

# Preliminary User's Manual

## **78K0/KB1**

### **8-Bit Single-Chip Microcontrollers**

---

**$\mu$ PD780101**

**$\mu$ PD780102**

**$\mu$ PD780103**

**$\mu$ PD78F0103**

[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.

**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.**

**Ethernet is a trademark of Xerox Corp.**

**OSF/Motif is a trademark of OpenSoftware Foundation, Inc.**

**TRON stands for The Realtime Operating system Nucleus.**

**ITRON is an abbreviation of Industrial TRON.**

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

License not needed:  $\mu$ PD78F0103

The customer must judge the need for a license:  $\mu$ PD780101, 780102, and 780103

- **The information contained in this document is being issued in advance of the production cycle for the device. The parameters for the device may change before final production or NEC Corporation, at its own discretion, may withdraw the device prior to its production.**
  - **Not all devices/types available in every country. Please check with local NEC 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 Corporation. NEC Corporation assumes no responsibility for any errors which may appear in this document.
  - NEC Corporation does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from use of a device described herein or any other liability arising from use of such device. No license, either express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Corporation 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 the customer's equipment shall be done under the full responsibility of the customer. NEC Corporation assumes no responsibility for any losses incurred by the customer or third parties arising from the use of these circuits, software, and information.
  - While NEC Corporation has been making continuous effort to enhance the reliability of its semiconductor devices, the possibility of defects cannot be eliminated entirely. To minimize risks of damage or injury to persons or property arising from a defect in an NEC semiconductor device, customers must incorporate sufficient safety measures in its design, such as redundancy, fire-containment, and anti-failure features.
  - NEC devices are classified into the following three quality grades:  
"Standard", "Special", and "Specific". The Specific quality grade applies only to devices developed based on a customer designated "quality assurance program" for a specific application. The recommended applications of a device depend on its quality grade, as indicated below. Customers must check the quality grade of each device 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 or medical equipment for life support, etc.
- The quality grade of NEC devices is "Standard" unless otherwise specified in NEC's Data Sheets or Data Books. If customers intend to use NEC devices for applications other than those specified for Standard quality grade, they should contact an NEC sales representative in advance.

# Regional Information

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

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

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

## **NEC Electronics Inc. (U.S.)**

Santa Clara, California  
Tel: 408-588-6000  
800-366-9782  
Fax: 408-588-6130  
800-729-9288

## **NEC do Brasil S.A.**

Electron Devices Division  
Guarulhos-SP, Brasil  
Tel: 11-6462-6810  
Fax: 11-6462-6829

## **NEC Electronics (Europe) GmbH**

Duesseldorf, Germany  
Tel: 0211-65 03 01  
Fax: 0211-65 03 327

### **• Sucursal en España**

Madrid, Spain  
Tel: 091-504 27 87  
Fax: 091-504 28 60

### **• Succursale Française**

Vélizy-Villacoublay, France  
Tel: 01-30-67 58 00  
Fax: 01-30-67 58 99

### **• Filiale Italiana**

Milano, Italy  
Tel: 02-66 75 41  
Fax: 02-66 75 42 99

### **• Branch The Netherlands**

Eindhoven, The Netherlands  
Tel: 040-244 58 45  
Fax: 040-244 45 80

### **• Branch Sweden**

Taeby, Sweden  
Tel: 08-63 80 820  
Fax: 08-63 80 388

### **• United Kingdom Branch**

Milton Keynes, UK  
Tel: 01908-691-133  
Fax: 01908-670-290

## **NEC Electronics Hong Kong Ltd.**

Hong Kong  
Tel: 2886-9318  
Fax: 2886-9022/9044

## **NEC Electronics Hong Kong Ltd.**

Seoul Branch  
Seoul, Korea  
Tel: 02-528-0303  
Fax: 02-528-4411

## **NEC Electronics Shanghai, Ltd.**

Shanghai, P.R. China  
Tel: 021-6841-1138  
Fax: 021-6841-1137

## **NEC Electronics Taiwan Ltd.**

Taipei, Taiwan  
Tel: 02-2719-2377  
Fax: 02-2719-5951

## **NEC Electronics Singapore Pte. Ltd.**

Novena Square, Singapore  
Tel: 253-8311  
Fax: 250-3583

## Major Revisions in This Edition (1/2)

Page	Description
Major revisions from U15836EJ1V0UD00 to U15836EJ2V0UD00	
Throughout	X1 input clock oscillation stabilization time $2^{12}/f_x, 2^{14}/f_x, 2^{15}/f_x, 2^{16}/f_x, 2^{17}/f_x \rightarrow 2^{11}/f_x, 2^{13}/f_x, 2^{14}/f_x, 2^{15}/f_x, 2^{16}/f_x$
p.78	Modification of <b>Figure 4-5 Block Diagram of P10</b>
p.89	Modification of <b>Table 4-3 Settings of Port Mode Register and Output Latch When Alternate-Function Is Used</b>
p.98	Modification of <b>Figure 5-6 Format of Oscillation Stabilization Time Counter Status Register (OSTC)</b>
p.99	Modification of <b>Figure 5-7 Format of Oscillation Stabilization Time Select Register (OSTS)</b>
p.108	Addition of <b>5.7 Clock Selection Flowchart and Register Settings</b>
p.216	Addition of <b>Remark to 12.1 Functions of Serial Interface UART6</b>
p.273	Addition of Reset to <b>Table 14-1 Interrupt Source List</b>
p.290	Modification of <b>Figure 15-2 Format of Oscillation Stabilization Time Counter Status Register (OSTC)</b>
p.291	Modification of <b>Figure 15-3 Format of Oscillation Stabilization Time Select Register (OSTS)</b>
p.368	Addition of <b>CHAPTER 25 RETRY</b>
Major revisions from U15836EJ2V0UD00 to U15836EJ2V1UD00 (1/2)	
p.59	Modification of reset value of the following register in <b>Table 3-5 Special Function Register List</b> <ul style="list-style-type: none"> <li>• Serial I/O shift register 10 (SIO10)</li> </ul>
p.60	Modification of manipulatable bit unit of the following registers in <b>Table 3-5 Special Function Register List</b> <ul style="list-style-type: none"> <li>• Oscillation stabilization time counter status register (OSTC)</li> <li>• Interrupt request flag register 1L (IF1L)</li> <li>• Interrupt mask flag register 1L (MK1L)</li> <li>• Priority specification flag register 1L (PR1L)</li> </ul>
p.61	
p.61	
p.61	
p.98	Modification of manipulatable bit unit in <b>5.3 (5) Oscillation stabilization time counter status register (OSTC)</b>
pp.104, 105	Modification of <b>Figure 5-11 Status Transition Diagram</b>
p.106	Modification of <b>Table 5-3 Relationship Between Operation Clocks in Each Operation Status</b>
p.106	Modification of <b>Table 5-4 Oscillation Control Flags and Clock Oscillation Status</b>
p.110	Modification of <b>Table 5-6 Clock and Register Settings</b>
p.114	Modification of reset value in <b>6.2 (2) 16-bit timer capture/compare register 000 (CR000)</b> and <b>(3) 16-bit timer capture/compare register 010 (CR010)</b>
p.119	Modification of manipulatable bit unit in <b>6.3 (4) Prescaler mode register 00 (PRM00)</b>
p.168	Modification of <b>Caution in 9.4.2 Watchdog timer operation when “Ring-OSC can be stopped by software” is selected by mask option</b>
pp.169 to 171	Modification of <b>9.4.3 Watchdog timer operation in STOP mode (when “Ring-OSC can be stopped by software” is selected by mask option)</b>
p.171	Addition of <b>9.4.4 Watchdog timer operation in HALT mode (when “Ring-OSC can be stopped by software” is selected by mask option)</b>
p.191	Addition of <b>(11) A/D converter sampling time and A/D conversion start delay time in 10.6 Cautions for A/D Converter</b>
p.259	Modification of reset value in <b>13.2 (2) Serial I/O shift register 10 (SIO10)</b>

The mark ★ shows major revised points.

## Major Revisions in This Edition (2/2)

Page	Description
Major revisions from U15836EJ2V0UD00 to U15836EJ2V1UD00 (2/2)	
p.290	Modification of manipulatable bit unit in <b>15.1.2 (1) Oscillation stabilization time counter status register (OSTC)</b>
p.292	Modification of A/D converter item in <b>Table 15-2 Operating Statuses in HALT mode</b>
pp.312, 313	Addition of <b>18.4 Cautions for Power-on-Clear Circuit</b>
p.317	Modification of <b>Figure 19-3 Format of Low-Voltage Detection Level Selection Register (LVIS)</b>
pp.322 to 325	Addition of <b>19.5 Cautions for Low-Voltage Detector</b>
p.351 p.353 pp.354 to 356 p.363 p.363 p.364 p.364  p.364 pp.365, 366	Modification of the following contents in <b>CHAPTER 23 ELECTRICAL SPECIFICATIONS (TARGET VALUES)</b> <ul style="list-style-type: none"> <li>• <b>Absolute Maximum Ratings</b></li> <li>• <b>X1 Oscillator Characteristics</b></li> <li>• <b>DC Characteristics</b></li> <li>• <b>A/D Converter Characteristics</b></li> <li>• <b>POC Circuit Characteristics</b></li> <li>• <b>LVI Circuit Characteristics</b></li> <li>• <b>Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics</b> (deletion of data retention supply current)</li> <li>• Deletion of <b>Ring-OSC Characteristics</b></li> <li>• <b>Flash Memory Programming Characteristics</b></li> </ul>
pp.368 to 370	Modification from <b>CHAPTER 25 RETRY</b> to <b>CHAPTER 25 CAUTIONS FOR WAIT</b>

The mark ★ shows major revised points.

# INTRODUCTION

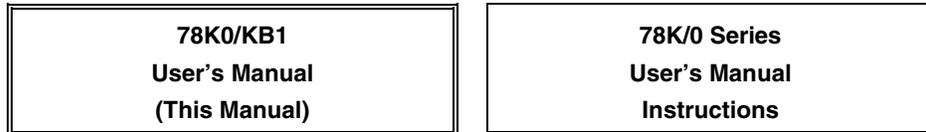
**Readers** This manual is intended for user engineers who wish to understand the functions of the 78K0/KB1 Series and design and develop application systems and programs for these devices.

The target products are as follows.

78K0/KB1 Series:  $\mu$ PD780101, 780102, 780103, and 78F0103

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

**Organization** The 78K0/KB1 Series manual is separated into two parts: this manual and the instructions edition (common to the 78K/0 Series).



- Pin functions
- Internal block functions
- Interrupts
- Other on-chip peripheral functions
- Electrical specifications
- CPU functions
- Instruction set
- Explanation of each instruction

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

- When using this manual as the manual for (A) products, (A1) products, and (A2) products:
  - Only the quality grade differs between standard products and (A), (A1), and (A2) products. Read the part number as follows.
    - $\mu$ PD780101 →  $\mu$ PD780101(A), 780101(A1), 780101(A2)
    - $\mu$ PD780102 →  $\mu$ PD780102(A), 780102(A1), 780102(A2)
    - $\mu$ PD780103 →  $\mu$ PD780103(A), 780103(A1), 780103(A2)
    - $\mu$ PD78F0103 →  $\mu$ PD78F0103(A), 78F0103(A1)
- To gain a general understanding of functions:
  - Read this manual in the order of the **CONTENTS**.
- How to interpret the register format:
  - For a bit number enclosed in square, the bit name is defined as a reserved word in the assembler, and is already defined in the header file named sfrbit.h in the C compiler.
- To check the details of a register when you know the register name.
  - Refer to **APPENDIX C REGISTER INDEX**.

- To know details of the 78K/0 Series instructions.  
→ Refer to the separate document **78K/0 Series Instructions User's Manual (U12326E)**.

**Caution** Examples in this manual employ the “standard” quality grade for general electronics. When using examples in this manual for the “special” quality grade, review the quality grade of each part and/or circuit actually used.

**Conventions**

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

**Related Documents**

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

**Documents Related to Devices**

Document Name	Document No.
78K0/KB1 User's Manual	This manual
78K/0 Series Instructions User's Manual	U12326E

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

Document Name	Document No.	
RA78K0 Assembler Package	Operation	U14445E
	Language	U14446E
	Structured Assembly Language	U11789E
CC78K0 C Compiler	Operation	U14297E
	Language	U14298E
SM78K0S, SM78K0 System Simulator Ver. 2.10 or Later	Operation (Windows™ Based)	U14611E
SM78K Series System Simulator Ver. 2.10 or Later	External Part User Open Interface Specifications	U15006E
ID78K0-NS Integrated Debugger Ver. 2.00 or Later	Operation (Windows Based)	U14379E
ID78K0 Integrated Debugger Windows Based	Reference	U11539E
	Guide	U11649E
RX78K0 Real-Time OS	Fundamentals	U11537E
	Installation	U11536E
Project Manager Ver. 3.12 or Later (Windows Based)		U14610E

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

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

Document Name	Document No.
IE-78K0-NS In-Circuit Emulator	U13731E
IE-78K0-NS-A In-Circuit Emulator	U14889E
IE-780148-NS-EM1 Emulation Board	To be prepared

**Documents Related to Flash Memory Programming**

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

**Other Documents**

Document Name	Document No.
SEMICONDUCTOR SELECTION GUIDE – Product & Packages –	X13769E
Semiconductor Device Mounting Technology Manual	C10535E
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

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

## CONTENTS

<b>CHAPTER 1 OUTLINE</b> .....	<b>25</b>
<b>1.1 Features</b> .....	<b>25</b>
<b>1.2 Applications</b> .....	<b>26</b>
<b>1.3 Ordering Information</b> .....	<b>27</b>
<b>1.4 Quality Grade</b> .....	<b>29</b>
<b>1.5 Pin Configuration (Top View)</b> .....	<b>30</b>
<b>1.6 78K0/Kxx Series Lineup</b> .....	<b>32</b>
<b>1.7 Block Diagram</b> .....	<b>33</b>
<b>1.8 Outline of Functions</b> .....	<b>34</b>
<b>CHAPTER 2 PIN FUNCTIONS</b> .....	<b>36</b>
<b>2.1 Pin Function List</b> .....	<b>36</b>
<b>2.2 Description of Pin Functions</b> .....	<b>38</b>
2.2.1 P00 to P03 (port 0) .....	38
2.2.2 P10 to P17 (port 1) .....	38
2.2.3 P20 to P23 (port 2) .....	39
2.2.4 P30 to P33 (port 3) .....	39
2.2.5 P120 (port 12) .....	39
2.2.6 P130 (port 13) .....	39
2.2.7 AVREF .....	40
2.2.8 AVSS .....	40
2.2.9 RESET .....	40
2.2.10 X1 and X2 .....	40
2.2.11 VDD .....	40
2.2.12 VSS .....	40
2.2.13 VPP (flash memory versions only) .....	40
2.2.14 IC (mask ROM versions only) .....	40
<b>2.3 Pin I/O Circuits and Recommended Connection of Unused Pins</b> .....	<b>41</b>
<b>CHAPTER 3 CPU ARCHITECTURE</b> .....	<b>43</b>
<b>3.1 Memory Space</b> .....	<b>43</b>
3.1.1 Internal program memory space .....	48
3.1.2 Internal data memory space .....	49
3.1.3 Special function register (SFR) area .....	49
3.1.4 Data memory addressing .....	50
<b>3.2 Processor Registers</b> .....	<b>54</b>
3.2.1 Control registers .....	54
3.2.2 General-purpose registers .....	57
3.2.3 Special Function Registers (SFRs) .....	58
<b>3.3 Instruction Address Addressing</b> .....	<b>62</b>
3.3.1 Relative addressing .....	62
3.3.2 Immediate addressing .....	63
3.3.3 Table indirect addressing .....	64
3.3.4 Register addressing .....	64
<b>3.4 Operand Address Addressing</b> .....	<b>65</b>

3.4.1	Implied addressing .....	65
3.4.2	Register addressing .....	66
3.4.3	Direct addressing .....	67
3.4.4	Short direct addressing .....	68
3.4.5	Special function register (SFR) addressing .....	69
3.4.6	Register indirect addressing .....	70
3.4.7	Based addressing .....	71
3.4.8	Based indexed addressing .....	72
3.4.9	Stack addressing .....	72
<b>CHAPTER 4 PORT FUNCTIONS .....</b>		<b>73</b>
<b>4.1</b>	<b>Port Functions .....</b>	<b>73</b>
<b>4.2</b>	<b>Port Configuration .....</b>	<b>74</b>
4.2.1	Port 0 .....	75
4.2.2	Port 1 .....	78
4.2.3	Port 2 .....	84
4.2.4	Port 3 .....	85
4.2.5	Port 12 .....	86
4.2.6	Port 13 .....	87
<b>4.3</b>	<b>Registers Controlling Port Function .....</b>	<b>87</b>
<b>4.4</b>	<b>Port Function Operations .....</b>	<b>92</b>
4.4.1	Writing to I/O port .....	92
4.4.2	Reading from I/O port .....	92
4.4.3	Operations on I/O port .....	92
<b>CHAPTER 5 CLOCK GENERATOR .....</b>		<b>93</b>
<b>5.1</b>	<b>Functions of Clock Generator .....</b>	<b>93</b>
<b>5.2</b>	<b>Configuration of Clock Generator .....</b>	<b>93</b>
<b>5.3</b>	<b>Registers Controlling Clock Generator .....</b>	<b>95</b>
<b>5.4</b>	<b>System Clock Oscillator .....</b>	<b>100</b>
5.4.1	X1 oscillator .....	100
5.4.2	Ring-OSC oscillator .....	102
5.4.3	Prescaler .....	102
<b>5.5</b>	<b>Clock Generator Operation .....</b>	<b>102</b>
<b>5.6</b>	<b>Time Required to Switch Between Ring-OSC Clock and X1 Input Clock .....</b>	<b>107</b>
★ <b>5.7</b>	<b>Clock Selection Flowchart and Register Settings .....</b>	<b>108</b>
5.7.1	Changing to X1 input clock from Ring-OSC clock .....	108
5.7.2	Changing from X1 input clock to Ring-OSC clock .....	109
5.7.3	Register settings .....	110
<b>CHAPTER 6 16-BIT TIMER/EVENT COUNTER 00 .....</b>		<b>111</b>
<b>6.1</b>	<b>Functions of 16-Bit Timer/Event Counter 00 .....</b>	<b>111</b>
<b>6.2</b>	<b>Configuration of 16-Bit Timer/Event Counter 00 .....</b>	<b>112</b>
<b>6.3</b>	<b>Registers Controlling 16-Bit Timer/Event Counter 00 .....</b>	<b>115</b>
<b>6.4</b>	<b>Operation of 16-Bit Timer/Event Counter 00 .....</b>	<b>121</b>
6.4.1	Interval timer operation .....	121
6.4.2	PPG output operations .....	122
6.4.3	Pulse width measurement operations .....	124

