Timer/Event Counter 80**

TCL801	TCL800	Minimum Interval Time			Maximum Interval Time			Resolution		
			At fx = 10.0 MHz Operation[Note]	At fx = 5.0 MHz Operation		At fx = 10.0 MHz Operation[Note]	At fx = 5.0 MHz Operation		At fx = 10.0 MHz Operation[Note]	At fx = 5.0 MHz Operation
0	0	1/fx	100 ns	200 ns	2^8/fx	25.6 μs	51.2 μs	1/fx	100 ns	200 ns
0	1	2^8/fx	25.6 μs	51.2 μs	2^{16}/fx	6.55 ms	13.1 ms	2^8/fx	25.6 μs	51.2 μs
1	0	TI80 input cycle			2^8 × TI80 input cycle			TI80 input edge cycle		
1	1	TI80 input cycle			2^8 × TI80 input cycle			TI80 input edge cycle		

Note Expanded-specification products only.

Remark fx: System clock oscillation frequency

Figure 7-4. Interval Timer Operation Timing


t
Count clock
TM80 count value
00H
01H
N
00H
01H
N
00H
01H
N
Clear
Clear
CR80
N
N
N
N
TCE80
Count start
INTTM80
Interrupt acknowledgment
Interrupt acknowledgment
TO80
Interval time
Interval time
Interval time

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

7.4.2 Operation as external event counter

The external event counter counts the number of external clock pulses input to the TI80/P27/TO80 pin by using 8-bit timer counter 80 (TM80).

To operate 8-bit timer/event counter 80 as an external event counter, settings must be made in the following sequence.

<1> Set P27 to input mode (PM27 = 1).
<2> Disable operation of 8-bit timer counter 80 (TM80) (TCE80 (bit 7 of 8-bit timer mode control register 80 (TMC80)) = 0).
<3> Specify the rising or falling edge of TI80 (see **Table 7-4**). Disable output of TO80 (TOE80 (bit 0 of TMC80) = 0) and PWM output (PWME80 (bit 6 of TMC80) = 0).
<4> Set a count value in CR80.
<5> Enable the operation of TM80 (TCE80 = 1).

Each time the valid edge specified by bit 1 (TCL800) of TMC80 is input, the value of 8-bit timer counter 80 (TM80) is incremented.

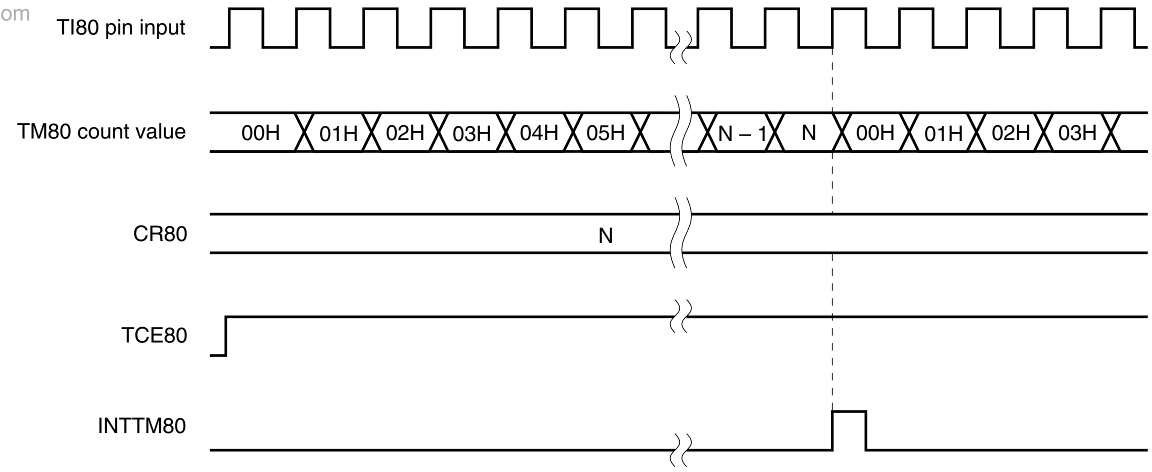
When the count value of TM80 matches the value set in CR80, TM80 is cleared to 0 and continues counting. At the same time, an interrupt request signal (INTTM80) is generated.

Figure 7-5 shows the timing of the external event counter operation (with rising edge specified).

Cautions 1. Before rewriting CR80, stop the timer operation. If CR80 is rewritten while the timer operation is enabled, a match interrupt request signal may be generated immediately at the point of rewrite.

2. If setting the count clock to TMC80 and enabling the operation of TM80 are performed at the same time with an 8-bit memory manipulation instruction, the error one cycle after the timer has been started may exceed one clock. To use 8-bit timer/event counter 80 as an external event counter, therefore, make the settings in the above sequence.

Figure 7-5. External Event Counter Operation Timing (with Rising Edge Specified)



Remark N = 00H to FFH

7.4.3 Operation as square-wave output

8-bit timer/event counter 80 can generate square-wave output of an arbitrary frequency at an interval specified by the count value preset in 8-bit compare register 80 (CR80).

To use 8-bit timer/event counter 80 for square-wave output, settings must be made in the following sequence.

<1> Set P27 to output mode (PM27 = 0). Set the output latch of P27 to 0.
<2> Disable operation of 8-bit timer counter 80 (TM80) (TCE80 = 0).
<3> Set a count clock for 8-bit timer/event counter 80 (see **Table 7-5**), enable output of TO80 (TOE80 = 1), and disable PWM output (PWME80 = 0).
<4> Set a count value in CR80.
<5> Enable the operation of TM80 (TCE80 = 1).

When the count value of 8-bit timer counter 80 (TM80) matches the value set in CR80, the TO80 pin output will be inverted. Through application of this mechanism, square waves of any frequency can be output. As soon as a match occurs, TM80 is cleared to 0 and continues counting, generating an interrupt request signal (INTTM80).

Setting bit 7 (TCE80) of TMC80 to 0 clears the square-wave output to 0.

Table 7-5 shows the square-wave output range, and Figure 7-6 shows the timing of square-wave output.

Cautions 1. Stop the timer operation before rewriting CR80. If CR80 is rewritten while the timer operation is enabled, a match interrupt request signal may be generated immediately at the point of rewrite.

2. If setting the count clock to TMC80 and enabling the operation of TM80 are performed at the same time with an 8-bit memory manipulation instruction, the error one cycle after the timer has been started may exceed one clock. To use 8-bit timer/event counter 80 as a square-wave output, therefore, make the settings in the above sequence.

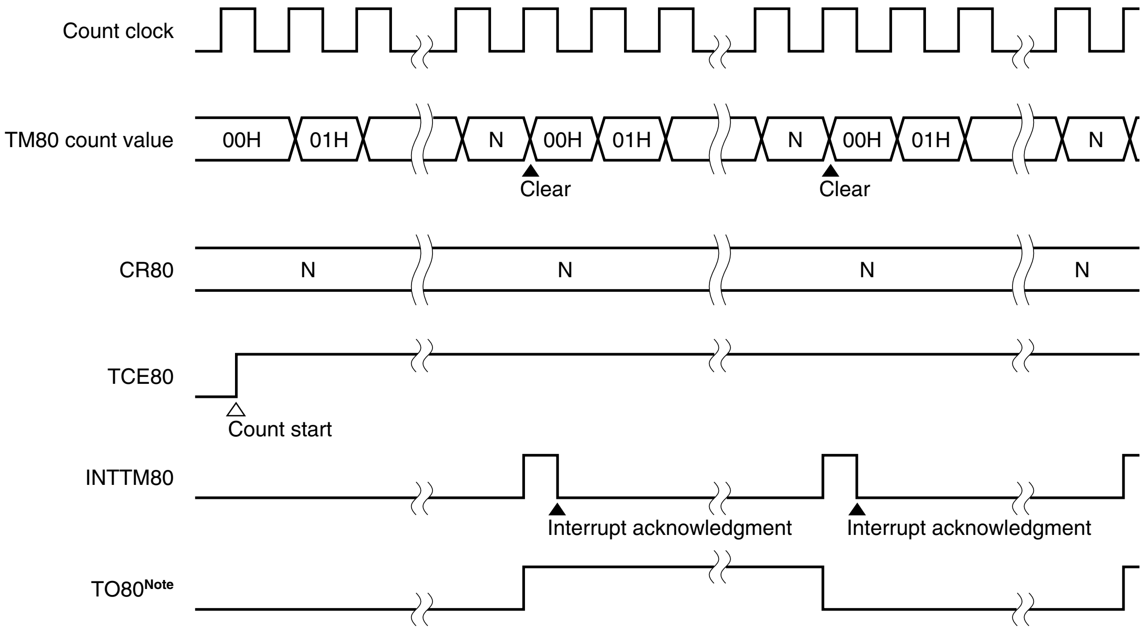
★ **Table 7-5. Square-Wave Output Range of 8-Bit Timer/Event Counter**

TCL801	TCL800	Minimum Pulse Width			Maximum Pulse Width			Resolution		
			At fx = 10.0 MHz Operation[Note]	At fx = 5.0 MHz Operation		At fx = 10.0 MHz Operation[Note]	At fx = 5.0 MHz Operation		At fx = 10.0 MHz Operation[Note]	At fx = 5.0 MHz Operation
0	0	1/fx	100 ns	200 ns	2^8/fx	25.6 μs	51.2 μs	1/fx	100 ns	200 ns
0	1	2^8/fx	25.6 μs	51.2 μs	2^{16}/fx	6.55 ms	13.1 ms	2^8/fx	25.6 μs	51.2 μs

Note Expanded-specification products only.

Remark fx: System clock oscillation frequency

Figure 7-6. Square-Wave Output Timing


Count clock
TM80 count value
00H
01H
N
00H
01H
N
00H
01H
N
Clear
Clear
CR80
N
N
N
N
TCE80
Count start
INTTM80
Interrupt acknowledgment
Interrupt acknowledgment
TO80Note

Note The initial value of TO80 is low when output is enabled (TOE80 = 1).

7.4.4 Operation as PWM output

PWM output enables an interrupt to be generated repeatedly at an interval specified by the count value preset in 8-bit compare register 80 (CR80).

To use 8-bit timer/event counter 80 for PWM output, the following settings are required.

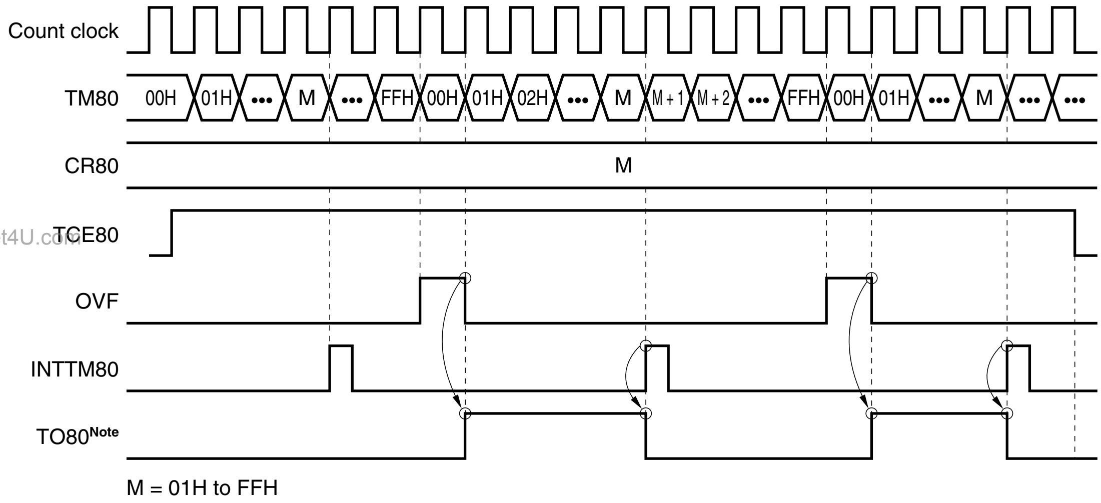
<1> Set P27 to output mode (PM27 = 0). Set the output latch of P27 to 0.
<2> Disable the operation of 8-bit timer counter 80 (TM80) (TCE80 = 0).
<3> Set a count clock for 8-bit timer/event counter (see **Table 7-4**), and enable output of TO80 (TOE80 = 1) and PWM output (PWME80 = 1).
<4> Set a count value in CR80.
<5> Enable the operation of TM80 (TCE80 = 1).

When the count value of 8-bit timer counter 80 (TM80) matches the value set in CR80, TM80 continues counting, and an interrupt request signal (INTTM80) is generated.

Cautions 1. If CR80 is rewritten during timer operation, a high level may be output during the next cycle (see 7.5 (2) Setting of 8-bit compare register 80).

2. If setting the count clock to TMC80 and enabling the operation of TM80 are performed at the same time with an 8-bit memory manipulation instruction, the error one cycle after the timer has been started may exceed one clock. To use 8-bit timer/event counter 80 as a PWM output, therefore, make the settings in the above sequence.

Figure 7-7. PWM Output Timing



Note The initial value of TO80 is low when output is enabled (TOE80 = 1).

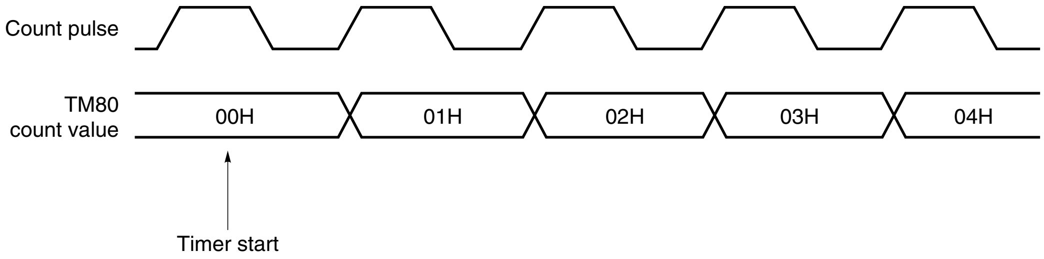
Caution Do not set CR80 to 00H in PWM output mode; otherwise, PWM may not be output normally.

7.5 Notes on Using 8-Bit Timer/Event Counter 80

(1) Error on starting timer

An error of up to 1 clock is included in the time between when the timer is started and a match signal is generated. This is because 8-bit timer counter 80 (TM80) is started asynchronously to the count pulse.

Figure 7-8. Start Timing of 8-Bit Timer Counter 80

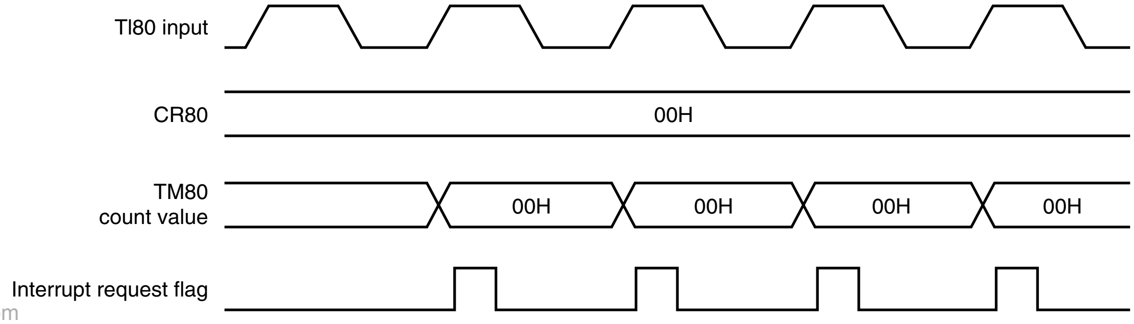

Count pulse
TM80 count value
00H
01H
02H
03H
04H
Timer start

(2) Setting of 8-bit compare register 80

8-bit compare register 80 (CR80) can be set to 00H.

Therefore, one pulse can be counted when 8-bit timer/event counter 80 operates as an event counter.

Figure 7-9. External Event Counter Operation Timing


TI80 input
CR80
00H
TM80 count value
00H
00H
00H
00H
Interrupt request flag

Cautions 1. Before rewriting CR80 in timer counter operation mode (PWME80 (bit 6 of 8-bit timer mode control register 80 (TMC80) = 0), stop the timer operation. If CR80 is rewritten while the timer operation is enabled, a match interrupt request signal may be generated immediately at the point of rewrite.

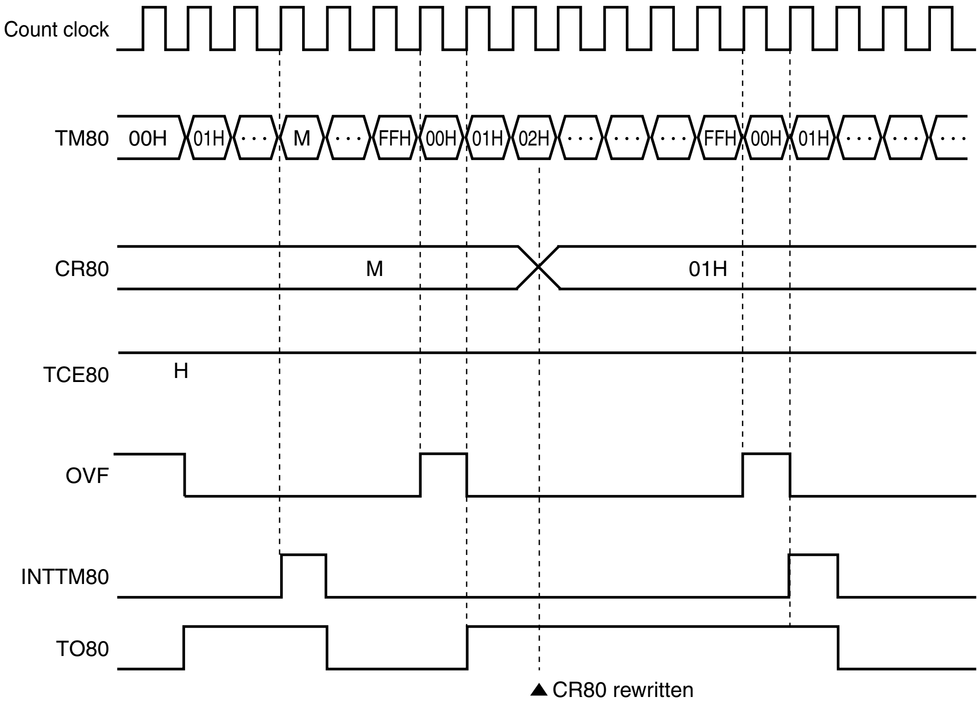
★ **2. If CR80 is rewritten during timer operation in PWM output operation mode (PWME80 = 1), pulses may not be generated for one cycle after the rewrite.**

3. Do not set CR80 to 00H in PWM operation mode (when PWME80 = 1) otherwise, PWM may not be output normally.

★ **(3) Timer operation after compare register is rewritten during PWM output**

When 8-bit compare register 80 (CR80) is rewritten during PWM output, if the new value is smaller than that of 8-bit timer/counter 80 (TM80), a high-level signal may be output for the next cycle (256 count pulses) after the CR80 value is rewritten. Figure 7-10 shows the timing at which the high-level signal is output.

Figure 7-10. Operation Timing After Compare Register Is Rewritten During PWM Output



M = 02H to FFH

★ **(4) Cautions when STOP mode is set**

Be sure to stop timer operations (TCE80 = 0) before executing the STOP instruction.

★ **(5) Start timing of external event counter**

When the rising edge of TI80 is selected as the count clock, start the timer when TI80 is low level (TCE80 = 0 → 1). Likewise, when the falling edge of TI80 is selected as the count clock, start the timer when TI80 is high level (TCE80 = 0 → 1).

CHAPTER 8 WATCHDOG TIMER

8.1 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 a $\overline{\text{RESET}}$ signal can be generated.

★ **Table 8-1. Inadvertent Loop Detection Time of Watchdog Timer**

Inadvertent Loop Detection Time	At fx = 10.0 MHz Operation^{Note}	At fx = 5.0 MHz Operation
$2^{11} \times 1/f_X$	205 μs	410 μs
$2^{13} \times 1/f_X$	819 μs	1.64 ms
$2^{15} \times 1/f_X$	3.28 ms	6.55 ms
$2^{17} \times 1/f_X$	13.1 ms	26.2 ms

Note Expanded-specification products only.

Remark fx: System clock oscillation frequency

(2) Interval timer

The interval timer generates an interrupt at an arbitrary preset interval.

★ **Table 8-2. Interval Time**

Interval Time	At fx = 10.0 MHz Operation^{Note}	At fx = 5.0 MHz Operation
$2^{11} \times 1/f_X$	205 μs	410 μs
$2^{13} \times 1/f_X$	819 μs	1.64 ms
$2^{15} \times 1/f_X$	3.28 ms	6.55 ms
$2^{17} \times 1/f_X$	13.1 ms	26.2 ms

Note Expanded-specification products only.

Remark fx: System clock oscillation frequency

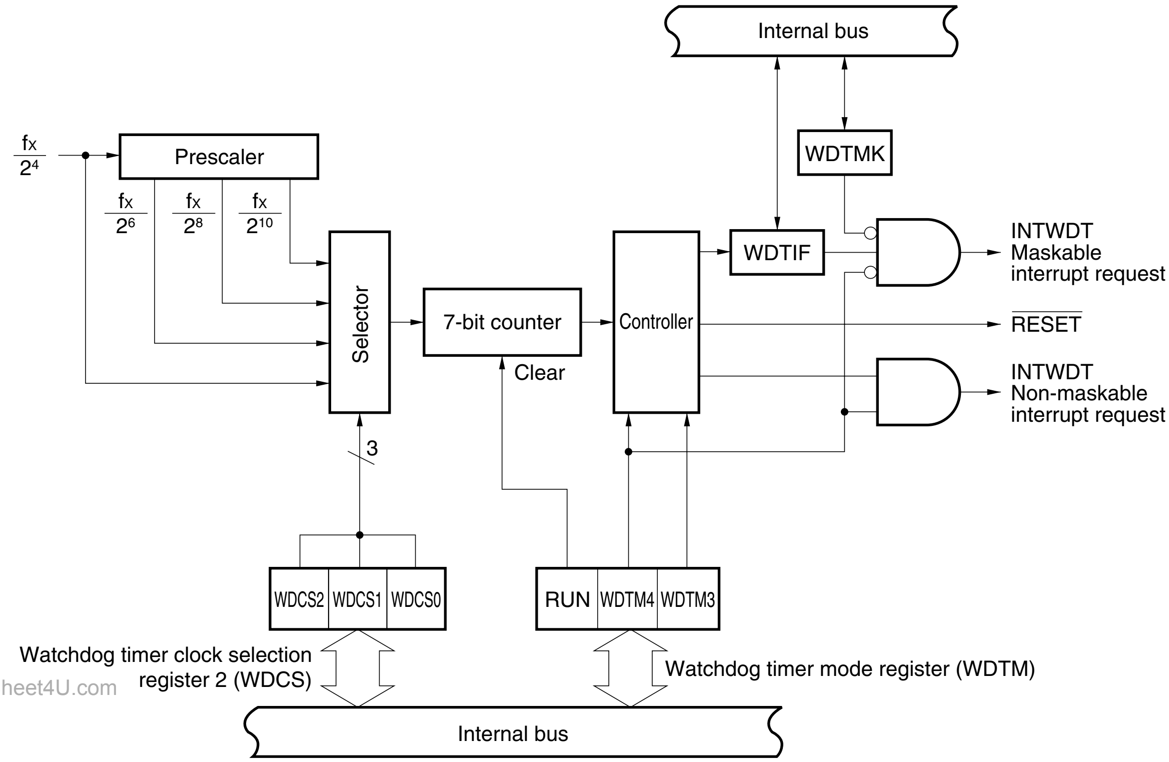
8.2 Watchdog Timer Configuration

The watchdog timer includes the following hardware.

Table 8-3. Configuration of Watchdog Timer

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

Figure 8-1. Block Diagram of Watchdog Timer



Internal bus
fx/2^4
Prescaler
fx/2^6
fx/2^8
fx/2^10
Selector
7-bit counter
Clear
Controller
WDTMK
WDTIF
INTWDT
Maskable
interrupt request
RESET
INTWDT
Non-maskable
interrupt request
3
WDCS2
WDCS1
WDCS0
RUN
WDTM4
WDTM3
Watchdog timer clock selection
register 2 (WDCS)
Watchdog timer mode register (WDTM)
Internal bus

8.3 Watchdog Timer Control Registers

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

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

(1) Watchdog timer clock selection register (WDCS)

This register sets the watchdog timer count clock.
WDCS is set with an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears WDCS to 00H.

★ **Figure 8-2. Format of Watchdog Timer Clock Selection 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	Count clock selection			Interval time		
				At f_X = 10.0 MHz operation$^{\text{Note}}$	At f_X = 5.0 MHz operation		At f_X = 10.0 MHz operation$^{\text{Note}}$	At f_X = 5.0 MHz operation
0	0	0	$2^4/f_X$	625 kHz	313 kHz	$2^{11}/f_X$	205 μs	410 μs
0	1	0	$2^6/f_X$	156 kHz	78.1 kHz	$2^{13}/f_X$	819 μs	1.64 ms
1	0	0	$2^8/f_X$	39.1 kHz	19.5 kHz	$2^{15}/f_X$	3.28 ms	6.55 ms
1	1	0	$2^{10}/f_X$	9.77 kHz	4.88 kHz	$2^{17}/f_X$	13.1 ms	26.2 ms
Other than above			Setting prohibited					

Note Expanded-specification products only.

Remark f_X: System clock oscillation frequency

(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 with a 1-bit or 8-bit memory manipulation instruction.

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

Figure 8-3. 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	Stop counting.
1	Clear counter and start counting.

WDTM4	WDTM3	Watchdog timer operation mode selection[Note 2]
0	0	Operation stop
0	1	Interval timer mode (a maskable interrupt is generated upon overflow occurrence)[Note 3]
1	0	Watchdog timer mode 1 (a non-maskable interrupt is generated upon overflow occurrence)
1	1	Watchdog timer mode 2 (a reset operation is started upon overflow occurrence)

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 operation 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 selection register (WDCS).

2. To set watchdog timer mode 1 or 2, set WDTM4 to 1 after confirming WDTIF (bit 0 of interrupt request flag register 0 (IF0)) is set to 0. When watchdog timer mode 1 or 2 is selected with WDTIF set to 1, a non-maskable interrupt is generated upon the completion of rewriting WDTM.

8.4 Watchdog Timer Operation

8.4.1 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 selection 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, a system reset signal or a non-maskable interrupt is generated, depending on the value of bit 3 (WDTM3) of WDTM.

The watchdog timer continues operation in HALT mode, but stops in STOP mode. Therefore, first set RUN to 1 to clear the watchdog timer before executing the STOP instruction.

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

★ **Table 8-4. Inadvertent Loop Detection Time of Watchdog Timer**

WDCS2	WDCS1	WDCS0	Inadvertent Loop Detection Time	At fx = 10.0 MHz Operation[Note]	At fx = 5.0 MHz Operation
0	0	0	$2^{11} \times 1/f_X$	205 μs	410 μs
0	1	0	$2^{13} \times 1/f_X$	819 μs	1.64 ms
1	0	0	$2^{15} \times 1/f_X$	3.28 ms	6.55 ms
1	1	0	$2^{17} \times 1/f_X$	13.1 ms	26.2 ms

Note Expanded-specification products only.

Remark fx: System clock oscillation frequency

8.4.2 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 operates as an interval timer that repeatedly generates an interrupt at an interval specified by a preset count value.

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

In 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 HALT mode, but stops in STOP mode. Therefore, first set RUN to 1 to clear the interval timer before executing the STOP instruction.

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

2. The interval time may be up to 0.8% shorter than the set time when WDTM has just been set.

★ **Table 8-5. Interval Generated Using Interval Timer**

WDCS2	WDCS1	WDCS0	Interval Time	At fx = 10.0 MHz Operation[Note]	At fx = 5.0 MHz Operation
0	0	0	$2^{11} \times 1/f_X$	205 μs	410 μs
0	1	0	$2^{13} \times 1/f_X$	819 μs	1.64 ms
1	0	0	$2^{15} \times 1/f_X$	3.28 ms	6.55 ms
1	1	0	$2^{17} \times 1/f_X$	13.1 ms	26.2 ms

Note Expanded-specification products only.

Remark fx: System clock oscillation frequency

CHAPTER 9 SERIAL INTERFACE 20

9.1 Functions of Serial Interface 20

Serial interface 20 has the following three modes.

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

(1) Operation stop 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 20 contains a UART-dedicated 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 clock input to the ASCK20 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 ($\overline{\text{SCK20}}$) line and two serial data lines (SI20 and SO20).

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 20 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 is useful for connecting peripheral I/O circuits and display controllers having conventional synchronous serial interfaces, such as those of the 75X/XL, 78K, and 17K Series devices.

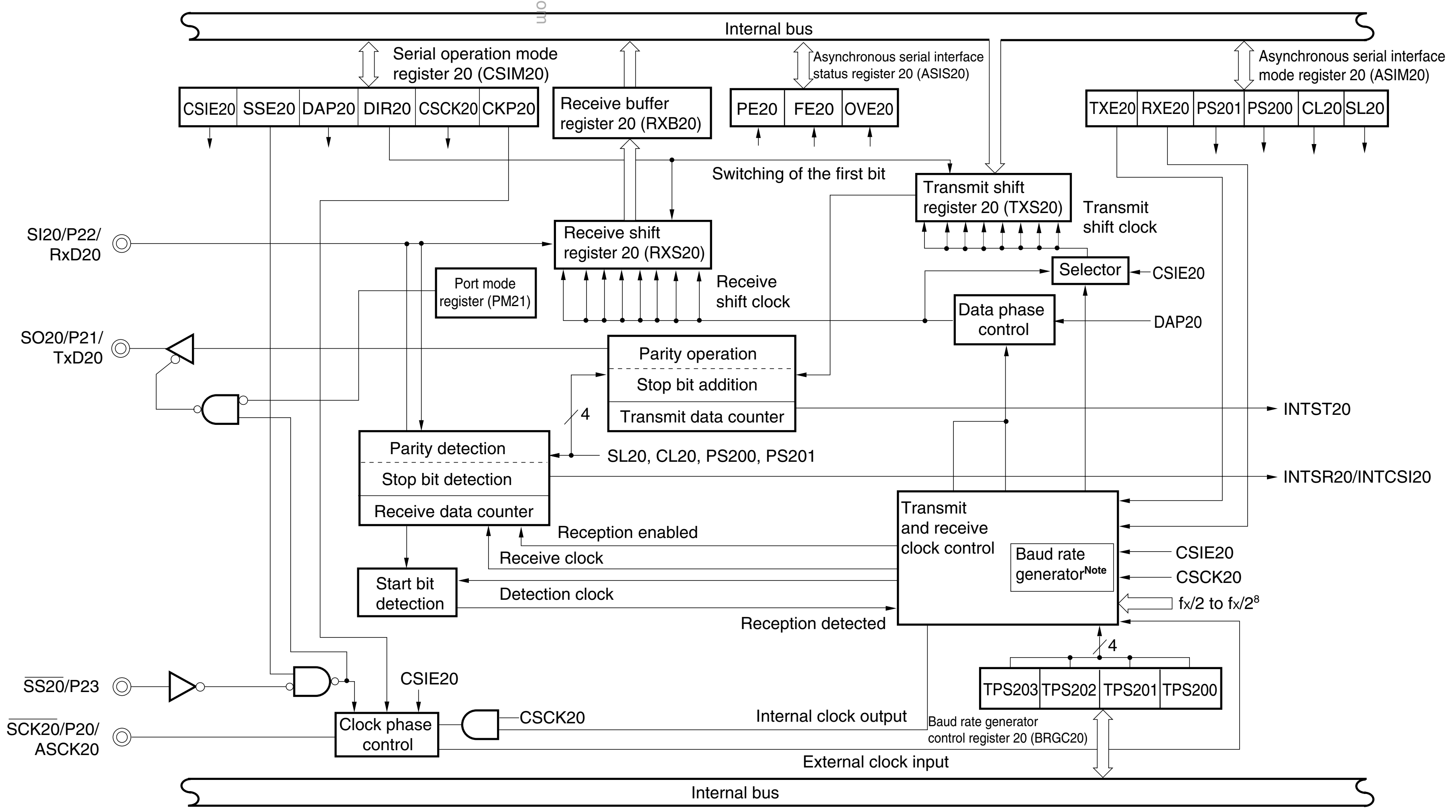
9.2 Configuration of Serial Interface 20

Serial interface 20 includes the following hardware.

Table 9-1. Configuration of Serial Interface 20

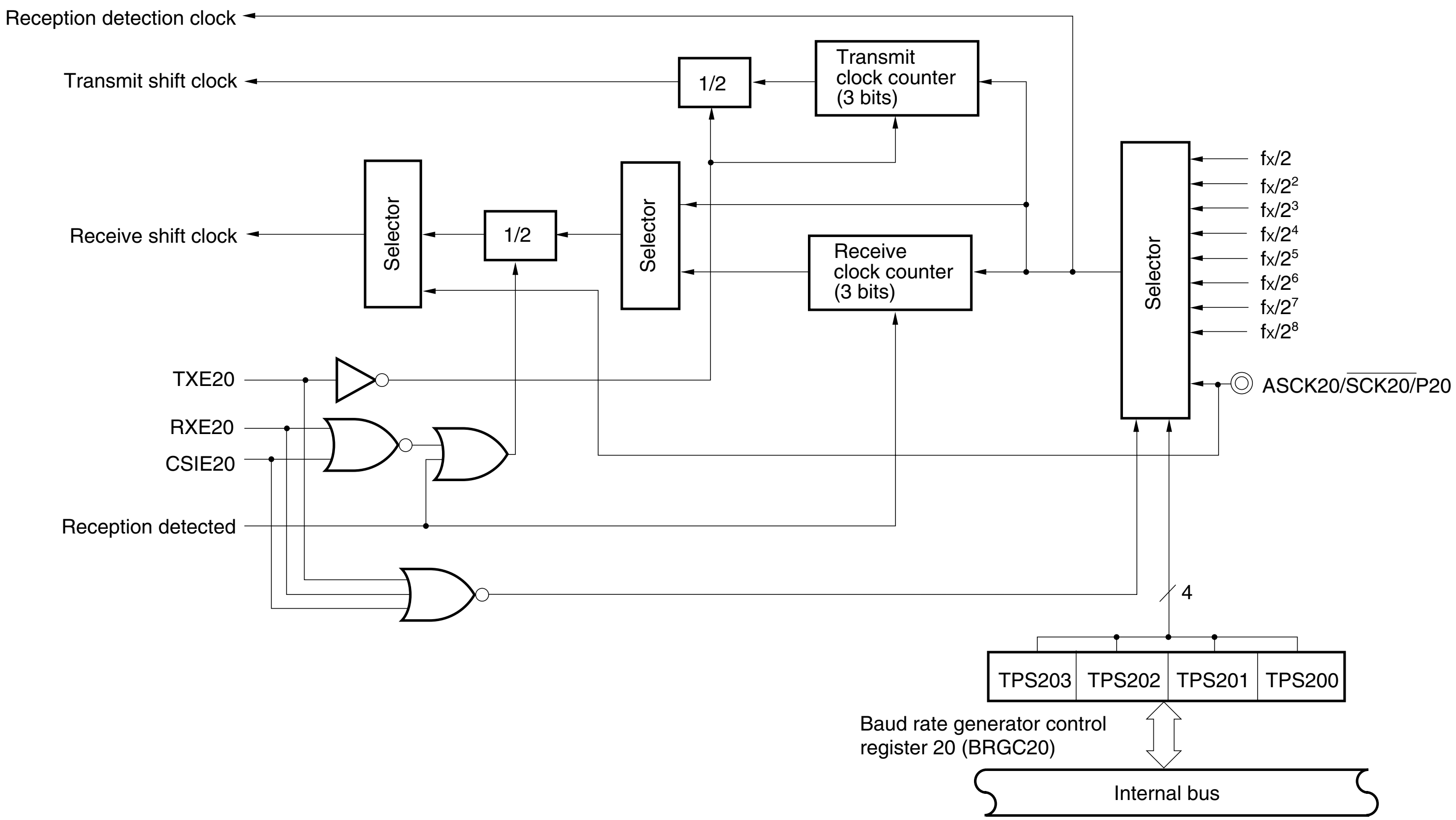
Item	Configuration
Registers	Transmission shift register 20 (TXS20) Reception shift register 20 (RXS20) Receive buffer register 20 (RXB20)
Control registers	Serial operation mode register 20 (CSIM20) Asynchronous serial interface mode register 20 (ASIM20) Asynchronous serial interface status register 20 (ASIS20) Baud rate generator control register 20 (BRGC20) Port mode register 2 (PM2) Port 2 (P2)

Figure 9-1. Block Diagram of Serial Interface 20


Internal bus
Serial operation mode register 20 (CSIM20)
CSIE20
SSE20
DAP20
DIR20
CSCK20
CKP20
Receive buffer register 20 (RXB20)
Asynchronous serial interface status register 20 (ASIS20)
PE20
FE20
OVE20
Asynchronous serial interface mode register 20 (ASIM20)
TXE20
RXE20
PS201
PS200
CL20
SL20
Switching of the first bit
Transmit shift register 20 (TXS20)
Transmit shift clock
SI20/P22/RxD20
Receive shift register 20 (RXS20)
Receive shift clock
Selector
CSIE20
Port mode register (PM21)
Data phase control
DAP20
SO20/P21/TxD20
Parity operation
Stop bit addition
Transmit data counter
4
INTST20
Parity detection
Stop bit detection
Receive data counter
SL20, CL20, PS200, PS201
INTSR20/INTCSI20
Transmit and receive clock control
Baud rate generatorNote
Reception enabled
Receive clock
CSIE20
CSCK20
Start bit detection
Detection clock
fx/2 to fx/2^8
Reception detected
4
TPS203
TPS202
TPS201
TPS200
SS20/P23
CSIE20
Clock phase control
CSCK20
Internal clock output
Baud rate generator control register 20 (BRGC20)
SCK20/P20/ASCK20
External clock input
Internal bus

Note See **Figure 9-2** for the configuration of the baud rate generator.