6.4.4	External event counter operation .....	131
6.4.5	Square-wave output operation .....	132
<b>6.5</b>	<b>Cautions on 16-Bit Timer/Event Counter 00 .....</b>	<b>134</b>
<b>CHAPTER 7</b>	<b>8-BIT TIMER/EVENT COUNTER 50 .....</b>	<b>137</b>
<b>7.1</b>	<b>Functions of 8-Bit Timer/Event Counter 50 .....</b>	<b>137</b>
<b>7.2</b>	<b>Configuration of 8-Bit Timer/Event Counter 50 .....</b>	<b>138</b>
<b>7.3</b>	<b>Registers Controlling 8-Bit Timer/Event Counter 50 .....</b>	<b>139</b>
<b>7.4</b>	<b>Operations of 8-Bit Timer/Event Counter 50 .....</b>	<b>142</b>
7.4.1	Operation as interval timer .....	142
7.4.2	Operation as external event counter .....	144
7.4.3	Operation as square-wave output .....	145
7.4.4	Operation as PWM output .....	146
<b>7.5</b>	<b>Cautions on 8-Bit Timer/Event Counter 50 .....</b>	<b>148</b>
<b>CHAPTER 8</b>	<b>8-BIT TIMERS H0 AND H1 .....</b>	<b>149</b>
<b>8.1</b>	<b>Functions of 8-Bit Timers H0 and H1 .....</b>	<b>149</b>
<b>8.2</b>	<b>Configuration of 8-Bit Timers H0 and H1 .....</b>	<b>149</b>
<b>8.3</b>	<b>Registers Controlling 8-Bit Timers H0 and H1 .....</b>	<b>151</b>
<b>8.4</b>	<b>Operation of 8-Bit Timers H0 and H1 .....</b>	<b>154</b>
8.4.1	Operation as interval timer .....	154
8.4.2	Operation as PWM pulse generator .....	157
<b>CHAPTER 9</b>	<b>WATCHDOG TIMER .....</b>	<b>163</b>
<b>9.1</b>	<b>Functions of Watchdog Timer .....</b>	<b>163</b>
<b>9.2</b>	<b>Configuration of Watchdog Timer .....</b>	<b>164</b>
<b>9.3</b>	<b>Registers Controlling Watchdog Timer .....</b>	<b>165</b>
<b>9.4</b>	<b>Operation of Watchdog Timer .....</b>	<b>167</b>
9.4.1	Watchdog timer operation when “Ring-OSC cannot be stopped” is selected by mask option .....	167
9.4.2	Watchdog timer operation when “Ring-OSC can be stopped by software” is selected by mask option .....	168
9.4.3	Watchdog timer operation in STOP mode (when “Ring-OSC can be stopped by software” is selected by mask option) .....	169
9.4.4	Watchdog timer operation in HALT mode (when “Ring-OSC can be stopped by software” is selected by mask option) .....	171
<b>CHAPTER 10</b>	<b>A/D CONVERTER .....</b>	<b>172</b>
<b>10.1</b>	<b>Function of A/D Converter .....</b>	<b>172</b>
<b>10.2</b>	<b>A/D Converter Configuration .....</b>	<b>174</b>
<b>10.3</b>	<b>Registers Controlling A/D Converter .....</b>	<b>176</b>
<b>10.4</b>	<b>A/D Converter Operations .....</b>	<b>179</b>
10.4.1	Basic operations of A/D converter .....	179
10.4.2	Input voltage and conversion results .....	181
10.4.3	A/D converter operation mode .....	182
<b>10.5</b>	<b>How to Read A/D Converter Characteristics Table .....</b>	<b>185</b>
<b>10.6</b>	<b>Cautions for A/D Converter .....</b>	<b>187</b>
<b>CHAPTER 11</b>	<b>SERIAL INTERFACE UART0 (μPD780102, 780103, AND 78F0103 ONLY) .....</b>	<b>192</b>

★

11.1	Functions of Serial Interface UART0.....	192
11.2	Configuration of Serial Interface UART0.....	193
11.3	Registers Controlling Serial Interface UART0.....	196
11.4	Operation of Serial Interface UART0 .....	200
11.4.1	Operation stop mode.....	200
11.4.2	Asynchronous serial interface (UART) mode .....	201
11.4.3	Dedicated baud rate generator.....	209
<b>CHAPTER 12 SERIAL INTERFACE UART6 .....</b>		<b>215</b>
12.1	Functions of Serial Interface UART6.....	215
12.2	Configuration of Serial Interface UART6.....	219
12.3	Registers Controlling Serial Interface UART6.....	222
12.4	Operation of Serial Interface UART6 .....	230
12.4.1	Operation stop mode.....	230
12.4.2	Asynchronous serial interface (UART) mode .....	231
12.4.3	Dedicated baud rate generator.....	249
<b>CHAPTER 13 SERIAL INTERFACE CSI10 .....</b>		<b>258</b>
13.1	Functions of Serial Interface CSI10.....	258
13.2	Configuration of Serial Interface CSI10.....	258
13.3	Registers Controlling Serial Interface CSI10.....	260
13.4	Operation of Serial Interface CSI10 .....	262
13.4.1	Operation stop mode.....	262
13.4.2	3-wire serial I/O mode .....	262
<b>CHAPTER 14 INTERRUPT FUNCTIONS.....</b>		<b>272</b>
14.1	Interrupt Function Types .....	272
14.2	Interrupt Sources and Configuration .....	272
14.3	Registers Controlling Interrupt Function.....	275
14.4	Interrupt Servicing Operations .....	281
14.4.1	Maskable interrupt acknowledgement.....	281
14.4.2	Software interrupt request acknowledgement.....	283
14.4.3	Multiple interrupt servicing .....	284
14.4.4	Interrupt request hold .....	287
<b>CHAPTER 15 STANDBY FUNCTION .....</b>		<b>288</b>
15.1	Standby Function and Configuration .....	288
15.1.1	Standby function .....	288
15.1.2	Registers controlling standby function.....	290
15.2	Standby Function Operation .....	292
15.2.1	HALT mode.....	292
15.2.2	STOP mode .....	295
<b>CHAPTER 16 RESET FUNCTION.....</b>		<b>298</b>
16.1	Register for Confirming Reset Source.....	303
<b>CHAPTER 17 CLOCK MONITOR .....</b>		<b>304</b>
17.1	Functions of Clock Monitor.....	304

	17.2 Configuration of Clock Monitor .....	304
	17.3 Registers Controlling Clock Monitor.....	305
	17.4 Operation of Clock Monitor.....	306
	<b>CHAPTER 18 POWER-ON-CLEAR CIRCUIT .....</b>	<b>310</b>
	18.1 Functions of Power-on-Clear Circuit.....	310
	18.2 Configuration of Power-on-Clear Circuit .....	311
	18.3 Operation of Power-on-Clear Circuit.....	311
★	18.4 Cautions for Power-on-Clear Circuit.....	312
	<b>CHAPTER 19 LOW-VOLTAGE DETECTOR .....</b>	<b>314</b>
	19.1 Functions of Low-Voltage Detector.....	314
	19.2 Configuration of Low-Voltage Detector .....	314
	19.3 Registers Controlling Low-Voltage Detector .....	315
	19.4 Operation of Low-Voltage Detector.....	318
★	19.5 Cautions for Low-Voltage Detector.....	322
	<b>CHAPTER 20 MASK OPTIONS .....</b>	<b>326</b>
	<b>CHAPTER 21 <math>\mu</math>PD78F0103.....</b>	<b>327</b>
	21.1 Internal Memory Size Switching Register.....	328
	21.2 Flash Memory Programming.....	329
	21.2.1 Selection of communication mode .....	329
	21.2.2 Flash memory programming function.....	330
	21.2.3 Connecting Flashpro III.....	331
	21.2.4 Connection on adapter for flash memory writing.....	333
	<b>CHAPTER 22 INSTRUCTION SET .....</b>	<b>338</b>
	22.1 Conventions Used in Operation List.....	338
	22.1.1 Operand identifiers and specification methods .....	338
	22.1.2 Description of operation column.....	339
	22.1.3 Description of flag operation column.....	339
	22.2 Operation List.....	340
	22.3 Instructions Listed by Addressing Type .....	348
	<b>CHAPTER 23 ELECTRICAL SPECIFICATIONS (TARGET VALUES).....</b>	<b>351</b>
	<b>CHAPTER 24 PACKAGE DRAWING .....</b>	<b>367</b>
★	<b>CHAPTER 25 CAUTIONS FOR WAIT .....</b>	<b>368</b>
	25.1 Cautions for Wait.....	368
	25.2 Peripheral Hardware That Generates Wait .....	369
	25.3 Example of Wait Occurrence .....	370
	<b>APPENDIX A DEVELOPMENT TOOLS .....</b>	<b>371</b>
	A.1 Software Package.....	373
	A.2 Language Processing Software.....	374
	A.3 Flash Memory Writing Tools .....	375

<b>A.4 Debugging Tools .....</b>	<b>376</b>
A.4.1 Hardware .....	376
A.4.2 Software.....	377
<b>APPENDIX B EMBEDDED SOFTWARE .....</b>	<b>378</b>
<b>APPENDIX C REGISTER INDEX .....</b>	<b>379</b>
C.1 Register Index (In Alphabetical Order with Respect to Register Names).....	379
C.2 Register Index (In Alphabetical Order with Respect to Register Symbol) .....	382
<b>★ APPENDIX D REVISION HISTORY .....</b>	<b>385</b>

## LIST OF FIGURES (1/6)

Figure No.	Title	Page
2-1	Pin I/O Circuit List .....	42
3-1	Memory Map ( $\mu$ PD780101).....	44
3-2	Memory Map ( $\mu$ PD780102).....	45
3-3	Memory Map ( $\mu$ PD780103).....	46
3-4	Memory Map ( $\mu$ PD78F0103) .....	47
3-5	Data Memory Addressing ( $\mu$ PD780101) .....	50
3-6	Data Memory Addressing ( $\mu$ PD780102) .....	51
3-7	Data Memory Addressing ( $\mu$ PD780103) .....	52
3-8	Data Memory Addressing ( $\mu$ PD78F0103).....	53
3-9	Format of Program Counter .....	54
3-10	Format of Program Status Word .....	54
3-11	Stack Pointer Format .....	56
3-12	Data to Be Saved to Stack Memory .....	56
3-13	Data to Be Restored from Stack Memory .....	56
3-14	Configuration of General-Purpose Registers .....	57
4-1	Port Types .....	73
4-2	Block Diagram of P00 .....	75
4-3	Block Diagram of P01 .....	76
4-4	Block Diagram of P02 and P03.....	77
4-5	Block Diagram of P10 .....	78
4-6	Block Diagram of P11 and P14.....	79
4-7	Block Diagram of P12 .....	80
4-8	Block Diagram of P13 .....	81
4-9	Block Diagram of P15 .....	82
4-10	Block Diagram of P16 and P17 .....	83
4-11	Block Diagram of P20 to P23.....	84
4-12	Block Diagram of P30 to P33.....	85
4-13	Block Diagram of P120 .....	86
4-14	Block Diagram of P130 .....	87
4-15	Format of Port Mode Register.....	88
4-16	Format of Pull-up Resistor Option Register .....	90
4-17	Format of Input Switch Control Register (ISC).....	91
5-1	Block Diagram of Clock Generator .....	94
5-2	Format of Processor Clock Control Register (PCC).....	95
5-3	Format of Ring-OSC Mode Register (RCM) .....	96
5-4	Format of Main Clock Mode Register (MCM).....	97
5-5	Format of Main OSC Control Register (MOC) .....	98
5-6	Format of Oscillation Stabilization Time Counter Status Register (OSTC) .....	98
5-7	Format of Oscillation Stabilization Time Select Register (OSTS) .....	99
5-8	External Circuit of X1 Oscillator .....	100
5-9	Examples of Incorrect Resonator Connection.....	101

## LIST OF FIGURES (2/6)

Figure No.	Title	Page
5-10	Timing Diagram of CPU Default Start Using Ring-OSC .....	103
5-11	Status Transition Diagram.....	104
★ 5-12	Changing to X1 Input Clock from Ring-OSC Clock (Flowchart) .....	108
★ 5-13	Changing from X1 Input Clock to Ring-OSC Clock (Flowchart) .....	109
6-1	Block Diagram of 16-Bit Timer/Event Counter 00 .....	112
6-2	Format of 16-Bit Timer Mode Control Register 00 (TMC00) .....	116
6-3	Format of Capture/Compare Control Register 00 (CRC00) .....	117
6-4	Format of 16-Bit Timer Output Control Register 00 (TOC00).....	118
6-5	Format of Prescaler Mode Register 00 (PRM00) .....	119
6-6	Format of Port Mode Register 0 (PM0) .....	120
6-7	Control Register Settings for Interval Timer Operation.....	121
6-8	Interval Timer Configuration Diagram .....	122
6-9	Timing of Interval Timer Operation.....	122
6-10	Control Register Settings for PPG Output Operation .....	123
6-11	Control Register Settings for Pulse Width Measurement with Free-Running Counter and One Capture Register .....	124
6-12	Configuration Diagram for Pulse Width Measurement by Free-Running Counter.....	125
6-13	Timing of Pulse Width Measurement Operation by Free-Running Counter and One Capture Register (with Both Edges Specified) .....	125
6-14	Control Register Settings for Measurement of Two Pulse Widths with Free-Running Counter.....	126
6-15	CR010 Capture Operation with Rising Edge Specified .....	127
6-16	Timing of Pulse Width Measurement Operation with Free-Running Counter (with Both Edges Specified) .....	127
6-17	Control Register Settings for Pulse Width Measurement with Free-Running Counter and Two Capture Registers .....	128
6-18	Timing of Pulse Width Measurement Operation by Free-Running Counter and Two Capture Registers (with Rising Edge Specified).....	129
6-19	Control Register Settings for Pulse Width Measurement by Means of Restart .....	130
6-20	Timing of Pulse Width Measurement Operation by Means of Restart (with Rising Edge Specified) .....	130
6-21	Control Register Settings in External Event Counter Mode.....	131
6-22	Configuration Diagram of External Event Counter .....	132
6-23	External Event Counter Operation Timing (with Rising Edge Specified) .....	132
6-24	Control Register Settings in Square-Wave Output Mode .....	133
6-25	Square-Wave Output Operation Timing .....	133
6-26	Start Timing of 16-Bit Timer Counter 00 (TM00) .....	134
6-27	Timings After Change of Compare Register During Timer Count Operation.....	134
6-28	Capture Register Data Retention Timing .....	135
6-29	Operation Timing of OVF00 Flag .....	135
7-1	Block Diagram of 8-Bit Timer/Event Counter 50 .....	137
7-2	Format of Timer Clock Selection Register 50 (TCL50).....	139
7-3	Format of 8-Bit Timer Mode Control Register 50 (TMC50) .....	140
7-4	Format of Port Mode Register 1 (PM1) .....	141

## LIST OF FIGURES (3/6)

Figure No.	Title	Page
7-5	Interval Timer Operation Timing .....	142
7-6	External Event Counter Operation Timing (with Rising Edge Specified).....	144
7-7	Square-Wave Output Operation Timing.....	145
7-8	PWM Output Operation Timing.....	147
7-9	Timing of Operation with CR50 Changed .....	148
7-10	8-Bit Timer Counter 50 Start Timing .....	148
8-1	Block Diagram of 8-Bit Timer H0 .....	149
8-2	Block Diagram of 8-Bit Timer H1 .....	150
8-3	Format of 8-Bit Timer H Mode Register 0 (TMHMD0).....	152
8-4	Format of 8-Bit Timer H Mode Register 1 (TMHMD1).....	153
8-5	Register Setting in Interval Timer Mode.....	154
8-6	Timing of Interval Timer Operation .....	155
8-7	Register Setting in PWM Pulse Generator Mode.....	157
8-8	Operation Timing in PWM Pulse Generator Mode.....	159
9-1	Block Diagram of Watchdog Timer .....	165
9-2	Format of Watchdog Timer Mode Register (WDTM).....	166
9-3	Format of Watchdog Timer Enable Register (WDTE).....	167
9-4	Operation in STOP Mode (CPU Clock and WDT Operation Clock: X1 Input Clock).....	169
9-5	Operation in STOP Mode (CPU Clock: X1 Input Clock, WDT Operation Clock: Ring-OSC Clock).....	169
9-6	Operation in STOP Mode (CPU Clock: Ring-OSC Clock, WDT Operation Clock: X1 Input Clock).....	170
9-7	Operation in STOP Mode (CPU Clock and WDT Operation Clock: Ring-OSC Clock).....	171
★ 9-8	Operation in HALT Mode .....	171
10-1	Block Diagram of A/D Converter.....	173
10-2	Block Diagram of Power-Fail Detection Function.....	173
10-3	Format of A/D Conversion Register (ADCR).....	174
10-4	Format of A/D Converter Mode Register (ADM) .....	176
10-5	Timing Chart When Boost Reference Voltage Generator Is Used.....	177
10-6	Format of Analog Input Channel Specification Register (ADS).....	178
10-7	Format of Power-Fail Comparison Mode Register (PFM) .....	178
10-8	Format of Power-Fail Comparison Threshold Register (PFT).....	179
10-9	Basic Operation of A/D Converter.....	180
10-10	Relationship Between Analog Input Voltage and A/D Conversion Result.....	181
10-11	A/D Conversion Operation .....	182
10-12	Power-Fail Detection (When PFEN = 1 and PFCM = 0) .....	183
10-13	Overall Error .....	185
10-14	Quantization Error.....	185
10-15	Zero-Scale Error .....	186
10-16	Full-Scale Error.....	186
10-17	Integral Linearity Error .....	186
10-18	Differential Linearity Error .....	186
10-19	Example of Method of Reducing Current Consumption in Standby Mode .....	187

## LIST OF FIGURES (4/6)

Figure No.	Title	Page
10-20	Storing Conversion Result in ADCR and Timing of Data Read from ADCR.....	188
10-21	Analog Input Pin Connection.....	189
10-22	Timing of A/D Conversion End Interrupt Request Generation.....	190
★ 10-23	Timing of A/D Converter Sampling and A/D Conversion Start Delay.....	191
11-1	Block Diagram of Serial Interface UART0.....	194
11-2	Format of Asynchronous Serial Interface Operation Mode Register 0 (ASIM0).....	196
11-3	Format of Asynchronous Serial Interface Reception Error Status Register 0 (ASIS0).....	198
11-4	Format of Baud Rate Generator Control Register 0 (BRGC0).....	199
11-5	Format of Normal UART Transmit/Receive Data.....	204
11-6	Example of Normal UART Transmit/Receive Data Format.....	204
11-7	Normal Transmission Completion Interrupt Request Timing.....	206
11-8	Reception Completion Interrupt Request Timing.....	207
11-9	Noise Filter Circuit.....	208
11-10	Configuration of Baud Rate Generator.....	209
11-11	Permissible Baud Rate Range During Reception.....	213
12-1	LIN Transmission Operation.....	216
12-2	LIN Reception Operation.....	217
12-3	Port Configuration for LIN Reception Operation.....	218
12-4	Block Diagram of Serial Interface UART6.....	220
12-5	Format of Asynchronous Serial Interface Operation Mode Register 6 (ASIM6).....	222
12-6	Format of Asynchronous Serial Interface Reception Error Status Register 6 (ASIS6).....	224
12-7	Format of Asynchronous Serial Interface Transmission Status Register 6 (ASIF6).....	225
12-8	Format of Clock Selection Register 6 (CKSR6).....	226
12-9	Format of Baud Rate Generator Control Register 6 (BRGC6).....	227
12-10	Format of Asynchronous Serial Interface Control Register 6 (ASICL6).....	228
12-11	Format of Normal UART Transmit/Receive Data.....	237
12-12	Example of Normal UART Transmit/Receive Data Format.....	238
12-13	Normal Transmission Completion Interrupt Request Timing.....	240
12-14	Processing Flow of Continuous Transmission.....	242
12-15	Timing of Starting Continuous Transmission.....	243
12-16	Timing of Ending Continuous Transmission.....	244
12-17	Reception Completion Interrupt Request Timing.....	245
12-18	Reception Error Interrupt.....	246
12-19	Noise Filter Circuit.....	247
12-20	SBF Transmission.....	247
12-21	SBF Reception.....	248
12-22	Configuration of Baud Rate Generator.....	250
12-23	Permissible Baud Rate Range During Reception.....	255
12-24	Transfer Rate During Continuous Transmission.....	257
13-1	Block Diagram of Serial Interface CSI10.....	259
13-2	Format of Serial Operation Mode Register 10 (CSIM10).....	260

## LIST OF FIGURES (5/6)

Figure No.	Title	Page
13-3	Format of Serial Clock Selection Register 10 (CSIC10).....	261
13-4	Timing in 3-Wire Serial I/O Mode.....	266
13-5	Timing of Clock/Data Phase .....	268
13-6	Output Operation of First Bit .....	269
13-7	Output Value of SO10 Pin (Last Bit) .....	270
14-1	Basic Configuration of Interrupt Function.....	274
14-2	Format of Interrupt Request Flag Register (IF0L, IF0H, IF1L) .....	276
14-3	Format of Interrupt Mask Flag Register (MK0L, MK0H, MK1L) .....	277
14-4	Format of Priority Specification Flag Register (PR0L, PR0H, PR1L) .....	278
14-5	Format of External Interrupt Rising Edge Enable Register (EGP) and External Interrupt Falling Edge Enable Register (EGN) .....	279
14-6	Format of Program Status Word .....	280
14-7	Interrupt Request Acknowledgement Processing Algorithm .....	282
14-8	Interrupt Request Acknowledgement Timing (Minimum Time) .....	283
14-9	Interrupt Request Acknowledgement Timing (Maximum Time) .....	283
14-10	Examples of Multiple Interrupt Servicing.....	285
14-11	Interrupt Request Hold.....	287
15-1	Operation Timing When STOP Mode Is Released .....	289
15-2	Format of Oscillation Stabilization Time Counter Status Register (OSTC) .....	290
15-3	Format of Oscillation Stabilization Time Select Register (OSTS) .....	291
15-4	HALT Mode Release by Interrupt Request Generation.....	293
15-5	HALT Mode Release by $\overline{\text{RESET}}$ Input.....	294
15-6	STOP Mode Release by Interrupt Request Generation .....	296
15-7	STOP Mode Release by $\overline{\text{RESET}}$ Input .....	297
16-1	Block Diagram of Reset Function .....	299
16-2	Timing of Reset by $\overline{\text{RESET}}$ Input.....	300
16-3	Timing of Reset Due to Watchdog Timer Overflow .....	300
16-4	Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input.....	300
16-5	Format of Reset Control Flag Register (RESF) .....	303
17-1	Block Diagram of Clock Monitor.....	304
17-2	Format of Clock Monitor Mode Register (CLM) .....	305
17-3	Timing of Clock Monitor .....	307
18-1	Block Diagram of Power-on-Clear Circuit .....	311
18-2	Timing of Internal Reset Signal Generation in Power-on-Clear Circuit .....	311
★ 18-3	Example of Software Processing After Release of Reset .....	312
19-1	Block Diagram of Low-Voltage Detector .....	314
19-2	Format of Low-Voltage Detection Register (LVIM) .....	316
19-3	Format of Low-Voltage Detection Level Selection Register (LVIS).....	317

## LIST OF FIGURES (6/6)

Figure No.	Title	Page
19-4	Timing of Low-Voltage Detector Internal Reset Signal Generation .....	319
19-5	Timing of Low-Voltage Detector Interrupt Signal Generation .....	321
★ 19-6	Example of Software Processing After Release of Reset .....	323
★ 19-7	Example of Software Processing of LVI Interrupt.....	325
21-1	Format of Internal Memory Size Switching Register (IMS).....	328
21-2	Communication Mode Selection Format .....	330
21-3	Connection of Flashpro III in 3-Wire Serial I/O Mode .....	331
21-4	Connection of Flashpro III in 3-Wire Serial I/O Mode (Using Handshake) .....	331
21-5	Connection of Flashpro III in UART (UART0) Mode.....	332
21-6	Connection of Flashpro III in UART (UART0) Mode (Using Handshake) .....	332
21-7	Connection of Flashpro III in UART (UART6) Mode.....	332
21-8	Example of Wiring Adapter for Flash Memory Writing in 3-Wire Serial I/O Mode .....	333
21-9	Example of Wiring Adapter for Flash Memory Writing in 3-Wire Serial I/O Mode (Using Handshake)....	334
21-10	Example of Wiring Adapter for Flash Memory Writing in UART Mode (UART0) .....	335
21-11	Example of Wiring Adapter for Flash Memory Writing in UART Mode (UART0) (Using Handshake) .....	336
21-12	Example of Wiring Adapter for Flash Memory Writing in UART Mode (UART6) .....	337
A-1	Development Tool Configuration.....	372

## LIST OF TABLES (1/2)

Table No.	Title	Page
1-1	Flash Memory Versions Corresponding to Mask Options of Mask ROM Versions .....	28
2-1	Pin I/O Circuit Types.....	41
3-1	Internal Memory Size Switching Register (IMS) Set Value .....	43
3-2	Internal Memory Capacity .....	48
3-3	Vector Table .....	48
3-4	Internal High-Speed RAM Capacity .....	49
3-5	Special Function Register List .....	59
4-1	Port Functions.....	74
4-2	Port Configuration .....	74
4-3	Settings of Port Mode Register and Output Latch When Alternate-Function Is Used .....	89
5-1	Configuration of Clock Generator .....	93
5-2	Relationship Between CPU Clock and Minimum Instruction Execution Time .....	96
5-3	Relationship Between Operation Clocks in Each Operation Status .....	106
5-4	Oscillation Control Flags and Clock Oscillation Status.....	106
5-5	Maximum Time Required to Switch Between Ring-OSC Clock and X1 Input Clock .....	107
★ 5-6	Clock and Register Settings.....	110
6-1	Configuration of 16-Bit Timer/Event Counter 00 .....	112
6-2	TI000/P00 Pin Valid Edge and CR000, CR010 Capture Trigger.....	113
6-3	TI010/TO00/P01 Pin Valid Edge and CR000 Capture Trigger.....	113
7-1	Configuration of 8-Bit Timer/Event Counter 50 .....	138
8-1	Configuration of 8-Bit Timers H0 and H1 .....	149
9-1	Loop Detection Time of Watchdog Timer.....	163
9-2	Mask Option Setting and Watchdog Timer Operation Mode.....	164
9-3	Configuration of Watchdog Timer .....	164
10-1	Configuration of A/D Converter.....	174
10-2	Settings of ADCS and ADCE .....	177
★ 10-3	A/D Converter Sampling Time and A/D Conversion Start Delay Time (ADM Set Value) .....	191
11-1	Configuration of Serial Interface UART0.....	193
11-2	Cause of Reception Error .....	208
11-3	Set Data of Baud Rate Generator.....	212
11-4	Maximum/Minimum Permissible Baud Rate Error .....	214
12-1	Configuration of Serial Interface UART6.....	219
12-2	Write Processing and Writing to TXB6 During Execution of Continuous Transmission .....	241

## LIST OF TABLES (2/2)

Table No.	Title	Page
12-3	Cause of Reception Error.....	246
12-4	Set Data of Baud Rate Generator .....	254
12-5	Maximum/Minimum Permissible Baud Rate Error.....	256
13-1	Configuration of Serial Interface CSI10.....	258
13-2	SO10 Pin Status .....	271
14-1	Interrupt Source List.....	273
14-2	Flags Corresponding to Interrupt Request Sources .....	275
14-3	Time from Generation of Maskable Interrupt Until Servicing.....	281
14-4	Interrupt Request Enabled for Multiple Interrupt During Interrupt Servicing .....	284
15-1	Relationship Between HALT and STOP Modes and Clock.....	288
15-2	Operating Statuses in HALT Mode.....	292
15-3	Operation After HALT Mode Release.....	294
15-4	Operating Statuses in STOP Mode.....	295
15-5	Operation After STOP Mode Release .....	297
16-1	Hardware Statuses After Reset.....	301
16-2	RESF Status When Reset Request Is Generated.....	303
17-1	Configuration of Clock Monitor.....	304
17-2	Operation Status of Clock Monitor (When CLME =1).....	306
20-1	Flash Memory Versions Supporting Mask Options of Mask ROM Versions.....	326
21-1	Differences Between $\mu$ PD78F0103 and Mask ROM Versions .....	327
21-2	Internal Memory Size Switching Register Settings.....	328
21-3	Communication Mode List.....	329
21-4	Main Functions of Flash Memory Programming.....	330
22-1	Operand Identifiers and Specification Methods.....	338
★ 25-1	Registers That Generate Wait and Number of CPU Wait Clocks.....	369
★ 25-2	Number of Wait Clocks and Number of Execution Clocks on Occurrence of Wait (A/D Converter) .....	370

## CHAPTER 1 OUTLINE

### 1.1 Features

- ROM, RAM capacities

Item \ Part Number	Program Memory (ROM)		Data Memory
			(Internal High-Speed RAM)
$\mu$ PD780101	Mask ROM	8 KB	512 bytes
$\mu$ PD780102		16 KB	768 bytes
$\mu$ PD780103		24 KB	
$\mu$ PD78F0103	Flash memory	24 KB <sup>Note</sup>	

**Note** The internal flash memory and internal high-speed RAM capacities can be changed using the internal memory size switching register (IMS).

- On-chip power-on-clear (POC) circuit and low-voltage detector (LVI)
- Short startup is possible via the CPU default start using the on-chip Ring-OSC
- On-chip clock monitor function using on-chip Ring-OSC
- On-chip watchdog timer (operable with Ring-OSC clock)
- On-chip UART supporting LIN (Local Interconnect Network) bus
- Minimum instruction execution time can be changed from high speed (0.2  $\mu$ s: @ 10 MHz operation with X1 input clock) to low-speed (3.2  $\mu$ s: @ 10 MHz operation with X1 input clock)
- I/O ports: 22
- Timer: 5 channels
- Serial interface
  - UART: 1 channel
  - CSI1/UART<sup>Note</sup>: 1 channel ( $\mu$ PD780101 only, CSI1: 1 channel)
- 10-bit resolution A/D converter: 4 channels
- Supply voltage:  $V_{DD} = 2.7$  to 5.5 V

★ **Note** Select either of the functions of these alternate-function pins.

## 1.2 Applications

- Automotive equipment
  - System control for body electricals (power windows, keyless entry reception, etc.)
  - Sub-microcontrollers for control
- Home audio, car audio
- AV equipment
- PC peripheral equipment (keyboards, etc.)
- Household electrical appliances
  - Outdoor air conditioner units
  - Microwave ovens, electric rice cookers
- Industrial equipment
  - Pumps
  - Vending machines
  - FA

## 1.3 Ordering Information

Part Number	Package	Internal ROM
$\mu$ PD780101MC-xxx-5A4	30-pin plastic SSOP (7.62 mm (300))	Mask ROM
$\mu$ PD780102MC-xxx-5A4	30-pin plastic SSOP (7.62 mm (300))	Mask ROM
$\mu$ PD780103MC-xxx-5A4	30-pin plastic SSOP (7.62 mm (300))	Mask ROM
$\mu$ PD780101MC(A)-xxx-5A4	30-pin plastic SSOP (7.62 mm (300))	Mask ROM
$\mu$ PD780102MC(A)-xxx-5A4	30-pin plastic SSOP (7.62 mm (300))	Mask ROM
$\mu$ PD780103MC(A)-xxx-5A4	30-pin plastic SSOP (7.62 mm (300))	Mask ROM
$\mu$ PD780101MC(A1)-xxx-5A4	30-pin plastic SSOP (7.62 mm (300))	Mask ROM
$\mu$ PD780102MC(A1)-xxx-5A4	30-pin plastic SSOP (7.62 mm (300))	Mask ROM
$\mu$ PD780103MC(A1)-xxx-5A4	30-pin plastic SSOP (7.62 mm (300))	Mask ROM
$\mu$ PD780101MC(A2)-xxx-5A4	30-pin plastic SSOP (7.62 mm (300))	Mask ROM
$\mu$ PD780102MC(A2)-xxx-5A4	30-pin plastic SSOP (7.62 mm (300))	Mask ROM
$\mu$ PD780103MC(A2)-xxx-5A4	30-pin plastic SSOP (7.62 mm (300))	Mask ROM
$\mu$ PD78F0103M1MC-5A4	30-pin plastic SSOP (7.62 mm (300))	Flash memory
$\mu$ PD78F0103M2MC-5A4	30-pin plastic SSOP (7.62 mm (300))	Flash memory
$\mu$ PD78F0103M3MC-5A4	30-pin plastic SSOP (7.62 mm (300))	Flash memory
$\mu$ PD78F0103M4MC-5A4	30-pin plastic SSOP (7.62 mm (300))	Flash memory
$\mu$ PD78F0103M5MC-5A4	30-pin plastic SSOP (7.62 mm (300))	Flash memory
$\mu$ PD78F0103M6MC-5A4	30-pin plastic SSOP (7.62 mm (300))	Flash memory
$\mu$ PD78F0103M1MC(A)-5A4	30-pin plastic SSOP (7.62 mm (300))	Flash memory
$\mu$ PD78F0103M2MC(A)-5A4	30-pin plastic SSOP (7.62 mm (300))	Flash memory
$\mu$ PD78F0103M3MC(A)-5A4	30-pin plastic SSOP (7.62 mm (300))	Flash memory
$\mu$ PD78F0103M4MC(A)-5A4	30-pin plastic SSOP (7.62 mm (300))	Flash memory
$\mu$ PD78F0103M5MC(A)-5A4	30-pin plastic SSOP (7.62 mm (300))	Flash memory
$\mu$ PD78F0103M6MC(A)-5A4	30-pin plastic SSOP (7.62 mm (300))	Flash memory
$\mu$ PD78F0103M1MC(A1)-5A4	30-pin plastic SSOP (7.62 mm (300))	Flash memory
$\mu$ PD78F0103M2MC(A1)-5A4	30-pin plastic SSOP (7.62 mm (300))	Flash memory
$\mu$ PD78F0103M3MC(A1)-5A4	30-pin plastic SSOP (7.62 mm (300))	Flash memory
$\mu$ PD78F0103M4MC(A1)-5A4	30-pin plastic SSOP (7.62 mm (300))	Flash memory
$\mu$ PD78F0103M5MC(A1)-5A4	30-pin plastic SSOP (7.62 mm (300))	Flash memory
$\mu$ PD78F0103M6MC(A1)-5A4	30-pin plastic SSOP (7.62 mm (300))	Flash memory

**Remark** xxx indicates ROM code suffix.

Mask ROM versions ( $\mu$ PD780101, 780102, and 780103) include a mask options. When ordering, it is possible to select “Power-on-clear (POC) circuit can be used/cannot be used” and “Ring-OSC clock can be stopped/cannot be stopped by software”.

Flash memory versions corresponding to the mask options of the mask ROM versions are as follows.