Figure 9-2. Block Diagram of Baud Rate Generator 20


Reception detection clock
Transmit shift clock
1/2
Transmit clock counter (3 bits)
Receive shift clock
Selector
1/2
Selector
Receive clock counter (3 bits)
Selector
fx/2
fx/2^2
fx/2^3
fx/2^4
fx/2^5
fx/2^6
fx/2^7
fx/2^8
ASCK20/$\overline{\text{SCK20}}$/P20
TXE20
RXE20
CSIE20
Reception detected
4
TPS203
TPS202
TPS201
TPS200
Baud rate generator control register 20 (BRGC20)
Internal bus

(1) Transmit shift register 20 (TXS20)

TXS20 is a register in which transmit data is prepared. The transmit data is output from TXS20 bit-serially. When the data length is seven bits, bits 0 to 6 of the data in TXS20 will be transmit data. Writing data to TXS20 triggers transmission.

TXS20 can be written with an 8-bit memory manipulation instruction, but cannot be read.

$\overline{\text{RESET}}$ input sets TXS20 to FFH.

Caution Do not write to TXS20 during transmission.
TXS20 and receive buffer register 20 (RXB20) are mapped at the same address, so that any attempt to read from TXS20 results in a value being read from RXB20.

(2) Receive shift register 20 (RXS20)

RXS20 is a register in which serial data, received at the RxD20 pin, is converted to parallel data. Once one entire byte has been received, RXS20 feeds the receive data to receive buffer register 20 (RXB20).

RXS20 cannot be manipulated directly by a program.

(3) Receive buffer register 20 (RXB20)

RXB20 holds a receive data. New receive data is transferred from receive shift register 20 (RXS20) at every 1-byte data reception.

When the data length is seven bits, the receive data is sent to bits 0 to 6 of RXB20, in which the MSB is always fixed to 0.

RXB20 can be read with an 8-bit memory manipulation instruction, but cannot be written.

$\overline{\text{RESET}}$ input makes RXB20 undefined.

Caution RXB20 and transmit shift register 20 (TXS20) are mapped at the same address, so that any attempt to write to RXB20 results in a value being written to TXS20.

(4) Transmission controller

The transmission controller controls transmission. For example, it adds start, parity, and stop bits to the data in transmit shift register 20 (TXS20), according to the setting of asynchronous serial interface mode register 20 (ASIM20).

(5) Reception controller

The reception controller controls reception according to the setting of asynchronous serial interface mode register 20 (ASIM20). It also checks for errors, such as parity errors, during reception. If an error is detected, asynchronous serial interface status register 20 (ASIS20) is set according to the status of the error.

9.3 Control Registers of Serial Interface 20

Serial interface 20 is controlled by the following six registers.

- Serial operation mode register 20 (CSIM20)
- Asynchronous serial interface mode register 20 (ASIM20)
- Asynchronous serial interface status register 20 (ASIS20)
- Baud rate generator control register 20 (BRGC20)
- Port mode register 2 (PM2)
- Port 2 (P2)

(1) Serial operation mode register 20 (CSIM20)

CSIM20 is set when serial interface 20 is used in 3-wire serial I/O mode.
CSIM20 is set with a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears CSIM20 to 00H.

Figure 9-3. Format of Serial Operation Mode Register 20

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
CSIM20	CSIE20	SSE20	0	0	DAP20	DIR20	CSCK20	CKP20	FF72H	00H	R/W

CSIE20	3-wire serial I/O mode operation control
0	Operation disabled
1	Operation enabled

SSE20	$\overline{\text{SS20}}$ pin selection	Function of $\overline{\text{SS20}}$/P23 pin	Communication status
0	Not used	Port function	Communication enabled
1	Used	0	Communication enabled
		1	Communication disabled

DAP20	3-wire serial I/O mode data phase selection
0	Outputs at the falling edge of $\overline{\text{SCK20}}$.
1	Outputs at the rising edge of $\overline{\text{SCK20}}$.

DIR20	First-bit specification
0	MSB
1	LSB

CSCK20	3-wire serial I/O mode clock selection
0	External clock input to the $\overline{\text{SCK20}}$ pin
1	Output of the dedicated baud rate generator

CKP20	3-wire serial I/O mode clock phase selection
0	Clock is active low, and $\overline{\text{SCK20}}$ is at high level in the idle state.
1	Clock is active high, and $\overline{\text{SCK20}}$ is at low level in the idle state.

Cautions 1. Bits 4 and 5 must both be set to 0.

2. CSIM20 must be cleared to 00H, if UART mode is selected.

(2) Asynchronous serial interface mode register 20 (ASIM20)

ASIM20 is set when serial interface 20 is used in asynchronous serial interface mode.

ASIM20 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ASIM20 to 00H.

Figure 9-4. Format of Asynchronous Serial Interface Mode Register 20

Symbol	<7>	<6>	5	4	3	2	1	0	Address	After reset	R/W
ASIM20	TXE20	RXE20	PS201	PS200	CL20	SL20	0	0	FF70H	00H	R/W

TXE20	Transmit operation control
0	Transmit operation stop
1	Transmit operation enable

RXE20	Receive operation control
0	Receive operation stop
1	Receive operation enable

PS201	PS200	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

CL20	Transmit data character length specification
0	7 bits
1	8 bits

SL20	Transmit data stop bit length
0	1 bit
1	2 bits

Cautions 1. Bits 0 and 1 must both be set to 0.

2. If 3-wire serial I/O mode is selected, ASIM20 must be cleared to 00H.

3. Switch operating modes after halting the serial transmit/receive operation.

Table 9-2. Operating Mode Settings of Serial Interface 20

(1) Operation stop mode

ASIM20		CSIM20			PM22	P22	PM21	P21	PM20	P20	First Bit	Shift Clock	P22/SI20/RxD20 Pin Function	P21/SO20/TxD20 Pin Function	P20/$\overline{\text{SCK20}}$/ASCK20 Pin Function
TXE20	RXE20	CSIE20	DIR20	CSCK20											
0	0	0	×	×	×[Note 1]	×[Note 1]	×[Note 1]	×[Note 1]	×[Note 1]	×[Note 1]	–	–	P22	P21	P20
Other than above											Setting prohibited				

(2) 3-wire serial I/O mode

ASIM20		CSIM20			PM22	P22	PM21	P21	PM20	P20	First Bit	Shift Clock	P22/SI20/RxD20 Pin Function	P21/SO20/TxD20 Pin Function	P20/$\overline{\text{SCK20}}$/ASCK20 Pin Function
TXE20	RXE20	CSIE20	DIR20	CSCK20											
0	0	1	0	0	1[Note 2]	×[Note 2]	0	1	1	×	MSB	External clock	SI20[Note 2]	SO20 (CMOS output)	$\overline{\text{SCK20}}$ input
				1					0	1		Internal clock			$\overline{\text{SCK20}}$ output
		1	1	0					1	×	LSB	External clock			$\overline{\text{SCK20}}$ input
				1					0	1		Internal clock			$\overline{\text{SCK20}}$ output
Other than above											Setting prohibited				

(3) Asynchronous serial interface mode

ASIM20		CSIM20			PM22	P22	PM21	P21	PM20	P20	First Bit	Shift Clock	P22/SI20/RxD20 Pin Function	P21/SO20/TxD20 Pin Function	P20/$\overline{\text{SCK20}}$/ASCK20 Pin Function
TXE20	RXE20	CSIE20	DIR20	CSCK20											
1	0	0	0	0	×[Note 1]	×[Note 1]	0	1	1	×	LSB	External clock	P22	TxD20 (CMOS output)	ASCK20 input
									×[Note 1]	×[Note 1]		Internal clock			P20
0	1	0	0	0	1	×	×[Note 1]	×[Note 1]	1	×		External clock	RxD20	P21	ASCK20 input
									×[Note 1]	×[Note 1]		Internal clock			P20
1	1	0	0	0	1	×	0	1	1	×		External clock		TxD20 (CMOS output)	ASCK20 input
									×[Note 1]	×[Note 1]		Internal clock			P20
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 P22 (CMOS I/O).

Remark ×: Don't care.

(3) Asynchronous serial interface status register 20 (ASIS20)

ASIS20 indicates the type of a reception error, if it occurs while asynchronous serial interface mode is set.

ASIS20 is read with a 1-bit or 8-bit memory manipulation instruction.

The contents of ASIS20 are undefined in 3-wire serial I/O mode.

$\overline{\text{RESET}}$ input clears ASIS20 to 00H.

Figure 9-5. Format of Asynchronous Serial Interface Status Register 20

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
ASIS20	0	0	0	0	0	PE20	FE20	OVE20	FF71H	00H	R

PE20	Parity error flag
0	No parity error occurred.
1	A parity error occurred (when the transmit parity and receive parity did not match).

FE20	Flaming error flag
0	No framing error occurred.
1	A framing error occurred (no stop bit detected).[Note 1]

OVE20	Overrun error flag
0	No overrun error occurred.
1	An overrun error occurred[Note 2]. (The subsequent receive operation was completed before data was read from the receive buffer register.)

Notes 1. Even when the stop bit length is set to 2 bits by setting bit 2 (SL20) of asynchronous serial interface mode register 20 (ASIM20), the stop bit detection at reception is performed with 1 bit.

2. Be sure to read receive buffer register 20 (RXB20) when an overrun error occurs. If not, every time the data is received an overrun error is generated.

(4) Baud rate generator control register 20 (BRGC20)

BRGC20 is used to specify the serial clock for serial interface 20.

BRGC20 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears BRGC20 to 00H.

★ **Figure 9-6. Format of Baud Rate Generator Control Register 20**

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
BRGC20	TPS203	TPS202	TPS201	TPS200	0	0	0	0	FF73H	00H	R/W

TPS203	TPS202	TPS201	TPS200	Selection of source clock for baud rate generator			n
					At f_X = 10.0 MHz operation[Note]	At f_X = 5.0 MHz operation	
0	0	0	0	$f_X/2$	5.0 MHz	2.5 MHz	1
0	0	0	1	$f_X/2^2$	2.5 MHz	1.25 MHz	2
0	0	1	0	$f_X/2^3$	1.25 MHz	625 kHz	3
0	0	1	1	$f_X/2^4$	625 kHz	313 kHz	4
0	1	0	0	$f_X/2^5$	313 kHz	156 kHz	5
0	1	0	1	$f_X/2^6$	156 kHz	78.1 kHz	6
0	1	1	0	$f_X/2^7$	78.1 kHz	39.1 kHz	7
0	1	1	1	$f_X/2^8$	39.1 kHz	19.5 kHz	8
1	0	0	0	External clock input to the ASCK20 pin[Note 2]			—
Other than above				Setting prohibited			

Notes 1. Expanded-specification products only.

2. An external clock can be used only in UART mode.

Cautions 1. When writing to BRGC20 is performed during a communication operation, the output of the baud rate generator is disrupted and communication cannot be performed normally. Be sure not to write to BRGC20 during a communication operation.

2. Be sure not to select n = 1 during operation at f_X > 2.5 MHz in UART mode because the resulting baud rate exceeds the rated range.

3. Be sure not to select n = 2 during operation at fx > 5.0 MHz in UART mode because the resulting serial clock exceeds the rated range.

4. Be sure not to select n = 1 during operation at fx > 5.0 MHz in 3-wire serial I/O mode because the resulting serial clock exceeds the rated range.

5. When the external input clock is selected, set port mode register 2 (PM2) to input mode.

Remarks 1. f_X: System clock oscillation frequency

2. n: Value determined by setting TPS200 through TPS203 ($1 \leq n \leq 8$)

The baud rate transmit/receive clock to be generated is either a signal generated by dividing the system clock, or a signal generated by dividing the clock input from the ASCK20 pin.

(a) Generation of baud rate transmit/receive clock from system clock

The transmit/receive clock is generated by dividing the system clock. The baud rate of a clock generated from the system clock is estimated by using the following expression.

$$[\text{Baud rate}] = \frac{f_X}{2^{n+1} \times 8} [\text{bps}]$$

f_X: System clock oscillation frequency

n: Value determined by settings of TPS200 through TPS203 as shown in Figure 9-6 ($2 \leq n \leq 8$)

★ **Table 9-3. Example of Relationship Between System Clock and Baud Rate**

Baud Rate (bps)	f_X = 10.0 MHz Note			f_X = 5.0 MHz			f_X = 4.9152 MHz		
	n	BRGC20 Set Value	Error (%)	n	BRGC20 Set Value	Error (%)	n	BRGC20 Set Value	Error (%)
1,200	–	–	1.73	8	70H	1.73	8	70H	0
2,400	8	70H		7	60H		7	60H	
4,800	7	60H		6	50H		6	50H	
9,600	6	50H		5	40H		5	40H	
19,200	5	40H		4	30H		4	30H	
38,400	4	30H		3	20H		3	20H	
76,800	3	20H		2	10H		2	10H	

Note Expanded-specification products only.

Cautions 1. Be sure not to select n = 1 during operation at fx > 2.5 MHz because the resulting baud rate exceeds the rated range.

2. Be sure not to select n = 2 during operation at fx > 5.0 MHz because the resulting baud rate exceeds the rated range.

(b) Generation of baud rate transmit/receive clock from external clock input from ASCK20 pin

The transmit/receive clock is generated by dividing the clock input from the ASCK20 pin. The baud rate of a clock generated from the clock input from the ASCK20 pin is estimated by using the following expression.

$$[\text{Baud rate}] = \frac{f_{\text{ASCK}}}{16}\ [\text{bps}]$$

f_{ASCK}: Frequency of clock input from the ASCK20 pin

Table 9-4. Relationship Between ASCK20 Pin Input Frequency and Baud Rate (When BRGC20 Is Set to 80H)

Baud Rate (bps)	ASCK20 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 in 3-wire serial I/O mode from system clock**

The serial clock is generated by dividing the system clock. The serial clock frequency is estimated by using the following expression. BRGC20 does not need to be set when an external serial clock is input to the $\overline{\text{SCK20}}$ pin.

$$\text{Serial clock frequency} = \frac{f_X}{2^{n+1}}\ [\text{Hz}]$$

f_X: System clock oscillation frequency

n: Value (shown in Figure 9-6) determined by setting TPS200 through TPS203 ($1 \leq n \leq 8$)

9.4 Operation of Serial Interface 20

Serial interface 20 provides the following three modes.

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

9.4.1 Operation stop mode

In operation stop mode, serial transfer is not executed; therefore, the power consumption can be reduced. The P20/$\overline{\text{SCK20}}$/ASCK20, P21/SO20/TxD20, and P22/SI20/RxD20 pins can be used as normal I/O ports.

(1) Register setting

Operation stop mode is set by serial operation mode register 20 (CSIM20) and asynchronous serial interface mode register 20 (ASIM20).

(a) Serial operation mode register 20 (CSIM20)

CSIM20 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM20 to 00H.

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
CSIM20	CSIE20	SSE20	0	0	DAP20	DIR20	CSCK20	CKP20	FF72H	00H	R/W

CSIE20	3-wire serial I/O mode operation control
0	Operation disabled
1	Operation enabled

Caution Bits 4 and 5 must both be set to 0.

(b) Asynchronous serial interface mode register 20 (ASIM20)

ASIM20 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ASIM20 to 00H.

Symbol	<7>	<6>	5	4	3	2	1	0	Address	After reset	R/W
ASIM20	TXE20	RXE20	PS201	PS200	CL20	SL20	0	0	FF70H	00H	R/W

TXE20	Transmit operation control
0	Transmit operation stop
1	Transmit operation enable

RXE20	Receive operation control
0	Receive operation stop
1	Receive operation enable

Caution Bits 0 and 1 must both be set to 0.

9.4.2 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 the desired baud rate from many options. In addition, the baud rate can also be defined by dividing the clock input to the ASCK20 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

UART mode is set by serial operation mode register 20 (CSIM20), asynchronous serial interface mode register 20 (ASIM20), asynchronous serial interface status register 20 (ASIS20), baud rate generator control register 20 (BRGC20), port mode register 2 (PM2), and port 2 (P2).

(a) Serial operation mode register 20 (CSIM20)

CSIM20 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM20 to 00H.

Set CSIM20 to 00H when UART mode is selected.

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
CSIM20	CSIE20	SSE20	0	0	DAP20	DIR20	CSCK20	CKP20	FF72H	00H	R/W

CSIE20	3-wire serial I/O mode operation control
0	Operation disabled
1	Operation enabled

SSE20	$\overline{\text{SS20}}$ pin selection	Function of $\overline{\text{SS20}}$/P23 pin	Communication status
0	Not used	Port function	Communication enabled
1	Used	0	Communication enabled
		1	Communication disabled

DAP20	3-wire serial I/O mode data phase selection
0	Outputs at the falling edge of $\overline{\text{SCK20}}$.
1	Outputs at the rising edge of $\overline{\text{SCK20}}$.

DIR20	First-bit specification
0	MSB
1	LSB

CSCK20	3-wire serial I/O mode clock selection
0	External clock input to the $\overline{\text{SCK20}}$ pin
1	Output of the dedicated baud rate generator

CKP20	3-wire serial I/O mode clock phase selection
0	Clock is active low, and $\overline{\text{SCK20}}$ is high level in the idle state.
1	Clock is active high, and $\overline{\text{SCK20}}$ is low level in the idle state.

Caution Bits 4 and 5 must both be set to 0.

(b) Asynchronous serial interface mode register 20 (ASIM20)

ASIM20 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ASIM20 to 00H.

Symbol	<7>	<6>	5	4	3	2	1	0	Address	After reset	R/W
ASIM20	TXE20	RXE20	PS201	PS200	CL20	SL20	0	0	FF70H	00H	R/W

TXE20	Transmit operation control
0	Transmit operation stopped
1	Transmit operation enabled

RXE20	Receive operation control
0	Receive operation stopped
1	Receive operation enabled

PS201	PS200	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 is generated.)
1	0	Odd parity
1	1	Even parity

CL20	Character length specification
0	7 bits
1	8 bits

SL20	Transmit data stop bit length specification
0	1 bit
1	2 bits

Cautions 1. Bits 0 and 1 must both be set to 0.

2. Switch operating modes after halting the serial transmit/receive operation.

(c) Asynchronous serial interface status register 20 (ASIS20)

ASIS20 is read with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ASIS20 to 00H.

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
ASIS20	0	0	0	0	0	PE20	FE20	OVE20	FF71H	00H	R

PE20	Parity error flag
0	Parity error did not occur
1	Parity error occurred (when the parity of transmit data did not match)

FE20	Flaming error flag
0	Framing error did not occur
1	Framing error occurred (when stop bit was not detected)Note 1

OVE20	Overrun error flag
0	Overrun error did not occur
1	Overrun error occurredNote 2 (when the next receive operation was completed before the data was read from the receive buffer register)

Notes 1. Even when the stop bit length is set to 2 bits by setting bit 2 (SL20) of asynchronous serial interface mode register 20 (ASIM20), the stop bit detection at reception is performed with 1 bit.

2. Be sure to read receive buffer register 20 (RXB20) when an overrun error occurs. If not, every time the data is received an overrun error is generated.

(d) Baud rate generator control register 20 (BRGC20)

BRGC20 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears BRGC20 to 00H.

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
BRGC20	TPS203	TPS202	TPS201	TPS200	0	0	0	0	FF73H	00H	R/W

★

TPS203	TPS202	TPS201	TPS200	Selection of source clock for baud rate generator			n
					At f_X = 10.0 MHz operation[Note]	At f_X = 5.0 MHz operation	
0	0	0	0	$f_X/2$	5.0 MHz	2.5 MHz	1
0	0	0	1	$f_X/2^2$	2.5 MHz	1.25 MHz	2
0	0	1	0	$f_X/2^3$	1.25 MHz	625 kHz	3
0	0	1	1	$f_X/2^4$	625 kHz	313 kHz	4
0	1	0	0	$f_X/2^5$	313 kHz	156 kHz	5
0	1	0	1	$f_X/2^6$	156 kHz	78.1 kHz	6
0	1	1	0	$f_X/2^7$	78.1 kHz	39.1 kHz	7
0	1	1	1	$f_X/2^8$	39.1 kHz	19.5 kHz	8
1	0	0	0	External clock input to ASCK20 pin			—
Other than above				Setting prohibited			

Note Expanded-specification products only.

Cautions 1. When writing to BRGC20 is performed during a communication operation, the output of the baud rate generator is disrupted and communication cannot be performed normally. Be sure not to write to BRGC20 during a communication operation.

2. Be sure not to select n = 1 during operation at f_X > 2.5 MHz because the resulting baud rate exceeds the rated range.

3. Be sure not to select n = 2 during operation at f_X > 5.0 MHz because the resulting baud rate exceeds the rated range.

4. When the external input clock is selected, set port mode register 2 (PM2) to input mode.

Remarks 1. f_X: System clock oscillation frequency

2. n: Value determined by setting TPS200 through TPS203 ($1 \le n \le 8$)

The baud rate transmit/receive clock to be generated is either a signal divided from the system clock, or a signal divided from the clock input from the ASCK20 pin.

(i) Generation of baud rate transmit/receive clock from system clock

The transmit/receive clock is generated by dividing the system clock. The baud rate of the clock generated from the system clock is estimated by using the following expression.

$$[\text{Baud rate}] = \frac{f_X}{2^{n+1} \times 8} \ [\text{bps}]$$

f_X: System clock oscillation frequency

n: Value determined by setting TPS200 through TPS203 as shown in the above table ($2 \le n \le 8$)

★ **Table 9-5. Example of Relationship Between System Clock and Baud Rate**

Baud Rate (bps)	f_X = 10.0 MHz [Note]			f_X = 5.0 MHz			f_X = 4.9152 MHz		
	n	BRGC20 Set Value	Error (%)	n	BRGC20 Set Value	Error (%)	n	BRGC20 Set Value	Error (%)
1,200	–	–	1.73	8	70H	1.73	8	70H	0
2,400	8	70H		7	60H		7	60H	
4,800	7	60H		6	50H		6	50H	
9,600	6	50H		5	40H		5	40H	
19,200	5	40H		4	30H		4	30H	
38,400	4	30H		3	20H		3	20H	
76,800	3	20H		2	10H		2	10H	

Note Expanded-specification products only.

Cautions 1. Be sure not to select n = 1 during operation at fx > 2.5 MHz because the resulting baud rate exceeds the rated range.

2. Be sure not to select n = 2 during operation at fx > 5.0 MHz because the resulting baud rate exceeds the rated range.

(ii) Generation of baud rate transmit/receive clock from external clock input from ASCK20 pin

The transmit/receive clock is generated by dividing the clock input from the ASCK20 pin. The baud rate of the clock generated from the clock input from the ASCK20 pin is estimated by using the following expression.

$$[\text{Baud rate}] = \frac{f_{ASCK}}{16}\ [\text{bps}]$$

f_{ASCK}: Frequency of clock input from the ASCK20 pin

Table 9-6. Relationship Between ASCK20 Pin Input Frequency and Baud Rate (When BRGC20 Is Set to 80H)

Baud Rate (bps)	ASCK20 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

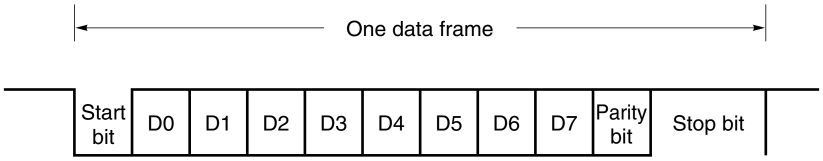
(2) Communication operation

(a) Data format

The transmit/receive data format is as shown in Figure 9-7. One data frame consists of a start bit, character bits, a parity bit, and stop bit(s).

The specification of the character bit length in one data frame, parity selection, and specification of the stop bit length is carried out with asynchronous serial interface mode register 20 (ASIM20).

Figure 9-7. Format of Asynchronous Serial Interface Transmit/Receive Data



- Start bits .... 1 bit
- Character bits .... 7 bits/8 bits
- Parity bits .... Even parity/odd parity/0 parity/no parity
- Stop bit(s) .... 1 bit/2 bits

When 7 bits is 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 baud rate generator control register 20 (BRGC20).

If a serial data receive error is generated, the receive error contents can be determined by reading the status of asynchronous serial interface status register 20 (ASIS20).

(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 one-bit (odd number) error can be detected. With 0 parity and no parity, an error cannot be detected.

(i) Even parity

- **At transmission**

  The parity bit is determined so that the number of bits with a value of "1" in the transmit data including the parity bit is 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

- **At reception**

  The number of bits with a value of "1" in the receive data including the parity bit is counted, and if the number is odd, a parity error occurs.

(ii) Odd parity

- **At transmission**

  Conversely to even parity, the parity bit is determined so that the number of bits with a value of "1" in the transmit data including the parity bit is 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

- **At reception**

  The number of bits with a value of "1" in the receive data including the 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.

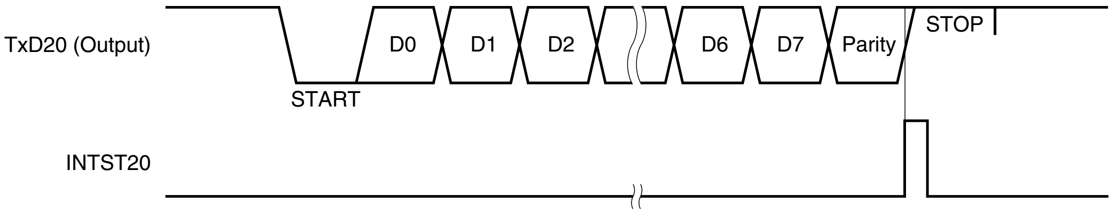
(c) Transmission

A transmit operation is started by writing transmit data to transmit shift register 20 (TXS20). The start bit, parity bit, and stop bit(s) are added automatically.

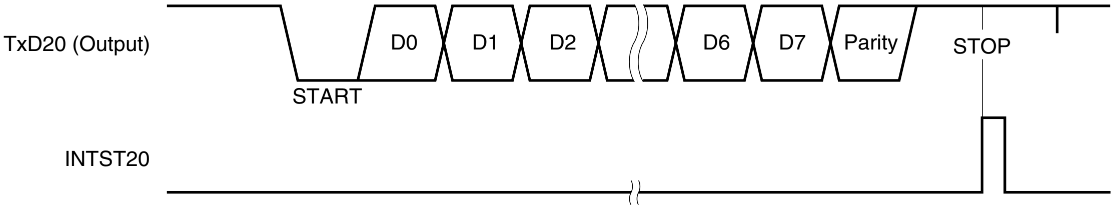
When the transmit operation starts, the data in TXS20 is shifted out, and when TXS20 is empty, a transmission completion interrupt (INTST20) is generated.

Figure 9-8. 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 20 (ASIM20) during a transmit operation. If the ASIM20 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 (RXE20) of asynchronous serial interface mode register 20 (ASIM20) is set to 1, a receive operation is enabled and sampling of the RxD20 pin input is performed.

RxD20 pin input sampling is performed using the serial clock specified by BRGC20.

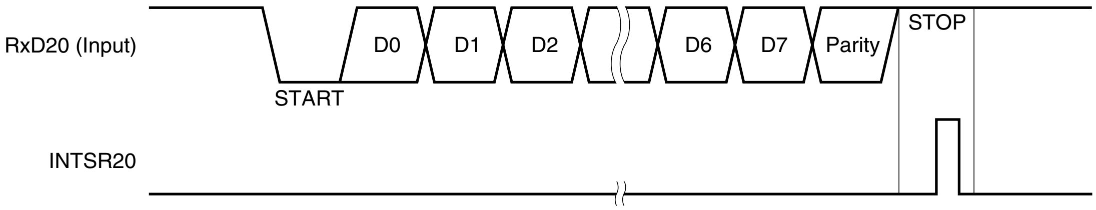
When the RxD20 pin input becomes low, the 3-bit counter starts counting, and at the time when half the time determined by the specified baud rate has passed, the data sampling start timing signal is output. If the RxD20 pin input sampled again as a result of this start timing signal is low, it is identified 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 20 (RXB20), and a reception completion interrupt (INTSR20) is generated.

If an error occurs, the receive data in which the error occurred is still transferred to RXB20, and INTSR20 is generated.

If the RXE20 bit is reset to 0 during the receive operation, the receive operation is stopped immediately. In this case, the contents of RXB20 and asynchronous serial interface status register 20 (ASIS20) are not changed, and INTSR20 is not generated.

Figure 9-9. Asynchronous Serial Interface Reception Completion Interrupt Timing



Caution **Be sure to read receive buffer register 20 (RXB20) even if a receive error occurs. If RXB20 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, a framing error, and an overrun error. After data reception, an error flag is set in asynchronous serial interface status register 20 (ASIS20). Receive error causes are shown in Table 9-7.

It is possible to determine what kind of error occurred during reception by reading the contents of ASIS20 in the reception error interrupt servicing (see **Table 9-7** and **Figure 9-10**).

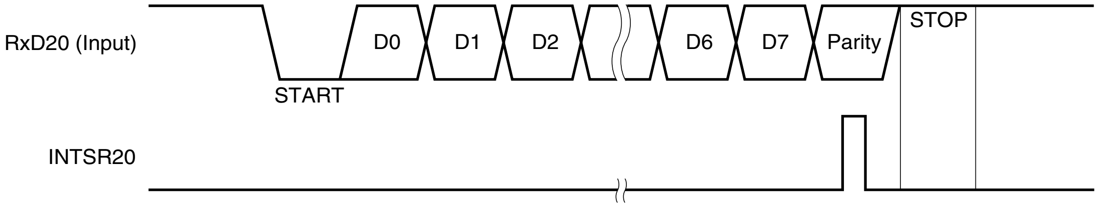
The contents of ASIS20 are reset to 0 by reading receive buffer register 20 (RXB20) or receiving the next data (if there is an error in the next data, the corresponding error flag is set).

Table 9-7. Receive Error Causes

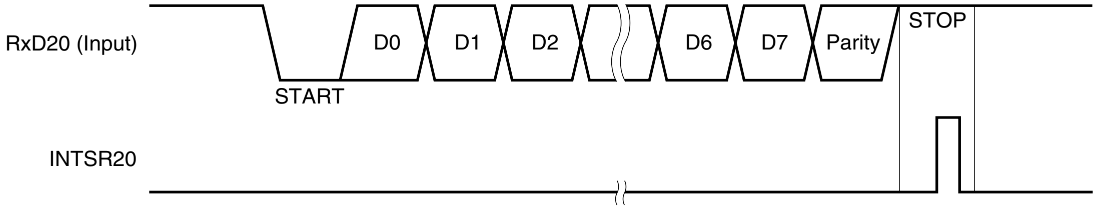
Receive Errors	Cause
Parity error	Transmission-time parity and reception data parity do not match.
Framing error	Stop bit not detected
Overrun error	Reception of next data is completed before data is read from receive buffer register.

Figure 9-10. Receive Error Timing

(a) Parity error occurred



(b) Framing error or overrun error occurred



Cautions 1. The contents of the ASIS20 register are reset to 0 by reading receive buffer register 20 (RXB20) or receiving the next data. To ascertain the error contents, read ASIS20 before reading RXB20.

2. Be sure to read receive buffer register 20 (RXB20) even if a receive error occurs. If RXB20 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

When the reception completion interrupt (INTSR20) is generated, read the value of receive buffer register 20 (RXB20) to read the receive data.

When reading the receive data stored in receive buffer register 20 (RXB20), enable the receive operation (RXE20 = 1).

Remark If the receive data must be read after the receive operation has been disabled (RXE20 = 0), use either method below.

(a) After waiting for 1 cycle or more of the source clock selected by BRGC20, set RXE20 to 0, and then read the receive data.

(b) Set bit 2 (DIR20) of serial operation mode register 20 (CSIM20) to 1, and read the receive data.

Example program for (a) (BRGC29 = 00H (source clock = fx/2))

```
INTRXE:                     ;Reception completion interrupt routine
         NOP                ;2 clocks
         CLR1  RXE20        ;Stop reception operation
         MOV   A,RXB20      ;Read receive data
```

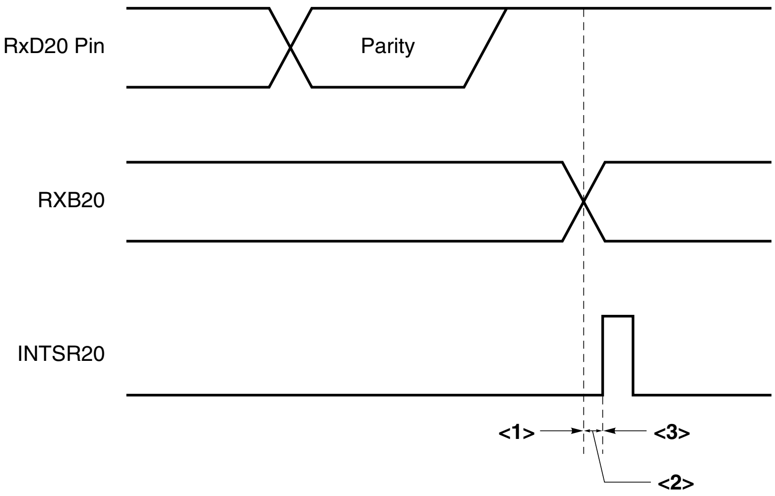
Example program for (b)

```
INTRXE:                     ;Reception completion interrupt routine
         SET1  CSIM20.2     ;Set the DIR20 flag to LSB first
         CLR1  RXE20        ;Stop reception operation
         MOV   A,RXB20      ;Read receive data
```

(3) Cautions related to UART mode

(a) When bit 7 (TXE20) of asynchronous serial interface mode register 20 (ASIM20) is cleared during transmission, be sure to set transmit shift register 20 (TXS20) to FFH, then set TXE20 to 1 before executing the next transmission.