**Table 1-1. Flash Memory Versions Corresponding to Mask Options of Mask ROM Versions**

Mask Option		Flash Memory Versions (Part Number)
POC Circuit	Ring-OSC	
POC cannot be used	Cannot be stopped	$\mu$ PD78F0103M1MC-5A4 $\mu$ PD78F0103M1MC(A)-5A4 $\mu$ PD78F0103M1MC(A1)-5A4
	Can be stopped by software	$\mu$ PD78F0103M2MC-5A4 $\mu$ PD78F0103M2MC(A)-5A4 $\mu$ PD78F0103M2MC(A1)-5A4
POC used ( $V_{POC} = 2.85\text{ V} \pm 0.15\text{ V}$ )	Cannot be stopped	$\mu$ PD78F0103M3MC-5A4 $\mu$ PD78F0103M3MC(A)-5A4 $\mu$ PD78F0103M3MC(A1)-5A4
	Can be stopped by software	$\mu$ PD78F0103M4MC-5A4 $\mu$ PD78F0103M4MC(A)-5A4 $\mu$ PD78F0103M4MC(A1)-5A4
POC used ( $V_{POC} = 3.5\text{ V} \pm 0.2\text{ V}$ )	Cannot be stopped	$\mu$ PD78F0103M5MC-5A4 $\mu$ PD78F0103M5MC(A)-5A4 $\mu$ PD78F0103M5MC(A1)-5A4
	Can be stopped by software	$\mu$ PD78F0103M6MC-5A4 $\mu$ PD78F0103M6MC(A)-5A4 $\mu$ PD78F0103M6MC(A1)-5A4

## 1.4 Quality Grade

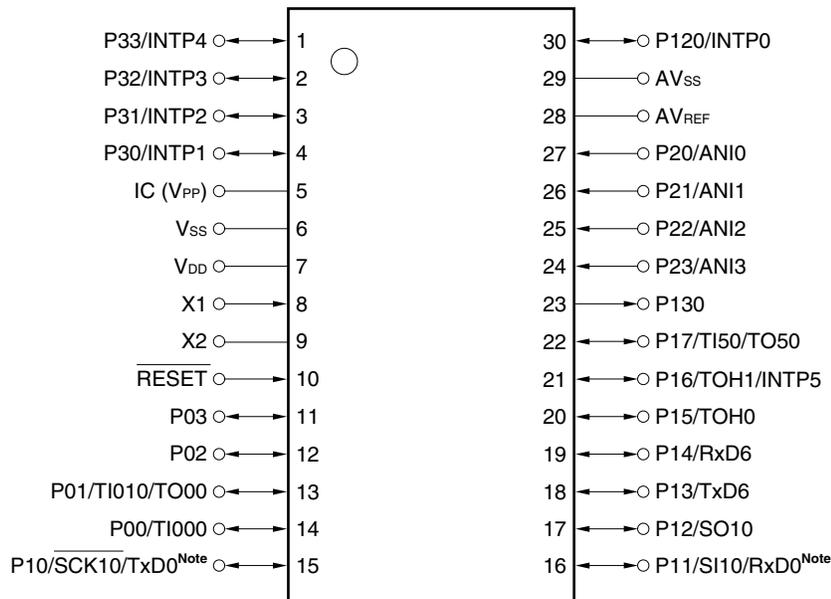
Part Number	Package	Quality Grade
$\mu$ PD780101MC-xxx-5A4	30-pin plastic SSOP (7.62 mm (300))	Standard
$\mu$ PD780102MC-xxx-5A4	30-pin plastic SSOP (7.62 mm (300))	Standard
$\mu$ PD780103MC-xxx-5A4	30-pin plastic SSOP (7.62 mm (300))	Standard
$\mu$ PD780101MC(A)-xxx-5A4	30-pin plastic SSOP (7.62 mm (300))	Special
$\mu$ PD780102MC(A)-xxx-5A4	30-pin plastic SSOP (7.62 mm (300))	Special
$\mu$ PD780103MC(A)-xxx-5A4	30-pin plastic SSOP (7.62 mm (300))	Special
$\mu$ PD780101MC(A1)-xxx-5A4	30-pin plastic SSOP (7.62 mm (300))	Special
$\mu$ PD780102MC(A1)-xxx-5A4	30-pin plastic SSOP (7.62 mm (300))	Special
$\mu$ PD780103MC(A1)-xxx-5A4	30-pin plastic SSOP (7.62 mm (300))	Special
$\mu$ PD780101MC(A2)-xxx-5A4	30-pin plastic SSOP (7.62 mm (300))	Special
$\mu$ PD780102MC(A2)-xxx-5A4	30-pin plastic SSOP (7.62 mm (300))	Special
$\mu$ PD780103MC(A2)-xxx-5A4	30-pin plastic SSOP (7.62 mm (300))	Special
$\mu$ PD78F0103M1MC-5A4	30-pin plastic SSOP (7.62 mm (300))	Standard
$\mu$ PD78F0103M2MC-5A4	30-pin plastic SSOP (7.62 mm (300))	Standard
$\mu$ PD78F0103M3MC-5A4	30-pin plastic SSOP (7.62 mm (300))	Standard
$\mu$ PD78F0103M4MC-5A4	30-pin plastic SSOP (7.62 mm (300))	Standard
$\mu$ PD78F0103M5MC-5A4	30-pin plastic SSOP (7.62 mm (300))	Standard
$\mu$ PD78F0103M6MC-5A4	30-pin plastic SSOP (7.62 mm (300))	Standard
$\mu$ PD78F0103M1MC(A)-5A4	30-pin plastic SSOP (7.62 mm (300))	Special
$\mu$ PD78F0103M2MC(A)-5A4	30-pin plastic SSOP (7.62 mm (300))	Special
$\mu$ PD78F0103M3MC(A)-5A4	30-pin plastic SSOP (7.62 mm (300))	Special
$\mu$ PD78F0103M4MC(A)-5A4	30-pin plastic SSOP (7.62 mm (300))	Special
$\mu$ PD78F0103M5MC(A)-5A4	30-pin plastic SSOP (7.62 mm (300))	Special
$\mu$ PD78F0103M6MC(A)-5A4	30-pin plastic SSOP (7.62 mm (300))	Special
$\mu$ PD78F0103M1MC(A1)-5A4	30-pin plastic SSOP (7.62 mm (300))	Special
$\mu$ PD78F0103M2MC(A1)-5A4	30-pin plastic SSOP (7.62 mm (300))	Special
$\mu$ PD78F0103M3MC(A1)-5A4	30-pin plastic SSOP (7.62 mm (300))	Special
$\mu$ PD78F0103M4MC(A1)-5A4	30-pin plastic SSOP (7.62 mm (300))	Special
$\mu$ PD78F0103M5MC(A1)-5A4	30-pin plastic SSOP (7.62 mm (300))	Special
$\mu$ PD78F0103M6MC(A1)-5A4	30-pin plastic SSOP (7.62 mm (300))	Special

**Remark** xxx indicates ROM code suffix.

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

### 1.5 Pin Configuration (Top View)

- 30-pin plastic SSOP (7.62 mm (300))
  - $\mu$ PD780101MC-xxx-5A4, 780102MC-xxx-5A4, 780103MC-xxx-5A4,
  - $\mu$ PD780101MC(A)-xxx-5A4, 780102MC(A)-xxx-5A4, 780103MC(A)-xxx-5A4,
  - $\mu$ PD780101MC(A1)-xxx-5A4, 780102MC(A1)-xxx-5A4, 780103MC(A1)-xxx-5A4,
  - $\mu$ PD780101MC(A2)-xxx-5A4, 780102MC(A2)-xxx-5A4, 780103MC(A2)-xxx-5A4,
  - $\mu$ PD78F0103M1MC-5A4, 78F0103M2MC-5A4, 78F0103M3MC-5A4, 78F0103M4MC-5A4,
  - $\mu$ PD78F0103M5MC-5A4, 78F0103M6MC-5A4, 78F0103M1MC(A)-5A4, 78F0103M2MC(A)-5A4,
  - $\mu$ PD78F0103M3MC(A)-5A4, 78F0103M4MC(A)-5A4, 78F0103M5MC(A)-5A4,
  - $\mu$ PD78F0103M6MC(A)-5A4, 78F0103M1MC(A1)-5A4, 78F0103M2MC(A1)-5A4,
  - $\mu$ PD78F0103M3MC(A1)-5A4, 78F0103M4MC(A1)-5A4, 78F0103M5MC(A1)-5A4,
  - $\mu$ PD78F0103M6MC(A1)-5A4



**Note** TxD0 and RxD0 are available only in the  $\mu$ PD780102, 780103, and 78F0103.

- Cautions**
1. Connect the IC (Internally Connected) pin directly to V<sub>SS</sub>.
  2. Connect the AV<sub>REF</sub> pin to V<sub>DD</sub>.
  3. Connect the AV<sub>SS</sub> pin to V<sub>SS</sub>.

**Remark** Figures in parentheses apply only to the  $\mu$ PD78F0103.

**Pin Identification**

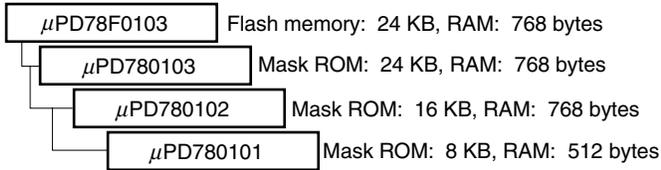
ANI0 to ANI3:	Analog input	RxD0 <sup>Note</sup> , RxD6:	Receive data
AV <sub>REF</sub> :	Analog reference voltage	$\overline{\text{SCK10}}$ :	Serial clock input/output
IC:	Internally connected	SI10:	Serial data input
INTP0 to INTP5:	External interrupt input	SO10:	Serial data output
P00 to P03:	Port 0	TI000, TI010, TI50:	Timer output
P10 to P17:	Port 1	TO00, TO50, TOH0, TOH1:	Timer output
P20 to P23:	Port 2	TxD0 <sup>Note</sup> , TxD6:	Transmit data
P30 to P33:	Port 3	V <sub>DD</sub> :	Power supply
P120:	Port 12	V <sub>PP</sub> :	Programming power supply
P130:	Port 13	V <sub>SS</sub> :	Ground
$\overline{\text{RESET}}$ :	Reset	X1, X2:	Crystal (X1 input clock)

**Note** TxD0 and RxD0 are available only in the  $\mu$ PD780102, 780103, and 78F0103.

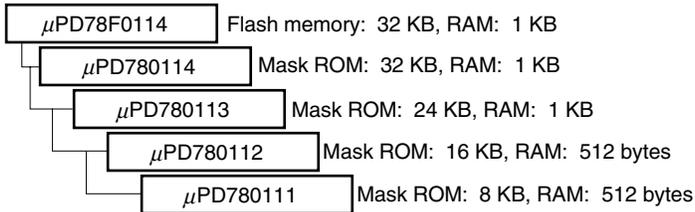
## 1.6 78K0/Kxx Series Lineup

The lineup of products in the 78K0/Kxx Series (under development or in planning) is shown below.

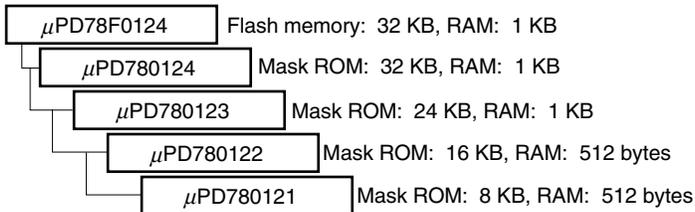
### ★ 78K0/KB1 Series: 30-pin (7.62 mm 0.65 mm pitch)



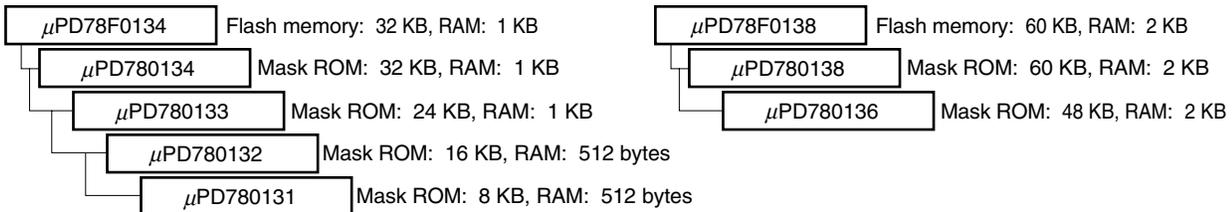
### 78K0/KC1 Series: 44-pin (10 × 10 mm 0.8 mm pitch)



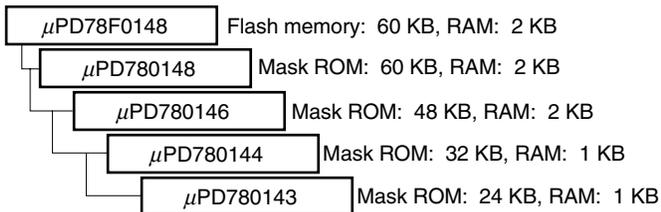
### 78K0/KD1 Series: 52-pin (10 × 10 mm 0.65 mm pitch)



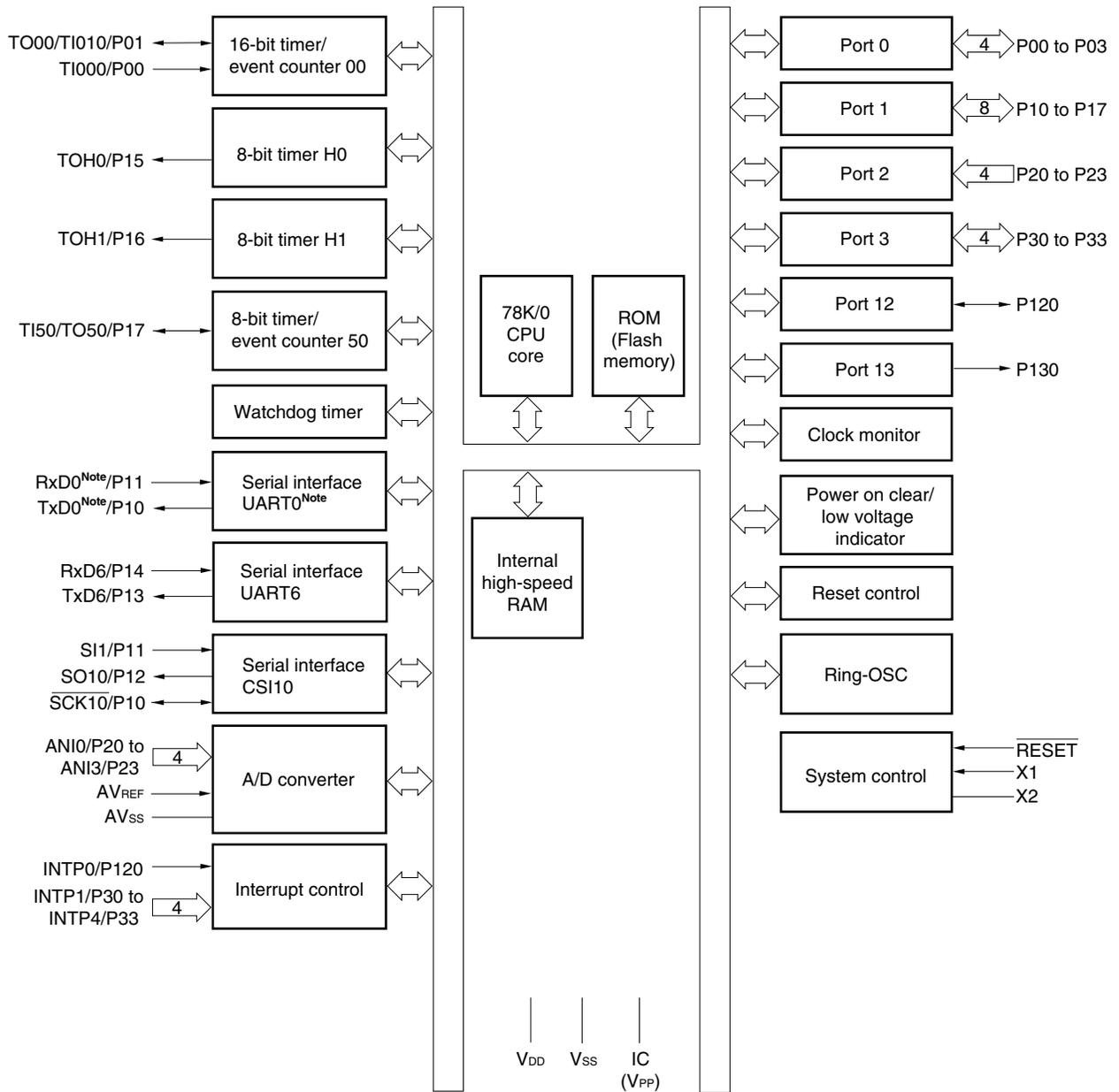
### 78K0/KE1 Series: 64-pin (10 × 10 mm 0.5 mm pitch, 12 × 12 mm 0.65 mm pitch, 14 × 14 mm 0.8 mm pitch)



### 78K0/KF1 Series: 80-pin (12 × 12 mm 0.5 mm pitch, 14 × 14 mm 0.65 mm pitch)



1.7 Block Diagram



**Note**  $\mu$ PD780102, 780103, and 78F0103 only.

**Remark** Items in parentheses are available only in the  $\mu$ PD78F0103.

1.8 Outline of Functions

Item		$\mu$ PD780101	$\mu$ PD780102	$\mu$ PD780103	$\mu$ PD78F0103
Internal memory	ROM	8 KB	16 KB	24 KB	24 KB (flash memory)
	High-speed RAM	512 bytes	768 bytes		
Memory space		64 KB			
X1 input clock (oscillation frequency)		Ceramic/crystal oscillation (10.0 MHz: $V_{DD} = 4.0$ to $5.5$ V, 8.38 MHz: $V_{DD} = 3.3$ to $5.5$ V, 5 MHz: $V_{DD} = 2.7$ to $5.5$ V)			
Ring-OSC clock (oscillation frequency)		On-chip Ring oscillation (240 kHz (TYP.))			
General-purpose registers		8 bits $\times$ 32 registers (8 bits $\times$ 8 registers $\times$ 4 banks)			
Minimum instruction execution time		0.2 $\mu$ s/0.4 $\mu$ s/0.8 $\mu$ s/1.6 $\mu$ s/3.2 $\mu$ s (X1 input clock: @ $f_{XP} = 10$ MHz operation)			
		8.3 $\mu$ s/16.6 $\mu$ s/33.2 $\mu$ s/66.4 $\mu$ s/132.8 $\mu$ s (TYP.) (Ring-OSC clock: @ $f_R = 240$ kHz (TYP.) operation)			
Instruction set		<ul style="list-style-type: none"> <li>• 16-bit operation</li> <li>• Multiply/divide (8 bits <math>\times</math> 8 bits <math>\times</math> 4 banks)</li> <li>• Bit manipulate (set, reset, test, and Boolean operation)</li> <li>• BCD adjust, etc.</li> </ul>			
I/O ports		Total: <u>22</u>			
		CMOS I/O 17			
		CMOS input 4			
		CMOS output 1			
Timers		<ul style="list-style-type: none"> <li>• 16-bit timer/event counter: 1 channel</li> <li>• 8-bit timer/event counter: 1 channel</li> <li>• 8-bit timer: 2 channels</li> <li>• Watchdog timer: 1 channel</li> </ul>			
A/D converter		10-bit resolution $\times$ 4 channels			
Serial interface		<ul style="list-style-type: none"> <li>• UART mode supporting LIN-bus: 1 channel</li> <li>• 3-wire serial I/O mode/UART mode<sup>Note</sup>: 1 channel (<math>\mu</math>PD780101 only, 3-wire serial I/O mode: 1 channel)</li> </ul>			
Vectored interrupt sources	Internal	10	12		
	External	6			
Reset		<ul style="list-style-type: none"> <li>• Reset using <math>\overline{\text{RESET}}</math> pin</li> <li>• Internal reset by watchdog timer</li> <li>• Internal reset by clock monitor</li> <li>• Internal reset by power-on-reset</li> <li>• Internal reset by low-voltage detector</li> </ul>			
Supply voltage		$V_{DD} = 2.7$ to $5.5$ V			
Operating ambient temperature		Standard products, (A) products: $T_A = -40$ to $+85^\circ\text{C}$ (A1) products: $T_A = -40$ to $+110^\circ\text{C}$ (A2) products: $T_A = -40$ to $+125^\circ\text{C}$ ( $\mu$ PD780101, 780102, and 780103 only)			
Package		30-pin plastic SSOP (7.62 mm (300))			

★ **Note** Select either of the functions of these alternate-function pins.

An outline of the timer is shown below.

		16-Bit Timer/Event Counter 00	8-Bit Timer/Event Counter 50	8-Bit Timers H0 and H1	Watchdog Timer
Operation mode	Interval timer	1 channel	1 channel	2 channels	1 channel
	External event counter	1 channel	1 channel	–	–
Function	Timer output	1 output	1 output	2 outputs	–
	PPG output	1 output	–	–	–
	PWM output	–	1 output	2 outputs	–
	Pulse width measurement	2 inputs	–	–	–
	Square-wave output	1 output	1 output	–	–
	Interrupt source	2	1	2	–

## CHAPTER 2 PIN FUNCTIONS

### 2.1 Pin Function List

#### (1) Port pins

Pin Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	Ti000
P01				Ti010/TO00
P02				–
P03				–
P10	I/O	Port 1. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	$\overline{\text{SCK10}}/\text{TxD0}^{\text{Note}}$
P11				SI10/RxD0 <sup>Note</sup>
P12				SO10
P13				TxD6
P14				RxD6
P15				TOH0
P16				TOH1/INTP5
P17				TI50/TO50
P20 to P23	Input	Port 2. 4-bit input-only port.	Input	ANI0 to ANI3
P30 to P33	I/O	Port 3. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	INTP1 to INTP4
P120	I/O	Port 12. 1-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	INTP0
P130	Output	Port 13. 1-bit output-only port.	Output	–

**Note** TxD0 and RxD0 are available only in the  $\mu$ PD780102, 780103, and 78F0103.

(2) Non-port pins

Pin Name	I/O	Function	After Reset	Alternate Function
INTP0	Input	External interrupt request input for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified	Input	P120
INTP1 to INTP4				P30 to P33
INTP5				P16/TOH1
SI10	Input	Serial data input to serial interface	Input	P11/RxD0 <sup>Note</sup>
SO10	Output	Serial data output from serial interface	Input	P12
SCK10	I/O	Clock input/output for serial interface	Input	P10/TxD0 <sup>Note</sup>
RxD0 <sup>Note</sup>	Input	Serial data input to asynchronous serial interface	Input	P11/SI10
RxD6				P14
TxD0 <sup>Note</sup>	Output	Serial data output from asynchronous serial interface	Input	P10/SCK10
TxD6				P13
TI000	Input	External count clock input to 16-bit timer/event counter 00 Capture trigger input to capture registers (CR000, CR010) of 16-bit timer/event counter 00	Input	P00
TI010		Capture trigger input to capture register (CR000) of 16-bit timer/event counter 00		P01/TO00
TO00	Output	16-bit timer/event counter 00 output	Input	P01/TO10
TI50	Input	External count clock input to 8-bit timer/event counter 50	Input	P17/TO50
TO50	Output	8-bit timer/event counter 50 output	Input	P17/TO50
TOH0	Output	8-bit timer H output	Input	P15
TOH1				P16/INTP5
ANI0 to ANI3	Input	A/D converter analog input	Input	P20 to P23
AV <sub>REF</sub>	Input	A/D converter reference voltage input	–	–
AV <sub>SS</sub>	–	A/D converter ground potential. Make the same potential as V <sub>SS</sub> .	–	–
RESET	Input	System reset input	–	–
X1	Input	Connecting crystal resonator for X1 input clock oscillation	–	–
X2	–		–	–
V <sub>DD</sub>	–	Positive power supply	–	–
V <sub>SS</sub>	–	Ground potential	–	–
IC	–	Internally connected. Connect directly to V <sub>SS</sub> .	–	–
V <sub>PP</sub>	–	Flash memory programming mode setting. High-voltage application for program write/verify. Connect directly to V <sub>SS</sub> in normal operation mode.	–	–

**Note** TxD0 and RxD0 are available only in the  $\mu$ PD780102, 780103, and 78F0103.

## 2.2 Description of Pin Functions

### 2.2.1 P00 to P03 (port 0)

P00 to P03 function as a 4-bit I/O port. These pins also function as timer I/O.

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

#### (1) Port mode

P00 to P03 function as a 4-bit I/O port. P00 to P03 can be set to input or output in 1-bit units using port mode register 0 (PM0). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 0 (PU0).

#### (2) Control mode

P00 to P03 function as timer I/O.

##### (a) TI000

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

##### (b) TI010

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

##### (c) TO00

This is a timer output pin.

### 2.2.2 P10 to P17 (port 1)

P10 to P17 function as an 8-bit I/O port. These pins also function as pins for external interrupt request input, serial interface data I/O, clock I/O, and timer I/O.

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

#### (1) Port mode

P10 to P17 function as an 8-bit I/O port. P10 to P17 can be set to input or output in 1-bit units using port mode register 1 (PM1). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 1 (PU1).

#### (2) Control mode

P10 to P17 function as external interrupt request input, serial interface data I/O, clock I/O, and timer I/O.

##### (a) SI10, SO10

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

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

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

##### (c) RxD0<sup>Note</sup>, RxD6, TxD0<sup>Note</sup>, and TxD6

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

**Note** TxD0 and RxD0 are available only in the  $\mu$ PD780102, 780103, and 78F0103.

**(d) TI50**

This is the pin for inputting an external count clock to 8-bit timer/event counter 50.

**(e) TO50, TOH0, and TOH1**

These are timer output pins.

**(f) INTP5**

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

**2.2.3 P20 to P23 (port 2)**

P20 to P23 function as a 4-bit input-only port. These pins also function as pins for A/D converter analog input. The following operation modes can be specified in 1-bit units.

**(1) Port mode**

P20 to P23 function as a 4-bit input-only port.

**(2) Control mode**

P20 to P23 function as A/D converter analog input pins (ANI0 to ANI3).

**2.2.4 P30 to P33 (port 3)**

P30 to P33 function as a 4-bit I/O port. These pins also function as pins for external interrupt request input. The following operation modes can be specified in 1-bit units.

**(1) Port mode**

P30 to P33 function as a 4-bit I/O port. P30 to P33 can be set to input or output in 1-bit units using port mode register 3 (PM3). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 3 (PU3).

**(2) Control mode**

P30 to P33 function as external interrupt request input pins (INTP1 to INTP4) for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

**2.2.5 P120 (port 12)**

P120 functions as a 1-bit I/O port. This pin also functions as a pin for external interrupt request input. The following operation modes can be specified in 1-bit units.

**(1) Port mode**

P120 functions as a 1-bit I/O port. P120 can be set to input or output in 1-bit units using port mode register 12 (PM12). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 12 (PU12).

**(2) Control mode**

P120 functions as an external interrupt request input pin (INTP0) for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

**2.2.6 P130 (port 13)**

P130 functions as a 1-bit output-only port.

**2.2.7 AV<sub>REF</sub>**

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

When A/D converter is not used, connect this pin to V<sub>DD</sub>.

**2.2.8 AV<sub>SS</sub>**

This is the A/D converter ground potential pin. Even when the A/D converter is not used, always use this pin with the same potential as the V<sub>SS</sub> pin.

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

This is the active-low system reset input pin.

**2.2.10 X1 and X2**

These are the pins for connecting a crystal resonator for X1 input clock oscillation.

When supplying an external clock, input a signal to the X1 pin and input the inverse signal to the X2 pin.

**2.2.11 V<sub>DD</sub>**

This is the positive power supply pin.

**2.2.12 V<sub>SS</sub>**

This is the ground potential pin.

**2.2.13 V<sub>PP</sub> (flash memory versions only)**

This is a pin for flash memory programming mode setting and high-voltage application for program write/verify.

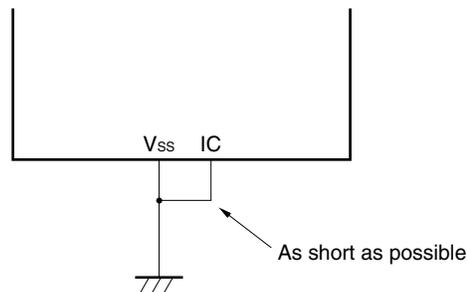
Connect directly to V<sub>SS</sub> in the normal operation mode.

**2.2.14 IC (mask ROM versions only)**

The IC (Internally Connected) pin is provided to set the test mode to check the 78K0/KB1 Series at shipment. Connect it directly to V<sub>SS</sub> with the shortest possible wire in the normal operation mode.

When a potential difference is produced between the IC pin and the V<sub>SS</sub> pin because the wiring between these two pins is too long or external noise is input to the IC pin, the user's program may not operate normally.

- Connect the IC pin directly to V<sub>SS</sub>.



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

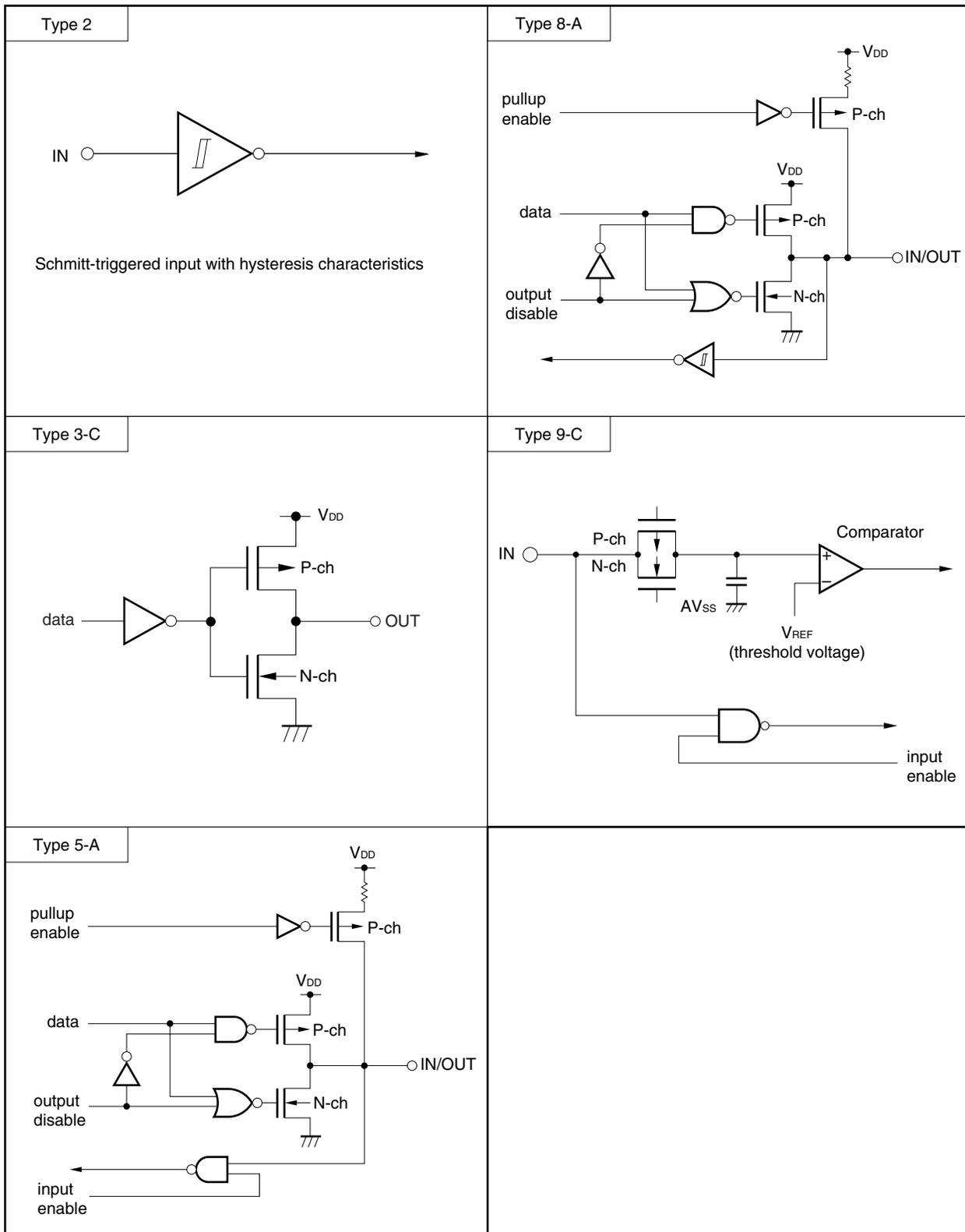
Table 2-1 shows the types of pin I/O circuit and the recommended connections of unused pins. Refer to Figure 2-1 for the configuration of the I/O circuits of each type.

**Table 2-1. Pin I/O Circuit Types**

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00/TI000	8-A	I/O	Input: Independently connect to $V_{DD}$ or $V_{SS}$ via a resistor. Output: Leave open.
P01/TI010/TO00			
P02			
P03			
P10/ $\overline{SCK10}$ /TxD0 <sup>Note</sup>			
P11/SI10/RxD0 <sup>Note</sup>			
P12/SO10	5-A	I/O	Input: Independently connect to $V_{DD}$ or $V_{SS}$ via a resistor. Output: Leave open.
P13/TxD6			
P14/RxD6	8-A		
P15/TOH0	5-A		
P16/TOH1/INTP5	8-A		
P17/TI50/TO50			
P20/ANI0 to P23/ANI3	9-C	Input	Connect to $V_{DD}$ or $V_{SS}$ .
P30/INTP1 to P33/INTP4	8-A	I/O	Input: Independently connect to $V_{SS}$ via a resistor. Output: Leave open.
P120/INTP0			Input: Independently connect to $V_{DD}$ or $V_{SS}$ via a resistor. Output: Leave open.
P130	3-C	Output	Leave open.
$\overline{RESET}$	2	Input	–
$AV_{REF}$	–	Input	Connect to $V_{DD}$ .
$AV_{SS}$	–	–	Connect to $V_{SS}$ .
IC	–	–	Connect directly to $V_{SS}$ .
$V_{PP}$			

**Note** TxD0 and RxD0 are available only in the  $\mu$ PD780102, 780103, and 78F0103.

Figure 2-1. Pin I/O Circuit List



## CHAPTER 3 CPU ARCHITECTURE

### 3.1 Memory Space

Products in the 78K0/KB1 Series can each access a 64 KB memory space. Figures 3-1 to 3-4 show the memory maps.

**Caution** Regardless of the internal memory capacity, the initial values of internal memory size switching register (IMS) of all products in the 78K0/KB1 Series are fixed (CFH). Therefore, set the value corresponding to each product as indicated below.

**Table 3-1. Internal Memory Size Switching Register (IMS) Set Value**

	Internal Memory Size Switching Register (IMS)
$\mu$ PD780101	42H
$\mu$ PD780102	04H
$\mu$ PD780103	06H
$\mu$ PD78F0103	Value corresponding to mask ROM product

Figure 3-1. Memory Map ( $\mu$ PD780101)

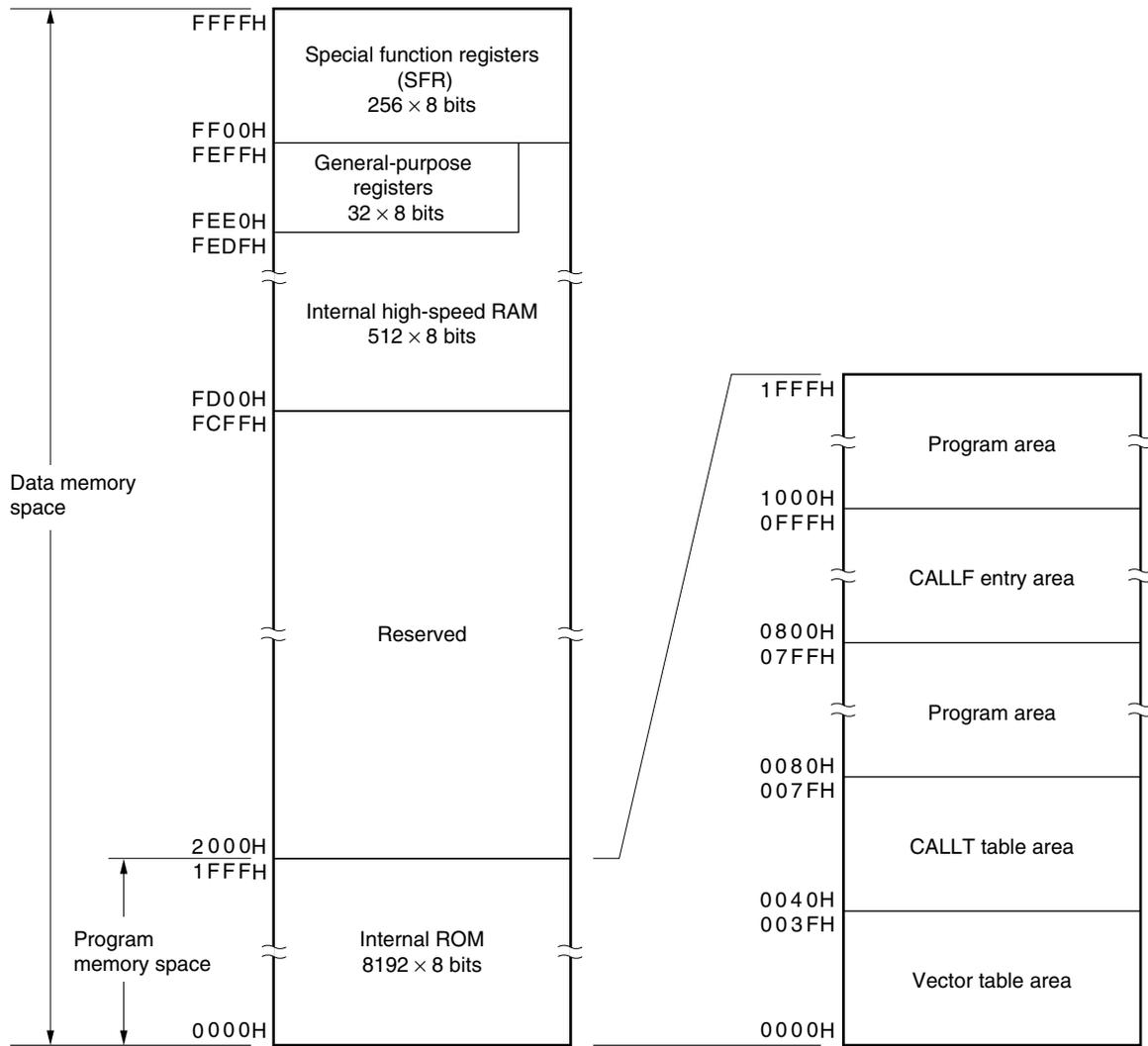


Figure 3-2. Memory Map ( $\mu$ PD780102)