(b) When bit 6 (RXE20) of asynchronous serial interface mode register 20 (ASIM20) is cleared during reception, receive buffer register 20 (RXB20) and the receive completion interrupt (INTSR20) are as follows.



When RXE20 is set to 0 at the time indicated by **<1>**, RXB20 holds the previous data and INTSR20 is not generated.
When RXE20 is set to 0 at the time indicated by **<2>**, RXB20 renews the data and INTSR20 is not generated.
When RXE20 is set to 0 at the time indicated by **<3>**, RXB20 renews the data and INTSR20 is generated.

9.4.3 3-wire serial I/O mode

The 3-wire serial I/O mode is useful for connection of peripheral I/Os and display controllers, etc., that incorporate a conventional synchronous serial interface, such as the 75XL Series, 78K Series, 17K Series, etc.

Communication is performed using three lines: the serial clock ($\overline{\text{SCK20}}$), serial output (SO20), and serial input (SI20).

(1) Register setting

3-wire serial I/O mode settings are performed using serial operation mode register 20 (CSIM20), asynchronous serial interface mode register 20 (ASIM20), baud rate generator control register 20 (BRGC20), port mode register 2 (PM2), and port 2 (P2).

(a) Serial operation mode register 20 (CSIM20)

CSIM20 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM20 to 00H.

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
CSIM20	CSIE20	SSE20	0	0	DAP20	DIR20	CSCK20	CKP20	FF72H	00H	R/W

CSIE20	3-wire serial I/O mode operation control
0	Operation disabled
1	Operation enabled

SSE20	$\overline{\text{SS20}}$ pin selection	Function of $\overline{\text{SS20}}$/P23 pin	Communication status
0	Not used	Port function	Communication enabled
1	Used	0	Communication enabled
		1	Communication disabled

DAP20	3-wire serial I/O mode data phase selection
0	Outputs at the falling edge of $\overline{\text{SCK20}}$.
1	Outputs at the rising edge of $\overline{\text{SCK20}}$.

DIR20	First-bit specification
0	MSB
1	LSB

CSCK20	3-wire serial I/O mode clock selection
0	External clock input to the $\overline{\text{SCK20}}$ pin
1	Output of the dedicated baud rate generator

CKP20	3-wire serial I/O mode clock phase selection
0	Clock is active low, and $\overline{\text{SCK20}}$ is at high level in the idle state.
1	Clock is active high, and $\overline{\text{SCK20}}$ is at low level in the idle state.

Caution Bits 4 and 5 must both be set to 0.

(b) Asynchronous serial interface mode register 20 (ASIM20)

ASIM20 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ASIM20 to 00H.

When 3-wire serial I/O mode is selected, ASIM20 must be set to 00H.

Symbol	<7>	<6>	5	4	3	2	1	0	Address	After reset	R/W
ASIM20	TXE20	RXE20	PS201	PS200	CL20	SL20	0	0	FF70H	00H	R/W

TXE20	Transmit operation control
0	Transmit operation stop
1	Transmit operation enable

RXE20	Receive operation control
0	Receive operation stop
1	Receive operation enable

PS201	PS200	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

CL20	Character length specification
0	7 bits
1	8 bits

SL20	Transmit data sop bit length specification
0	1 bit
1	2 bits

Cautions 1. Bits 0 and 1 must both be set to 0.

2. Switch operating modes after halting the serial transmit/receive operation.

(c) Baud rate generator control register 20 (BRGC20)

BRGC20 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears BRGC20 to 00H.

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
BRGC20	TPS203	TPS202	TPS201	TPS200	0	0	0	0	FF73H	00H	R/W

★

TPS203	TPS202	TPS201	TPS200	Selection of source clock for baud rate generator			n
					At f_X = 10.0 MHz operation[Note]	At f_X = 5.0 MHz operation	
0	0	0	0	$f_X/2$	5.0 MHz	2.5 MHz	1
0	0	0	1	$f_X/2^2$	2.5 MHz	1.25 MHz	2
0	0	1	0	$f_X/2^3$	1.25 MHz	625 kHz	3
0	0	1	1	$f_X/2^4$	625 kHz	313 kHz	4
0	1	0	0	$f_X/2^5$	313 kHz	156 kHz	5
0	1	0	1	$f_X/2^6$	156 kHz	78.1 kHz	6
0	1	1	0	$f_X/2^7$	78.1 kHz	39.1 kHz	7
0	1	1	1	$f_X/2^8$	39.1 kHz	19.5 kHz	8
Other than above				Setting prohibited			

Note Expanded-specification products only.

Cautions 1. When writing to BRGC20 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 BRGC20 during a communication operation.

2. Be sure not to select n = 1 during operation at $f_X > 5.0$ MHz in 3-wire serial I/O mode because the resulting serial clock exceeds the rated range.

3. When the external input clock is selected, set port mode register 2 (PM2) to input mode.

Remarks 1. f_X: System clock oscillation frequency

2. n: Value determined by setting TPS200 through TPS203 ($1 \le n \le 8$)

If the internal clock is used as the serial clock for 3-wire serial I/O mode, set bits TPS200 to TPS203 to set the frequency of the serial clock. To obtain the frequency to be set, use the following expression. When an external serial clock is used, setting BRGC20 is not necessary.

$$\text{Serial clock frequency} = \frac{f_X}{2^{n+1}} \text{ [Hz]}$$

f_X: System clock oscillation frequency

n: Value determined by setting TPS200 to TPS203 as shown in the above table ($1 \le n \le 8$)

(2) Communication operation

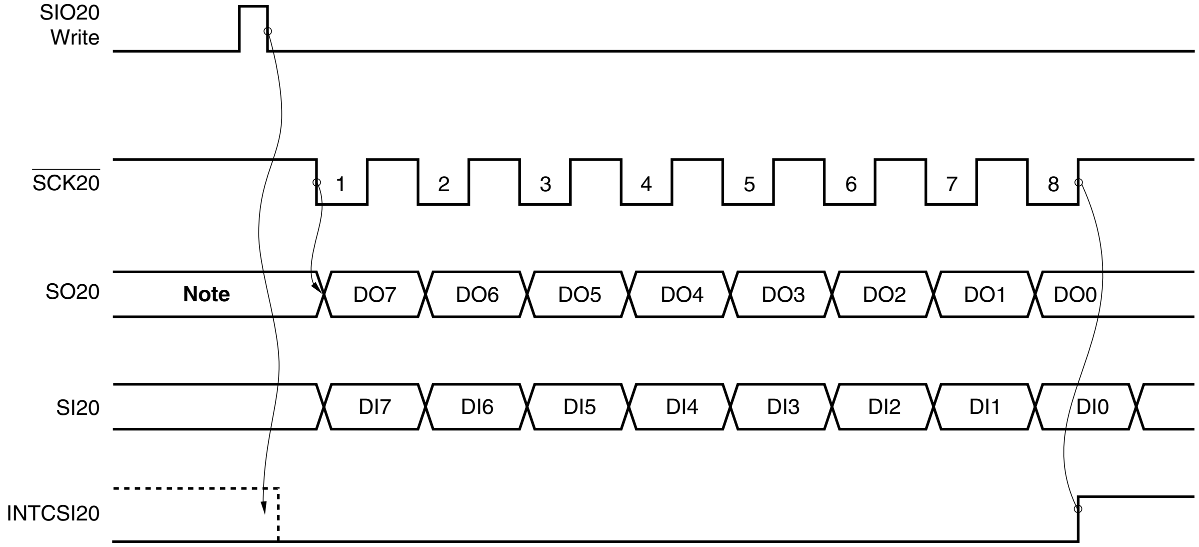
In 3-wire serial I/O mode, data transmission/reception is performed in 8-bit units. Data is transmitted/received bit by bit in synchronization with the serial clock.

Transmit shift register 20 (TXS20/SIO20) and receive shift register 20 (RXS20) shift operations are performed in synchronization with the fall of the serial clock ($\overline{\text{SCK20}}$). Then transmit data is held in the SO20 latch and output from the SO20 pin. Also, receive data input to the SI20 pin is latched in receive buffer register 20 (RXB20/SIO20) on the rise of $\overline{\text{SCK20}}$.

At the end of an 8-bit transfer, the operation of TXS20/SIO20 and RXS20 stops automatically, and the interrupt request signal (INTCSI20) is generated.

Figure 9-11. 3-Wire Serial I/O Mode Timing (1/7)

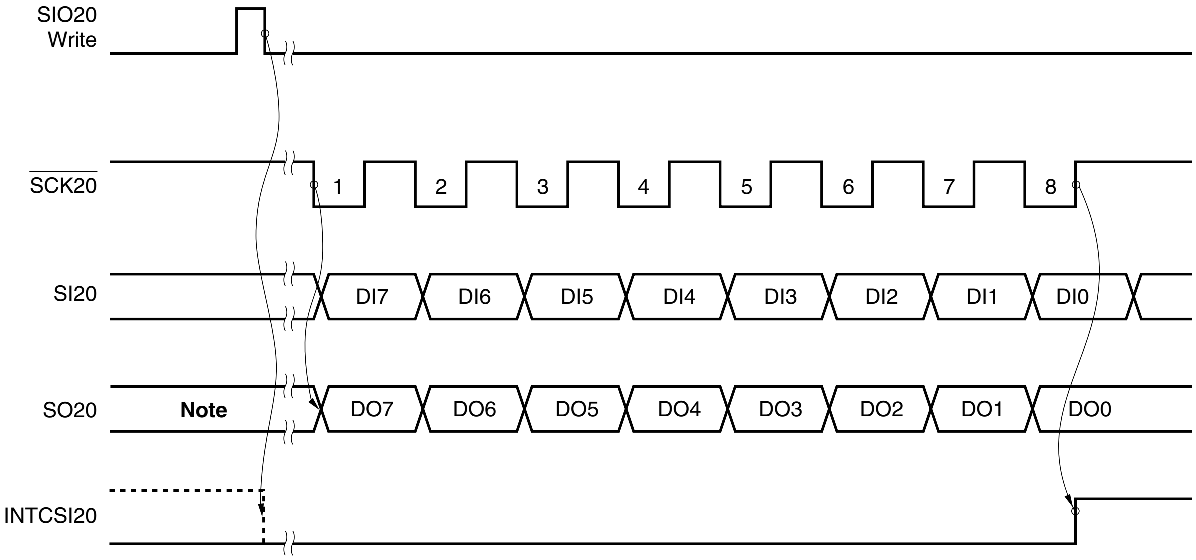
(i) Master operation timing (when DAP20 = 0, CKP20 = 0, SSE20 = 0)


SIO20
Write
$\overline{\text{SCK20}}$
1
2
3
4
5
6
7
8
SO20
Note
DO7
DO6
DO5
DO4
DO3
DO2
DO1
DO0
SI20
DI7
DI6
DI5
DI4
DI3
DI2
DI1
DI0
INTCSI20

Note The value of the last bit previously output is output.

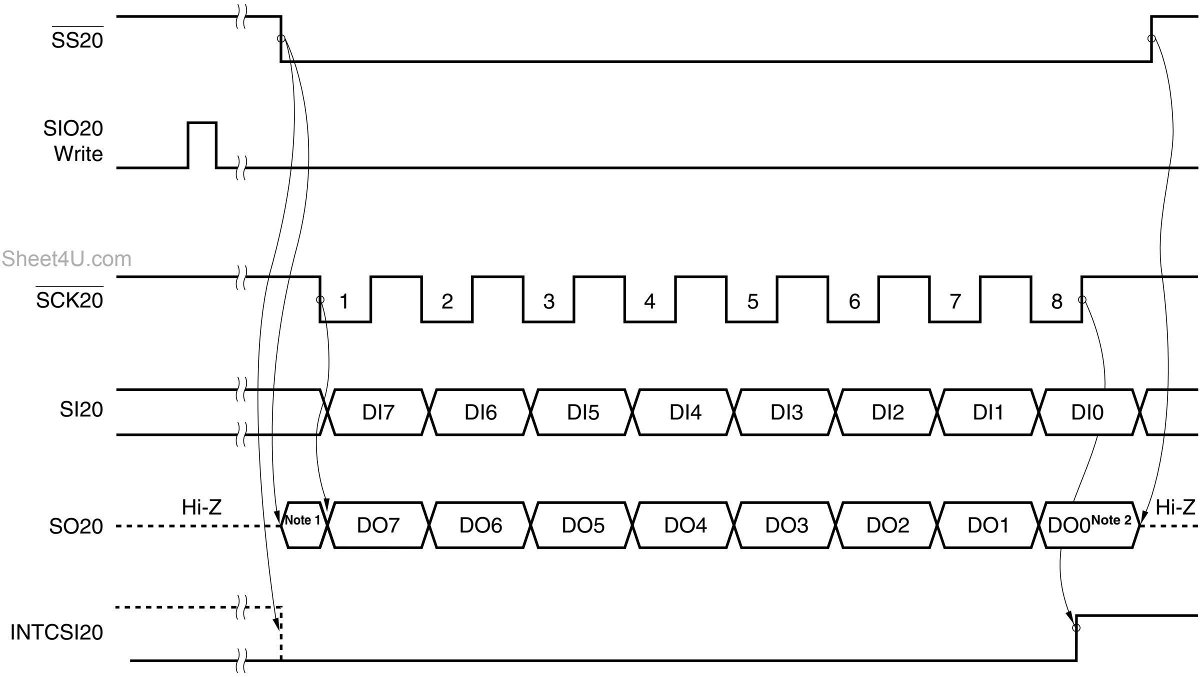
Figure 9-11. 3-Wire Serial I/O Mode Timing (2/7)

(ii) Slave operation timing (when DAP20 = 0, CKP20 = 0, SSE20 = 0)


SIO20 Write
SCK20
1
2
3
4
5
6
7
8
SI20
DI7
DI6
DI5
DI4
DI3
DI2
DI1
DI0
SO20
Note
DO7
DO6
DO5
DO4
DO3
DO2
DO1
DO0
INTCSI20

Note The value of the last bit previously output is output.

(iii) Slave operation (when DAP20 = 0, CKP20 = 0, SSE20 = 1)

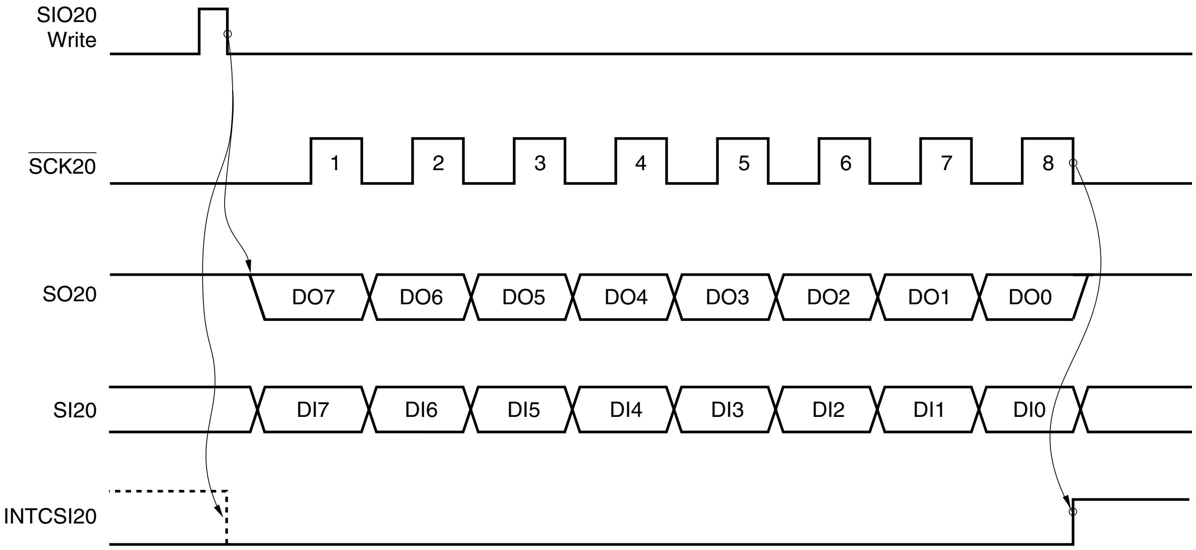

SS20
SIO20 Write
SCK20
1
2
3
4
5
6
7
8
SI20
DI7
DI6
DI5
DI4
DI3
DI2
DI1
DI0
SO20
Hi-Z
Note 1
DO7
DO6
DO5
DO4
DO3
DO2
DO1
DO0
Note 2
Hi-Z
INTCSI20

Notes 1. The value of the last bit previously output is output.

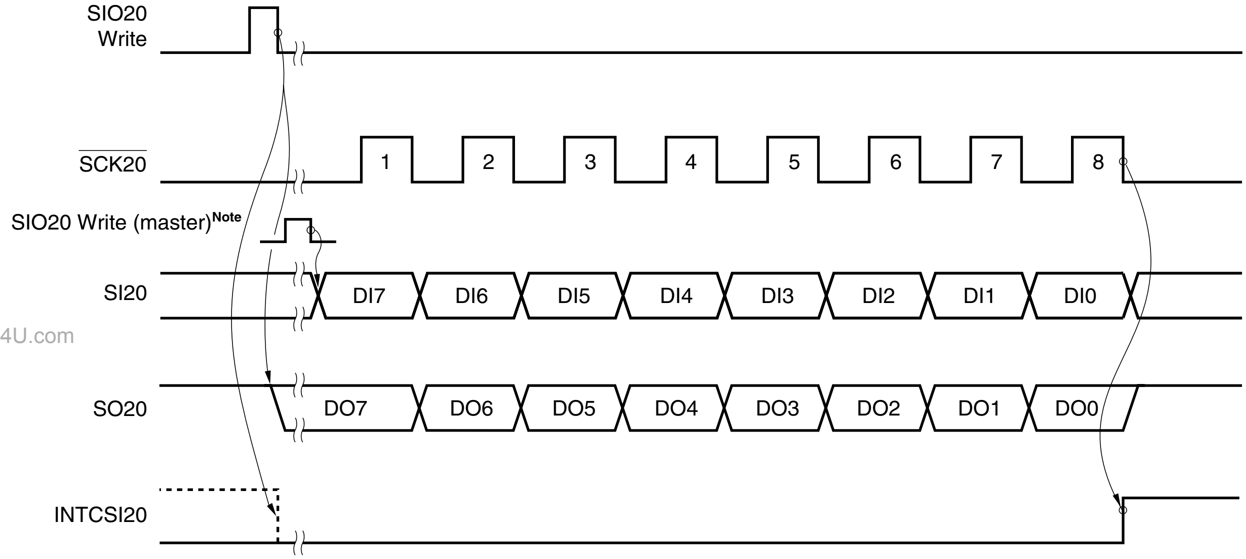
2. DO0 is output until $\overline{SS20}$ rises.
When $\overline{SS20}$ is high, SO20 is in a high-impedance state.

Figure 9-11. 3-Wire Serial I/O Mode Timing (3/7)

(iv) Master operation (when DAP20 = 0, CKP20 = 1, SSE20 = 0)


SIO20 Write
SCK20
1
2
3
4
5
6
7
8
SO20
DO7
DO6
DO5
DO4
DO3
DO2
DO1
DO0
SI20
DI7
DI6
DI5
DI4
DI3
DI2
DI1
DI0
INTCSI20

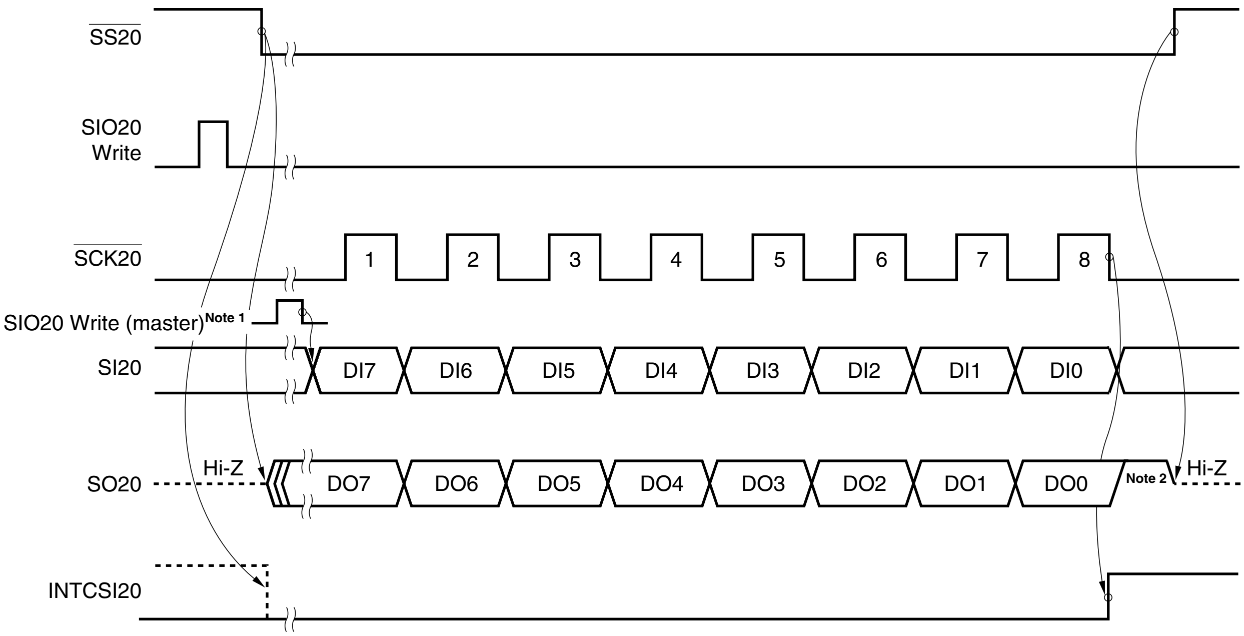
(v) Slave operation (when DAP20 = 0, CKP20 = 1, SSE20 = 0)


SIO20 Write
SCK20
1
2
3
4
5
6
7
8
SIO20 Write (master)Note
SI20
DI7
DI6
DI5
DI4
DI3
DI2
DI1
DI0
SO20
DO7
DO6
DO5
DO4
DO3
DO2
DO1
DO0
INTCSI20

Note The data of SI20 is loaded at the first rising edge of $\overline{\text{SCK20}}$. Make sure that the master outputs the first bit before the first rising of $\overline{\text{SCK20}}$.

Figure 9-11. 3-Wire Serial I/O Mode Timing (4/7)

(vi) Slave operation (when DAP20 = 0, CKP20 = 1, SSE20 = 1)


SS20
SIO20 Write
SCK20
1
2
3
4
5
6
7
8
SIO20 Write (master)Note 1
SI20
DI7
DI6
DI5
DI4
DI3
DI2
DI1
DI0
SO20
Hi-Z
DO7
DO6
DO5
DO4
DO3
DO2
DO1
DO0
Note 2
Hi-Z
INTCSI20

Notes 1. The data of SI20 is loaded at the first rising edge of $\overline{\text{SCK20}}$. Make sure that the master outputs the first bit before the first rising of $\overline{\text{SCK20}}$.

2. SO20 is high until $\overline{\text{SS20}}$ rises after completion of DO0 output. When $\overline{\text{SS20}}$ is high, SO20 is in a high-impedance state.

(vii) Master operation (when DAP20 = 1, CKP20 = 0, SSE20 = 0)

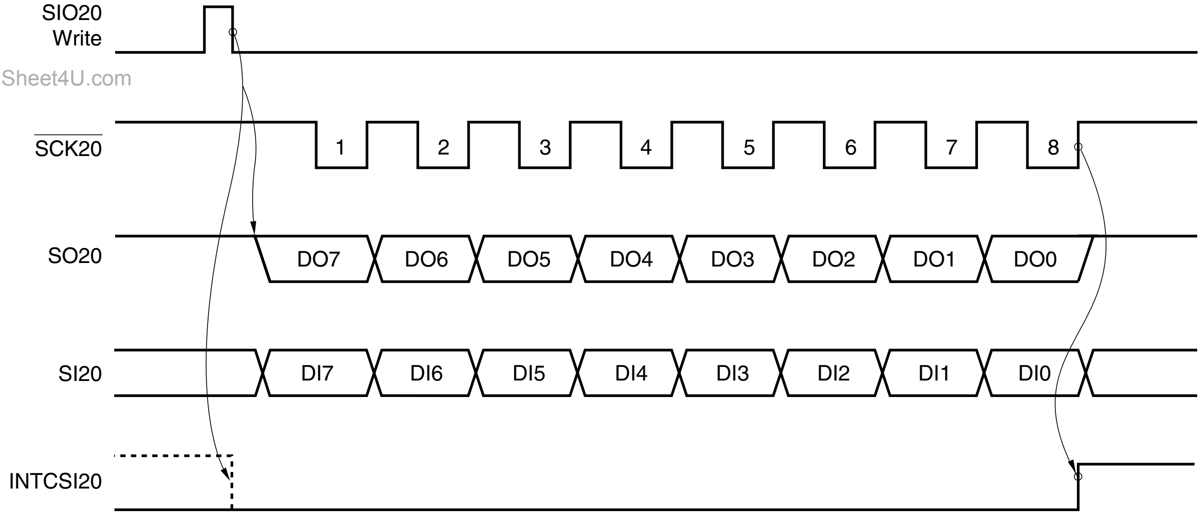
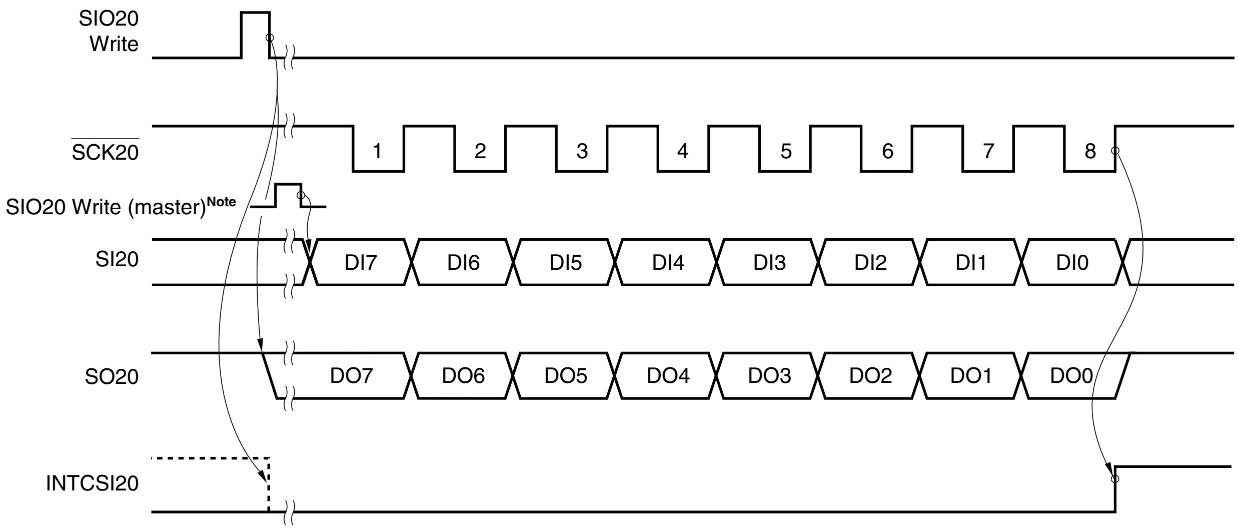

SIO20 Write
SCK20
1
2
3
4
5
6
7
8
SO20
DO7
DO6
DO5
DO4
DO3
DO2
DO1
DO0
SI20
DI7
DI6
DI5
DI4
DI3
DI2
DI1
DI0
INTCSI20

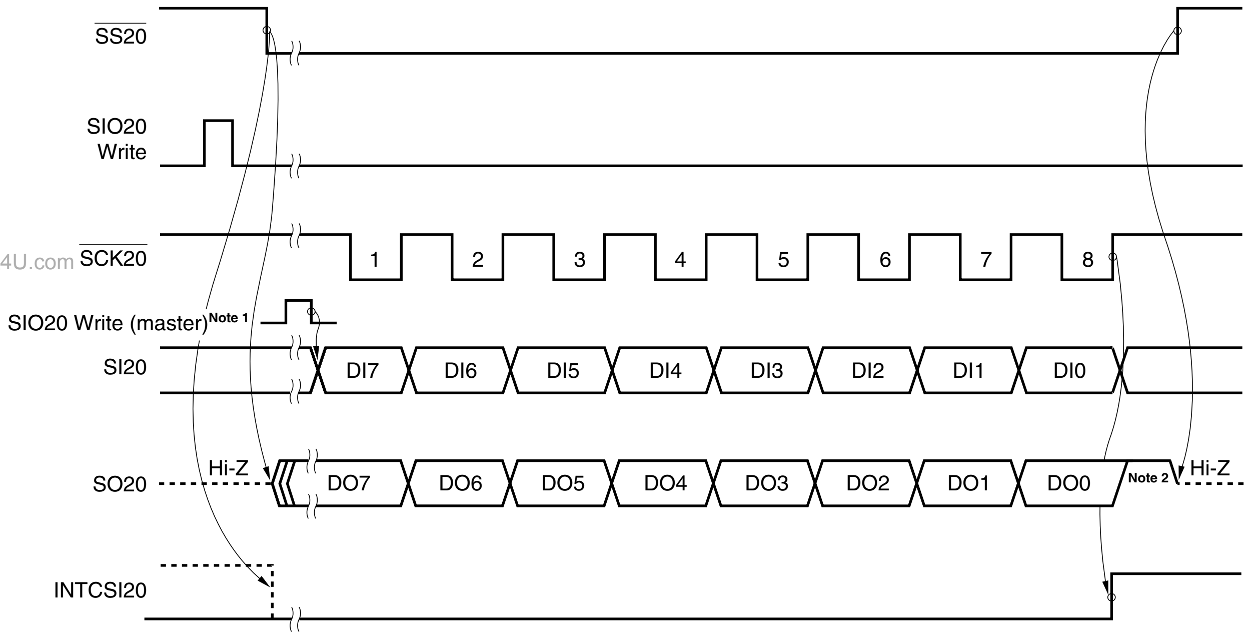
Figure 9-11. 3-Wire Serial I/O Mode Timing (5/7)

(viii) Slave operation (when DAP20 = 1, CKP20 = 0, SSE20 = 0)


SIO20 Write
SCK20
1
2
3
4
5
6
7
8
SIO20 Write (master)Note
SI20
DI7
DI6
DI5
DI4
DI3
DI2
DI1
DI0
SO20
DO7
DO6
DO5
DO4
DO3
DO2
DO1
DO0
INTCSI20

Note The data of SI20 is loaded at the first falling edge of $\overline{\text{SCK20}}$. Make sure that the master outputs the first bit before the first falling of $\overline{\text{SCK20}}$.

(ix) Slave operation (when DAP20 = 1, CKP20 = 0, SSE20 = 1)

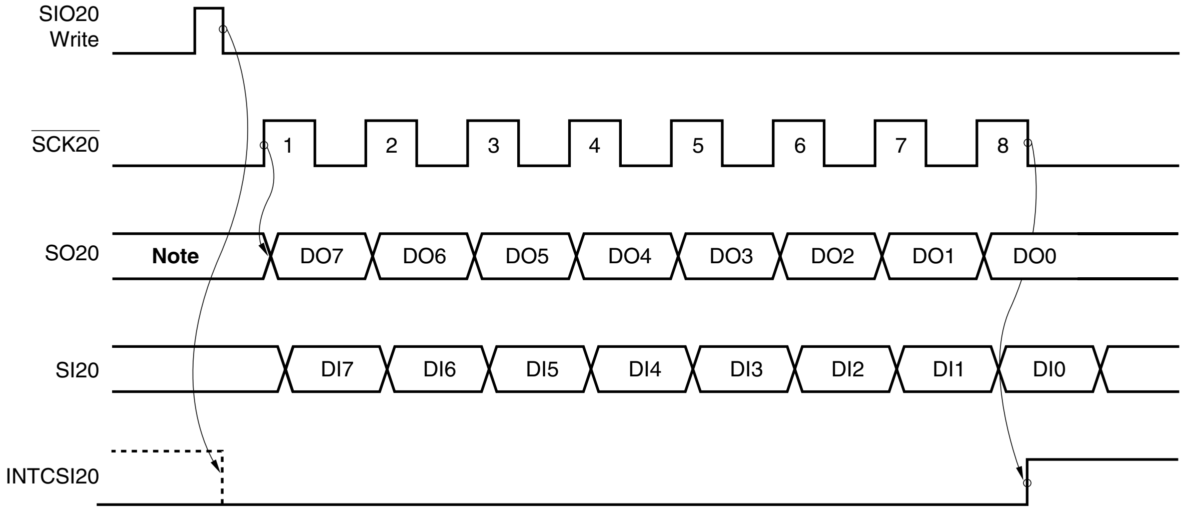

SS20
SIO20 Write
SCK20
1
2
3
4
5
6
7
8
SIO20 Write (master)Note 1
SI20
DI7
DI6
DI5
DI4
DI3
DI2
DI1
DI0
SO20
Hi-Z
DO7
DO6
DO5
DO4
DO3
DO2
DO1
DO0
Note 2
Hi-Z
INTCSI20

Notes 1. The data of SI20 is loaded at the first falling edge of $\overline{\text{SCK20}}$. Make sure that the master outputs the first bit before the first falling of $\overline{\text{SCK20}}$.

2. SO20 is high until $\overline{\text{SS20}}$ rises after completion of DO0 output. When $\overline{\text{SS20}}$ is high, SO20 is in a high-impedance state.

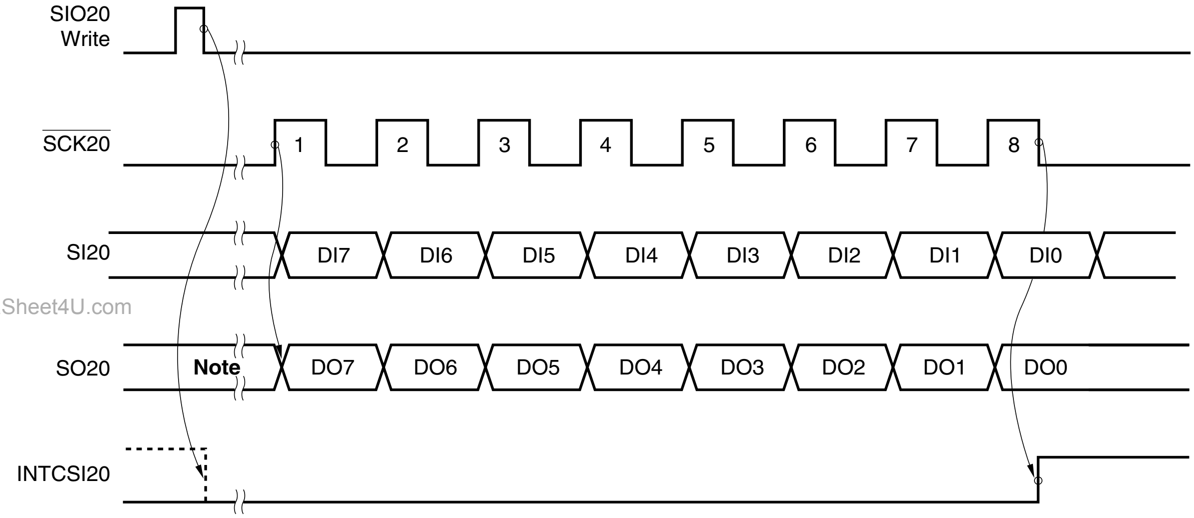
Figure 9-11. 3-Wire Serial I/O Mode Timing (6/7)

(x) Master operation (when DAP20 = 1, CKP20 = 1, SSE20 = 0)


SIO20 Write
$\overline{\text{SCK20}}$
1
2
3
4
5
6
7
8
SO20
Note
DO7
DO6
DO5
DO4
DO3
DO2
DO1
DO0
SI20
DI7
DI6
DI5
DI4
DI3
DI2
DI1
DI0
INTCSI20

Note The value of the last bit previously output is output.