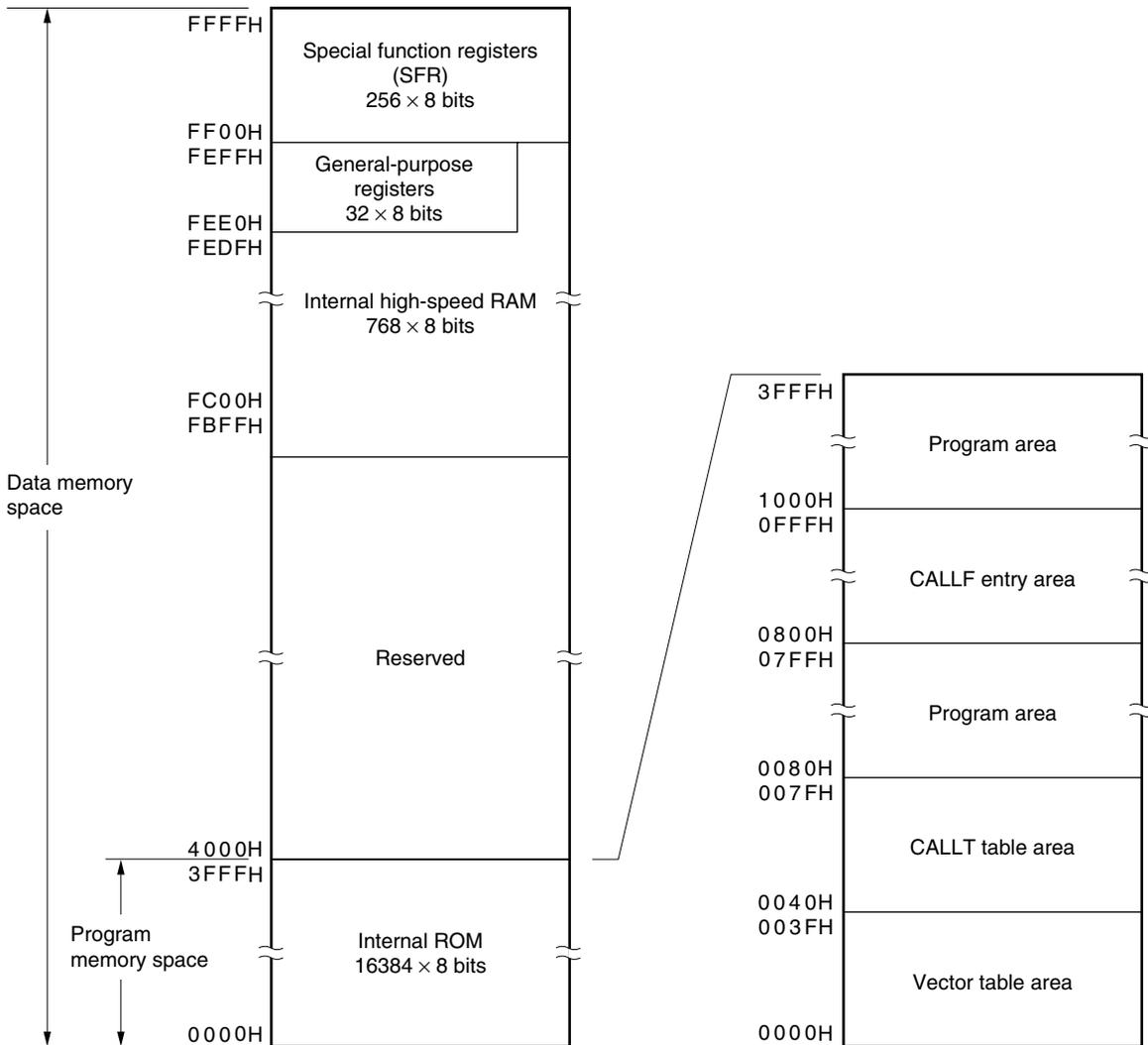


Figure 3-3. Memory Map ( $\mu$ PD780103)

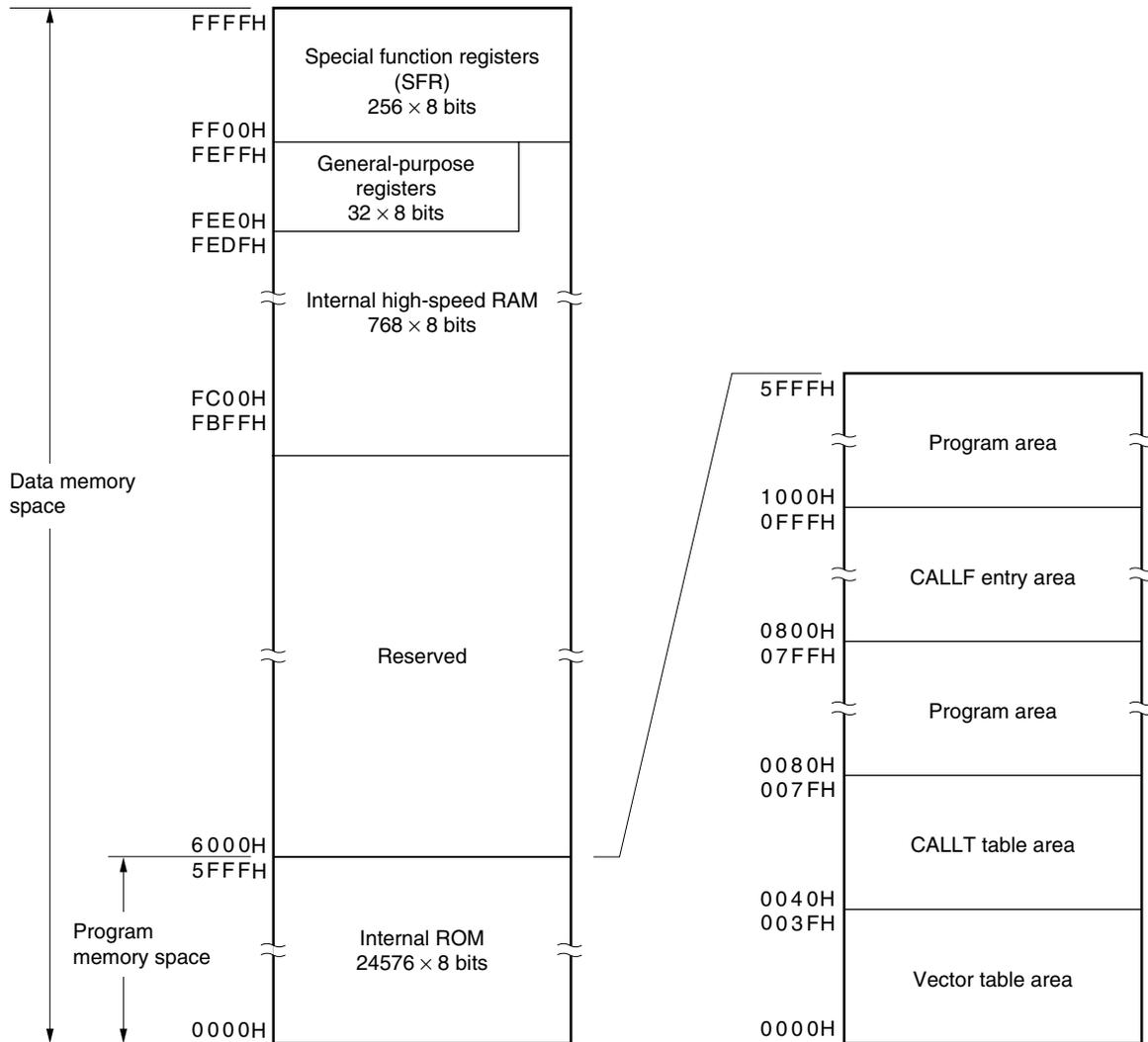
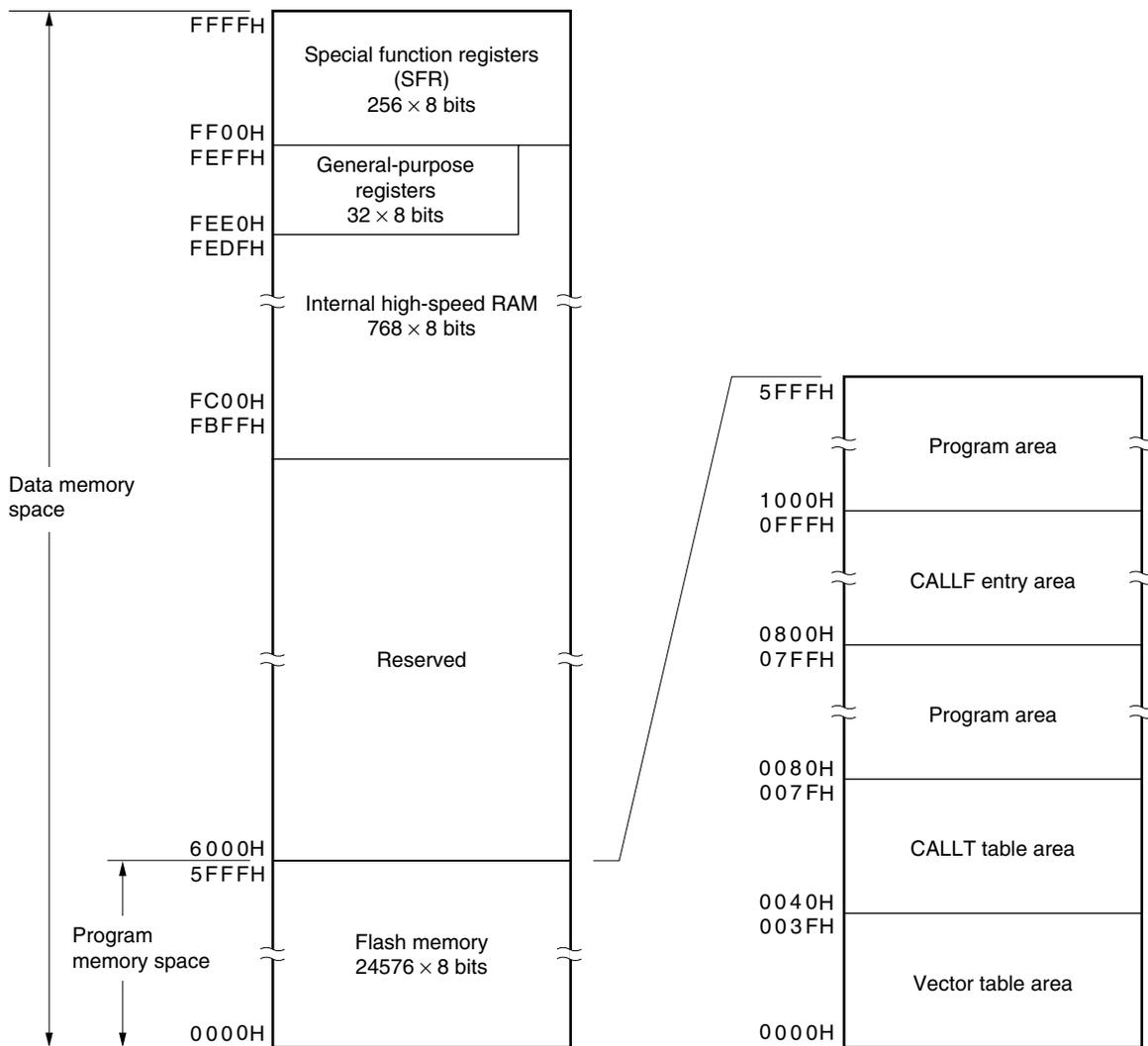


Figure 3-4. Memory Map ( $\mu$ PD78F0103)



### 3.1.1 Internal program memory space

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

78K0/KB1 Series products incorporate internal ROM (or flash memory), as shown below.

**Table 3-2. Internal Memory Capacity**

Part Number	Internal ROM	
	Structure	Capacity
$\mu$ PD780101	Mask ROM	8192 $\times$ 8 bits (0000H to 1FFFH)
$\mu$ PD780102		16384 $\times$ 8 bits (0000H to 3FFFH)
$\mu$ PD780103		24576 $\times$ 8 bits (0000H to 5FFFH)
$\mu$ PD78F0103	Flash memory	24576 $\times$ 8 bits (0000H to 5FFFH)

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

#### (1) Vector table area

The 64-byte area 0000H to 003FH is reserved as a vector table area. The program start addresses for branch upon  $\overline{\text{RESET}}$  input or generation of each interrupt request are stored in the vector table area.

Of the 16-bit address, the lower 8 bits are stored at even addresses and the higher 8 bits are stored at odd addresses.

**Table 3-3. Vector Table**

Vector Table Address	Interrupt Source	Vector Table Address	Interrupt Source
0000H	$\overline{\text{RESET}}$ input	0016H	INTST6
0004H	INTLVI	0018H	INTCSI10/INTST0 <sup>Note</sup>
0006H	INTP0	001AH	INTTMH1
0008H	INTP1	001CH	INTTMH0
000AH	INTP2	001EH	INTTM50
000CH	INTP3	0020H	INTTM000
000EH	INTP4	0022H	INTTM010
0010H	INTP5	0024H	INTAD
0012H	INTSRE6	0026H	INTSR0 <sup>Note</sup>
0014H	INTSR6		

**Note** Available only in the  $\mu$ PD780102, 780103, and 78F0103.

#### (2) CALLT instruction table area

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

#### (3) CALLF instruction entry area

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

### 3.1.2 Internal data memory space

78K0/KB1 Series products incorporate the following internal high-speed RAM.

**Table 3-4. Internal High-Speed RAM Capacity**

Part Number	Internal High-Speed RAM
$\mu$ PD780101	512 $\times$ 8 bits (FD00H to FEFFH)
$\mu$ PD780102	768 $\times$ 8 bits (FC00H to FEFFH)
$\mu$ PD780103	
$\mu$ PD78F0103	

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

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

### 3.1.3 Special function register (SFR) area

On-chip peripheral hardware special function registers (SFRs) are allocated in the area FF00H to FFFFH (refer to **Table 3-5 Special Function Register List** in **3.2.3 Special Function Registers (SFRs)**).

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

### 3.1.4 Data memory addressing

Addressing refers to the method of specifying the address of the instruction to be executed next or the address of the register or memory relevant to the execution of instructions. The address of the instruction to be executed next is addressed by the program counter (PC) (for details, refer to **3.3 Instruction Address Addressing**).

Several addressing modes are provided for addressing the memory relevant to the execution of instructions for the 78K0/KB1 Series, based on operability and other considerations. For areas containing data memory in particular, special addressing methods designed for the functions of special function registers (SFR) and general-purpose registers are available for use. Data memory addressing is illustrated in Figures 3-5 to 3-8. For details of each addressing mode, refer to **3.4 Operand Address Addressing**.

**Figure 3-5. Data Memory Addressing ( $\mu$ PD780101)**

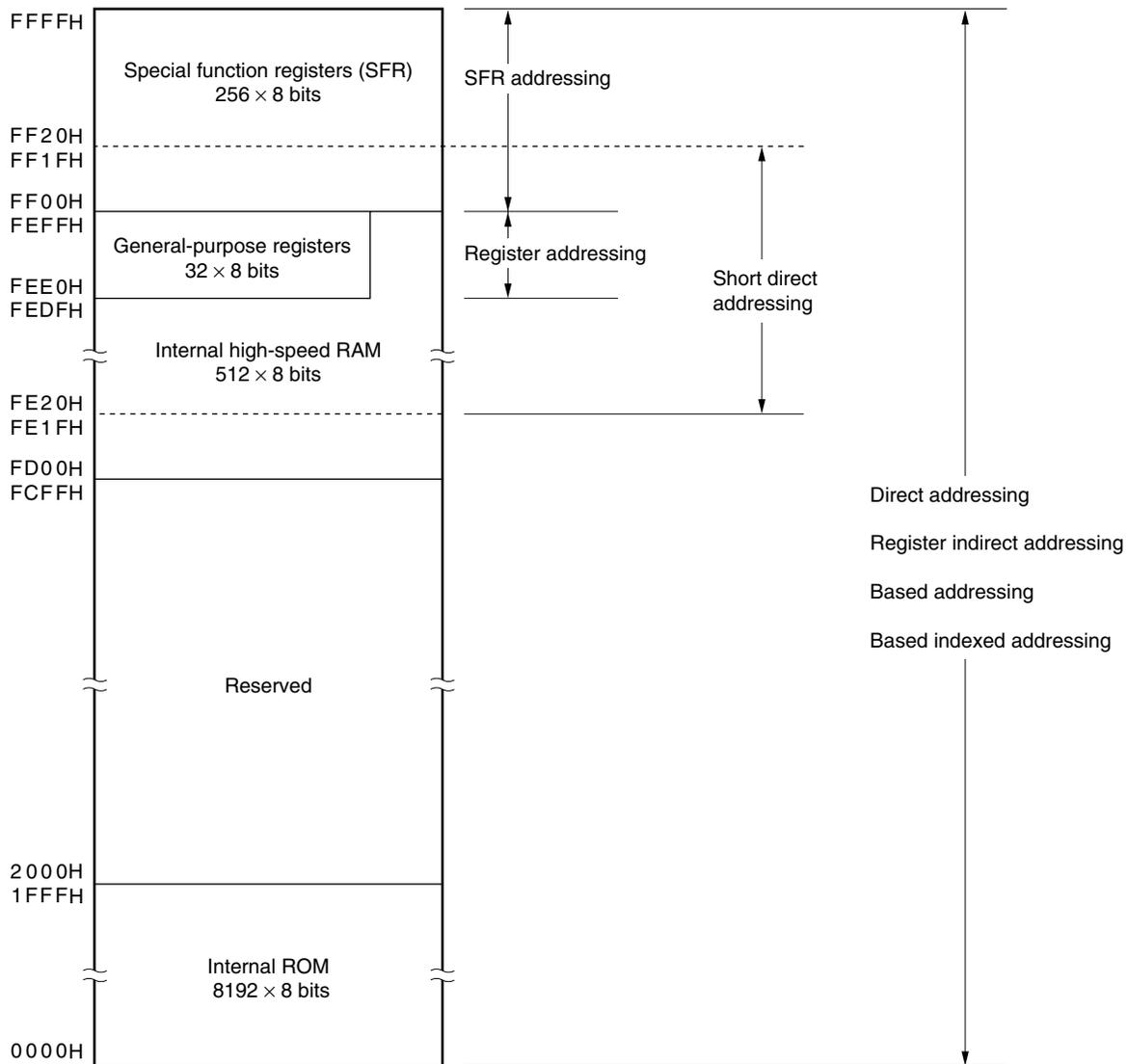


Figure 3-6. Data Memory Addressing ( $\mu$ PD780102)

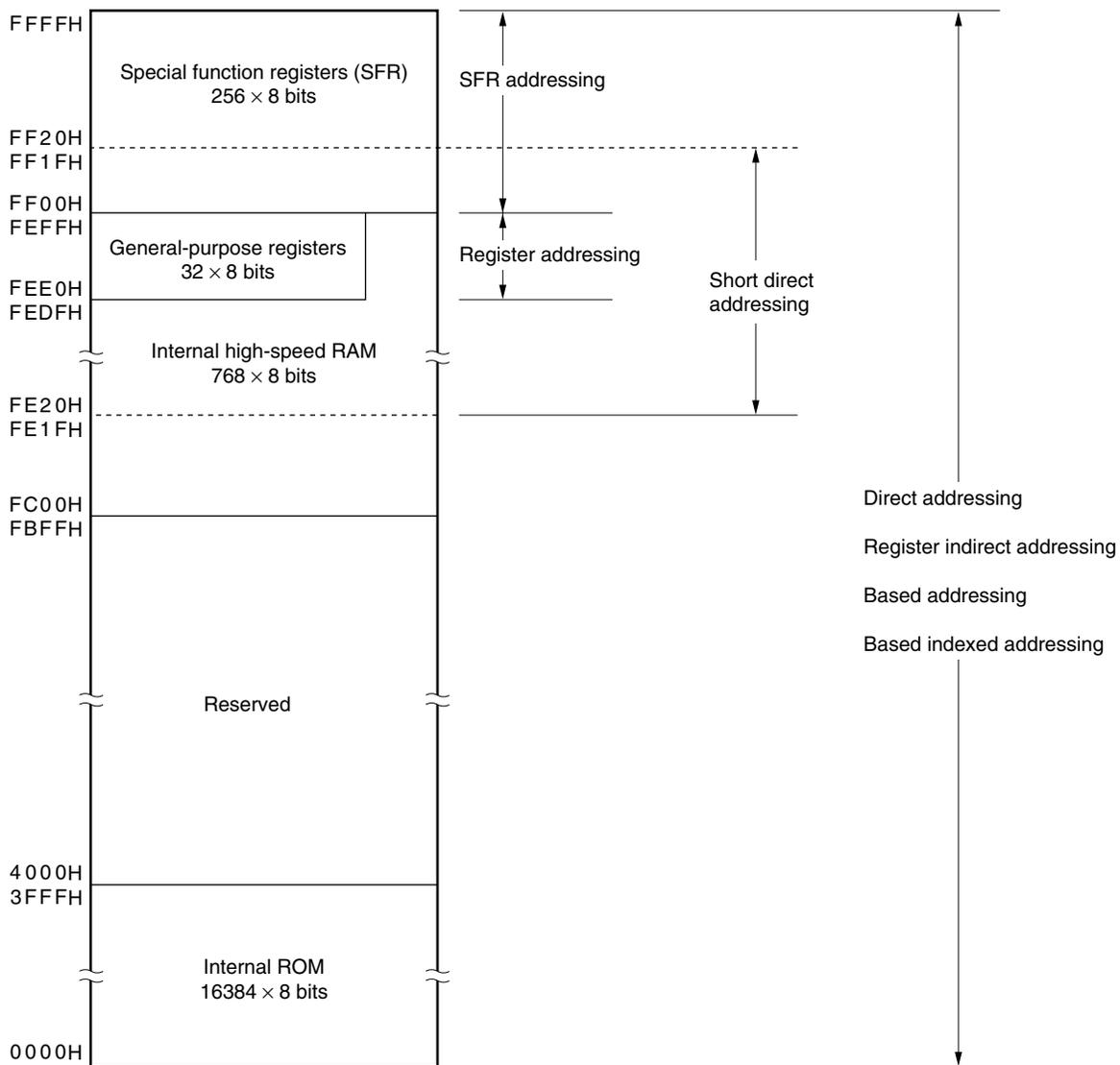


Figure 3-7. Data Memory Addressing ( $\mu$ PD780103)

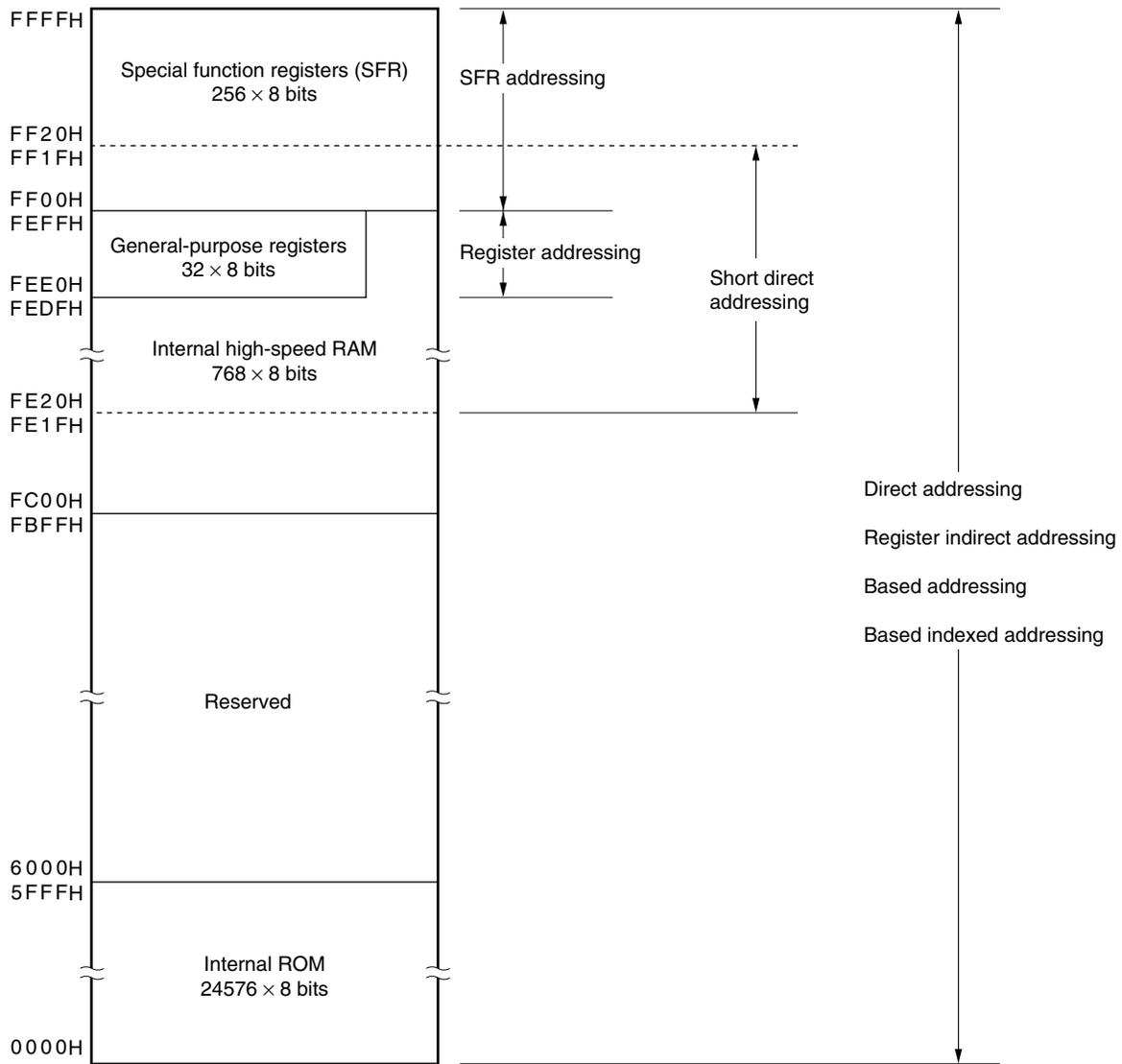
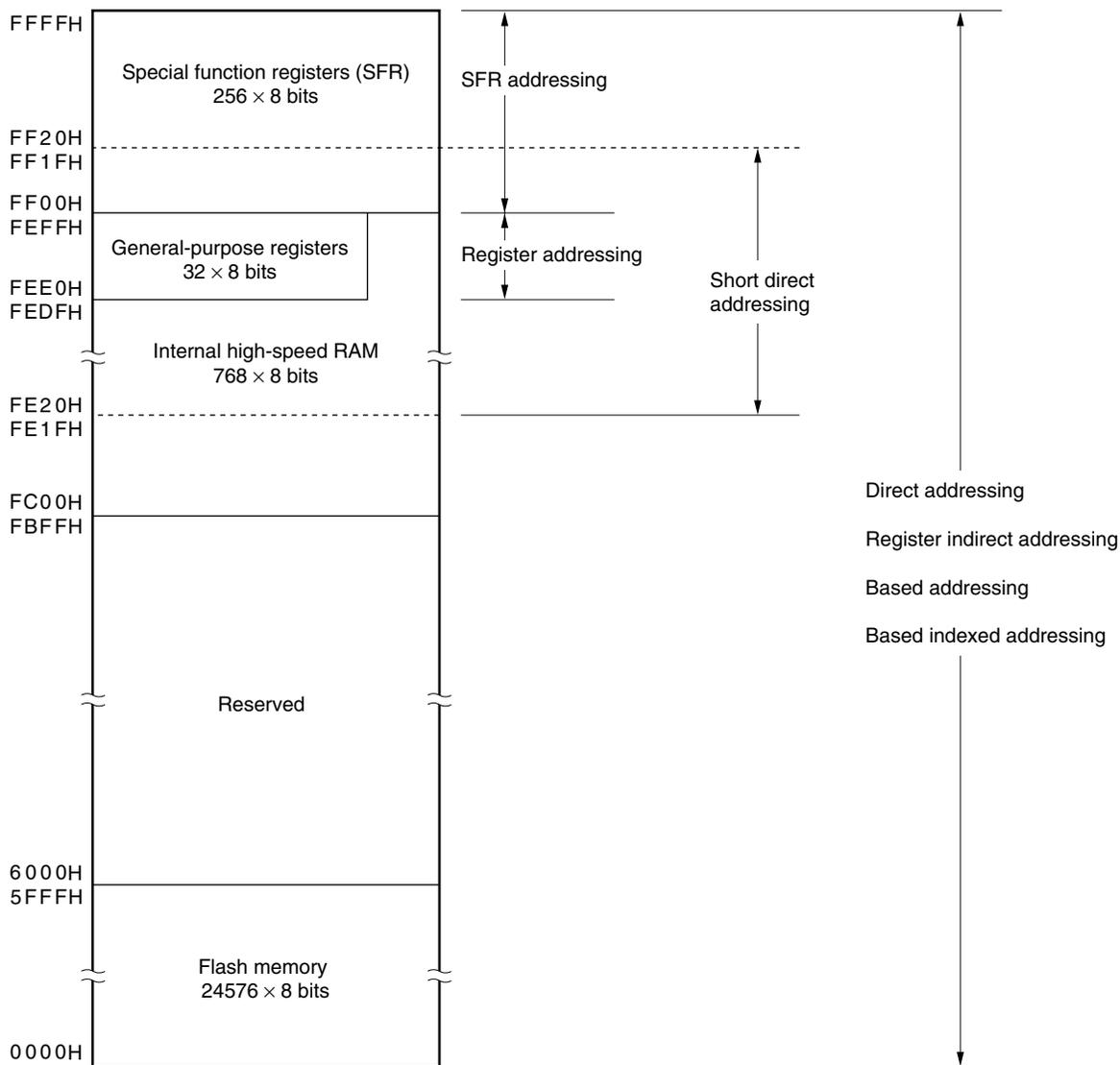


Figure 3-8. Data Memory Addressing ( $\mu$ PD78F0103)



### 3.2 Processor Registers

The 78K0/KB1 Series products incorporate the following processor registers.

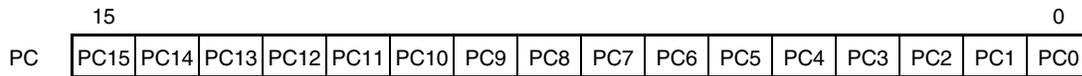
#### 3.2.1 Control registers

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

##### (1) Program counter (PC)

The program counter is a 16-bit register that holds the address information of the next program to be executed. In normal operation, the PC is automatically incremented according to the number of bytes of the instruction to be fetched. When a branch instruction is executed, immediate data and register contents are set.  $\overline{\text{RESET}}$  input sets the reset vector table values at addresses 0000H and 0001H to the program counter.

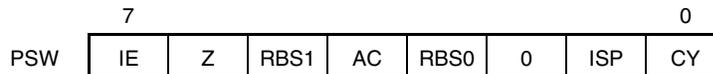
**Figure 3-9. Format of Program Counter**



##### (2) Program status word (PSW)

The program status word is an 8-bit register consisting of various flags to be set/reset by instruction execution. Program status word contents are automatically stacked upon interrupt request generation or PUSH PSW instruction execution and are automatically reset upon execution of the RETB, RETI and POP PSW instructions.  $\overline{\text{RESET}}$  input sets the PSW to 02H.

**Figure 3-10. Format of Program Status Word**



**(a) Interrupt enable flag (IE)**

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

When 0, the IE is set to the interrupt disabled (DI) state, and only non-maskable interrupt request becomes acknowledgeable. Other interrupt requests are all disabled.

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

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

**(b) Zero flag (Z)**

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

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

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

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

**(d) Auxiliary carry flag (AC)**

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

**(e) In-service priority flag (ISP)**

This flag manages the priority of acknowledgeable maskable vectored interrupts. When this flag is 0, low-level vectored interrupts request specified with a priority specification flag register (PR0L, PR0H, PR1L) (refer to **14.3 (3) Priority specification flag registers (PR0L, PR0H, PR1L)**) are disabled for acknowledgement. When it is 1, all interrupts are acknowledgeable. Actual request acknowledgement is controlled with the interrupt enable flag (IE).

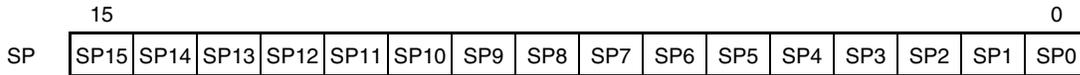
**(f) Carry flag (CY)**

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

**(3) Stack pointer (SP)**

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

Figure 3-11. Stack Pointer Format



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

Each stack operation saves/resets data as shown in Figures 3-12 and 3-13.

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

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

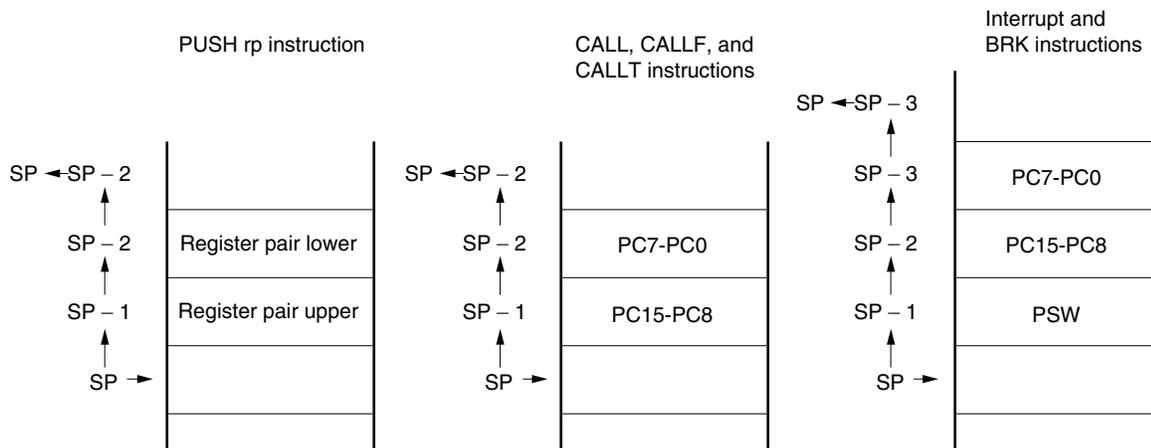
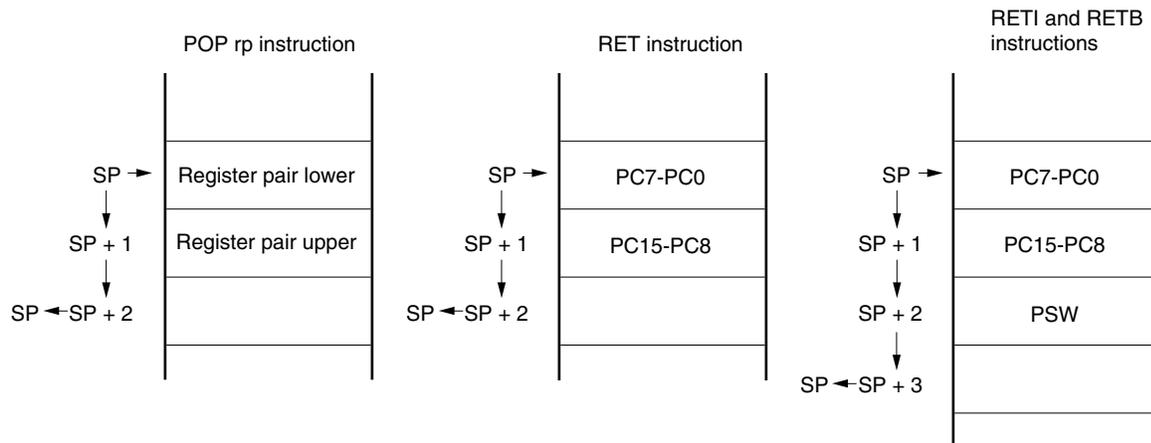


Figure 3-13. Data to Be Restored from Stack Memory



**3.2.2 General-purpose registers**

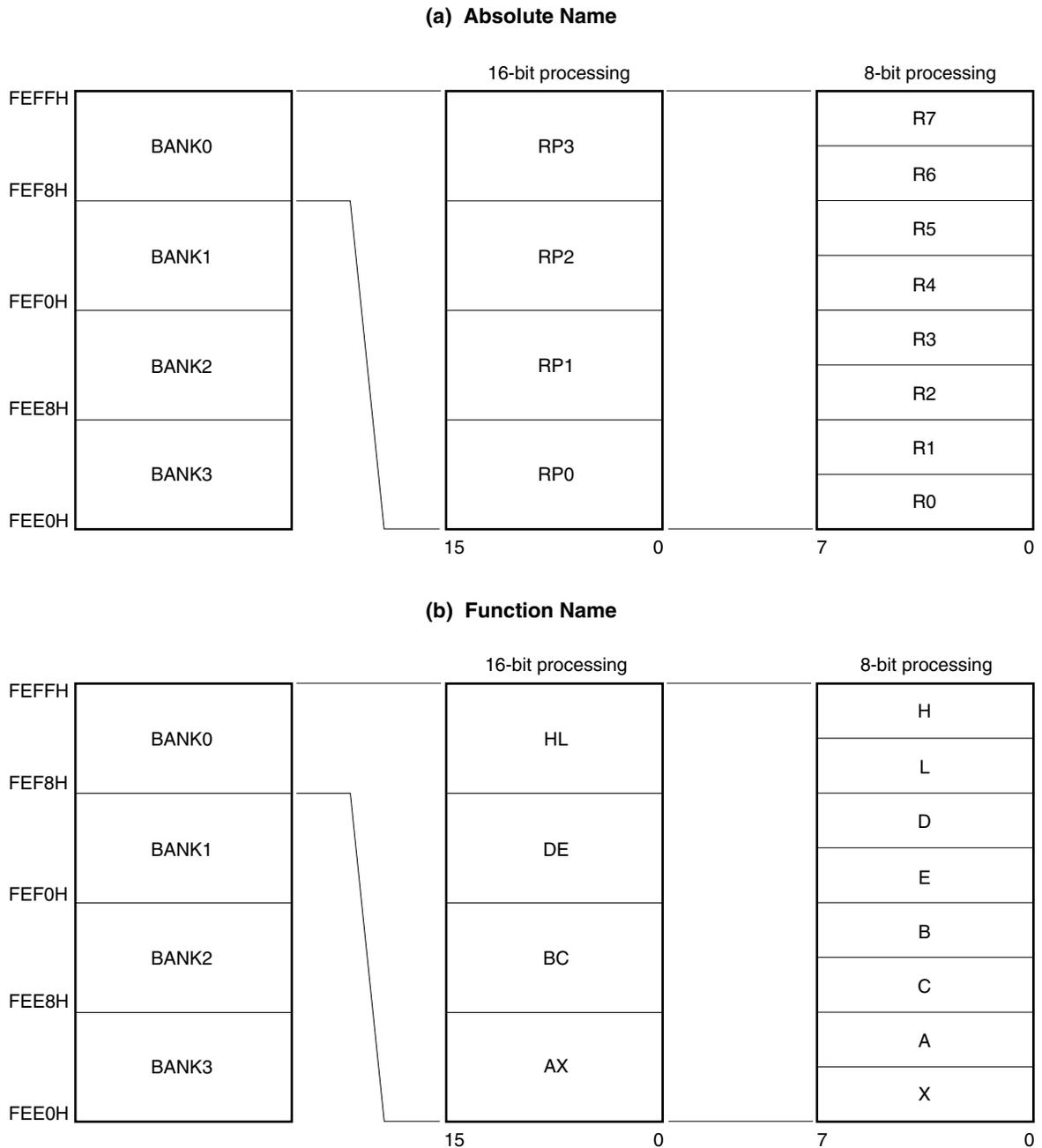
General-purpose registers are mapped at particular addresses (FEE0H to FEFFH) of the data memory. The general-purpose registers consists of 4 banks, each bank consisting of eight 8-bit registers (X, A, C, B, E, D, L, and H).