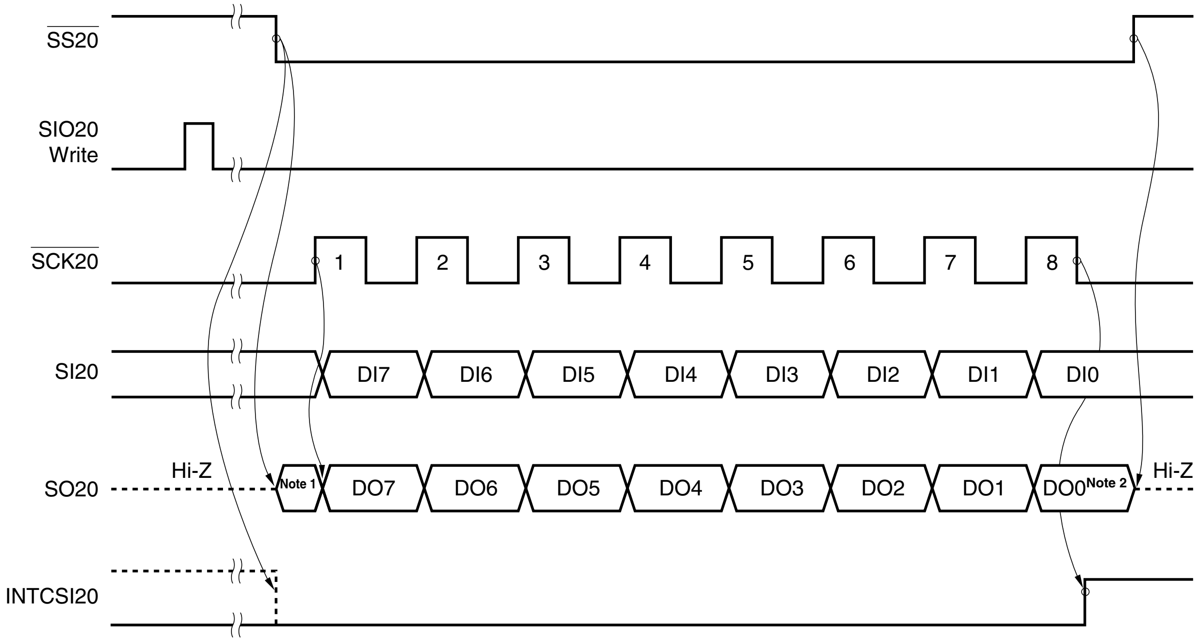
(xi) Slave operation (when DAP20 = 1, CKP20 = 1, SSE20 = 0)


SIO20 Write
$\overline{\text{SCK20}}$
1
2
3
4
5
6
7
8
SI20
DI7
DI6
DI5
DI4
DI3
DI2
DI1
DI0
SO20
Note
DO7
DO6
DO5
DO4
DO3
DO2
DO1
DO0
INTCSI20

Note The value of the last bit previously output is output.

Figure 9-11. 3-Wire Serial I/O Mode Timing (7/7)

(xii) Slave operation (when DAP20 = 1, CKP20 = 1, SSE20 = 1)


SS20
SIO20 Write
SCK20
1
2
3
4
5
6
7
8
SI20
DI7
DI6
DI5
DI4
DI3
DI2
DI1
DI0
Hi-Z
SO20
Note 1
DO7
DO6
DO5
DO4
DO3
DO2
DO1
DO0Note 2
Hi-Z
INTCSI20

Notes 1. The value of the last bit previously output is output.

2. DO0 is output until $\overline{SS20}$ rises. When $\overline{SS20}$ is high, SO20 is in a high-impedance state.

(3) Transfer start

Serial transfer is started by setting transfer data to the transmit shift register (TXS20/SIO20) when the following two conditions are satisfied.

- Serial operation mode register 20 (CSIM20) bit 7 (CSIE20) = 1
- Internal serial clock is stopped or $\overline{SCK20}$ is high after 8-bit serial transfer.

Caution If CSIE20 is set to "1" after data is written to TXS20/SIO20, transfer does not start.

The termination of 8-bit transfer stops the serial transfer automatically and generates the interrupt request signal (INTCSI20).

CHAPTER 10 INTERRUPT FUNCTIONS

10.1 Interrupt Function Types

The following two types of interrupt functions are used.

(1) Non-maskable interrupt

This interrupt is acknowledged unconditionally even if interrupts are disabled. It does not undergo interrupt priority control and is given top priority over all other interrupt requests.

A standby release signal is generated.

An interrupt from the watchdog timer is the only non-maskable interrupt source.

(2) Maskable interrupt

These interrupts undergo mask control. If two or more interrupts are simultaneously generated, each interrupt has a predetermined priority as shown in Table 10-1.

A standby release signal is generated.

There are three external sources and five internal sources of maskable interrupts.

10.2 Interrupt Sources and Configuration

There are a total of 9 non-maskable and maskable interrupt sources (see **Table 10-1**).

Table 10-1. Interrupt Sources

Interrupt Type	Priority[Note 1]	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Non-maskable interrupt	–	INTWDT	Watchdog timer overflow (when watchdog timer mode 1 is selected)	Internal	0004H	(A)
Maskable interrupt	0	INTWDT	Watchdog timer overflow (when interval timer mode is selected)			(B)
	1	INTP0	Pin input edge detection	External	0006H	(C)
	2	INTP1			0008H	
	3	INTP2			000AH	
	4	INTSR20	End of UART reception on serial interface 20	Internal	000CH	(B)
		INTCSI20	End of 3-wire SIO transfer reception on serial interface 20			
	5	INTST20	End of UART transmission on serial interface 20		000EH	
	6	INTTM80	Generation of match signal for 8-bit timer/event counter 80		0014H	
	7	INTTM90	Generation of match signal for 16-bit timer 90		0016H	

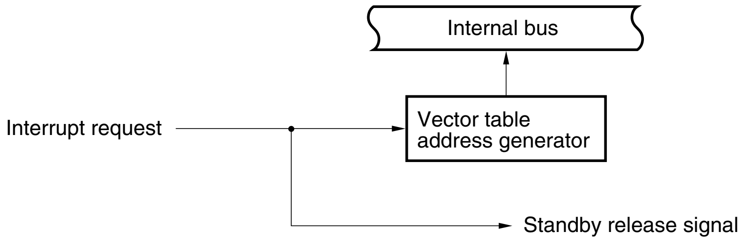
Notes 1. Priority is the priority order when several maskable interrupt requests are generated at the same time. 0 is the highest and 7 is the lowest.

2. Basic configuration types (A), (B), and (C) correspond to (A), (B), and (C) in Figure 10-1.

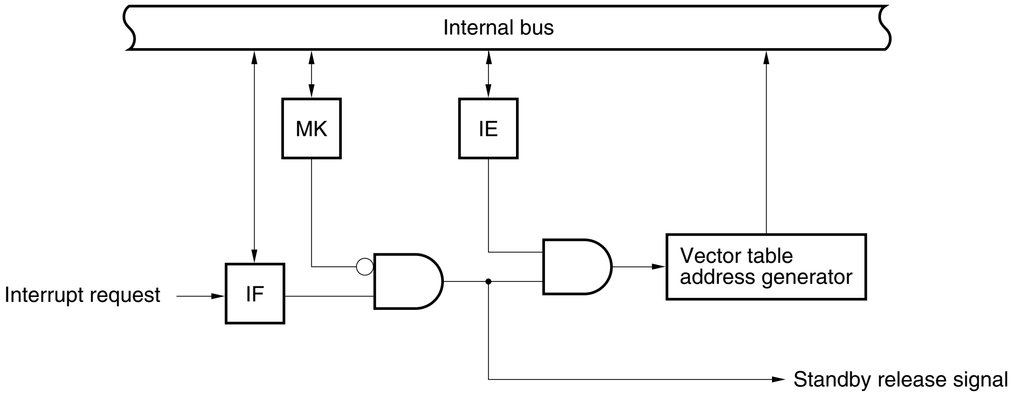
Remark There are two interrupt sources for the watchdog timer (INTWDT): non-maskable interrupts and maskable interrupts. Either one (but not both) should be selected for actual use.

Figure 10-1. Basic Configuration of Interrupt Function

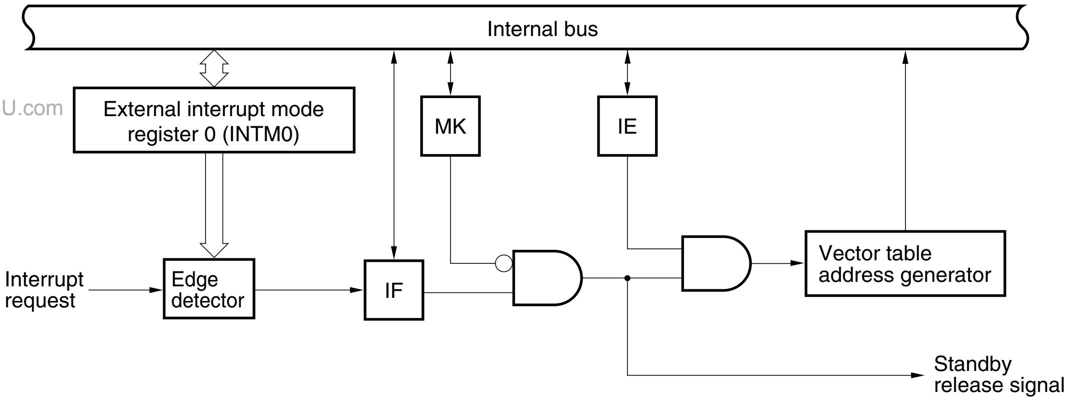
(A) Internal non-maskable interrupt


Internal bus
Interrupt request
Vector table address generator
Standby release signal

(B) Internal maskable interrupt


Internal bus
MK
IE
Interrupt request
IF
Vector table address generator
Standby release signal

(C) External maskable interrupt


Internal bus
External interrupt mode register 0 (INTM0)
MK
IE
Interrupt request
Edge detector
IF
Vector table address generator
Standby release signal

IF: Interrupt request flag

IE: Interrupt enable flag

MK: Interrupt mask flag

10.3 Interrupt Function Control Registers

The interrupt functions are controlled by the following four types of registers.

- Interrupt request flag registers 0 and 1 (IF0 and IF1)
- Interrupt mask flag registers 0 and 1 (MK0 and MK1)
- External interrupt mode register 0 (INTM0)
- Program status word (PSW)

Table 10-2 lists interrupt requests, the corresponding interrupt request flags, and interrupt mask flags.

Table 10-2. Interrupt Request Signals and Corresponding Flags

Interrupt Request Signal	Interrupt Request Flag	Interrupt Mask Flag
INTWDT	WDTIF	WDTMK
INTP0	PIF0	PMK0
INTP1	PIF1	PMK1
INTP2	PIF2	PMK2
INTSR20/INTCSI20	SRIF20	SRMK20
INTST20	STIF20	STMK20
INTTM80	TMIF80	TMMK80
INTTM90	TMIF90	TMMK90

(1) Interrupt request flag registers 0 and 1 (IF0 and IF1)

An interrupt request flag is set to 1 when the corresponding interrupt request is issued, or when the related instruction is executed. It is cleared to 0 when the interrupt request is acknowledged, when a $\overline{\text{RESET}}$ signal is input, or when a related instruction is executed.

IF0 and IF1 are set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears IF0 and IF1 to 00H.

Figure 10-2. Format of Interrupt Request Flag Register

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
IF0	0	0	STIF20	SRIF20	PIF2	PIF1	PIF0	WDTIF	FFE0H	00H	R/W
	7	6	5	4	3	2	<1>	<0>			
IF1	0	0	0	0	0	0	TMIF90	TMIF80	FFE1H	00H	R/W

××IF×	Interrupt request flag
0	No interrupt request signal has been issued.
1	An interrupt request signal has been issued; an interrupt request has been made.

Cautions 1. Bits 6 and 7 of IF0 and bits 2 to 7 of IF1 must all be set to 0.

2. The WDTIF flag can be read- and write-accessed only when the watchdog timer is being used as an interval timer. It must be cleared to 0 if the watchdog timer is used in watchdog timer mode 1 or 2.

3. When port 2 is being used as an output port, and its output level is changed, an interrupt request flag is set, because this port is also used as an external interrupt input. To use port 2 in output mode, therefore, the interrupt mask flag must be preset to 1.

(2) Interrupt mask flag registers 0 and 1 (MK0 and MK1)

The interrupt mask flags are used to enable and disable the corresponding maskable interrupts.

MK0 and MK1 are set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets MK0 and MK1 to FFH.

Figure 10-3. Format of Interrupt Mask Flag Register

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
MK0	1	1	STMK20	SRMK20	PMK2	PMK1	PMK0	WDTMK	FFE4H	FFH	R/W
	7	6	5	4	3	2	<1>	<0>			
MK1	1	1	1	1	1	1	TMMK90	TMMK80	FFE5H	FFH	R/W

××MK	Interrupt servicing control
0	Enable interrupt servicing.
1	Disable interrupt servicing.

Cautions 1. Bits 6 and 7 of MK0 and bits 2 to 7 of MK1 must all be set to 1.

2. When the watchdog timer is being used in watchdog timer mode 1 or 2, any attempt to read the WDTMK flag results in an undefined value being detected.

3. When port 2 is being used as an output port, and its output level is changed, an interrupt request flag is set, because this port is also used as an external interrupt input. To use port 2 in output mode, therefore, the interrupt mask flag must be preset to 1.

(3) External interrupt mode register 0 (INTM0)

INTM0 is used to specify a valid edge for INTP0 to INTP2.

INTM0 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears INTM0 to 00H.

Figure 10-4. Format of External Interrupt Mode Register 0

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
INTM0	ES21	ES20	ES11	ES10	ES01	ES00	0	0	FFECH	00H	R/W

ES21	ES20	INTP2 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

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. Bits 0 and 1 must both be set to 0.

2. Before setting INTM0, set the corresponding interrupt mask flag to 1 to disable interrupts.

To enable interrupts, clear to 0 the corresponding interrupt request flag, then the corresponding interrupt mask flag.

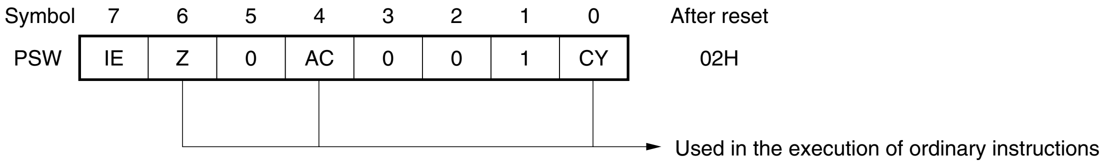
(4) Program status word (PSW)

The program status word is used to hold the instruction execution result and the current status of the interrupt requests. The IE flag, used to enable and disable maskable interrupts, is mapped to the PSW.

The PSW can be read- and write-accessed in 8-bit units, as well as using bit manipulation instructions and dedicated instructions (EI and DI). When a vector interrupt is acknowledged, the PSW is automatically saved to a stack, and the IE flag is reset to 0.

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

Figure 10-5. Program Status Word Configuration


Symbol
7
6
5
4
3
2
1
0
After reset
PSW
IE
Z
0
AC
0
0
1
CY
02H
Used in the execution of ordinary instructions

IE	Whether to enable/disable interrupt acknowledgment
0	Disabled
1	Enabled

10.4 Interrupt Processing Operation

10.4.1 Non-maskable interrupt request acknowledgment operation

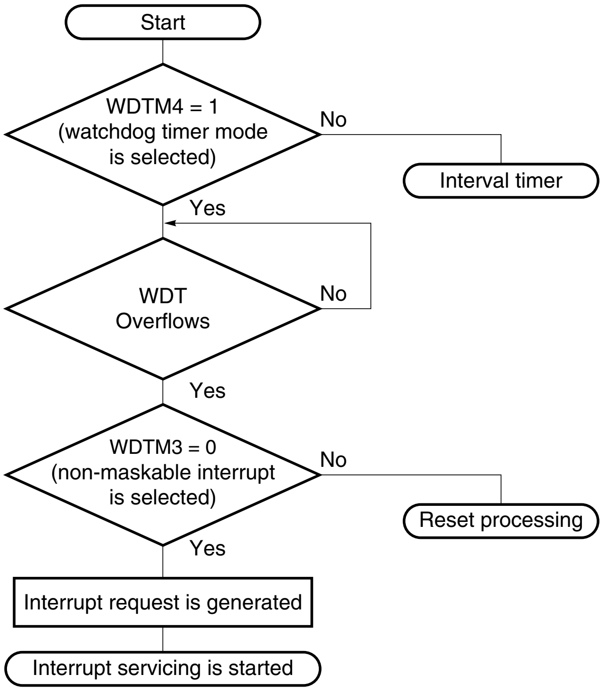
The non-maskable interrupt request is unconditionally acknowledged even when interrupts are disabled. It is not subject to interrupt priority control and takes precedence over all other interrupts.

When the 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 10-6 shows the flowchart from non-maskable interrupt request generation to acknowledgment. Figure 10-7 shows the timing of non-maskable interrupt request acknowledgment. Figure 10-8 shows the acknowledgment operation if multiple non-maskable interrupts are generated.

Caution During a non-maskable interrupt service program execution, do not input another non-maskable interrupt request; if it is input, the service program will be interrupted and the new interrupt request will be acknowledged.

Figure 10-6. Flowchart from Non-Maskable Interrupt Request Generation to Acknowledgment


Start
WDTM4 = 1
(watchdog timer mode
is selected)
No
Interval timer
Yes
WDT
Overflows
No
Yes
WDTM3 = 0
(non-maskable interrupt
is selected)
No
Reset processing
Yes
Interrupt request is generated
Interrupt servicing is started

WDTM: Watchdog timer mode register
WDT: Watchdog timer

Figure 10-7. Timing of Non-Maskable Interrupt Request Acknowledgment

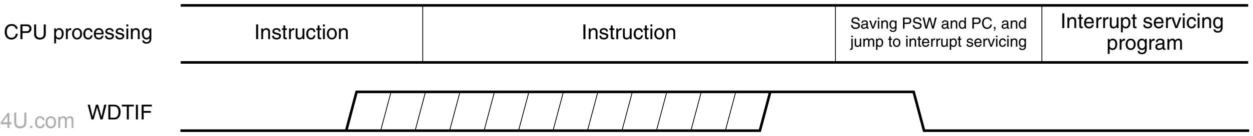
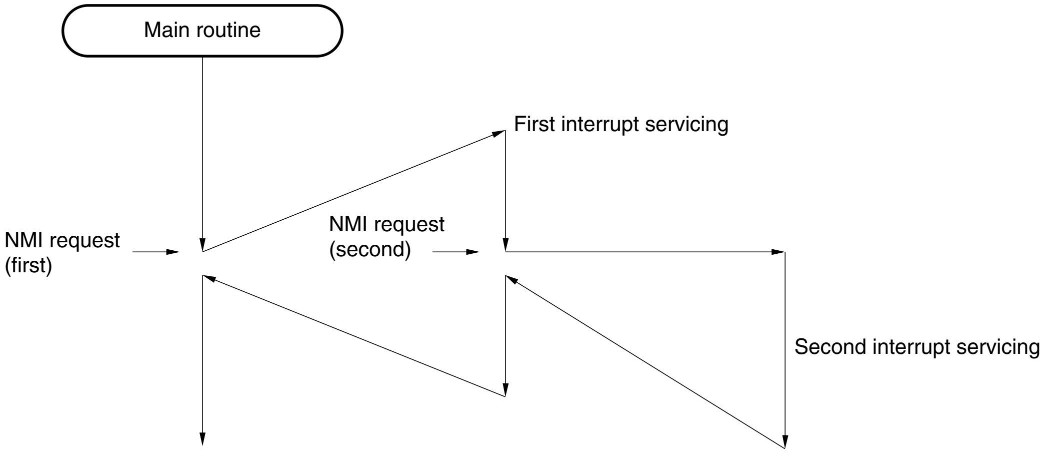

CPU processing
Instruction
Instruction
Saving PSW and PC, and
jump to interrupt servicing
Interrupt servicing
program
WDTIF

Figure 10-8. Acknowledgment of Non-Maskable Interrupt Request


Main routine
First interrupt servicing
NMI request
(first)
NMI request
(second)
Second interrupt servicing

10.4.2 Maskable interrupt request acknowledgment 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 10-3.

See Figures 10-10 and 10-11 for the interrupt request acknowledgment timing.

Table 10-3. 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 BT and BF instruction.

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.

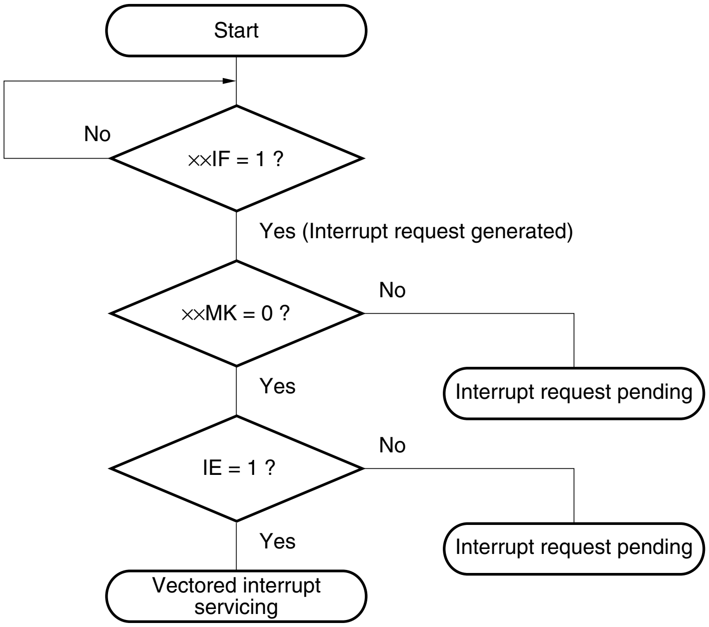
A pending interrupt is acknowledged when a status in which it can be acknowledged is set.

Figure 10-9 shows the algorithm of interrupt requests acknowledgment.

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.

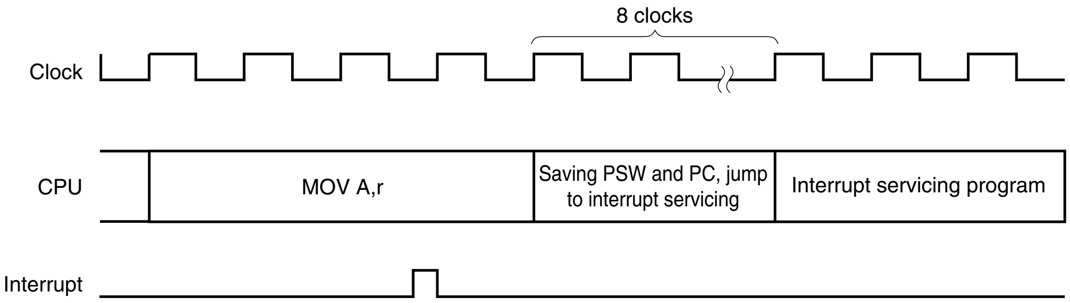
To return from interrupt servicing, use the RETI instruction.

Figure 10-9. Interrupt Request Acknowledgment Processing Algorithm


Start
No
××IF = 1 ?
Yes (Interrupt request generated)
××MK = 0 ?
No
Interrupt request pending
Yes
IE = 1 ?
No
Interrupt request pending
Yes
Vectored interrupt servicing

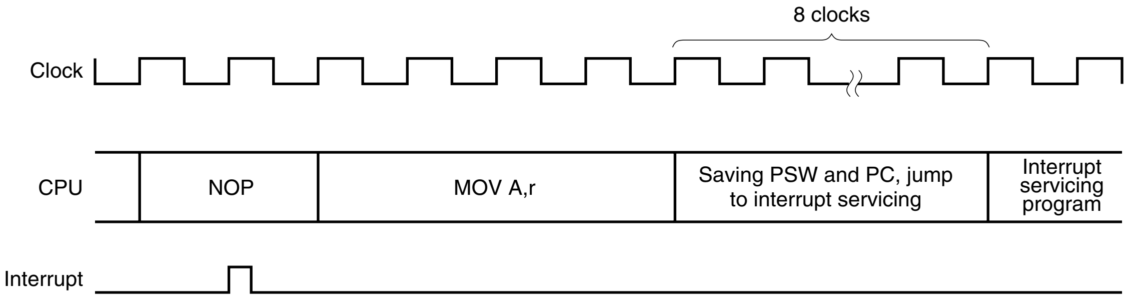
××IF:	Interrupt request flag
××MK:	Interrupt mask flag
IE:	Flag to control maskable interrupt request acknowledgment (1 = enable, 0 = disable)

Figure 10-10. Interrupt Request Acknowledgment Timing (Example of MOV A,r)


8 clocks
Clock
CPU
MOV A,r
Saving PSW and PC, jump to interrupt servicing
Interrupt servicing program
Interrupt

If an interrupt request flag (××IF) is set before an instruction clock n (n = 4 to 10) under execution becomes n – 1, the interrupt is acknowledged after the instruction under execution is complete. Figure 10-10 shows an example of the interrupt request acknowledgment 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 acknowledgment processing is performed after the MOV A,r instruction is executed.

Figure 10-11. Interrupt Request Acknowledgment Timing (When Interrupt Request Flag Is Set at Last Clock During Instruction Execution)


8 clocks
Clock
CPU
NOP
MOV A,r
Saving PSW and PC, jump to interrupt servicing
Interrupt servicing program
Interrupt

If an interrupt request flag (××IF) is set at the last clock of the instruction, the interrupt acknowledgment processing starts after the next instruction is executed. Figure 10-11 shows an example of the interrupt acknowledgment 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 acknowledgment processing is performed.

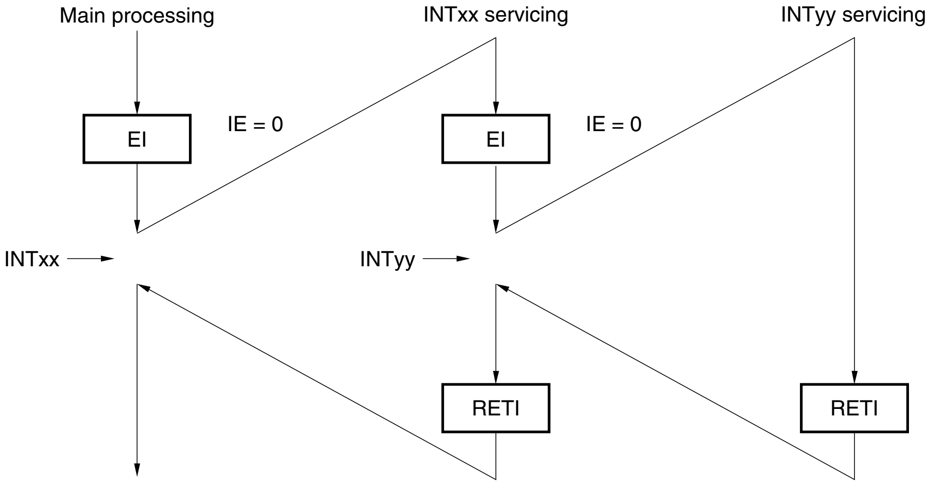
Caution Interrupt requests are held pending while interrupt request flag register 0 or 1 (IF0 or IF1) or interrupt mask flag register 0 or 1 (MK0 or MK1) is being accessed.

10.4.3 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 10-1**).

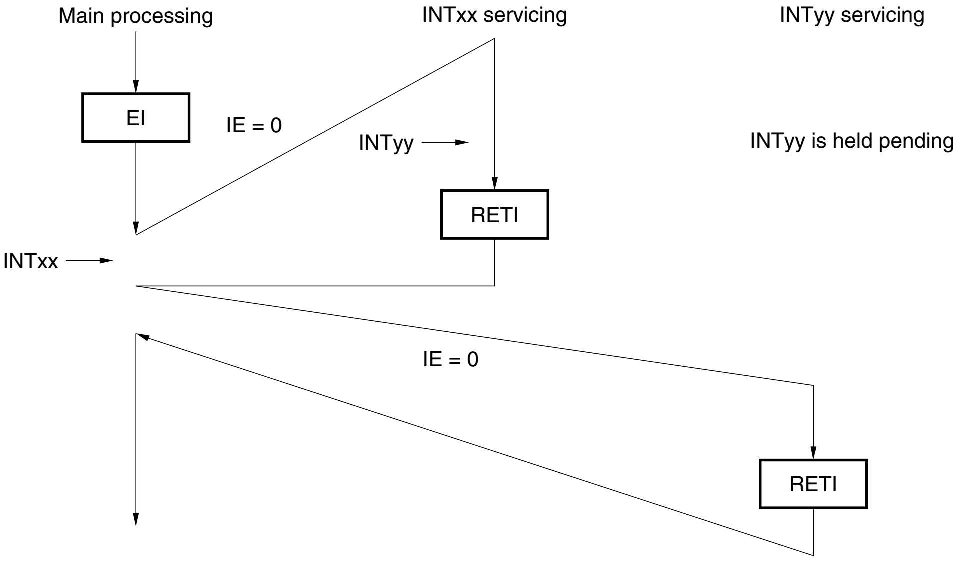
Figure 10-12. Example of Multiple Interrupts

Example 1. A multiple interrupt is acknowledged


Main processing
INTxx servicing
INTyy servicing
EI
IE = 0
EI
IE = 0
INTxx
INTyy
RETI
RETI

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 acknowledgment enable state is set.

Example 2. Multiple interrupts are not generated because interrupts are not enabled


Main processing
INTxx servicing
INTyy servicing
EI
IE = 0
INTyy
INTyy is held pending
RETI
INTxx
IE = 0
RETI

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 reserved and acknowledged after the INTxx servicing is performed.

IE = 0: Interrupt request acknowledgment disabled

10.4.4 Interrupt request hold

Some instructions may hold the acknowledgment of an instruction request pending 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 hold instructions).

- Manipulation instruction for interrupt request flag registers 0 and 1 (IF0 and IF1)
- Manipulation instruction for interrupt mask flag registers 0 and 1 (MK0 and MK1)

CHAPTER 11 STANDBY FUNCTION

11.1 Standby Function and Configuration

11.1.1 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 power 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. The STOP mode stops the main system clock oscillator and stops the entire system. The power consumption of the CPU can be substantially reduced in this mode.

The low voltage (V_{DD} = 1.8 V max.) of the data memory can be retained. Therefore, this mode is useful for retaining the contents of the data memory at an extremely low power consumption.

STOP mode can be released by an interrupt request, so that this mode can be used for intermittent operation. 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.

11.1.2 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 with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets OSTS to 04H. However, the oscillation stabilization time after $\overline{\text{RESET}}$ input is $2^{15}/f_X$, instead of $2^{17}/f_X$.

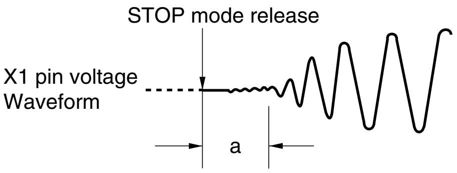
★ **Figure 11-1. Format of Oscillation Stabilization Time Selection Register**

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		
				At f_X = 10.0 MHz operation[Note]	At f_X = 5.0 MHz operation
0	0	0	$2^{12}/f_X$	410 μs	819 μs
0	1	0	$2^{15}/f_X$	3.28 ms	6.55 ms
1	0	0	$2^{17}/f_X$	13.1 ms	26.2 ms
Other than above			Setting prohibited		

Note Expanded-specification products only.

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 whether STOP mode is released by $\overline{\text{RESET}}$ input or by interrupt generation.



Remark f_X: System clock oscillation frequency

11.2 Operation of Standby Function

11.2.1 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 11-1. Operation Statuses in HALT Mode

Item	HALT Mode Operation Status
System clock generator	System clock oscillation enabled Clock supply to CPU stopped
CPU	Operation disabled
Port (output latch)	Remains in the state existing before the selection of HALT mode
16-bit timer 90	Operation enabled
8-bit timer/event counter 80	Operation enabled
Watchdog timer	Operation enabled
Serial interface 20	Operation enabled
External interrupt	Operation enabled[Note]

Note Maskable interrupt that is not masked

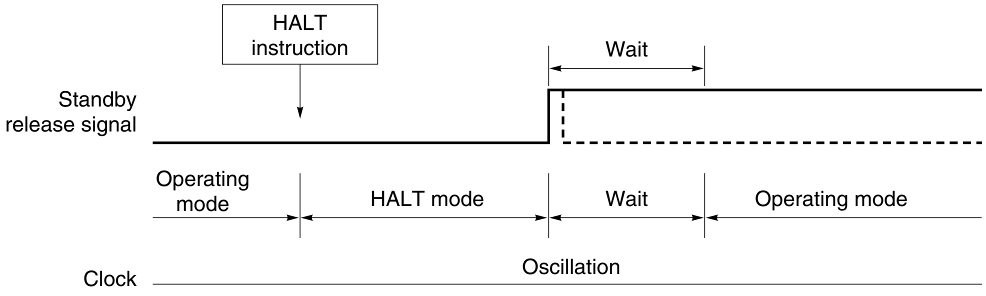
(2) Releasing HALT mode

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 acknowledgment is enabled, vectored interrupt servicing is performed. If interrupt acknowledgment is disabled, the instruction at the next address is executed.

Figure 11-2. 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

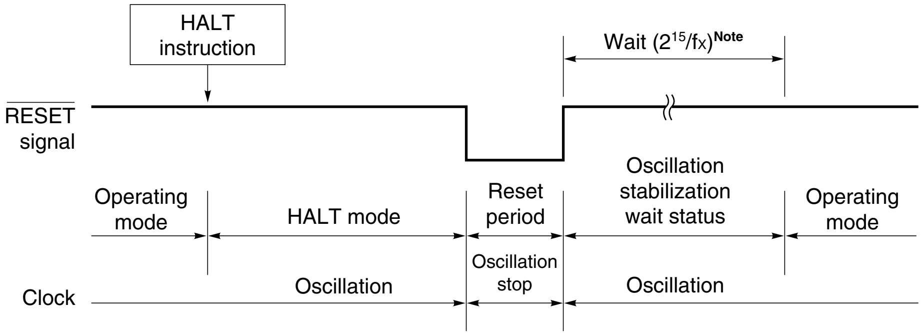
(b) Releasing by non-maskable interrupt request

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 11-3. Releasing HALT Mode by $\overline{\text{RESET}}$ Input**


HALT instruction
Wait (2^{15}/fx)Note
$\overline{\text{RESET}}$ signal
Operating mode
HALT mode
Reset period
Oscillation stabilization wait status
Operating mode
Clock
Oscillation
Oscillation stop
Oscillation

Note 3.28 ms (at fx = 10.0 MHz operation), 6.55 ms (at fx = 5.0 MHz operation)

Remark fx: System clock oscillation frequency

Table 11-2. 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

11.2.2 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. When STOP mode is set, therefore, HALT mode is set immediately after the STOP instruction has been executed, the wait time set by the oscillation stabilization time selection register (OSTS) elapses, and then the operation mode is set.

The operation statuses in STOP mode are shown in the following table.

Table 11-3. Operation Statuses in STOP Mode

Item	STOP Mode Operation Status
Clock generator	System clock oscillation stopped
CPU	Operation stopped
Port (output latch)	Remains in the state existing before STOP mode was set
16-bit timer 90	Operation stopped
8-bit timer/event counter 80	Operation enabled only when TI80 is selected for count clock
Watchdog timer	Operation stopped
Serial interface 20	Operation enabled only when external clock is input to serial clock
External interrupt	Operation enabled[Note]

Note Maskable interrupt that is not masked

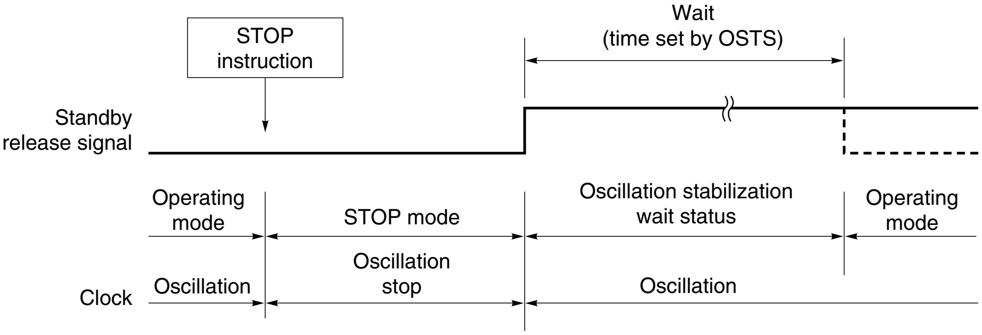
(2) Releasing STOP mode

STOP mode can be released by the following two sources.

(a) Releasing by unmasked interrupt request

STOP mode can be released by an unmasked interrupt request. In this case, vectored interrupt servicing is performed if interrupt acknowledgment is enabled after the oscillation stabilization time has elapsed. If interrupt acknowledgment is disabled, the instruction at the next address is executed.

Figure 11-4. Releasing STOP Mode by Interrupt

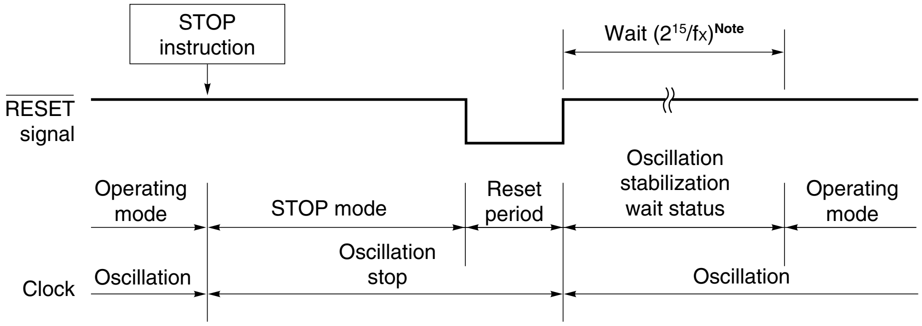

STOP instruction
Wait
(time set by OSTS)
Standby release signal
Operating mode
STOP mode
Oscillation stabilization wait status
Operating mode
Clock
Oscillation
Oscillation stop
Oscillation

Remark The broken lines indicate the case where the interrupt request that has released standby mode is acknowledged.

(b) 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 11-5. Releasing STOP Mode by $\overline{\text{RESET}}$ Input**


STOP instruction
Wait (2^{15}/fx)Note
$\overline{\text{RESET}}$ signal
Operating mode
STOP mode
Reset period
Oscillation stabilization wait status
Operating mode
Clock
Oscillation
Oscillation stop
Oscillation

Note 3.28 ms (at fx = 10.0 MHz operation), 6.55 ms (at fx = 5.0 MHz operation)

Remark fx: System clock oscillation frequency

Table 11-4. 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.
$\overline{\text{RESET}}$ input	–	–	Reset processing

×: Don't care

CHAPTER 12 RESET FUNCTION

The following two operations are available to generate reset signals.

(1) External reset input by $\overline{\text{RESET}}$ signal input
(2) Internal reset by watchdog timer program loop time detection

External 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 or the watchdog timer overflows, a reset is applied and each hardware is set to the status shown in Table 12-1. Each pin is high impedance during reset input or during the oscillation stabilization time just after reset clear.

When a high level is input to the $\overline{\text{RESET}}$ pin, the reset is cleared and program execution is started after the oscillation stabilization time has elapsed. The reset applied by the watchdog timer overflow is automatically cleared after reset, and program execution is started after the oscillation stabilization time has elapsed (see **Figures 12-2** through **12-4**).

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 cleared by reset, the STOP mode contents are held during reset input. However, the port pins become high impedance.

Figure 12-1. Block Diagram of Reset Function

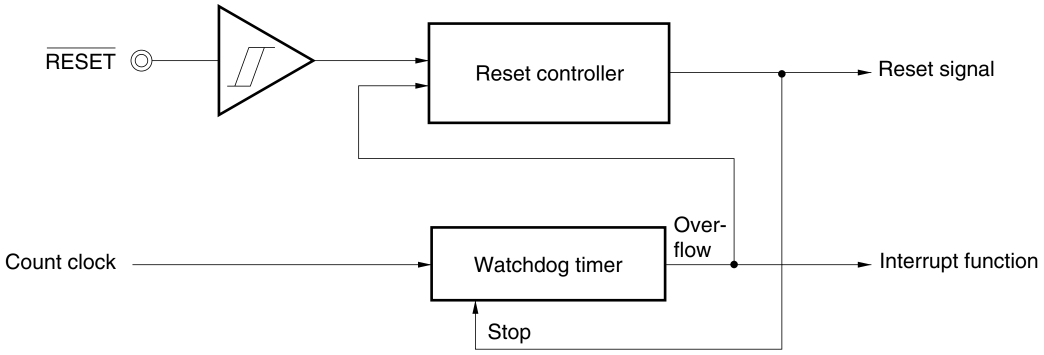


Figure 12-2. Reset Timing by $\overline{\text{RESET}}$ Input

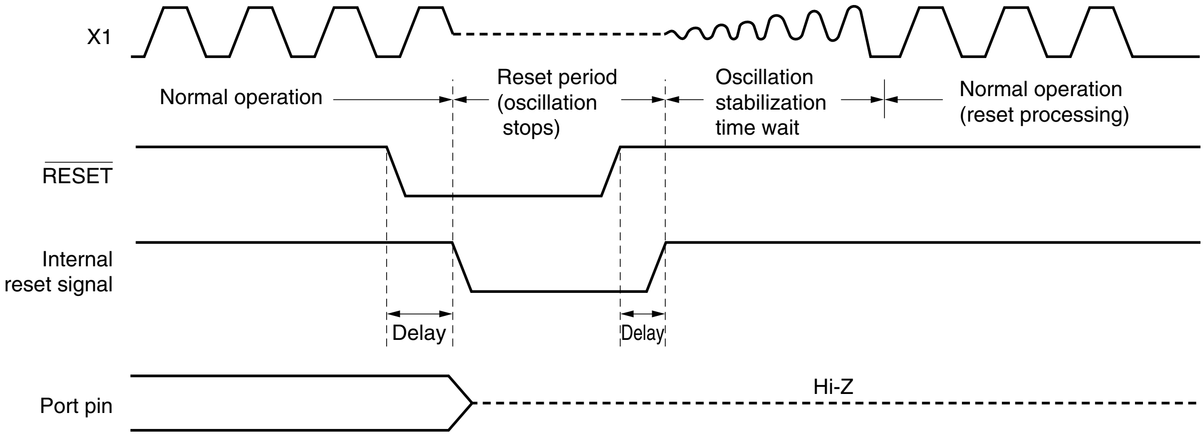

X1
Normal operation
Reset period (oscillation stops)
Oscillation stabilization time wait
Normal operation (reset processing)
RESET
Internal reset signal
Delay
Delay
Port pin
Hi-Z

Figure 12-3. Reset Timing by Watchdog Timer Overflow

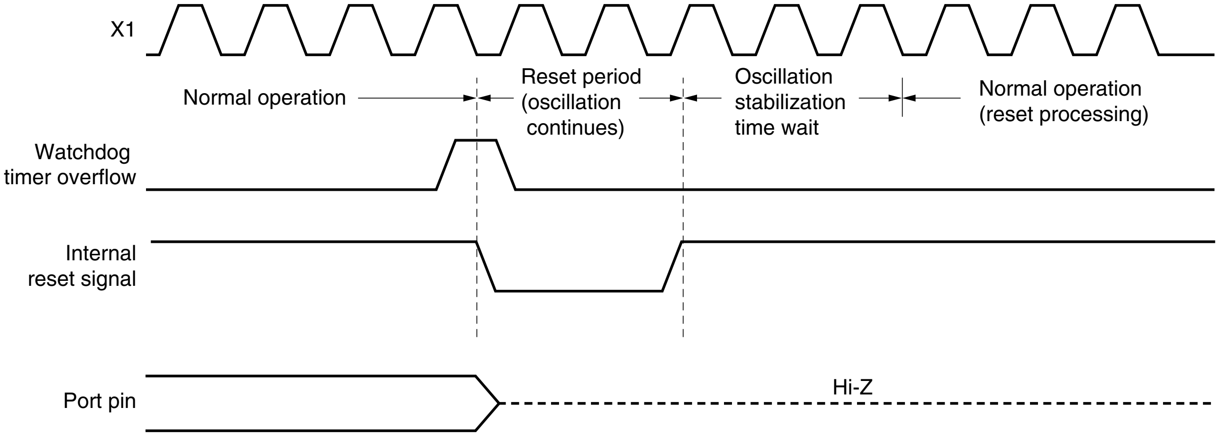

X1
Normal operation
Reset period (oscillation continues)
Oscillation stabilization time wait
Normal operation (reset processing)
Watchdog timer overflow
Internal reset signal
Port pin
Hi-Z

Figure 12-4. Reset Timing by $\overline{\text{RESET}}$ Input in STOP Mode

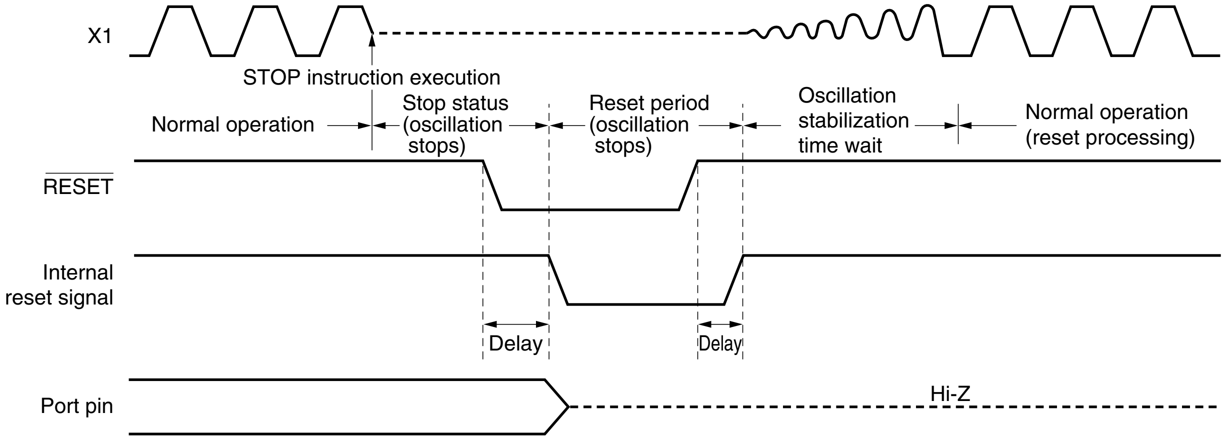

X1
STOP instruction execution
Normal operation
Stop status (oscillation stops)
Reset period (oscillation stops)
Oscillation stabilization time wait
Normal operation (reset processing)
RESET
Internal reset signal
Delay
Delay
Port pin
Hi-Z

Table 12-1. Status of Hardware After Reset

Hardware		Status After Reset
Program counter (PC)[Note 1]		Loaded with the contents of the reset vector table (0000H, 0001H)
Stack pointer (SP)		Undefined
Program status word (PSW)		02H
RAM	Data memory	Undefined[Note 2]
	General-purpose registers	Undefined[Note 2]
Ports (P0 to P3) (output latch)		00H
Port mode registers (PM0 to PM3)		FFH
Pull-up resistor option registers (PU0, PUB2)		00H
Processor clock control register (PCC)		02H
Oscillation stabilization time selection register (OSTS)		04H
16-bit timer	Timer counter (TM90)	0000H
	Compare register (CR90)	FFFFH
	Mode control register (TMC90)	00H
	Capture register (TCP90)	Undefined
	Buzzer output control register (BZC90)	00H
8-bit timer/event counter	Timer counter (TM80)	00H
	Compare register (CR80)	Undefined
	Mode control register (TMC80)	00H
Watchdog timer	Clock selection register (WDCS)	00H
	Mode register (WDTM)	00H
Serial interface	Serial operation mode register (CSIM20)	00H
	Asynchronous serial interface mode register (ASIM20)	00H
	Asynchronous serial interface status register (ASIS20)	00H
	Baud rate generator control register (BRGC20)	00H
	Transmit shift register (TXS20)	FFH
	Receive buffer register (RXB20)	Undefined
Interrupts	Request flag registers (IF0, IF1)	00H
	Mask flag registers (MK0, MK1)	FFH
	External interrupt mode register (INTM0)	00H

Notes 1. While a reset signal is being input, and during the oscillation stabilization period, the contents of the PC will be undefined, while the remainder of the hardware will be the same as after the reset.

2. In standby mode, the RAM enters the hold state after a reset.

★

CHAPTER 13 μPD78F9076

The μPD78F9076 replaces the internal ROM of the μPD789071, 789072, 789074, 789071(A), 789072(A), and 789074(A), with flash memory. The differences between the flash memory and the mask ROM versions are shown in Table 13-1.

Table 13-1. Differences Between Flash Memory and Mask ROM Versions

Item		Flash Memory Version	Mask ROM Version		
		μPD78F9076	μPD789071 μPD789071(A)	μPD789072 μPD789072(A)	μPD789074 μPD789074(A)
Internal memory	ROM structure	Flash memory	Mask ROM		
	ROM capacity	16 KB	2 KB	4 KB	8 KB
	High-speed RAM	256 bytes			
V_{PP} pin		Provided	Not provided		
Electrical characteristics		Refer to **CHAPTER 15 ELECTRICAL SPECIFICATIONS (EXPANDED-SPECIFICATION PRODUCTS)** and **CHAPTER 16 ELECTRICAL SPECIFICATIONS (CONVENTIONAL PRODUCTS)**.			

Caution There are differences in the noise immunity and noise radiation between flash memory and mask ROM versions. When pre-producing an application set with the flash memory version and the mass producing it with the mask ROM version, be sure to conduct sufficient evaluations on the commercial sample (CS), not engineering sample (ES), of the mask ROM version.

13.1 Flash Memory Characteristics

Flash memory programming is performed by connecting a dedicated flash programmer (Flashpro III (part no. FL-PR3, PG-FP3)/Flashpro IV (part no. FL-PR4, PG-FP4)) to the target system with the μPD78F9076 mounted on the target system (on-board). A flash memory program adapter (FA adapter), which is a target board used exclusively for programming, is also provided.

Remark FL-PR3, FL-PR4, and the program adapter are products made by Naito Densei Machida Mfg. Co., Ltd. (TEL +81-45-475-4191).

Programming using flash memory 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

13.1.1 Programming environment

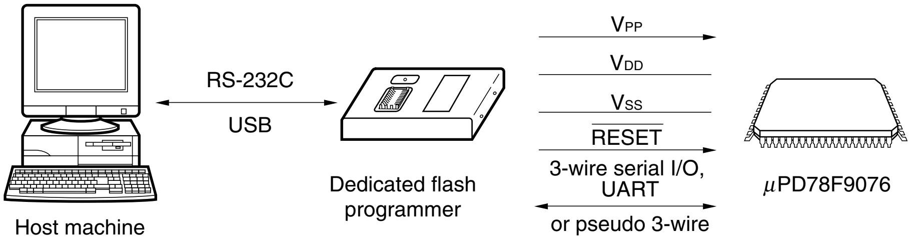
The following shows the environment required for μPD78F9076 flash memory programming.

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 13-1. Environment for Writing Program to Flash Memory



13.1.2 Communication mode

Use the communication mode shown in Table 13-2 to perform communication between the dedicated flash programmer and μPD78F9076.

Table 13-2. 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[Note 2]	1, 2, 4, 5 MHz[Notes 2, 3]	1 to 5 MHz[Note 2]	1.0	SI20/RxD20/P22 SO20/TxD20/P21 $\overline{\text{SCK20}}$/ASCK20/P20	0
UART	UART ch-0 (Async.)	4,800 to 76,800 bps[Notes 2, 4]	5 MHz[Note 5]	4.91 or 5 MHz[Note 2]	1.0	RxD20/SI20/P22 TxD20/SO20/P21	8
Pseudo 3-wire	Port A (Pseudo-3 wire)	100 Hz to 1 kHz	1, 2, 4, 5 MHz[Notes 2, 3]	1 to 5 MHz[Note 2]	1.0	P01 P02 P00	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. The possible setting range differs depending on the voltage. For details, refer to **CHAPTER 15 ELECTRICAL SPECIFICATIONS (EXPANDED-SPECIFICATION PRODUCTS)** and **CHAPTER 16 ELECTRICAL SPECIFICATIONS (CONVENTIONAL PRODUCTS)**.

3. 2 or 4 MHz only for Flashpro III

4. Because signal wave slew also affects UART communication, in addition to the baud rate error, thoroughly evaluate the slew.

5. Only for Flashpro IV. However, when using Flashpro III, be sure to select the clock of the resonator on the board. UART cannot be used with the clock supplied by Flashpro III.

Figure 13-2. Communication Mode Selection Format

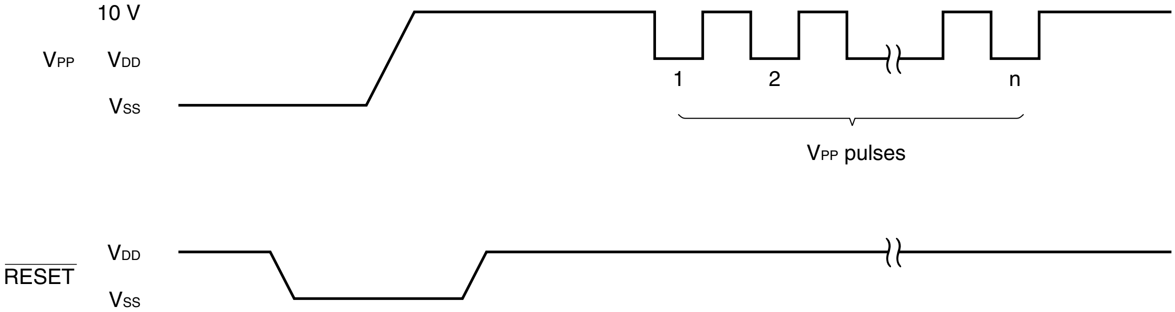
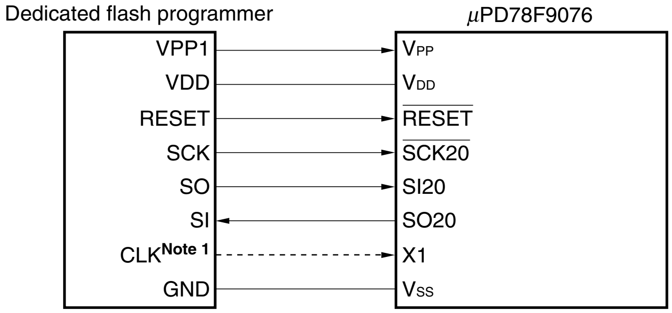
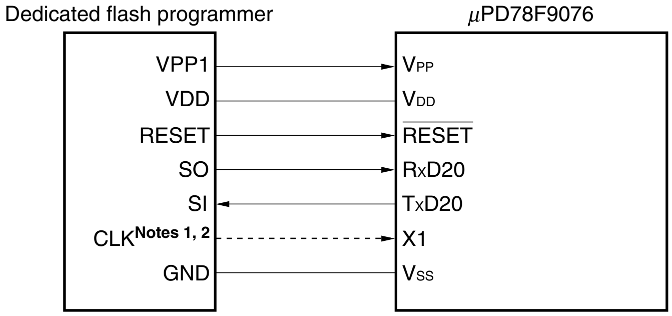


Figure 13-3. Example of Connection with Dedicated Flash Programmer

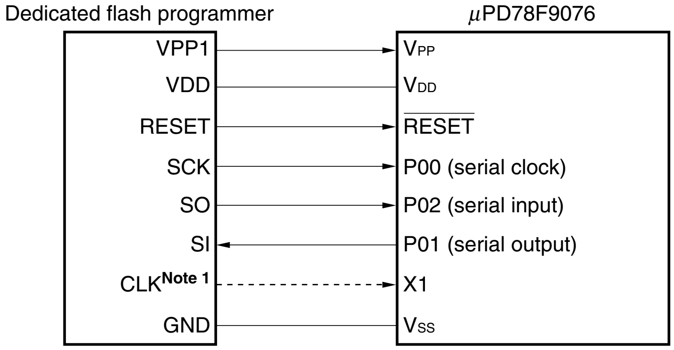
(a) 3-wire serial I/O


Dedicated flash programmer
μPD78F9076
VPP1
VDD
RESET
SCK
SO
SI
CLK^Note 1
GND
VPP
VDD
RESET
SCK20
SI20
SO20
X1
VSS

(b) UART


Dedicated flash programmer
μPD78F9076
VPP1
VDD
RESET
SO
SI
CLK^Notes 1, 2
GND
VPP
VDD
RESET
RxD20
TxD20
X1
VSS

(c) Pseudo 3-wire (when P0 is used)


Dedicated flash programmer
μPD78F9076
VPP1
VDD
RESET
SCK
SO
SI
CLK^Note 1
GND
VPP
VDD
RESET
P00 (serial clock)
P02 (serial input)
P01 (serial output)
X1
VSS

Notes 1. When supplying the system clock from a dedicated flash programmer, connect the CLK and X1 pins and cut off the resonator on the board. When using the clock oscillated by the on-board resonator, do not connect the CLK pin.

2. When using UART with Flashpro III, the clock of the resonator connected to the X1 pin must be used, so do not connect the CLK pin.

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 V_{DD} 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 μPD78F9076. For details, refer to the manual of Flashpro III/Flashpro IV.

Table 13-3. 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	SO20/TxD20/P01	◎	◎	◎
SO	Output	Transmit signal	SI20/RxD20/P02	◎	◎	◎
SCK	Output	Transfer clock	$\overline{\text{SCK20}}$/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.

13.1.3 On-board pin connections

When programming on the target system, provide a connector on the target system to connect to the dedicated flash programmer.

There may be cases in which an on-board function that switches from the normal operation mode to flash memory programming mode is required.

<V_{PP} pin>

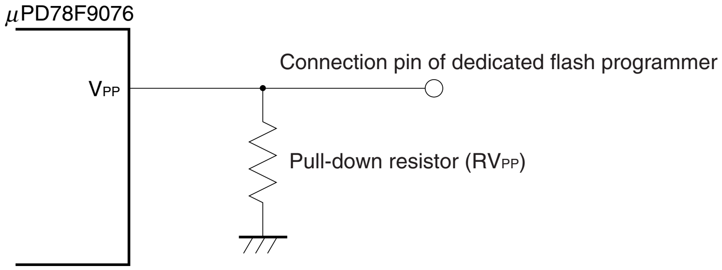
Input 0 V to the V_{PP} pin in the normal operation mode. A writing voltage of 10.0 V (TYP.) is supplied to the V_{PP} pin in the flash memory programming mode. Therefore, connect the V_{PP} pin as follows.

(1) Connect a pull-down resistor of RV_{PP} = 10 kΩ to the V_{PP} pin.

(2) Set the jumper on the board to switch the input of V_{PP} pin to the programmer side or directly to GND.

The following shows an example of V_{PP} pin connection.

Figure 13-4 V_{PP} Pin Connection Example


μPD78F9076
VPP
Connection pin of dedicated flash programmer
Pull-down resistor (RVPP)

<Serial interface pins>

The following shows the pins used by each serial interface.

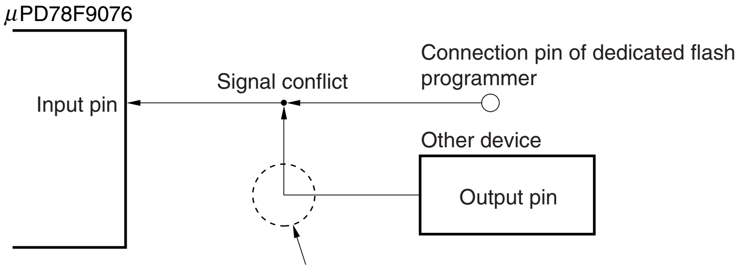
Serial Interface	Pins Used
3-wire serial I/O	SI20, SO20, $\overline{\text{SCK20}}$
UART	RxD20, TxD20
Pseudo 3-wire	P00, P01, P02

Note that signal conflict or malfunction of other devices may occur when an on-board serial interface pin that is connected to another device is connected to the dedicated flash programmer.

(1) Signal conflict

A signal conflict occurs if the dedicated flash programmer (output) is connected to a serial interface pin (input) connected to another device (output). To prevent this signal conflict, isolate the connection with the other device or put the other device in the output high impedance status.

Figure 13-5. Signal Conflict (Serial Interface Input Pin)

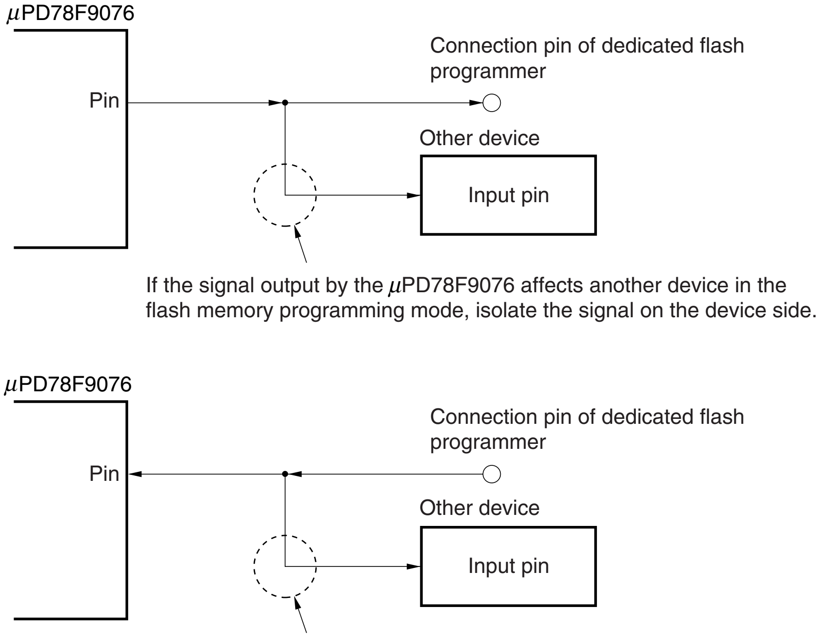

µPD78F9076
Connection pin of dedicated flash programmer
Signal conflict
Input pin
Other device
Output pin

In the flash memory programming mode, the signal output by another device and the signal sent by the dedicated flash programmer conflict. To prevent this, isolate the signal on the device side.

(2) Malfunction of another device

When the dedicated flash programmer (output or input) is connected to a serial interface pin (input or output) connected to another device (input), a signal may be output to the device, causing a malfunction. To prevent such malfunction, isolate the connection with other device or set so that the input signal to the device is ignored.

Figure 13-6. Malfunction of Another Device


µPD78F9076
Connection pin of dedicated flash programmer
Pin
Other device
Input pin
µPD78F9076
Connection pin of dedicated flash programmer
Pin
Other device
Input pin

If the signal output by the µPD78F9076 affects another device in the flash memory programming mode, isolate the signal on the device side.

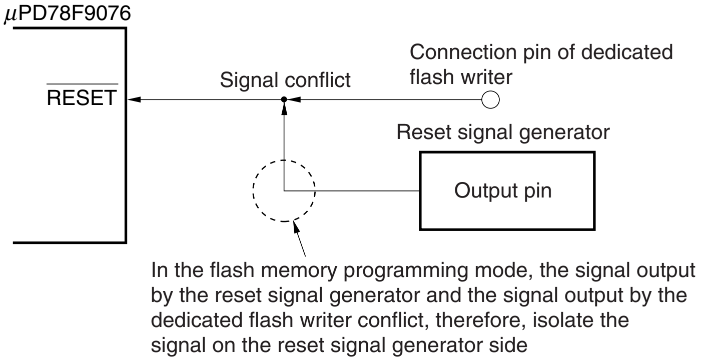
If the signal output by the dedicated flash programmer affects another device, isolate the signal on the device side.

<$\overline{\text{RESET}}$ pin>

When the reset signal of the dedicated flash programmer is connected to the $\overline{\text{RESET}}$ signal connected to the reset signal generator on the board, a signal conflict occurs. To prevent this signal conflict, isolate the connection with the reset signal generator.

If a reset signal is input from the user system in the flash memory programming mode, a normal programming operation will not be performed. Do not input signals other than reset signals from the dedicated flash programmer during this period.

Figure 13-7. Signal Conflict ($\overline{\text{RESET}}$ Pin)



In the flash memory programming mode, the signal output by the reset signal generator and the signal output by the dedicated flash writer conflict, therefore, isolate the signal on the reset signal generator side

<Port pins>

Shifting to the flash memory programming mode sets all the pins except those used for flash memory programming communication to the status immediately after reset.

Therefore, if the external device does not acknowledge an initial status such as the output high impedance status, connect the external device to V_{DD} or V_{SS} via a resistor.

<Oscillation pins>

When using an on-board clock, connection of X1 and X2 must conform to the methods in the normal operation mode.

When using the clock output of the flash programmer, directly connect it to the X1 pin with the on-board oscillator disconnected, and leave the X2 pin open.

<Power supply>

To use the power output of the flash programmer, connect the V_{DD} and V_{SS} pins to V_{DD} and GND of the flash programmer, respectively.

To use the on-board power supply, connection must conform to that in the normal operation mode. However, because the voltage is monitored by the flash programmer, therefore, V_{DD} of the flash programmer must be connected.

13.1.4 Connection of adapter for flash writing

The following figures show examples of the recommended connection when the adapter for flash writing is used.

Figure 13-8. Wiring Example for Flash Writing Adapter Using 3-Wire Serial I/O

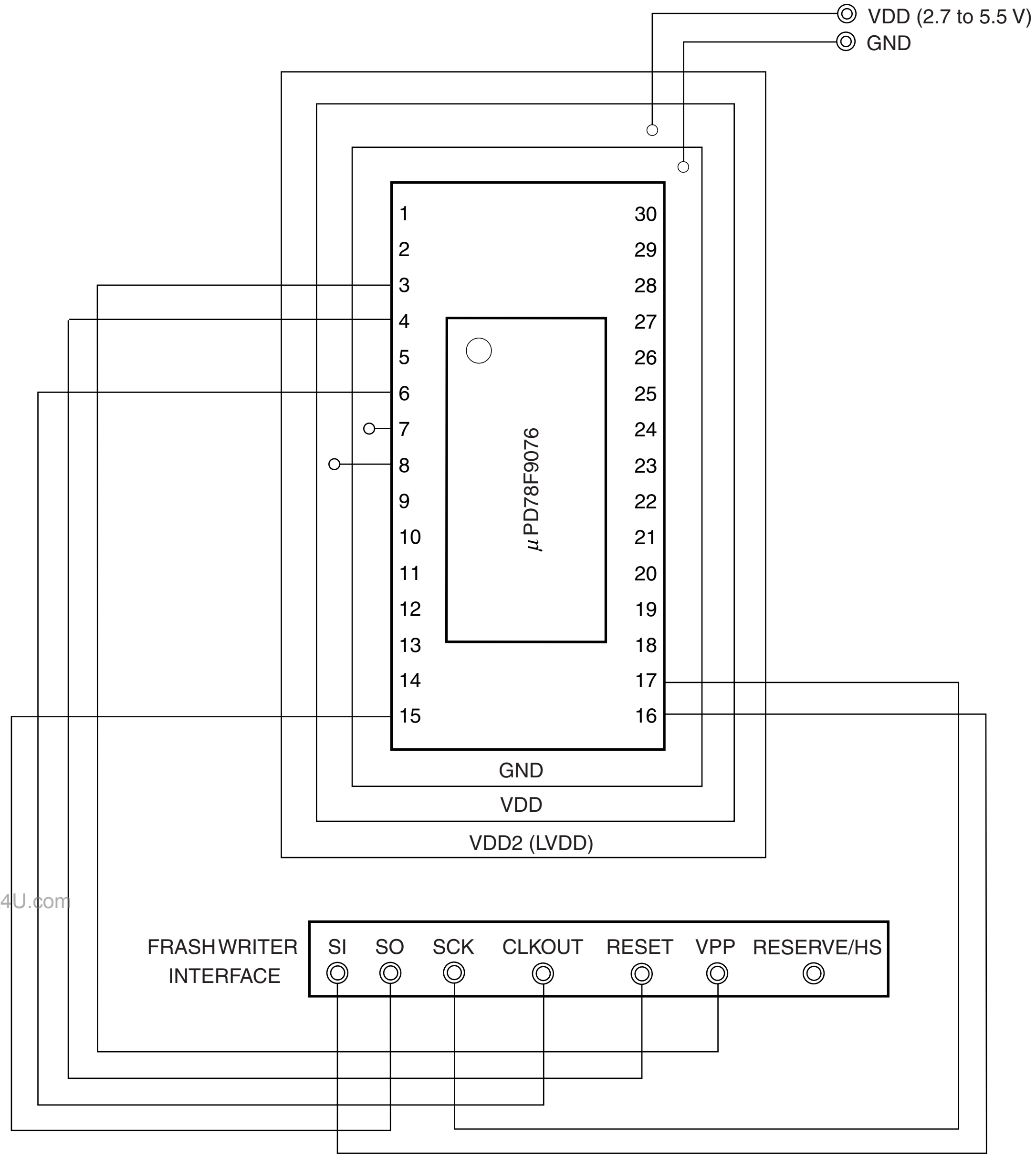

VDD (2.7 to 5.5 V)
GND
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
μPD78F9076
GND
VDD
VDD2 (LVDD)
FRASHWRITER
INTERFACE
SI
SO
SCK
CLKOUT
RESET
VPP
RESERVE/HS

Figure 13-9. Wiring Example for Flash Writing Adapter Using UART

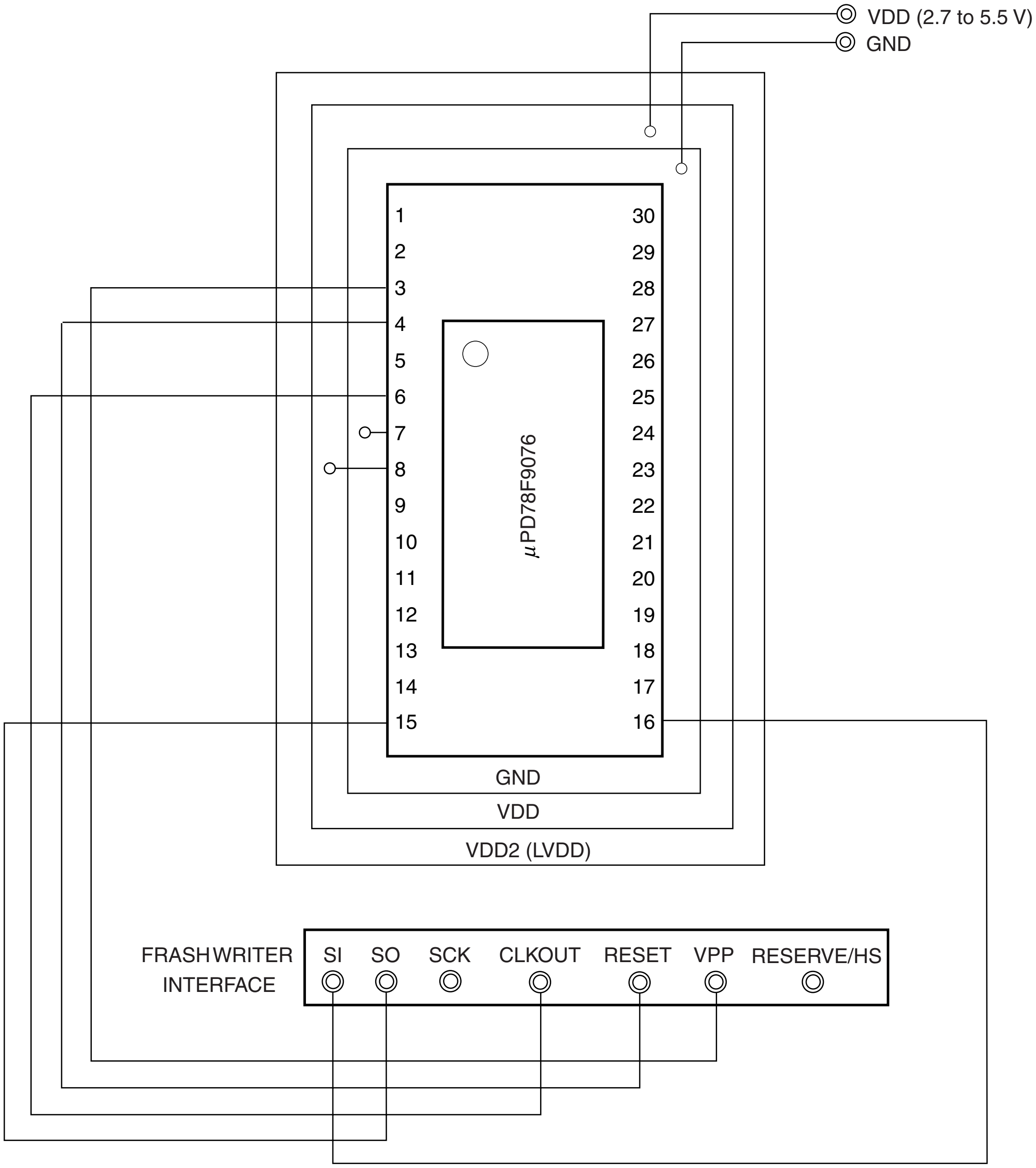
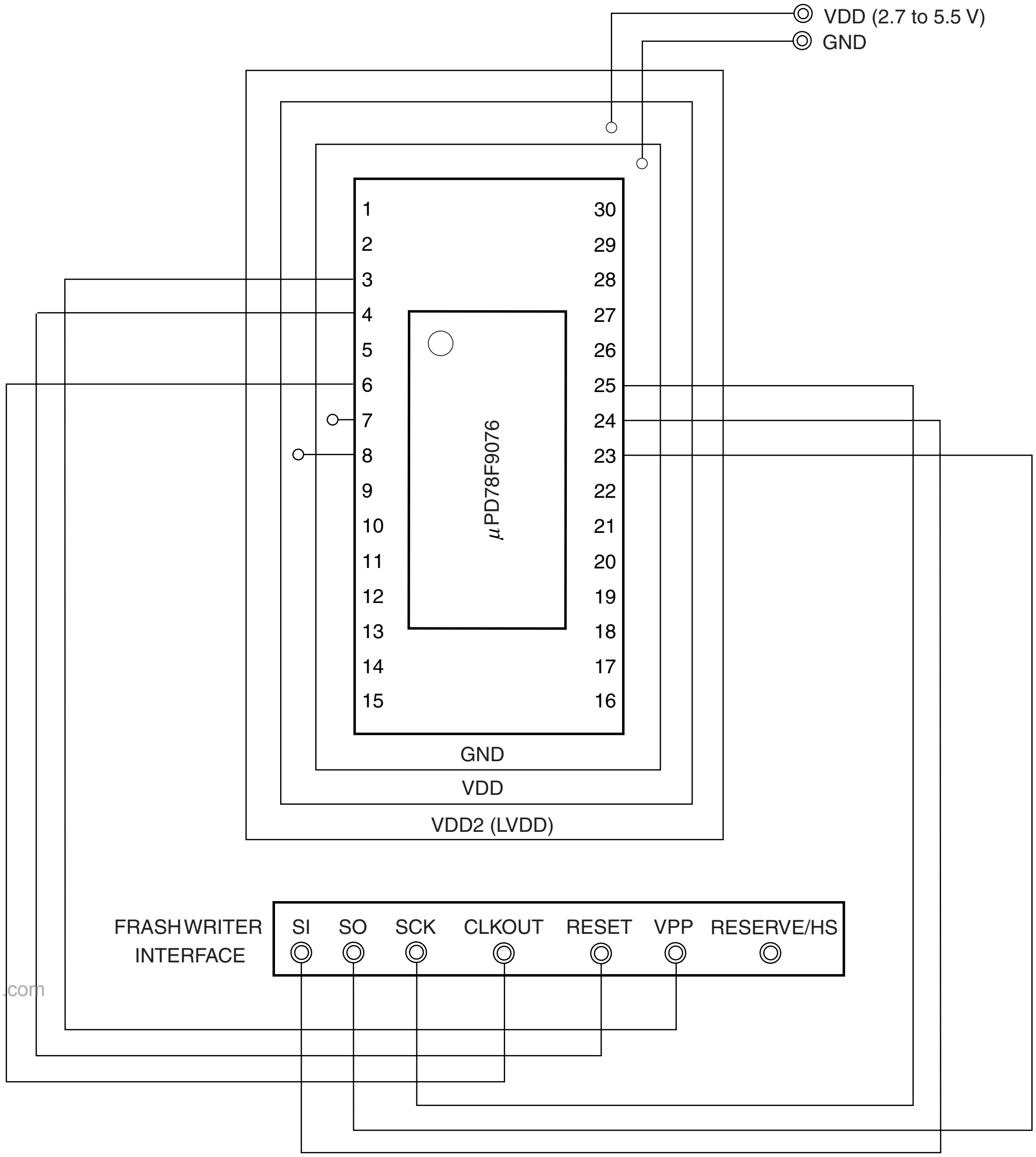

VDD (2.7 to 5.5 V)
GND
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
μPD78F9076
GND
VDD
VDD2 (LVDD)
FRASHWRITER
INTERFACE
SI
SO
SCK
CLKOUT
RESET
VPP
RESERVE/HS

Figure 13-10. Wiring Example for Flash Writing Adapter Using Pseudo 3-Wire


VDD (2.7 to 5.5 V)
GND
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
μPD78F9076
GND
VDD
VDD2 (LVDD)
FRASH WRITER INTERFACE
SI
SO
SCK
CLKOUT
RESET
VPP
RESERVE/HS

CHAPTER 14 INSTRUCTION SET OVERVIEW

This chapter lists the instruction set of the μPD789074 Subseries. For details of the operation and machine language (instruction code) of each instruction, refer to **78K/0S Series Instructions User's Manual (U11047E)**.

14.1 Operation

14.1.1 Operand identifiers and description methods

Operands are described in the "Operand" column of each instruction in accordance with the description method of the instruction operand identifier (refer to the assembler specifications for details). When there are two or more description methods, select one of them. Uppercase letters and the symbols #, !, $, and [ ] are keywords 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 14-1. Operand Identifiers and Description Methods

Identifier	Description Method
r	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7)
rp	AX (RP0), BC (RP1), DE (RP2), HL (RP3)
sfr	Special-function register symbol
saddr	FE20H to FF1FH Immediate data or labels
saddrp	FE20H to FF1FH Immediate data or labels (even addresses only)
addr16	0000H to FFFFH Immediate data or labels (only even addresses for 16-bit data transfer instructions)
addr5	0040H to 007FH Immediate data or labels (even addresses only)
word	16-bit immediate data or label
byte	8-bit immediate data or label
bit	3-bit immediate data or label

Remark For symbols of special function registers, see **Table 3-3 Special Function Registers**.

14.1.2 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
×H, ×L: Higher 8 bits and lower 8 bits of 16-bit register
∧: Logical product (AND)
∨: Logical sum (OR)
⊻: Exclusive logical sum (exclusive OR)
—: Inverted data
addr16: 16-bit immediate data or label
jdisp8: Signed 8-bit data (displacement value)

14.1.3 Description of "Flag" column

(Blank): Unchanged
0: Cleared to 0
1: Set to 1
×: Set/cleared according to the result
R: Previously saved value is stored

14.2 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	$(CY, A_7 \leftarrow A_0, A_{m-1} \leftarrow A_m) \times 1$			×
ROL	A, 1	1	2	$(CY, A_0 \leftarrow A_7, A_{m+1} \leftarrow A_m) \times 1$			×
RORC	A, 1	1	2	$(CY \leftarrow A_0, A_7 \leftarrow CY, A_{m-1} \leftarrow A_m) \times 1$			×
ROLC	A, 1	1	2	$(CY \leftarrow A_7, A_0 \leftarrow CY, A_{m+1} \leftarrow A_m) \times 1$			×
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) ← (PC + 3)$_H$, (SP − 2) ← (PC + 3)$_L$, PC ← addr16, SP ← SP − 2			
CALLT	[addr5]	1	8	(SP − 1) ← (PC + 1)$_H$, (SP − 2) ← (PC + 1)$_L$, PC$_H$ ← (00000000, addr5 + 1), PC$_L$ ← (00000000, addr5), SP ← SP − 2			
RET		1	6	PC$_H$ ← (SP + 1), PC$_L$ ← (SP), SP ← SP + 2			
RETI		1	8	PC$_H$ ← (SP + 1), PC$_L$ ← (SP), PSW ← (SP + 2), SP ← SP + 3, NMIS ← 0	R	R	R
PUSH	PSW	1	2	(SP − 1) ← PSW, SP ← SP − 1			
	rp	1	4	(SP − 1) ← rp$_H$, (SP − 2) ← rp$_L$, SP ← SP − 2			
POP	PSW	1	4	PSW ← (SP), SP ← SP + 1	R	R	R
	rp	1	6	rp$_H$ ← (SP + 1), rp$_L$ ← (SP), SP ← SP + 2			
MOVW	SP, AX	2	8	SP ← AX			
	AX, SP	2	6	AX ← SP			
BR	!addr16	3	6	PC ← addr16			
	$addr16	2	6	PC ← PC + 2 + jdisp8			
	AX	1	6	PC$_H$ ← A, PC$_L$ ← X			
BC	$saddr16	2	6	PC ← PC + 2 + jdisp8 if CY = 1			
BNC	$saddr16	2	6	PC ← PC + 2 + jdisp8 if CY = 0			
BZ	$saddr16	2	6	PC ← PC + 2 + jdisp8 if Z = 1			
BNZ	$saddr16	2	6	PC ← PC + 2 + jdisp8 if Z = 0			
BT	saddr.bit, $addr16	4	10	PC ← PC + 4 + jdisp8 if (saddr.bit) = 1			
	sfr.bit, $addr16	4	10	PC ← PC + 4 + jdisp8 if sfr.bit = 1			
	A.bit, $addr16	3	8	PC ← PC + 3 + jdisp8 if A.bit = 1			
	PSW.bit, $addr16	4	10	PC ← PC + 4 + jdisp8 if PSW.bit = 1			
BF	saddr.bit, $addr16	4	10	PC ← PC + 4 + jdisp8 if (saddr.bit) = 0			
	sfr.bit, $addr16	4	10	PC ← PC + 4 + jdisp8 if sfr.bit = 0			
	A.bit, $addr16	3	8	PC ← PC + 3 + jdisp8 if A.bit = 0			
	PSW.bit, $addr16	4	10	PC ← PC + 4 + jdisp8 if PSW.bit = 0			
DBNZ	B, $addr16	2	6	B ← B − 1, then PC ← PC + 2 + jdisp8 if B ≠ 0			
	C, $addr16	2	6	C ← C − 1, then PC ← PC + 2 + jdisp8 if C ≠ 0			
	saddr, $addr16	3	8	(saddr) ← (saddr) − 1, then PC ← PC + 3 + jdisp8 if (saddr) ≠ 0			
NOP		1	2	No Operation			
EI		3	6	IE ← 1 (Enable Interrupt)			
DI		3	6	IE ← 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).

14.3 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

2nd Operand / 1st 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

2nd Operand 1st 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

2nd Operand 1st 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

2nd Operand / 1st 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

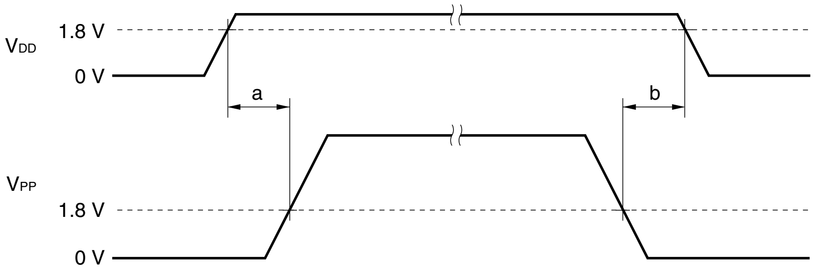
★ CHAPTER 15 ELECTRICAL SPECIFICATIONS (EXPANDED-SPECIFICATION PRODUCTS)

Absolute Maximum Ratings (T_A = 25°C)

Parameter	Symbol	Conditions		Ratings	Unit
Supply voltage	V_{DD}			–0.3 to +6.5	V
	V_{PP}	μPD78F9076 only, **Note**		–0.3 to +10.5	V
Input voltage	V_I			–0.3 to V_{DD} + 0.3	V
Output voltage	V_O			–0.3 to V_{DD} + 0.3	V
Output current, high	I_{OH}	Per pin	μPD78907x, 78F9076	–10	mA
		Total for all pins		–30	mA
		Per pin	μPD78907x(A)	–7	mA
		Total for all pins		–22	mA
Output current, low	I_{OL}	Per pin	μPD78907x, 78F9076	30	mA
		Total for all pins		160	mA
		Per pin	μPD78907x(A)	10	mA
		Total for all pins		120	mA
Operating ambient temperature	T_A	During normal operation		–40 to +85	°C
		During flash memory programming		10 to 40	°C
Storage temperature	T_{stg}	Mask ROM version		–65 to +150	°C
		μPD78F9076		–40 to +125	°C

Note Make sure that the following conditions of the V_{PP} voltage application timing are satisfied when the flash memory is written.

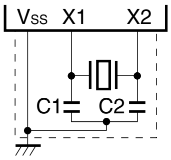
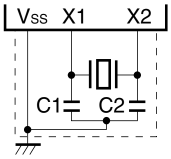
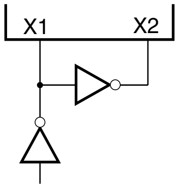
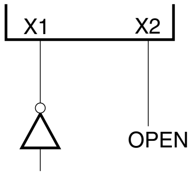
- **When supply voltage rises**
  V_{PP} must exceed V_{DD} 10 μs or more after V_{DD} has reached the lower-limit value (1.8 V) of the operating voltage range (see a in the figure below).
- **When supply voltage drops**
  V_{DD} must be lowered 10 μs or more after V_{PP} falls below the lower-limit value (1.8 V) of the operating voltage range of V_{DD} (see b in the figure below).



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} = 4.5 to 5.5 V	1.0		10.0	MHz
			V_{DD} = 3.0 to 5.5 V	1.0		6.0	MHz
			V_{DD} = 1.8 to 5.5 V	1.0		5.0	MHz
		Oscillation stabilization time[Note 2]	After V_{DD} reaches oscillation voltage range MIN.			4	ms
Crystal resonator		Oscillation frequency (f_X)[Note 1]	V_{DD} = 4.5 to 5.5 V	1.0		10.0	MHz
			V_{DD} = 3.0 to 5.5 V	1.0		6.0	MHz
			V_{DD} = 1.8 to 5.5 V	1.0		5.0	MHz
		Oscillation stabilization time[Note 2]	V_{DD} = 4.5 to 5.5 V			10	ms
			V_{DD} = 1.8 to 5.5 V			30	ms
External clock		X1 input frequency (f_X)[Note 1]	V_{DD} = 4.5 to 5.5 V	1.0		10.0	MHz
			V_{DD} = 3.0 to 5.5 V	1.0		6.0	MHz
			V_{DD} = 1.8 to 5.5 V	1.0		5.0	MHz
		X1 input high-/low-level width (t_{XH}, t_{XL})	V_{DD} = 4.5 to 5.5 V	45		500	ns
			V_{DD} = 3.0 to 5.5 V	75		500	ns
			V_{DD} = 1.8 to 5.5 V	85		500	ns
		X1 input frequency (f_X)[Note 1]	V_{DD} = 2.7 to 5.5 V	1.0		5.0	MHz
		X1 input high-/low-level width (t_{XH}, t_{XL})	V_{DD} = 2.7 to 5.5 V	85		500	ns

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.

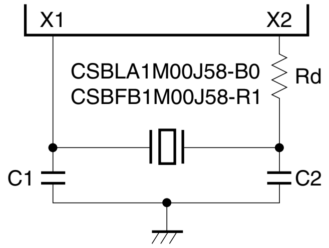
- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines.**
- **Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as V_{SS}.**
-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

Manufacturer	Part Number	Frequency (MHz)	Recommended Circuit Constant (pF)		Oscillation Voltage Range (V_{DD})		Remarks
			C1	C2	MIN.	MAX.	
Murata Mfg. Co., Ltd. (standard product)	CSBLA1M00J58-B0[Note]	1.0	100	100	1.9	5.5	Rd = 1.0 kΩ
	CSBFB1M00J58-R1[Note]						
	CSTCC2M00G56-R0	2.0	–	–	1.8		On-chip capacitor
	CSTCC2M00G56-R0						
	CSTCR4M00G53-R0	4.0					
	CSTLS4M00GG53-B0						
	CSTCR4M19G53-R0	4.194					
	CSTLS4M19GG53-B0						
	CSTCR4M91G53-R0	4.915					
	CSTLS4M91GG53-B0						
	CSTCR5M00G53-R0	5.0					
	CSTLS5M00GG53-B0						
	CSTCR6M00G53-R0	6.0			1.9		
	CSTLS6M00GG53-B0						
	CSTCE8M00G52-R0	8.0			1.8		
	CSTLS8M00G53-B0				2.0		
	CSTCE8M38G52-R0	8.388			1.8		
	CSTLS8M38G53-B0				2.0		
	CSTCE10M0G52-R0	10.0			1.8		
	CSTLS10M0G53-B0				2.0		
Murata Mfg. Co., Ltd. (low-voltage drive type)	CSTCR6M00G53093-R0	6.0	–	–	1.8	5.5	On-chip capacitor
	CSTLS6M00GG53093-B0						
	CSTLS8M00G53093-B0	8.0					
	CSTLS8M38G53093-B0	8.388					
	CSTLS10M0G53093-B0	10.0					

Note A limiting resistor (Rd = 1.0 kΩ) is required when CSBLA1M00J58-B0 or CSBFB1M00J58-R1 (1.0 MHz) manufactured by Murata Mfg. Co., Ltd. is used as the ceramic resonator (see the figure below). This is not necessary when using one of the other recommended resonators.

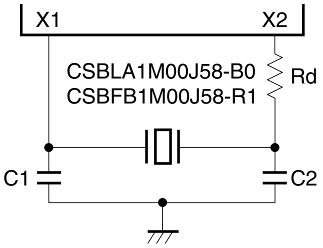


Caution The oscillator constant is a reference value based on evaluation in specific environments by the resonator manufacturer. If the oscillator characteristics need to be optimized in the actual application, request the resonator manufacturer for evaluation on the implementation circuit. Note that the oscillation voltage and oscillation frequency merely indicate the characteristics of the oscillator. Use the internal operation conditions of the μPD789071, 789072, and 789074 within the specifications of the DC and AC characteristics.

Ceramic resonator (T_A = –40 to +85°C) (μPD78F9076)

Manufacturer	Part Number	Frequency (MHz)	Recommended Circuit Constant (pF)		Oscillation Voltage Range (V_{DD})		Remarks
			C1	C2	MIN.	MAX.	
Murata Mfg. Co., Ltd.	CSBLA1M00J58-B0[Note]	1.0	100	100	2.1	5.5	Rd = 1.0 kΩ
	CSBFB1M00J58-R1[Note]						
	CSTCC2M00G56-R0	2.0	–	–	1.9		On-chip capacitor
	CSTLS2M00G56-B0						
	CSTCR4M00G53-R0	4.0					
	CSTLS4M00GG53-B0						
	CSTCR4M19G53-R0	4.194					
	CSTLS4M19GG53-B0						
	CSTCR4M91G53-R0	4.915			2.0		
	CSTLS4M91GG53-B0						
	CSTCR5M00G53-R0	5.0					
	CSTLS5M00GG53-B0						
	CSTCR6M00G53-R0	6.0			2.1		
	CSTLS6M00GG53-B0						
	CSTCE8M00G52-R0	8.0			1.8		
	CSTLS8M00G53-B0				2.2		
	CSTCE8M38G52-R0	8.388			1.8		
	CSTLS8M38G53-B0				2.2		
	CSTCE10M0G52-R0	10.0			2.0		
	CSTLS10M0G53-B0				2.1		
Murata Mfg. Co., Ltd. (low-voltage drive type)	CSTCR4M00G53093-R0	4.0	–	–	1.8	5.5	On-chip capacitor
	CSTLS4M00GG53093-B0						
	CSTCR4M19G53093-R0	4.194					
	CSTLS4M19GG53093-B0						
	CSTCR4M91G53093-R0	4.915					
	CSTLS4M91GG53093-B0						
	CSTCR5M00G53093-R0	5.0					
	CSTLS5M00GG53093-B0						
	CSTCR6M00G53093-R0	6.0					
	CSTLS6M00GG53093-B0						
	CSTLS8M00G53093-B0	8.0					
	CSTLS8M38G53093-B0	8.388					
	CSTLS10M0G53U-B0	10.0					

Note A limiting resistor (Rd = 1.0 kΩ) is required when CSBLA1M00J58-B0 or CSBFB1M00J58-R1 (1.0 MHz) manufactured by Murata Mfg. Co., Ltd. is used as the ceramic resonator (see the figure below). This is not necessary when using one of the other recommended resonators.



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 μPD78F9076 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)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high	I_{OH}	Per pin	μPD78907x, 78F9076			–1	mA
		Total for all pins				–15	mA
		Per pin	μPD78907x(A)			–1	mA
		Total for all pins				–11	mA
Output current, low	I_{OL}	Per pin	μPD78907x, 78F9076			10	mA
		Total for all pins				80	mA
		Per pin	μPD78907x(A)			3	mA
		Total for all pins				60	mA
Input voltage, high	V_{IH1}	P00 to P07, P10 to P15, P30, P31	V_{DD} = 2.7 to 5.5 V	0.7V_{DD}		V_{DD}	V
			V_{DD} = 1.8 to 5.5 V	0.9V_{DD}		V_{DD}	V
	V_{IH2}	$\overline{\text{RESET}}$, P20 to P27	V_{DD} = 2.7 to 5.5 V	0.8V_{DD}		V_{DD}	V
			V_{DD} = 1.8 to 5.5 V	0.9V_{DD}		V_{DD}	V
	V_{IH3}	X1, X2	V_{DD} = 4.5 to 5.5 V	V_{DD} – 0.5		V_{DD}	V
			V_{DD} = 1.8 to 5.5 V	V_{DD} – 0.1		V_{DD}	V
Input voltage, low	V_{IL1}	P00 to P07, P10 to P15, P30, P31	V_{DD} = 2.7 to 5.5 V	0		0.3V_{DD}	V
			V_{DD} = 1.8 to 5.5 V	0		0.1V_{DD}	V
	V_{IL2}	$\overline{\text{RESET}}$, P20 to P27	V_{DD} = 2.7 to 5.5 V	0		0.2V_{DD}	V
			V_{DD} = 1.8 to 5.5 V	0		0.1V_{DD}	V
	V_{IL3}	X1, X2	V_{DD} = 4.5 to 5.5 V	0		0.4	V
			V_{DD} = 1.8 to 5.5 V	0		0.1	V
Output voltage, high	V_{OH}	V_{DD} = 4.5 to 5.5 V, I_{OH} = –1 mA		V_{DD} – 1.0			V
		V_{DD} = 1.8 to 5.5 V, I_{OH} = –100 μA		V_{DD} – 0.5			V
Output voltage, low	V_{OL}	V_{DD} = 4.5 to 5.5 V, I_{OL} = 10 mA (μPD78907x, 78F9076)				1.0	V
		V_{DD} = 4.5 to 5.5 V, I_{OL} = 3 mA (μPD78907x(A))				1.0	V
		V_{DD} = 1.8 to 5.5 V, I_{OL} = 400 μA				0.5	V
Input leakage current, high	I_{LIH1}	V_{IN} = V_{DD}	Pins other than X1, X2			3	μA
	I_{LIH2}		X1, X2			20	μA
Input leakage current, low	I_{LIL1}	V_{IN} = 0 V	Pins other than X1, X2			–3	μA
	I_{LIL2}		X1, X2			–20	μA
Output leakage current, high	I_{LOH}	V_{OUT} = V_{DD}				3	μA
Output leakage current, low	I_{LOL}	V_{OUT} = 0 V				–3	μA
Software pull-up resistor	R_1	V_{IN} = 0 V		50	100	200	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)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Power supply current$^{\text{Note 1}}$ (mask ROM version)	I_{DD1}	10.0 MHz crystal oscillation operating mode	V_{DD} = 5.0 V ±10%$^{\text{Note 2}}$		3.0	7.5	mA
		6.0 MHz crystal oscillation operating mode	V_{DD} = 5.0 V ±10%$^{\text{Note 2}}$		1.7	3.9	mA
		5.0 MHz crystal oscillation operating mode (C1 = C2 = 22 pF)	V_{DD} = 5.0 V ±10%$^{\text{Note 2}}$		1.3	2.6	mA
			V_{DD} = 3.0 V ±10%$^{\text{Note 3}}$		0.26	0.5	mA
			V_{DD} = 2.0 V ±10%$^{\text{Note 3}}$		0.14	0.30	mA
	I_{DD2}	10.0 MHz crystal oscillation HALT mode	V_{DD} = 5.0 V ±10%$^{\text{Note 2}}$		1.0	2.0	mA
		6.0 MHz crystal oscillation HALT mode	V_{DD} = 5.0 V ±10%$^{\text{Note 2}}$		0.8	1.6	mA
		5.0 MHz crystal oscillation HALT mode (C1 = C2 = 22 pF)	V_{DD} = 5.0 V ±10%$^{\text{Note 2}}$		0.5	1.0	mA
			V_{DD} = 3.0 V ±10%$^{\text{Note 3}}$		0.17	0.35	mA
			V_{DD} = 2.0 V ±10%$^{\text{Note 3}}$		0.08	0.2	mA
	I_{DD3}	STOP mode	V_{DD} = 5.0 V ±10%		0.1	10	μA
			V_{DD} = 3.0 V ±10%		0.05	5.0	μA
			V_{DD} = 2.0 V ±10%		0.05	3.0	μA
Power supply current$^{\text{Note 1}}$ (μPD78F9076)	I_{DD1}	10.0 MHz crystal oscillation operating mode	V_{DD} = 5.0 V ±10%$^{\text{Note 2}}$		9.0	18.0	mA
		6.0 MHz crystal oscillation operating mode	V_{DD} = 5.0 V ±10%$^{\text{Note 2}}$		5.0	10.0	mA
		5.0 MHz crystal oscillation operating mode (C1 = C2 = 22 pF)	V_{DD} = 5.0 V ±10%$^{\text{Note 2}}$		4.0	8.0	mA
			V_{DD} = 3.0 V ±10%$^{\text{Note 3}}$		1.0	2.5	mA
			V_{DD} = 2.0 V ±10%$^{\text{Note 3}}$		0.8	2.0	mA
	I_{DD2}	10.0 MHz crystal oscillation HALT mode	V_{DD} = 5.0 V ±10%$^{\text{Note 2}}$		1.2	6.0	mA
		6.0 MHz crystal oscillation HALT mode	V_{DD} = 5.0 V ±10%$^{\text{Note 2}}$		0.9	2.8	mA
		5.0 MHz crystal oscillation HALT mode (C1 = C2 = 22 pF)	V_{DD} = 5.0 V ±10%$^{\text{Note 2}}$		0.8	2.5	mA
			V_{DD} = 3.0 V ±10%$^{\text{Note 3}}$		0.5	2.0	mA
			V_{DD} = 2.0 V ±10%$^{\text{Note 3}}$		0.3	1.0	mA
	I_{DD3}	STOP mode	V_{DD} = 5.0 V ±10%		0.1	10	μA
			V_{DD} = 3.0 V ±10%		0.05	5.0	μA
			V_{DD} = 2.0 V ±10%		0.05	3.0	μA

Notes 1. The port current (including the current flowing through the on-chip pull-up resistors) is not included.

2. High-speed mode operation (when processor clock control register (PCC) is set to 00H)

3. Low-speed mode operation (when PCC is set to 02H)

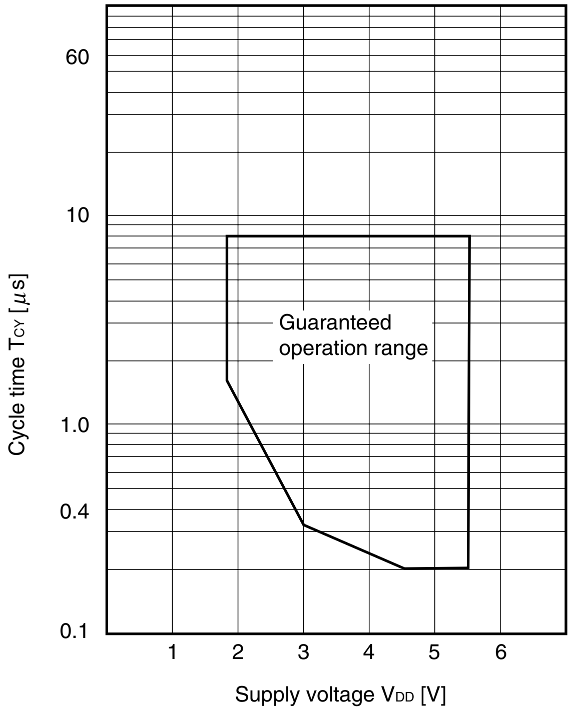
Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

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
Cycle time (Minimum instruction execution time)	T_{CY}	V_{DD} = 4.5 to 5.5 V	0.2		8	μs
		V_{DD} = 3.0 to 5.5 V	0.33		8	μs
		V_{DD} = 2.7 to 5.5 V	0.4		8	μs
		V_{DD} = 1.8 to 5.5 V	1.6		8	μs
TI80 input frequency	f_{TI}	V_{DD} = 2.7 to 5.5 V	0		4	MHz
		V_{DD} = 1.8 to 5.5 V	0		275	kHz
TI80 input high-/low-level width	t_{TIH}, t_{TIL}	V_{DD} = 2.7 to 5.5 V	0.1			μs
		V_{DD} = 1.8 to 5.5 V	1.8			μs
Interrupt input high-/low-level width	t_{INTH}, t_{INTL}	INTP0 to INTP2	10			μs
$\overline{\text{RESET}}$ input low-level width	t_{RSL}		10			μs
CPT90 input high-/low-level width	t_{CPH}, t_{CPL}		10			μs

T_{CY} vs V_{DD}



(2) Serial interface (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

(a) 3-wire serial I/O mode ($\overline{\text{SCK20}}$...Internal clock)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{\text{SCK20}}$ cycle time	t_{KCY1}	V_{DD} = 2.7 to 5.5 V		800			ns
		V_{DD} = 1.8 to 5.5 V		3,200			ns
$\overline{\text{SCK20}}$ high-/low-level width	t_{KH1}, t_{KL1}	V_{DD} = 2.7 to 5.5 V		t_{KCY1}/2–50			ns
		V_{DD} = 1.8 to 5.5 V		t_{KCY1}/2–150			ns
SI20 setup time (to $\overline{\text{SCK20}}$ ↑)	t_{SIK1}	V_{DD} = 2.7 to 5.5 V		150			ns
		V_{DD} = 1.8 to 5.5 V		500			ns
SI20 hold time (from $\overline{\text{SCK20}}$ ↑)	t_{KSI1}	V_{DD} = 2.7 to 5.5 V		400			ns
		V_{DD} = 1.8 to 5.5 V		600			ns
Delay time from $\overline{\text{SCK20}}$ ↓ to SO20 output	t_{KSO1}	R = 1 kΩ, C = 100 pF[Note]	V_{DD} = 2.7 to 5.5 V	0		250	ns
			V_{DD} = 1.8 to 5.5 V	0		1,000	ns

Note R and C are the load resistance and load capacitance of the SO20 output line.

(b) 3-wire serial I/O mode ($\overline{\text{SCK20}}$...External clock)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{\text{SCK20}}$ cycle time	t_{KCY2}	V_{DD} = 2.7 to 5.5 V		800			ns
		V_{DD} = 1.8 to 5.5 V		3,200			ns
$\overline{\text{SCK20}}$ high-/low-level width	t_{KH2}, t_{KL2}	V_{DD} = 2.7 to 5.5 V		400			ns
		V_{DD} = 1.8 to 5.5 V		1,600			ns
SI20 setup time (to $\overline{\text{SCK20}}$ ↑)	t_{SIK2}	V_{DD} = 2.7 to 5.5 V		100			ns
		V_{DD} = 1.8 to 5.5 V		150			ns
SI20 hold time (from $\overline{\text{SCK20}}$ ↑)	t_{KSI2}	V_{DD} = 2.7 to 5.5 V		400			ns
		V_{DD} = 1.8 to 5.5 V		600			ns
Delay time from $\overline{\text{SCK20}}$ ↓ to SO20 output	t_{KSO2}	R = 1 kΩ, C = 100 pF[Note]	V_{DD} = 2.7 to 5.5 V	0		300	ns
			V_{DD} = 1.8 to 5.5 V	0		1000	ns
SO20 setup time (when using $\overline{\text{SS20}}$, to $\overline{\text{SS20}}$ ↓)	t_{KAS2}	V_{DD} = 2.7 to 5.5 V				120	ns
		V_{DD} = 1.8 to 5.5 V				400	ns
SO20 disable time (when using $\overline{\text{SS20}}$, from $\overline{\text{SS20}}$↑)	t_{KDS2}	V_{DD} = 2.7 to 5.5 V				240	ns
		V_{DD} = 1.8 to 5.5 V				800	ns

Note R and C are the load resistance and load capacitance of the SO20 output line.

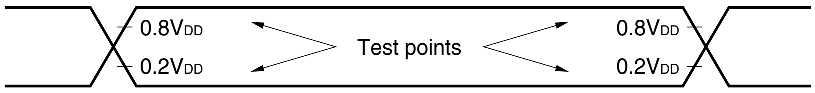
(c) UART mode (Dedicated baud rate generator output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		V_{DD} = 2.7 to 5.5 V			78,125	bps
		V_{DD} = 1.8 to 5.5 V			19,531	bps

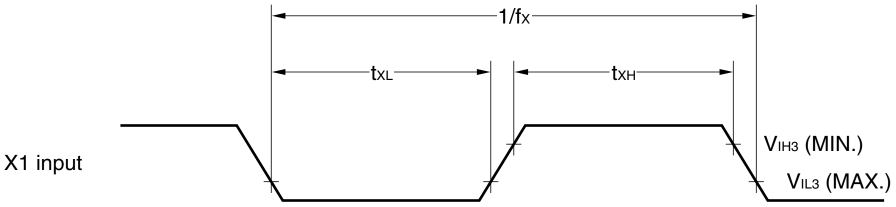
(d) UART mode (External clock input)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
ASCK20 cycle time	t_{KCY3}	V_{DD} = 2.7 to 5.5 V	800			ns
		V_{DD} = 1.8 to 5.5 V	3,200			ns
ASCK20 high-/low-level width	t_{KH3}, t_{KL3}	V_{DD} = 2.7 to 5.5 V	400			ns
		V_{DD} = 1.8 to 5.5 V	1,600			ns
Transfer rate		V_{DD} = 2.7 to 5.5 V			39,063	bps
		V_{DD} = 1.8 to 5.5 V			9,766	bps
ASCK20 rise/fall time	t_R, t_F				1	μs

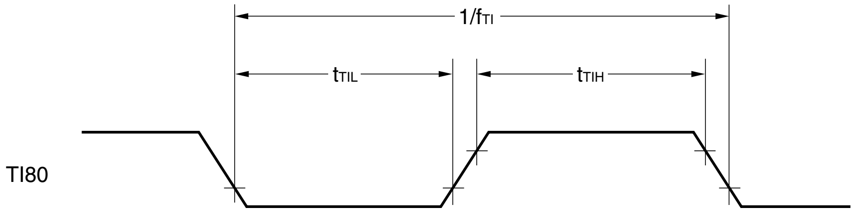
AC Timing Test Points (Excluding X1 Input)


$0.8V_{DD}$
$0.2V_{DD}$
Test points
$0.8V_{DD}$
$0.2V_{DD}$

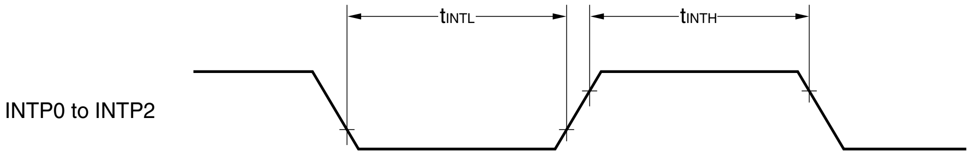
Clock Timing


$1/f_X$
t_{XL}
t_{XH}
X1 input
V_{IH3} (MIN.)
V_{IL3} (MAX.)

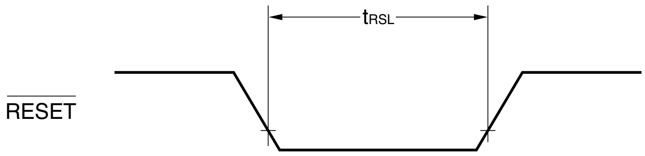
TI Timing


$1/f_{TI}$
t_{TIL}
t_{TIH}
TI80

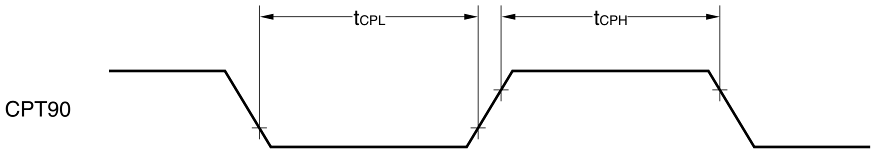
Interrupt Input Timing


t_{INTL}
t_{INTH}
INTP0 to INTP2

$\overline{\text{RESET}}$ Input Timing

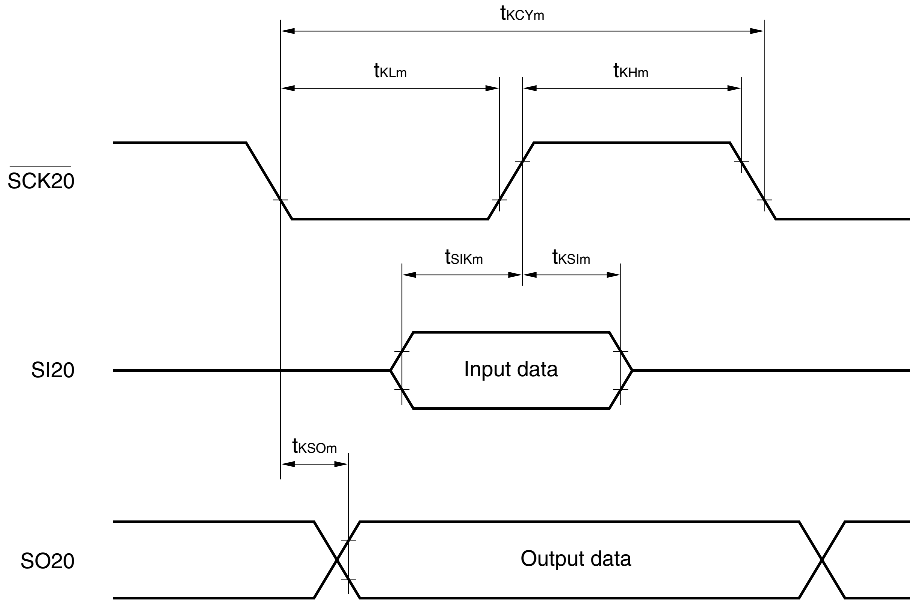

t_{RSL}
$\overline{\text{RESET}}$

CPT90 Input Timing


t_{CPL}
t_{CPH}
CPT90

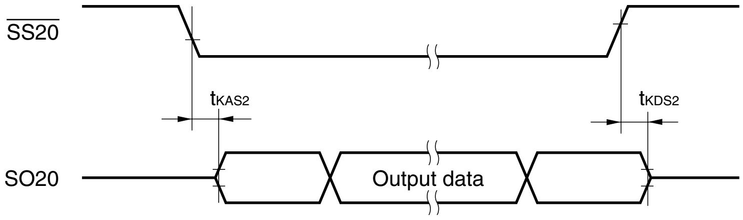
Serial Transfer Timing

3-wire serial I/O mode:

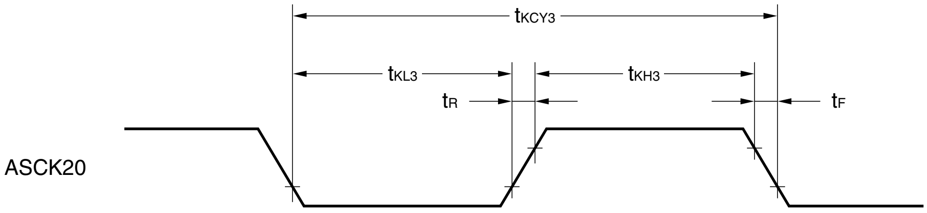

tKCYm
tKLm
tKHm
SCK20
tSIKm
tKSIm
SI20
Input data
tKSOm
SO20
Output data

Remark m = 1, 2

3-wire serial I/O mode (when using $\overline{\text{SS20}}$):


SS20
tKAS2
tKDS2
SO20
Output data

UART mode (external clock input):


tKCY3
tKL3
tKH3
tR
tF
ASCK20

Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics (T_A = –40 to +85°C)

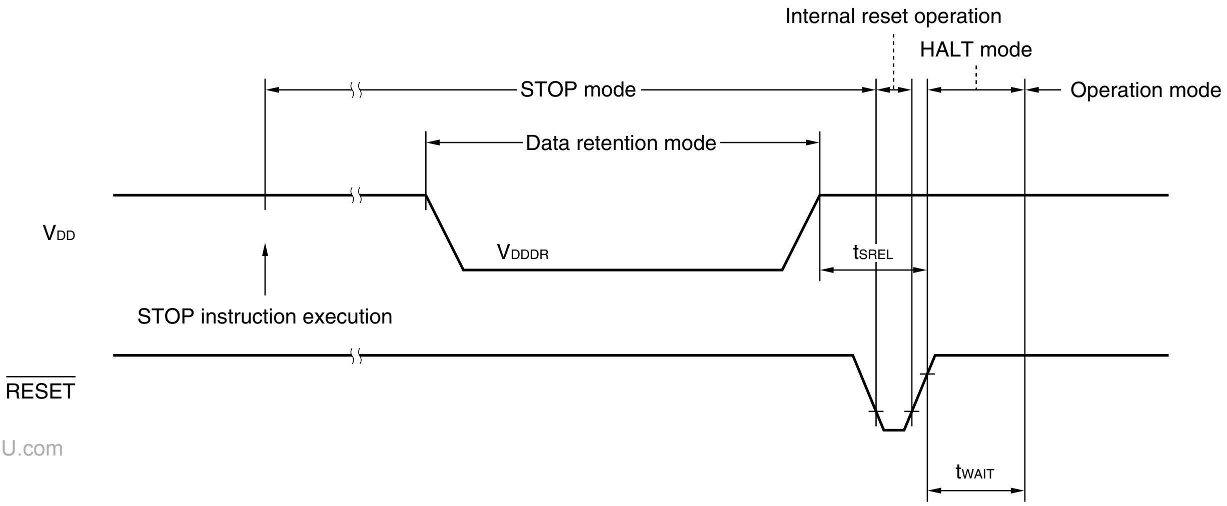
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention power supply voltage	V_{DDDR}		1.8		5.5	V
Release signal set time	t_{SREL}		0			μs
Oscillation stabilization wait time[Note 1]	t_{WAIT}	Release by $\overline{\text{RESET}}$		2^{15}/fx		ms
		Release by interrupt request		**Note 2**		ms

Notes 1. Oscillation stabilization wait time is the time in which the CPU operation is stopped to prevent unstable operation when oscillation is started.

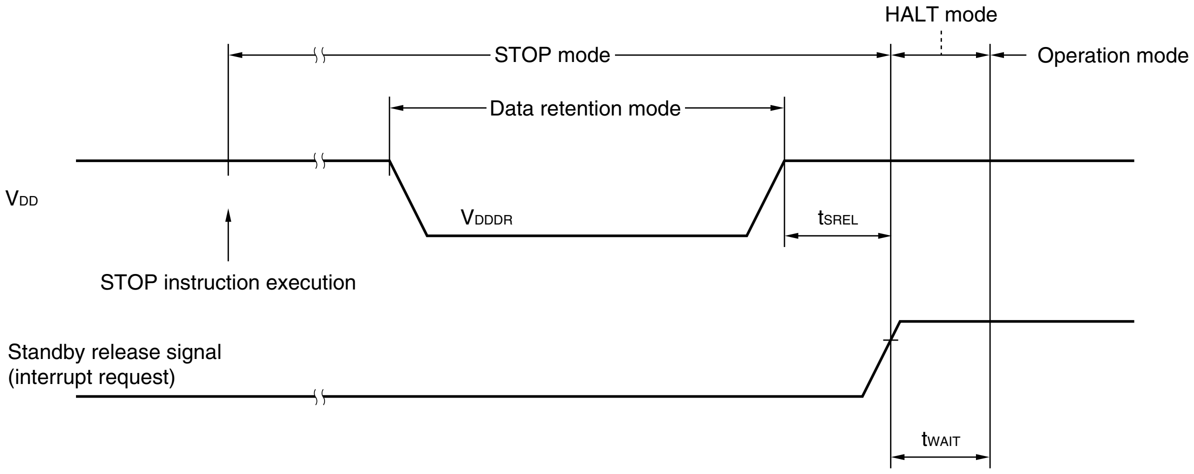
2. Selection of 2^{12}/fx, 2^{15}/fx, and 2^{17}/fx is possible using bits 0 to 2 (OSTS0 to OSTS2) of the oscillation stabilization time select register (OSTS).

Remark fx: System clock oscillation frequency

Data Retention Timing (STOP Mode Release by $\overline{\text{RESET}}$)


Internal reset operation
HALT mode
STOP mode
Operation mode
Data retention mode
V_{DD}
V_{DDDR}
t_{SREL}
STOP instruction execution
$\overline{\text{RESET}}$
t_{WAIT}

Data Retention Timing (Standby Release Signal: STOP Mode Release by Interrupt Signal)


HALT mode
STOP mode
Operation mode
Data retention mode
V_{DD}
V_{DDDR}
t_{SREL}
STOP instruction execution
Standby release signal
(interrupt request)
t_{WAIT}

Flash Memory Write/Erase Characteristics (T_A = 10 to 40°C, V_{DD} = 1.8 to 5.5 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Operating frequency	f_X	V_{DD} = 2.7 to 5.5 V		1.0		5	MHz
		V_{DD} = 1.8 to 5.5 V		1.0		1.25	MHz
Write current[Note] (V_{DD} pin)	I_{DDW}	When V_{PP} supply voltage = V_{PP1}	During f_X = 5.0 MHz operation			18	mA
Write current[Note] (V_{PP} pin)	I_{PPW}	When V_{PP} supply voltage = V_{PP1}				7.5	mA
Erase current[Note] (V_{DD} pin)	I_{DDE}	When V_{PP} supply voltage = V_{PP1}	During f_X = 5.0 MHz operation			18	mA
Erase current[Note] (V_{PP} pin)	I_{PPE}	When V_{PP} supply voltage = V_{PP1}				100	mA
Unit erase time	t_{er}			0.5	1	1	s
Total erase time	t_{era}					20	s
Write count		Erase/write are regarded as 1 cycle				20	Times
V_{PP} supply voltage	V_{PP0}	In normal operation		0		$0.2V_{DD}$	V
	V_{PP1}	During flash memory programming		9.7	10.0	10.3	V

Note The port current (including the current that flows to the on-chip pull-up resistors) is not included.

CHAPTER 16 ELECTRICAL SPECIFICATIONS (CONVENTIONAL PRODUCTS)

★ Absolute Maximum Ratings (T_A = 25°C)

Parameter	Symbol	Conditions		Ratings	Unit
Supply voltage	V_{DD}			–0.3 to +6.5	V
	V_{PP}	μPD78F9076 only, **Note**		–0.3 to +10.5	V
Input voltage	V_I			–0.3 to V_{DD} + 0.3	V
Output voltage	V_O			–0.3 to V_{DD} + 0.3	V
Output current, high	I_{OH}	Per pin	μPD78907x, 78F9076	–10	mA
		Total for all pins		–30	mA
		Per pin	μPD78907x(A)	–7	mA
		Total for all pins		–22	mA
Output current, low	I_{OL}	Per pin	μPD78907x, 78F9076	30	mA
		Total for all pins		160	mA
		Per pin	μPD78907x(A)	10	mA
		Total for all pins		120	mA
Operating ambient temperature	T_A	During normal operation		–40 to +85	°C
		During flash memory programming		10 to 40	°C
Storage temperature	T_{stg}	Mask ROM version		–65 to +150	°C
		μPD78F9076		–40 to +125	°C

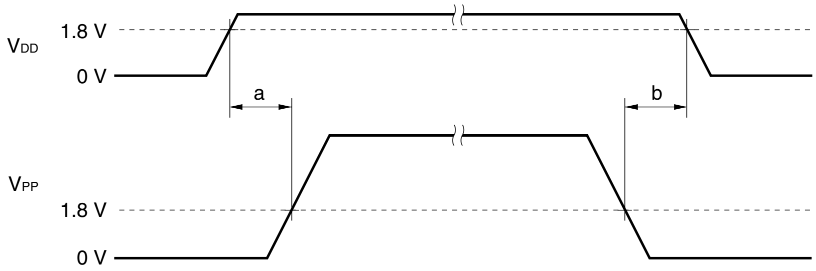
Note Make sure that the following conditions of the V_{PP} voltage application timing are satisfied when the flash memory is written.

- **When supply voltage rises**

  V_{PP} must exceed V_{DD} 10 μs or more after V_{DD} has reached the lower-limit value (1.8 V) of the operating voltage range (see a in the figure below).

- **When supply voltage drops**

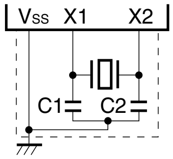
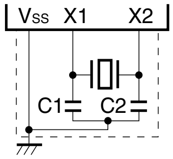
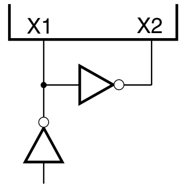
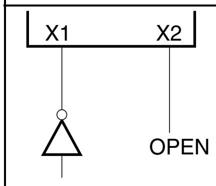
  V_{DD} must be lowered 10 μs or more after V_{PP} falls below the lower-limit value (1.8 V) of the operating voltage range of V_{DD} (see b in the figure below).



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	1.0		5.0	MHz
		Oscillation stabilization time[Note 2]	After V_{DD} reaches oscillation voltage range MIN.			4	ms
Crystal resonator		Oscillation frequency (f_X)[Note 1]		1.0		5.0	MHz
		Oscillation stabilization time[Note 2]	V_{DD} = 4.5 to 5.5 V			10	ms
			V_{DD} = 1.8 to 5.5 V			30	ms
External clock		X1 input frequency (f_X)[Note 1]		1.0		5.0	MHz
		X1 input high-/low-level width (t_{XH}, t_{XL})		85		500	ns
		X1 input frequency (f_X)[Note 1]	V_{DD} = 2.7 to 5.5 V	1.0		5.0	MHz
		X1 input high-/low-level width (t_{XH}, t_{XL})	V_{DD} = 2.7 to 5.5 V	85		500	ns

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.

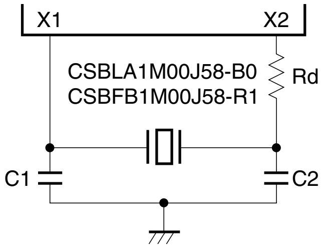
- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines.**
- **Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as V_{SS}.**
-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

Manufacturer	Part Number	Frequency (MHz)	Recommended Circuit Constant (pF)		Oscillation Voltage Range (V_{DD})		Remarks
			C1	C2	MIN.	MAX.	
Murata Mfg. Co., Ltd. (standard product)	CSBLA1M00J58-B0[Note]	1.0	100	100	1.9	5.5	Rd = 1.0 kΩ
	CSBFB1M00J58-R1[Note]						
	CSTCC2M00G56-R0	2.0	–	–	1.8		On-chip capacitor
	CSTLS2M00G56-B0						
	CSTCR4M00G53-R0	4.0					
	CSTLS4M00GG53-B0						
	CSTCR4M19G53-R0	4.194					
	CSTLS4M19GG53-B0						
	CSTCR4M91G53-R0	4.915					
	CSTLS4M91GG53-B0						
	CSTCR5M00G53-R0	5.0					
	CSTLS5M00GG53-B0						

Note A limiting resistor (Rd = 1.0 kΩ) is required when CSBLA1M00J58-B0 or CSBFB1M00J58-R1 (1.0 MHz) manufactured by Murata Mfg. Co., Ltd. is used as the ceramic resonator (see the figure below). This is not necessary when using one of the other recommended resonators.



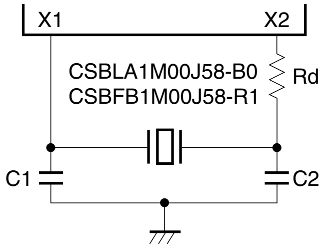
Caution The oscillator constant is a reference value based on evaluation in specific environments by the resonator manufacturer. If the oscillator characteristics need to be optimized in the actual application, request the resonator manufacturer for evaluation on the implementation circuit.

Note that the oscillation voltage and oscillation frequency merely indicate the characteristics of the oscillator. Use the internal operation conditions of the μPD789071, 789072, and 789074 within the specifications of the DC and AC characteristics.

Ceramic resonator (T_A = –40 to +85°C) (μPD78F9076)

Manufacturer	Part Number	Frequency (MHz)	Recommended Circuit Constant (pF)		Oscillation Voltage Range (V_{DD})		Remarks
			C1	C2	MIN.	MAX.	
Murata Mfg. Co., Ltd.	CSBLA1M00J58-B0[Note]	1.0	100	100	2.1	5.5	Rd = 1.0 kΩ
	CSBFB1M00J58-R1[Note]						
	CSTCC2M00G56-R0	2.0	–	–	1.9		On-chip capacitor
	CSTLS2M00G56-B0						
	CSTCR4M00G53-R0	4.0					
	CSTLS4M00GG53-B0						
	CSTCR4M19G53-R0	4.194					
	CSTLS4M19GG53-B0						
	CSTCR4M91G53-R0	4.915			2.0		
	CSTLS4M91GG53-B0						
	CSTCR5M00G53-R0	5.0					
	CSTLS5M00GG53-B0						
Murata Mfg. Co., Ltd. (low-voltage drive type)	CSTCR4M00G53093-R0	4.0	–	–	1.8	5.5	On-chip capacitor
	CSTLS4M00GG53093-B0						
	CSTCR4M19G53093-R0	4.194					
	CSTLS4M19GG53093-B0						
	CSTCR4M91G53093-R0	4.915					
	CSTLS4M91GG53093-B0						
	CSTCR5M00G53093-R0	5.0					
	CSTLS5M00GG53093-B0						

Note A limiting resistor (Rd = 1.0 kΩ) is required when CSBLA1M00J58-B0 or CSBFB1M00J58-R1 (1.0 MHz) manufactured by Murata Mfg. Co., Ltd. is used as the ceramic resonator (see the figure below). This is not necessary when using one of the other recommended resonators.



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 μPD78F9076 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)**

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high	I_{OH}	Per pin	μPD78907x, 78F9076			–1	mA
		Total for all pins				–15	mA
		Per pin	μPD78907x(A)			–1	mA
		Total for all pins				–11	mA
Output current, low	I_{OL}	Per pin	μPD78907x, 78F9076			10	mA
		Total for all pins				80	mA
		Per pin	μPD78907x(A)			3	mA
		Total for all pins				60	mA
Input voltage, high	V_{IH1}	P00 to P07, P10 to P15, P30, P31	V_{DD} = 2.7 to 5.5 V	0.7V_{DD}		V_{DD}	V
			V_{DD} = 1.8 to 5.5 V	0.9V_{DD}		V_{DD}	V
	V_{IH2}	$\overline{\text{RESET}}$, P20 to P27	V_{DD} = 2.7 to 5.5 V	0.8V_{DD}		V_{DD}	V
			V_{DD} = 1.8 to 5.5 V	0.9V_{DD}		V_{DD}	V
	V_{IH3}	X1, X2	V_{DD} = 4.5 to 5.5 V	V_{DD} – 0.5		V_{DD}	V
			V_{DD} = 1.8 to 5.5 V	V_{DD} – 0.1		V_{DD}	V
Input voltage, low	V_{IL1}	P00 to P07, P10 to P15, P30, P31	V_{DD} = 2.7 to 5.5 V	0		0.3V_{DD}	V
			V_{DD} = 1.8 to 5.5 V	0		0.1V_{DD}	V
	V_{IL2}	$\overline{\text{RESET}}$, P20 to P27	V_{DD} = 2.7 to 5.5 V	0		0.2V_{DD}	V
			V_{DD} = 1.8 to 5.5 V	0		0.1V_{DD}	V
	V_{IL3}	X1, X2	V_{DD} = 4.5 to 5.5 V	0		0.4	V
			V_{DD} = 1.8 to 5.5 V	0		0.1	V
Output voltage, high	V_{OH}	V_{DD} = 4.5 to 5.5 V, I_{OH} = –1 mA		V_{DD} – 1.0			V
		V_{DD} = 1.8 to 5.5 V, I_{OH} = –100 μA		V_{DD} – 0.5			V
Output voltage, low	V_{OL}	V_{DD} = 4.5 to 5.5 V, I_{OL} = 10 mA (μPD78907x,78F9076)				1.0	V
		V_{DD} = 4.5 to 5.5 V, I_{OL} = 3 mA (μPD78907x(A))				1.0	V
		V_{DD} = 1.8 to 5.5 V, I_{OL} = 400 μA				0.5	V
Input leakage current, high	I_{LIH1}	V_{IN} = V_{DD}	Pins other than X1, X2			3	μA
	I_{LIH2}		X1, X2			20	μA
Input leakage current, low	I_{LIL1}	V_{IN} = 0 V	Pins other than X1, X2			–3	μA
	I_{LIL2}		X1, X2			–20	μA
Output leakage current, high	I_{LOH}	V_{OUT} = V_{DD}				3	μA
Output leakage current, low	I_{LOL}	V_{OUT} = 0 V				–3	μA
Software pull-up resistor	R_1	V_{IN} = 0 V		50	100	200	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)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Power supply current[Note 1] (mask ROM version)	I_{DD1}	5.0 MHz crystal oscillation operating mode (C1 = C2 = 22 pF)	V_{DD} = 5.0 V ±10%[Note 2]		1.3	2.6	mA
			V_{DD} = 3.0 V ±10%[Note 3]		0.26	0.5	mA
			V_{DD} = 2.0 V ±10%[Note 3]		0.14	0.30	mA
	I_{DD2}	5.0 MHz crystal oscillation HALT mode (C1 = C2 = 22 pF)	V_{DD} = 5.0 V ±10%[Note 2]		0.5	1.0	mA
			V_{DD} = 3.0 V ±10%[Note 3]		0.17	0.35	mA
			V_{DD} = 2.0 V ±10%[Note 3]		0.08	0.2	mA
	I_{DD3}	STOP mode	V_{DD} = 5.0 V ±10%		0.1	10	μA
			V_{DD} = 3.0 V ±10%		0.05	5.0	μA
			V_{DD} = 2.0 V ±10%		0.05	3.0	μA
Power supply current[Note 1] (μPD78F9076)	I_{DD1}	5.0 MHz crystal oscillation operating mode (C1 = C2 = 22 pF)	V_{DD} = 5.0 V ±10%[Note 2]		4.0	8.0	mA
			V_{DD} = 3.0 V ±10%[Note 3]		1.0	2.5	mA
			V_{DD} = 2.0 V ±10%[Note 3]		0.8	2.0	mA
	I_{DD2}	5.0 MHz crystal oscillation HALT mode (C1 = C2 = 22 pF)	V_{DD} = 5.0 V ±10%[Note 2]		0.8	2.5	mA
			V_{DD} = 3.0 V ±10%[Note 3]		0.5	2.0	mA
			V_{DD} = 2.0 V ±10%[Note 3]		0.3	1.0	mA
	I_{DD3}	STOP mode	V_{DD} = 5.0 V ±10%		0.1	10	μA
			V_{DD} = 3.0 V ±10%		0.05	5.0	μA
			V_{DD} = 2.0 V ±10%		0.05	3.0	μA

Notes 1. The port current (including the current flowing through the on-chip pull-up resistors) is not included.

2. High-speed mode operation (when processor clock control register (PCC) is set to 00H)

3. Low-speed mode operation (when PCC is set to 02H)

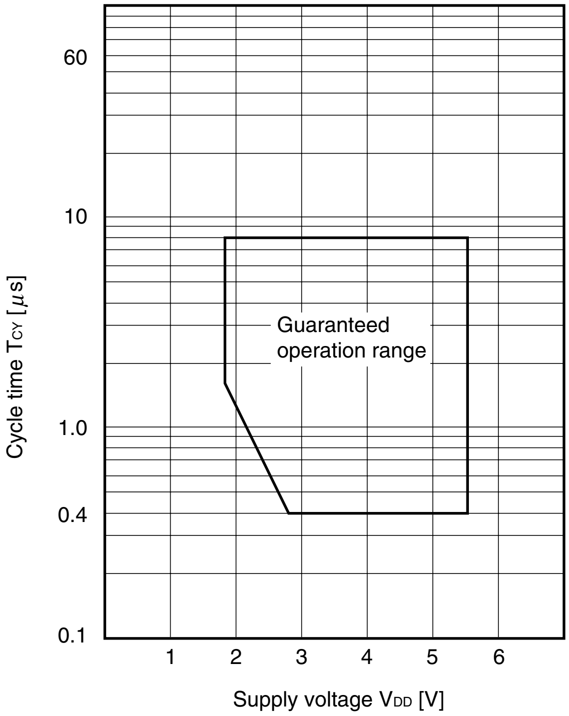
Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

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
Cycle time (Minimum instruction execution time)	T_{CY}	V_{DD} = 2.7 to 5.5 V	0.4		8	μs
		V_{DD} = 1.8 to 5.5 V	1.6		8	μs
TI80 input frequency	f_{TI}	V_{DD} = 2.7 to 5.5 V	0		4	MHz
		V_{DD} = 1.8 to 5.5 V	0		275	kHz
TI80 input high-/low-level width	t_{TIH}, t_{TIL}	V_{DD} = 2.7 to 5.5 V	0.1			μs
		V_{DD} = 1.8 to 5.5 V	1.8			μs
Interrupt input high-/low-level width	t_{INTH}, t_{INTL}	INTP0 to INTP2	10			μs
$\overline{\text{RESET}}$ input low-level width	t_{RSL}		10			μs
CPT90 input high-/low-level width	t_{CPH}, t_{CPL}		10			μs

T_{CY} vs V_{DD}



(2) Serial interface (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

(a) 3-wire serial I/O mode ($\overline{SCK20}$...Internal clock)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{SCK20}$ cycle time	t_{KCY1}	V_{DD} = 2.7 to 5.5 V		800			ns
		V_{DD} = 1.8 to 5.5 V		3,200			ns
$\overline{SCK20}$ high-/low-level width	t_{KH1}, t_{KL1}	V_{DD} = 2.7 to 5.5 V		$t_{KCY1}/2-50$			ns
		V_{DD} = 1.8 to 5.5 V		$t_{KCY1}/2-150$			ns
SI20 setup time (to $\overline{SCK20}$ ↑)	t_{SIK1}	V_{DD} = 2.7 to 5.5 V		150			ns
		V_{DD} = 1.8 to 5.5 V		500			ns
SI20 hold time (from $\overline{SCK20}$ ↑)	t_{KSI1}	V_{DD} = 2.7 to 5.5 V		400			ns
		V_{DD} = 1.8 to 5.5 V		600			ns
Delay time from $\overline{SCK20}$ ↓ to SO20 output	t_{KSO1}	R = 1 kΩ, C = 100 pF$^{\text{Note}}$	V_{DD} = 2.7 to 5.5 V	0		250	ns
			V_{DD} = 1.8 to 5.5 V	0		1,000	ns

Note R and C are the load resistance and load capacitance of the SO20 output line.

(b) 3-wire serial I/O mode ($\overline{SCK20}$...External clock)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{SCK20}$ cycle time	t_{KCY2}	V_{DD} = 2.7 to 5.5 V		800			ns
		V_{DD} = 1.8 to 5.5 V		3,200			ns
$\overline{SCK20}$ high-/low-level width	t_{KH2}, t_{KL2}	V_{DD} = 2.7 to 5.5 V		400			ns
		V_{DD} = 1.8 to 5.5 V		1,600			ns
SI20 setup time (to $\overline{SCK20}$ ↑)	t_{SIK2}	V_{DD} = 2.7 to 5.5 V		100			ns
		V_{DD} = 1.8 to 5.5 V		150			ns
SI20 hold time (from $\overline{SCK20}$ ↑)	t_{KSI2}	V_{DD} = 2.7 to 5.5 V		400			ns
		V_{DD} = 1.8 to 5.5 V		600			ns
Delay time from $\overline{SCK20}$ ↓ to SO20 output	t_{KSO2}	R = 1 kΩ, C = 100 pF$^{\text{Note}}$	V_{DD} = 2.7 to 5.5 V	0		300	ns
			V_{DD} = 1.8 to 5.5 V	0		1000	ns
SO20 setup time (when using $\overline{SS20}$, to $\overline{SS20}$ ↓)	t_{KAS2}	V_{DD} = 2.7 to 5.5 V				120	ns
		V_{DD} = 1.8 to 5.5 V				400	ns
SO20 disable time (when using $\overline{SS20}$, from $\overline{SS20}$↑)	t_{KDS2}	V_{DD} = 2.7 to 5.5 V				240	ns
		V_{DD} = 1.8 to 5.5 V				800	ns

Note R and C are the load resistance and load capacitance of the SO20 output line.

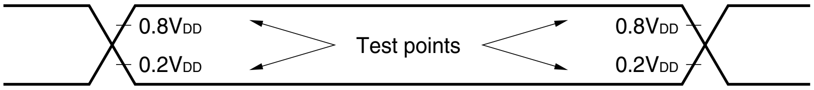
(c) UART mode (Dedicated baud rate generator output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		V_{DD} = 2.7 to 5.5 V			78,125	bps
		V_{DD} = 1.8 to 5.5 V			19,531	bps

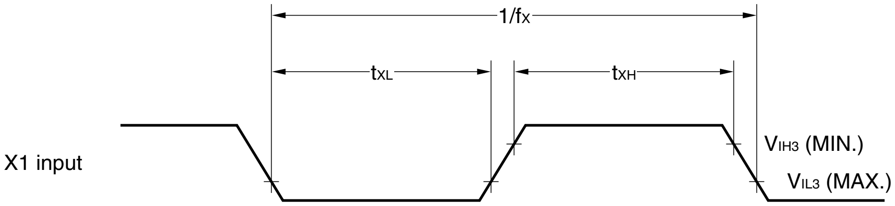
(d) UART mode (External clock input)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
ASCK20 cycle time	t_{KCY3}	V_{DD} = 2.7 to 5.5 V	800			ns
		V_{DD} = 1.8 to 5.5 V	3,200			ns
ASCK20 high-/low-level width	t_{KH3}, t_{KL3}	V_{DD} = 2.7 to 5.5 V	400			ns
		V_{DD} = 1.8 to 5.5 V	1,600			ns
Transfer rate		V_{DD} = 2.7 to 5.5 V			39,063	bps
		V_{DD} = 1.8 to 5.5 V			9,766	bps
ASCK20 rise/fall time	t_R, t_F				1	μs

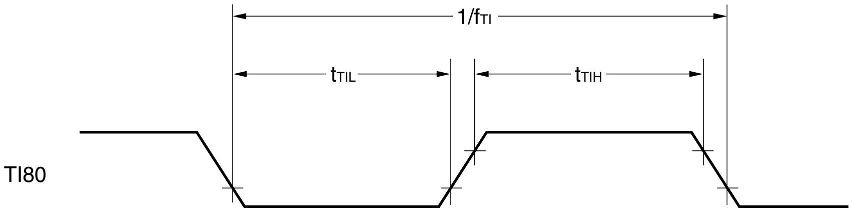
AC Timing Test Points (Excluding X1 Input)


0.8VDD
0.2VDD
Test points
0.8VDD
0.2VDD

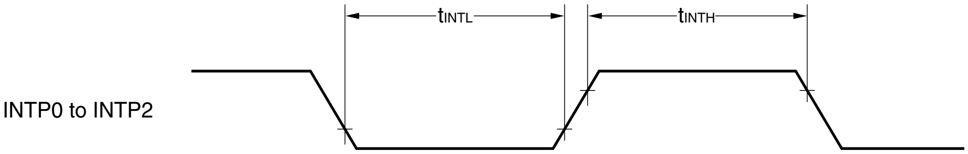
Clock Timing


1/fX
tXL
tXH
X1 input
VIH3 (MIN.)
VIL3 (MAX.)

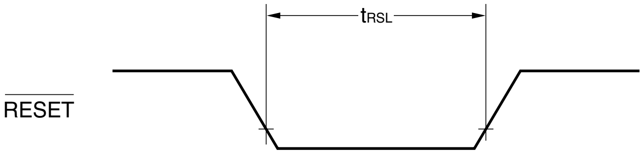
TI Timing


1/fTI
tTIL
tTIH
TI80

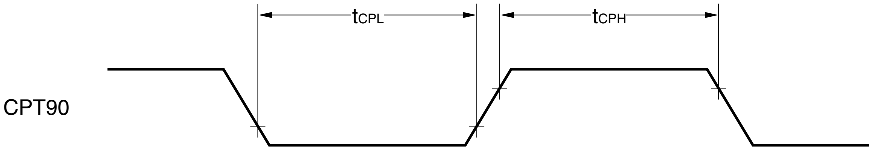
Interrupt Input Timing


tINTL
tINTH
INTP0 to INTP2

RESET Input Timing

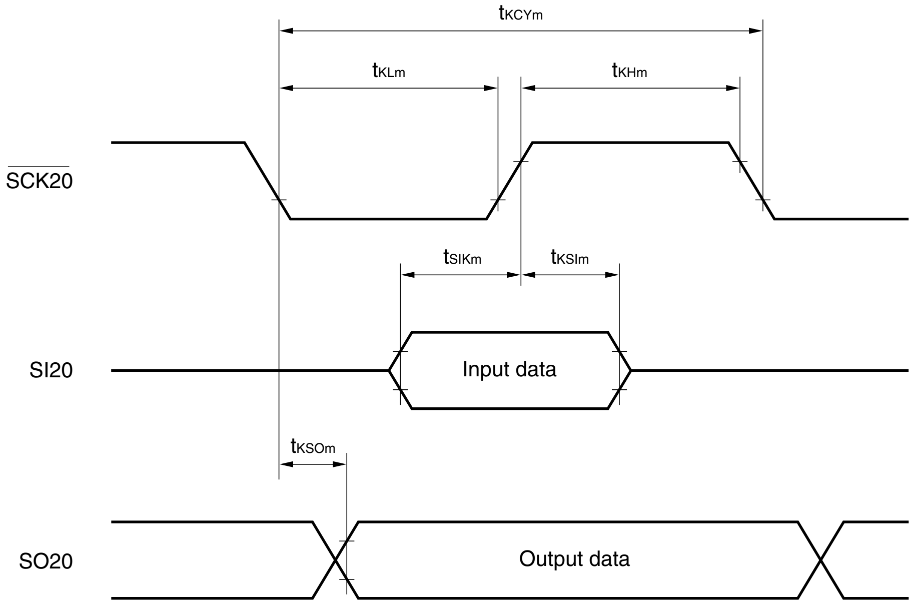


CPT90 Input Timing


tCPL
tCPH
CPT90

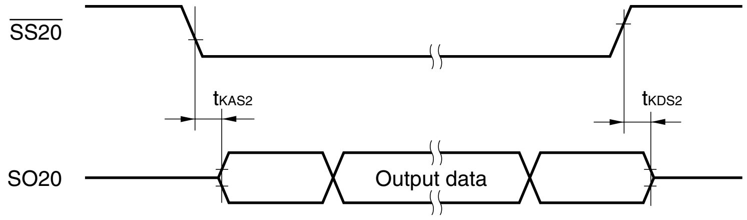
Serial Transfer Timing

3-wire serial I/O mode:

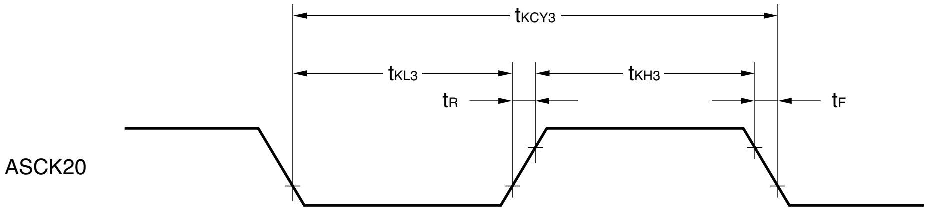

tKCYm
tKLm
tKHm
SCK20
tSIKm
tKSIm
SI20
Input data
tKSOm
SO20
Output data

Remark m = 1, 2

3-wire serial I/O mode (when using $\overline{\text{SS20}}$):


SS20
tKAS2
tKDS2
SO20
Output data

UART mode (external clock input):


tKCY3
tKL3
tKH3
tR
tF
ASCK20

Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics (T_A = –40 to +85°C)

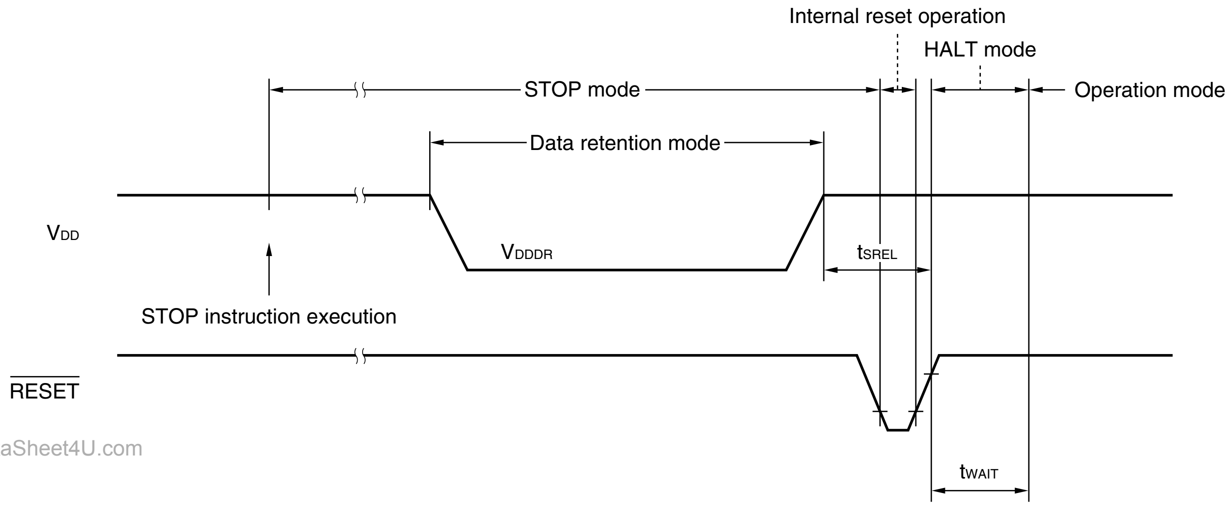
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention power supply voltage	V_{DDDR}		1.8		5.5	V
Release signal set time	t_{SREL}		0			μs
Oscillation stabilization wait time[Note 1]	t_{WAIT}	Release by $\overline{\text{RESET}}$		$2^{15}/f_X$		ms
		Release by interrupt request		**Note 2**		ms

Notes 1. Oscillation stabilization wait time is the time in which the CPU operation is stopped to prevent unstable operation when oscillation is started.

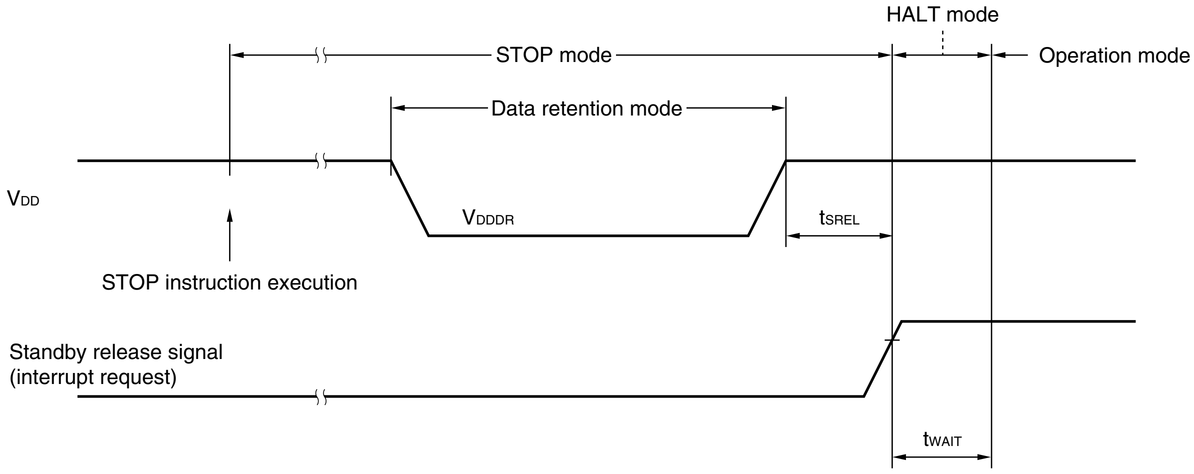
2. Selection of $2^{12}/f_X$, $2^{15}/f_X$, and $2^{17}/f_X$ is possible using bits 0 to 2 (OSTS0 to OSTS2) of the oscillation stabilization time select register (OSTS).

Remark f_X: System clock oscillation frequency

Data Retention Timing (STOP Mode Release by $\overline{\text{RESET}}$)


Internal reset operation
HALT mode
STOP mode
Operation mode
Data retention mode
VDD
VDDDR
tSREL
STOP instruction execution
RESET
tWAIT

Data Retention Timing (Standby Release Signal: STOP Mode Release by Interrupt Signal)

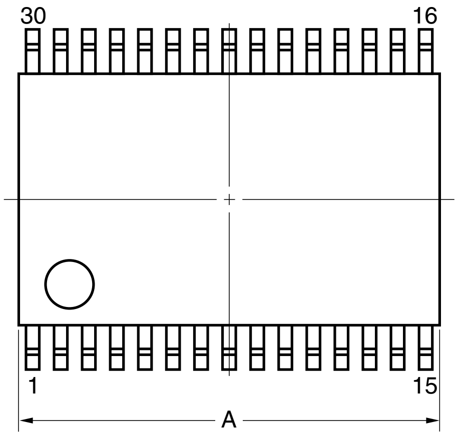

HALT mode
STOP mode
Operation mode
Data retention mode
VDD
VDDDR
tSREL
STOP instruction execution
Standby release signal
(interrupt request)
tWAIT

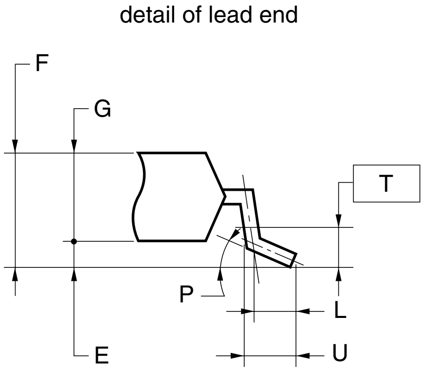
Flash Memory Write/Erase Characteristics (T_A = 10 to 40°C, V_{DD} = 1.8 to 5.5 V)

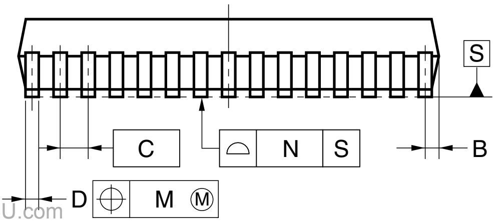
Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Operating frequency	f_X	V_{DD} = 2.7 to 5.5 V		1.0		5	MHz
		V_{DD} = 1.8 to 5.5 V		1.0		1.25	MHz
Write current[Note] (V_{DD} pin)	I_{DDW}	When V_{PP} supply voltage = V_{PP1}	During f_X = 5.0 MHz operation			18	mA
Write current[Note] (V_{PP} pin)	I_{PPW}	When V_{PP} supply voltage = V_{PP1}				7.5	mA
Erase current[Note] (V_{DD} pin)	I_{DDE}	When V_{PP} supply voltage = V_{PP1}	During f_X = 5.0 MHz operation			18	mA
Erase current[Note] (V_{PP} pin)	I_{PPE}	When V_{PP} supply voltage = V_{PP1}				100	mA
Unit erase time	t_{er}			0.5	1	1	s
Total erase time	t_{era}					20	s
★ Write count		Erase/write are regarded as 1 cycle				20	Times
V_{PP} supply voltage	V_{PP0}	In normal operation		0		0.2V_{DD}	V
	V_{PP1}	During flash memory programming		9.7	10.0	10.3	V

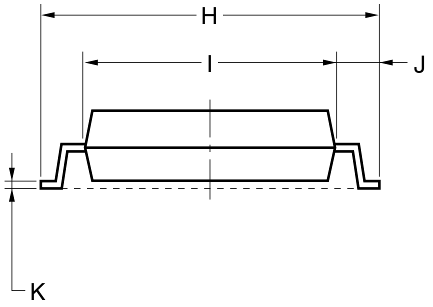
Note The port current (including the current that flows to the on-chip pull-up resistors) is not included.

30-PIN PLASTIC SSOP (7.62 mm (300))




detail of lead end
F
G
T
P
L
E
U





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 18 RECOMMENDED SOLDERING CONDITIONS

The μPD789071, 789072, and 789074 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 18-1. Surface Mounting Type Soldering Conditions

μPD789071MC-××× -5A4: 30-pin plastic SSOP (7.62 mm (300))
μPD789072MC-××× -5A4: 30-pin plastic SSOP (7.62 mm (300))
μPD789074MC-××× -5A4: 30-pin plastic SSOP (7.62 mm (300))
μPD789071MC(A)-××× -5A4: 30-pin plastic SSOP (7.62 mm (300))
μPD789072MC(A)-××× -5A4: 30-pin plastic SSOP (7.62 mm (300))
μPD789074MC(A)-××× -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	IR35-00-3
VPS	Package peak temperature: 215°C, Time: 40 seconds max. (at 200°C or higher), Count: Three times or less	VP15-00-3
Wave soldering	Solder bath temperature: 260°C max., Time: 10 seconds max., Count: Once, Preheating temperature: 120°C max. (package surface temperature)	WS60-00-1
Partial heating	Pin temperature: 300°C max., Time: 3 seconds max. (per pin row)	–

μPD78F9076MC-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, prebaking is necessary 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, prebaking is necessary 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).

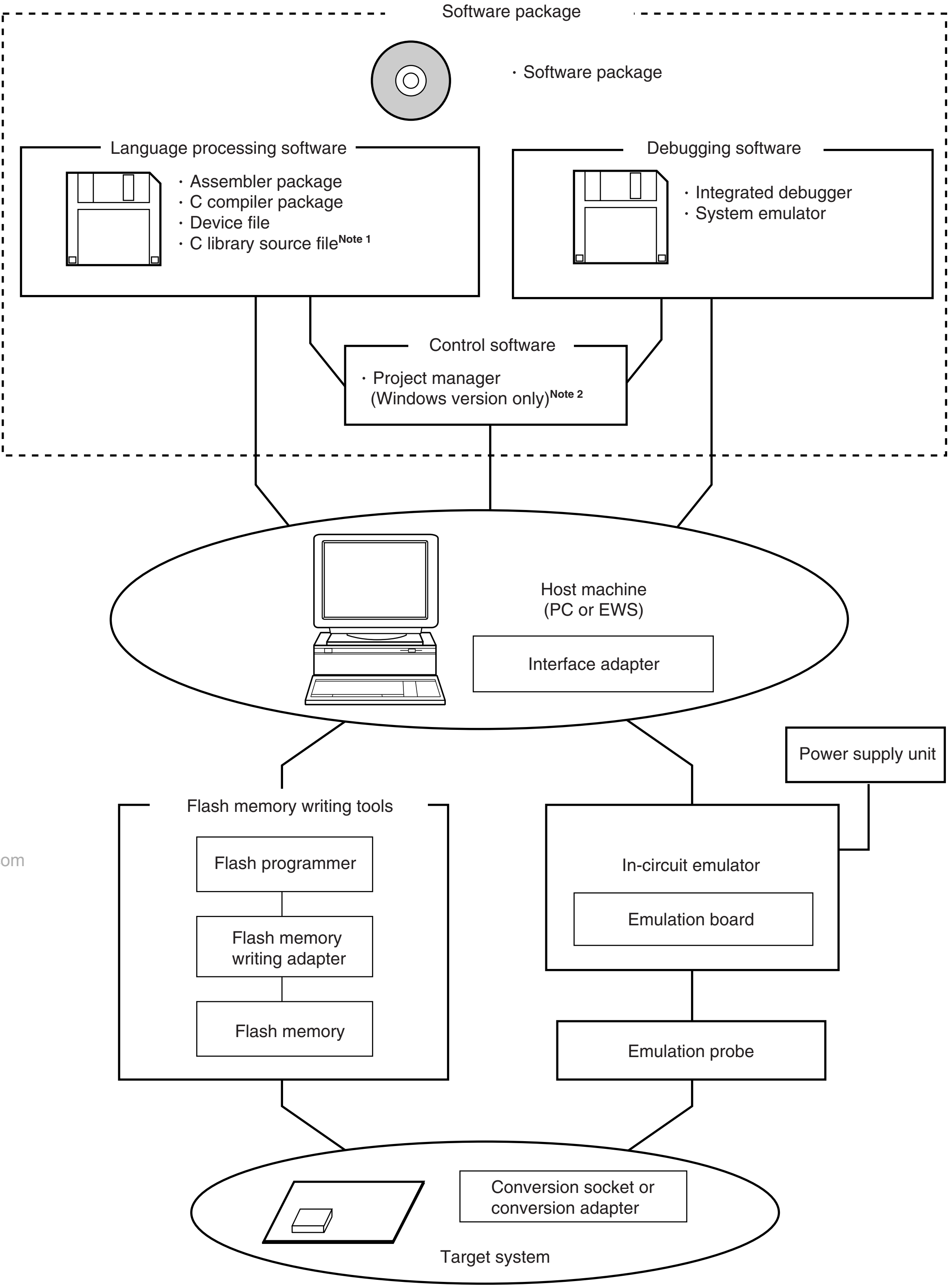
APPENDIX A DEVELOPMENT TOOLS

The following development tools are available for development of systems using the μPD789074 Subseries. Figure A-1 shows the development tools.

- Compatibility with PC98-NX Series
  Unless stated otherwise, products which are supported by 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, 98, 2000
  - Windows NT™ Ver. 4.0

Figure A-1. Development Tools


Software package
· Software package
Language processing software
· Assembler package
· C compiler package
· Device file
· C library source file Note 1
Debugging software
· Integrated debugger
· System emulator
Control software
· Project manager
(Windows version only) Note 2
Host machine
(PC or EWS)
Interface adapter
Power supply unit
Flash memory writing tools
Flash programmer
Flash memory writing adapter
Flash memory
In-circuit emulator
Emulation board
Emulation probe
Conversion socket or conversion adapter
Target system

Notes 1. The C library source file is not included in the software package.

2. The project manager is included in the assembler package and is available only for Windows.

A.1 Software Package

SP78K0S Software package	Various software tools for 78K/0S development are integrated in one package. The following tools are included. RA78K0S, CC78K0S, ID78K0-NS, SM78K0S, various device files
	Part number: μS××××SP78K0S

Remark ×××× in the part number differs depending on the operating system used.

μS×××× SP78K0S

××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT and compatibles	Japanese Windows	CD-ROM
BB17		English Windows	

Note Also operates under the DOS environment

A.2 Language Processing Software

RA78K0S Assembler package	Program that converts program written in mnemonic into object codes that can be executed by a microcontroller. In addition, automatic functions to generate a symbol table and optimize branch instructions are also provided. Used in combination with a device file (DF789076) (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 the project manager of Windows (included in the 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 a microcontroller. Used in combination with an assembler package (RA78K0S) and device file (DF789076) (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 the project manager of Windows (included in the assembler package).
	Part number: μS××××CC78K0S
DF789076[Note 1] Device file	File containing the information inherent to the device. Used in combination with other tools (RA78K0S, CC78K0S, ID78K0S-NS, SM78K0S) (all sold separately).
	Part number: μS××××DF789076
CC78K0S-L[Note 2] C library source file	Source file of functions constituting the object library included in the C compiler package. Necessary for changing the object library included in the C compiler package according to the customer's specifications. Since this is a source file, its working environment does not depend on any particular operating system.
	Part number: μS××××CC78K0S-L

Notes 1. DF789076 is a common file that can be used with RA78K0S, CC78K0S, ID78K0S-NS, and SM78K0S.

2. CC78K0S-L is not included in the software package (SP78K0S).

Remark ×××× in the part number differs depending on the host machine and operating system used.

μS××××RA78K0S
μS××××CC78K0S

××××	Host Machine	OS	Supply Media
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××××DF789076
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 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 Flash Memory Writing Tools

Flashpro III (FL-PR3, PG-FP3) Flashpro IV (FL-PR4, PG-FP4) Flash writer	Flash programmer dedicated to microcontrollers incorporating flash memory.
FA-30MC Flash memory writing adapter	Flash memory writing adapter. Used in connection with Flashpro III or Flashpro IV. 30-pin plastic SSOP (MC-5A4 type)

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 (Hardware)

IE-78K0S-NS In-circuit emulator	In-circuit emulator for debugging hardware and software of application system using the 78K/0S Series. Can be used with an integrated debugger (ID78K0S-NS). Used in combination with an AC adapter, emulation probe, and interface adapter for connecting the host machine.
IE-78K0S-NS-A In-circuit emulator	In-circuit emulator with enhanced functions of the IE-78K0S-NS. The debug function is further enhanced by adding a coverage function and enhancing the tracer and timer functions.
IE-70000-MC-PS-B AC adapter	Adapter for supplying power from a 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 (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 (PCMICA socket supported).
IE-70000-PC-IF-C Interface adapter	Adapter required when using an IBM PC/AT or compatible as the host machine (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.
IE-789046-NS-EM1 + NP-K907 Emulation board	Emulation board for emulating the peripheral hardware inherent to the device. Used in combination with an in-circuit emulator.
★ NP-30MC Emulation probe	Probe for connecting the in-circuit emulator and target system. Used in combination with NSPACK30BK and YSPACK30BK.
NSPACK30BK YSPACK30BK Conversion adapter	Conversion adapter used to connect a target system board designed to allow mounting a 30-pin plastic SSOP and the NP-30MC.

Remarks 1. NP-30MC and NP-K907 are products 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 made by TOKYO ELETECH CORPORATION.
For further information, contact: Daimaru Kogyo, Ltd.
Tokyo Electronics Department (TEL +81-3-3820-7112)
Osaka Electronics Department (TEL +81-6-6244-6672)

A.6 Debugging Tools (Software)

ID78K0S-NS Integrated debugger	This debugger supports the in-circuit emulators IE-78K0S-NS and IE-78K0S-NS-A for the 78K/0S Series. The ID78K0S-NS is Windows-based software. It has improved C-compatible debugging functions and can display the results of tracing with the source program using an integrating window function that associates the source program, disassemble display, and memory display with the trace result. Used in combination with a device file (DF789076) (sold separately).
	Part number: μS××××ID78K0S-NS
SM78K0S System simulator	This is a system simulator for the 78K/0S Series. The SM78K0S is Windows-based software. It can be used to debug the target system at C source level of assembler level while simulating the operation of the target system on the host machine. Using SM78K0S, the logic and performance of the application can be verified independently of hardware development. Therefore, the development efficiency can be enhanced and the software quality can be improved. Used in combination with a device file (DF789076) (sold separately).
	Part number: μS××××SM78K0S
DF789076[Note] Device file	File containing the information inherent to the device. Used in combination with other tools (RA78K0S, CC78K0S, ID78K0S-NS, SM78K0S) (all sold separately).
	Part number: μS××××DF789076

Note DF789076 is a common file that can be used with RA78K0S, CC78K0S, ID78K0S-NS, and SM78K0S.

Remark ×××× in the part number differs depending on the operating system used and the supply medium.

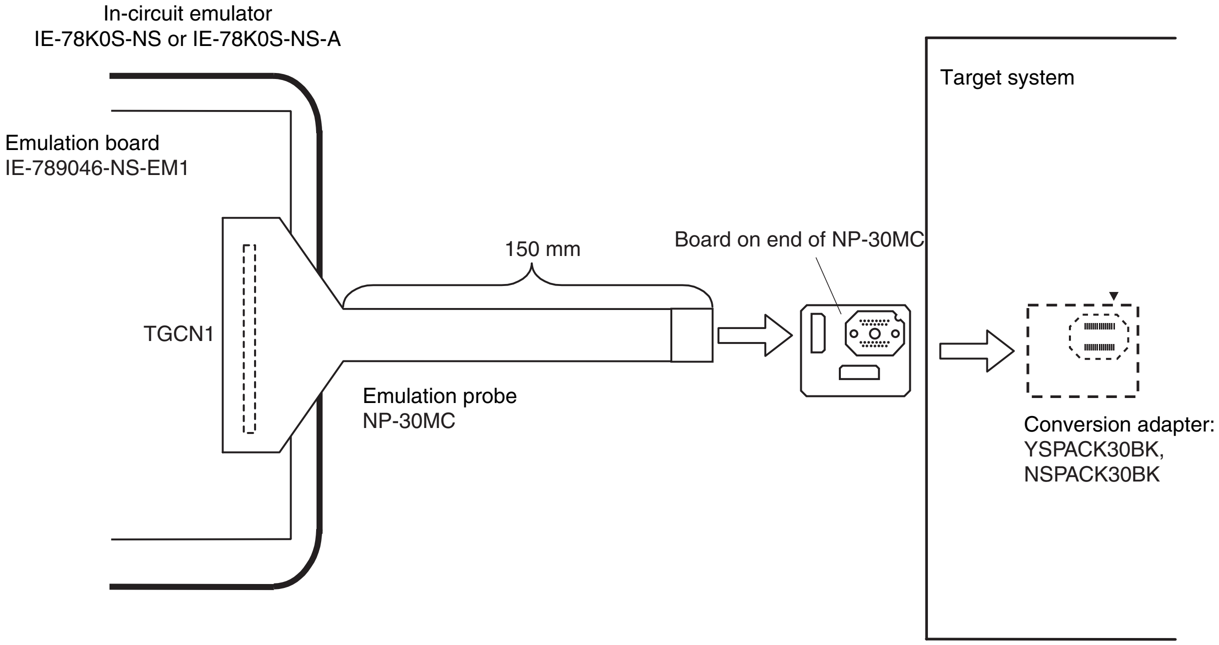
μS××××ID78K0S-NS
μS××××SM78K0S

××××	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	

★ APPENDIX B 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.

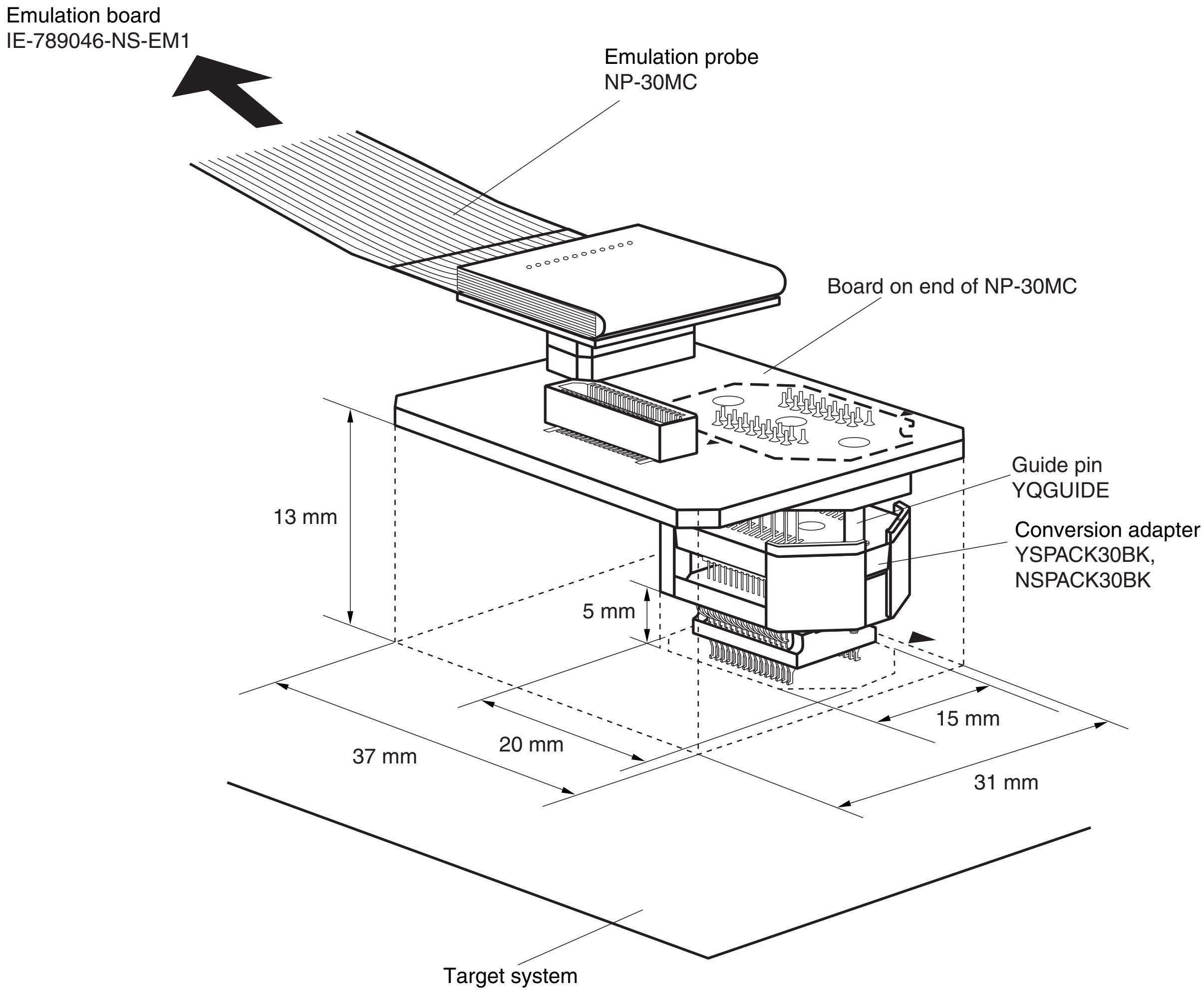
Figure B-1. Distance Between In-Circuit Emulator and Conversion Adapter


In-circuit emulator
IE-78K0S-NS or IE-78K0S-NS-A
Emulation board
IE-789046-NS-EM1
TGCN1
150 mm
Emulation probe
NP-30MC
Board on end of NP-30MC
Target system
Conversion adapter:
YSPACK30BK,
NSPACK30BK

Remarks 1. NP-30MC is a product of Naito Densei Machida Mfg. Co., Ltd.

2. YSPACK30BK and NSPACK30BK are products of TOKYO ELETECH CORPORATION.

Figure B-2. Connection Condition of Target System


Emulation board
IE-789046-NS-EM1
Emulation probe
NP-30MC
Board on end of NP-30MC
Guide pin
YQGUIDE
Conversion adapter
YSPACK30BK,
NSPACK30BK
13 mm
5 mm
15 mm
20 mm
37 mm
31 mm
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 C REGISTER INDEX

C.1 Register Name Index (Alphabetic Order)

16-bit capture register 90 (TCP90) 84
16-bit compare register 90 (CR90) 84
16-bit timer counter 90 (TM90) 84
16-bit timer mode control register 90 (TMC90) 85
8-bit compare register 80 (CR80) 98
8-bit timer counter 80 (TM80) 98
8-bit timer mode control register 80 (TMC80) 99

[A]
Asynchronous serial interface mode register 20 (ASIM20) 120
Asynchronous serial interface status register 20 (ASIS20) 122

[B]
Baud rate generator control register 20 (BRGC20) 123
Buzzer output control register 90 (BZC90) 87

[E]
External interrupt mode register 0 (INTM0) 156

[I]
Interrupt mask flag register 0, 1 (MK0, MK1) 155
Interrupt request flag register 0, 1 (IF0, IF1) 154

[O]
Oscillation stabilization time selection register (OSTS) 166

[P]
Port 0 (P0) 64
Port 1 (P1) 65
Port 2 (P2) 66
Port 3 (P3) 70
Port mode register 0 (PM0) 71
Port mode register 1 (PM1) 71
Port mode register 2 (PM2) 71
Port mode register 3 (PM3) 71
Processor clock control register (PCC) 76
Pull-up resistor option register 0 (PU0) 72
Pull-up resistor option register B2 (PUB2) 73

[R]
Receive buffer register 20 (RXB20) 118

[S]

Serial operation mode register 20 (CSIM20) 119

[T]

Transmission shift register 20 (TXS20) 118

[W]

Watchdog timer clock selection register (WDCS) 111
Watchdog timer mode register (WDTM) 112

C.2 Register Symbol Index (Alphabetic Order)

[A]

ASIM20: Asynchronous serial interface mode register 20 ... 120
ASIS20 : Asynchronous serial interface status register 20 ... 122

[B]

BRGC20: Baud rate generator control register 20 ... 123
BZC90: Buzzer output control register 90 ... 87

[C]

CR80: 8-bit compare register 80 ... 98
CR90: 16-bit compare register 90 ... 84
CSIM20: Serial operation mode register 20 ... 119

[I]

IF0: Interrupt request flag register 0 ... 154
IF1: Interrupt request flag register 1 ... 154
INTM0: External interrupt mode register 0 ... 156

[M]

MK0: Interrupt mask flag register 0 ... 155
MK1: Interrupt mask flag register 1 ... 155

[O]

OSTS: Oscillation stabilization time selection register ... 166

[P]

P0: Port 0 ... 64
P1: Port 1 ... 65
P2: Port 2 ... 66
P3: Port 3 ... 70
PCC: Processor clock control register ... 76
PM0: Port mode register 0 ... 71
PM1: Port mode register 1 ... 71
PM2: Port mode register 2 ... 71
PM3: Port mode register 3 ... 71
PU0: Pull-up resistor option register 0 ... 72
PUB2: Pull-up resistor option register B2 ... 73

[R]

RXB20: Receive buffer register 20 ... 118

[T]

TCP90: 16-bit capture register 90 ... 84
TM80: 8-bit timer counter 80 ... 98
TM90: 16-bit timer counter 90 ... 84
TMC80: 8-bit timer mode control register 80 ... 99

TMC90: 16-bit timer mode control register 90 85
TXS20: Transmission shift register 20 118

[W]

WDCS: Watchdog timer clock selection register 111
WDTM: Watchdog timer mode register 112

APPENDIX D REVISION HISTORY

The following shows the revision history. "Chapter" refers to the chapters in the respective edition.

(1/2)

Edition	Description	Chapter
2nd edition	Change of μPD789071, 789072, 789074, and 78F9076 from under development to development complete.	Throughout
	Modification of description of VPP pin connection	**CHAPTER 2 PIN FUNCTION**
	Modification of **Caution** on rewriting CR90 in **6.4.1 Operation as timer interrupt**	**CHAPTER 6 16-BIT TIMER 90**
	Addition of description on reading receive data of UART	**CHAPTER 9 SERIAL INTERFACE 20**
	Addition of **Note** on unused pins in **Table 13-2 Communication Mode**	**CHAPTER 13 μPD78F9076**
	Addition of **Note** and **Remark** to **Figures 13-2 Flashpro III Connection Example in 3-Wire Serial I/O Mode**, **13-3 Flashpro III Connection Example in UART Mode**, and **13-4 Flashpro III Connection Example in Pseudo 3-Wire Mode**	
	Modification of value of UART in **Table 13-4 Setting Example with PG-FP3**	
	Addition of **13.1.5 On-board pin connections**	
	Addition of electrical specifications	**CHAPTER 15 ELECTRICAL SPECIFICATIONS**
	Addition of package drawing	**CHAPTER 16 PACKAGE DRAWING**
	Addition of recommended soldering conditions	**CHAPTER 17 RECOMMENDED SOLDERING CONDITIONS**
	Overall modification of description on development tools Deletion of Embedded Software	**APPENDIX A DEVELOPMENT TOOLS**
3rd edition	• Addition of μPD789071(A), 789072(A), and 789074(A) • Addition of description of expanded-specification products	Throughout
	• Addition of **1.1 Expanded-Specification Products and Conventional Products** • Addition of **1.5 Quality Grades** • Addition of **1.10 Differences Between Standard Quality Grade Products and (A) Products**	**CHAPTER 1 GENERAL**
	• Modification of description of **6.4.1 Operation as timer interrupt** • Modification of description of **6.4.2 Operation as timer output** • Addition of **6.5 Notes on Using 16-Bit Timer 90** • Modification of **Figure 6-6. Timing of Timer Interrupt Operation** • Modification of **Figure 6-8. Timer Output Timing**	**CHAPTER 6 16-BIT TIMER 90**
	• Addition of **7.5 (3) Timer operation after compare register is rewritten during PWM output** • Addition of **7.5 (4) Cautions when STOP mode is set** • Addition of **7.5 (5) Start timing of external event counter**	**CHAPTER 7 8-BIT TIMER/EVENT COUNTER 80**
	Total revision of description of flash memory programming	**CHAPTER 13 μPD78F9076**

(2/2)

Edition	Description	Chapter
3rd edition	Addition of chapter	**CHAPTER 15 ELECTRICAL SPECIFICATIONS (EXPANDED-SPECIFICATION PRODUCTS)**
	Modification of table of recommended oscillator constant	**CHAPTER 16 ELECTRICAL SPECIFICATIONS (CONVENTIONAL PRODUCTS)**
	Change of recommended soldering conditions of μPD78F9076	**CHAPTER 18 RECOMMENDED SOLDERING CONDITIONS**
	Modification of description of **A.5 Debugging Tools (Hardware)**	**APPENDIX A DEVELOPMENT TOOLS**
	Addition of chapter	**APPENDIX B NOTES ON TARGET SYSTEM DESIGN**