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

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

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

**Figure 3-14. Configuration of General-Purpose Registers**



### 3.2.3 Special Function Registers (SFRs)

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

SFRs are allocated in the FF00H to FFFFH area.

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

Each manipulation bit unit can be specified as follows.

- 1-bit manipulation  
Describe the symbol reserved by the assembler for the 1-bit manipulation instruction operand (sfr.bit).  
This manipulation can also be specified with an address.
- 8-bit manipulation  
Describe the symbol reserved by the assembler for the 8-bit manipulation instruction operand (sfr).  
This manipulation can also be specified with an address.
- 16-bit manipulation  
Describe the symbol reserved by the assembler for the 16-bit manipulation instruction operand (sfrp).  
When specifying an address, describe an even address.

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

- Symbol  
Symbol indicating the address of a special function register. It is a reserved word in the RA78K0, and is defined by the header file "sfrbit.h" in the CC78K0. When using the RA78K0, ID78K0-NS, ID78K0, or SM78K0, symbols can be written as an instruction operand.
- R/W  
Indicates whether the corresponding special function register can be read or written.  
R/W: Read/write enable  
R: Read only  
W: Write only
- Manipulatable bit units  
Indicates the manipulatable bit unit (1, 8, or 16). "—" indicates a bit unit for which manipulation is not possible.
- After reset  
Indicates each register status upon  $\overline{\text{RESET}}$  input.

Table 3-5. Special Function Register List (1/3)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset
				1 Bit	8 Bits	16 Bits	
FF00H	Port 0	P0	R/W	√	√	–	00H
FF01H	Port 1	P1	R/W	√	√	–	00H
FF02H	Port 2	P2	R	√	√	–	00H
FF03H	Port 3	P3	R/W	√	√	–	00H
FF08H	A/D conversion result register	ADCR	R	–	–	√	0000H
FF09H							
FF0AH	Receive buffer register 6	RXB6	R	–	√	–	FFH
FF0BH	Transmit buffer register 6	TXB6	R/W	–	√	–	FFH
FF0CH	Port 12	P12	R/W	√	√	–	00H
FF0DH	Port 13	P13	R/W	√	√	–	00H
FF0FH	Serial I/O shift register 10	SIO10	R	–	√	–	00H
FF10H	16-bit timer counter 00	TM00	R	–	–	√	0000H
FF11H							
FF12H	16-bit timer capture/compare register 000	CR000	R/W	–	–	√	0000H
FF13H							
FF14H	16-bit timer capture/compare register 010	CR010	R/W	–	–	√	0000H
FF15H							
FF16H	8-bit timer counter 50	TM50	R	–	√	–	00H
FF17H	8-bit timer compare register 50	CR50	R/W	–	√	–	00H
FF18H	8-bit timer H compare register 00	CMP00	R/W	–	√	–	00H
FF19H	8-bit timer H compare register 10	CMP10	R/W	–	√	–	00H
FF1AH	8-bit timer H compare register 01	CMP01	R/W	–	√	–	00H
FF1BH	8-bit timer H compare register 11	CMP11	R/W	–	√	–	00H
FF20H	Port mode register 0	PM0	R/W	√	√	–	FFH
FF21H	Port mode register 1	PM1	R/W	√	√	–	FFH
FF23H	Port mode register 3	PM3	R/W	√	√	–	FFH
FF28H	A/D converter mode register	ADM	R/W	√	√	–	00H
FF29H	Analog input channel specification register	ADS	R/W	√	√	–	00H
FF2AH	Power-fail comparison mode register	PFM	R/W	√	√	–	00H
FF2BH	Power-fail comparison threshold register	PFT	R/W	–	√	–	00H
FF2CH	Port mode register 12	PM12	R/W	√	√	–	FFH
FF30H	Pull-up resistor option register 0	PU0	R/W	√	√	–	00H
FF31H	Pull-up resistor option register 1	PU1	R/W	√	√	–	00H
FF33H	Pull-up resistor option register 3	PU3	R/W	√	√	–	00H
FF3CH	Pull-up resistor option register 12	PU12	R/W	√	√	–	00H
FF48H	External interrupt rising edge enable register	EGP	R/W	√	√	–	00H
FF49H	External interrupt falling edge enable register	EGN	R/W	√	√	–	00H
FF4FH	Input switch control register	ISC	R/W	√	√	–	00H

★

Table 3-5. Special Function Register List (2/3)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset
				1 Bit	8 Bits	16 Bits	
FF50H	Asynchronous serial interface operation mode register 6	ASIM6	R/W	√	√	–	01H
FF53H	Asynchronous serial interface reception error status register 6	ASIS6	R	–	√	–	00H
FF55H	Asynchronous serial interface transmission status register 6	ASIF6	R	–	√	–	00H
FF56H	Clock selection register 6	CKSR6	R/W	–	√	–	00H
FF57H	Baud rate generator control register 6	BRGC6	R/W	–	√	–	FFH
FF58H	Asynchronous serial interface control register 6	ASICL6	R/W	√	√	–	16H
FF69H	8-bit timer H mode register 0	TMHMD0	R/W	√	√	–	00H
FF6AH	Timer clock selection register 50	TCL50	R/W	–	√	–	00H
FF6BH	8-bit timer mode control register 50	TMC50	R/W	√	√	–	00H
FF6CH	8-bit timer H mode register 1	TMHMD1	R/W	√	√	–	00H
FF70H	Asynchronous serial interface operation mode register 0 <sup>Note 1</sup>	ASIM0	R/W	√	√	–	01H
FF71H	Baud rate generator control register 0 <sup>Note 1</sup>	BRGC0	R/W	–	√	–	1FH
FF72H	Receive buffer register 0 <sup>Note 1</sup>	RXB0	R	–	√	–	FFH
FF73H	Asynchronous serial interface reception error status register 0 <sup>Note 1</sup>	ASIS0	R	–	√	–	00H
FF74H	Transmit shift register 0 <sup>Note 1</sup>	TXS0	W	–	√	–	FFH
FF80H	Serial operation mode register 10	CSIM10	R/W	√	√	–	00H
FF81H	Serial clock selection register 10	CSIC10	R/W	√	√	–	00H
FF84H	Transmit buffer register 10	SOTB10	R/W	–	√	–	Undefined
FF98H	Watchdog timer mode register	WDTM	R/W	–	√	–	67H
FF99H	Watchdog timer enable register	WDTE	R/W	–	√	–	9AH
FFA0H	Ring-OSC mode register	RCM	R/W	√	√	–	00H
FFA1H	Main clock mode register	MCM	R/W	√	√	–	00H
FFA2H	Main OSC control register	MOC	R/W	√	√	–	00H
★ FFA3H	Oscillation stabilization time counter status register	OSTC	R	√	√	–	00H
FFA4H	Oscillation stabilization time select register	OSTS	R/W	–	√	–	05H
FFA9H	Clock monitor mode register	CLM	R/W	√	√	–	00H
FFACH	Reset control flag register	RESF	R	–	√	–	00H <sup>Note 2</sup>
FFBAH	16-bit timer mode control register 00	TMC00	R/W	√	√	–	00H
FFBBH	Prescaler mode register 00	PRM00	R/W	√	√	–	00H

- Notes**
1.  $\mu$ PD780102, 780103, and 78F0103 only.
  2. This value varies depending on the reset source.

Table 3-5. Special Function Register List (3/3)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset
					1 Bit	8 Bits	16 Bits	
FFBCH	Capture/compare control register 00	CRC00		R/W	√	√	–	00H
FFBDH	16-bit timer output control register 00	TOC00		R/W	√	√	–	00H
FFBEH	Low-voltage detection register	LVIM		R/W	√	√	–	00H
FFBFH	Low-voltage detection level selection register	LVIS		R/W	–	√	–	00H
FFE0H	Interrupt request flag register 0L	IF0	IF0L	R/W	√	√	√	00H
FFE1H	Interrupt request flag register 0H		IF0H	R/W	√	√		00H
★ FFE2H	Interrupt request flag register 1L	1F1L		R/W	√	√	–	00H
FFE4H	Interrupt mask flag register 0L	MK0	MK0L	R/W	√	√	√	FFH
FFE5H	Interrupt mask flag register 0H		MK0H	R/W	√	√		FFH
★ FFE6H	Interrupt mask flag register 1L	MK1L		R/W	√	√	–	FFH
FFE8H	Priority specification flag register 0L	PR0	PR0L	R/W	√	√	√	FFH
FFE9H	Priority specification flag register 0H		PR0H	R/W	√	√		FFH
★ FFEAH	Priority specification flag register 1L	PR1L		R/W	√	√	–	FFH
FFF0H	Internal memory size switching register <sup>Note</sup>	IMS		R/W	–	√	–	CFH
FFFBH	Processor clock control register	PCC		R/W	√	√	–	00H

**Note** The default value of IMS is fixed (CFH) in all products in the 78K0/KB1 Series regardless of the internal memory capacity.

Therefore, set the following value to each product.

	Internal Memory Size Switching Register (IMS)
μPD780101	42H
μPD780102	04H
μPD780103	06H
μPD78F0103	Value corresponding to mask ROM version

### 3.3 Instruction Address Addressing

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

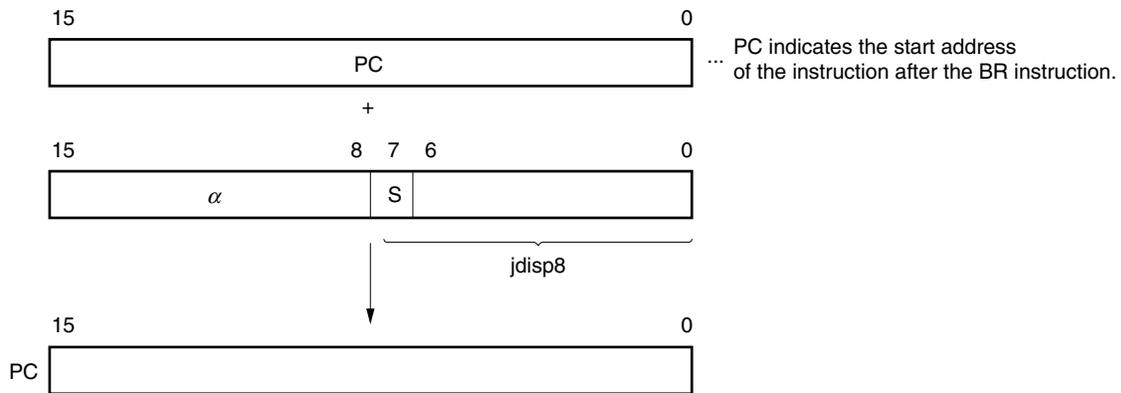
#### 3.3.1 Relative addressing

**[Function]**

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

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

**[Illustration]**



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

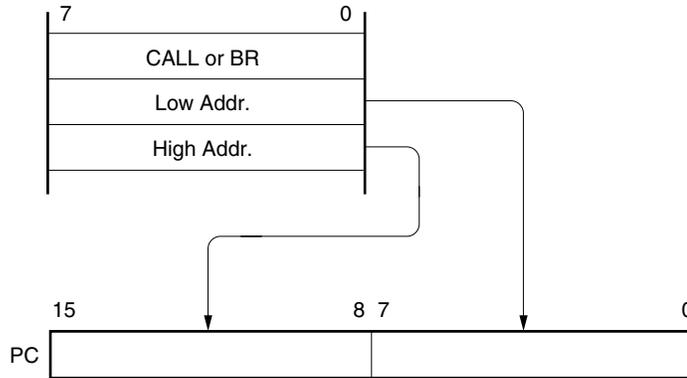
3.3.2 Immediate addressing

[Function]

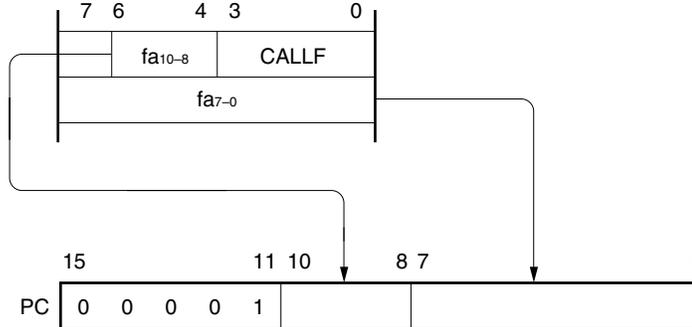
Immediate data in the instruction word is transferred to the program counter (PC) and branched. This function is carried out when the CALL !addr16 or BR !addr16 or CALLF !addr11 instruction is executed. CALL !addr16 and BR !addr16 instructions can be branched to the entire memory space. The CALLF !addr11 instruction is branched to the 0800H to 0FFFH area.

[Illustration]

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



In the case of CALLF !addr11 instruction



3.3.3 Table indirect addressing

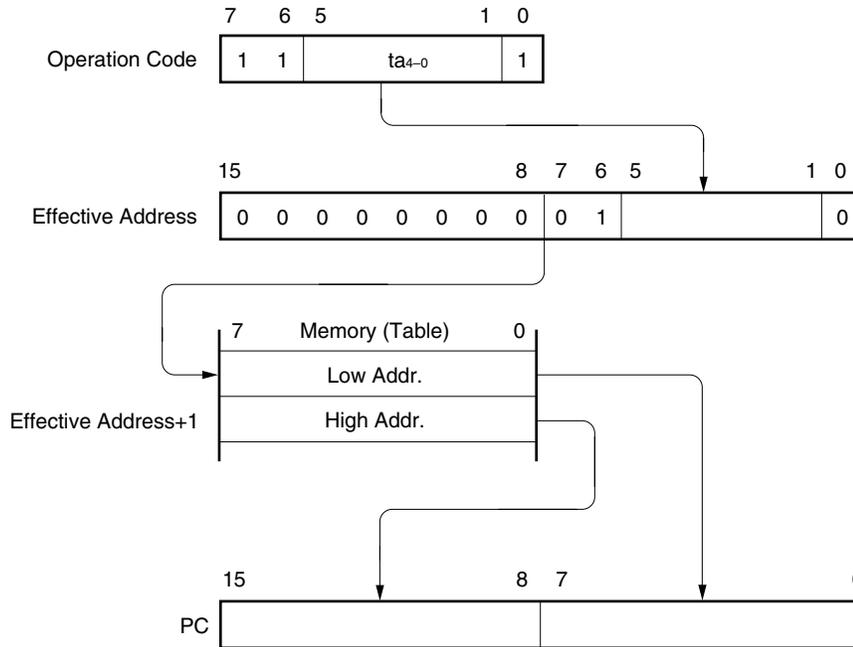
[Function]

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

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

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

[Illustration]



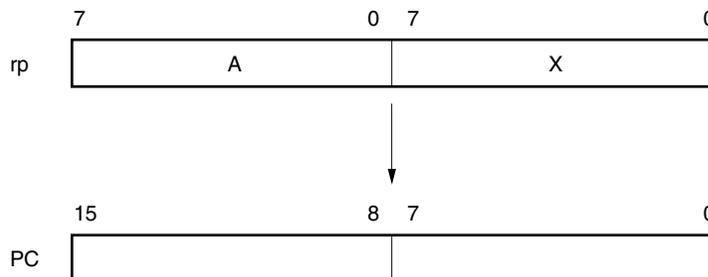
3.3.4 Register addressing

[Function]

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

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

[Illustration]



### 3.4 Operand Address Addressing

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

#### 3.4.1 Implied addressing

##### [Function]

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

Of the 78K0/KB1 Series instruction words, the following instructions employ implied addressing.

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

##### [Operand format]

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

##### [Description example]

In the case of MULU X

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

3.4.2 Register addressing

[Function]

The general-purpose register to be specified is accessed as an operand with the register specify code (Rn and RPN) of an instruction word in the registered bank specified with the register bank select flag (RBS0 to RBS1).

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

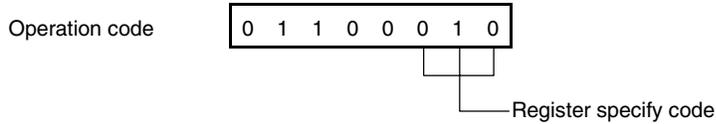
[Operand format]

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

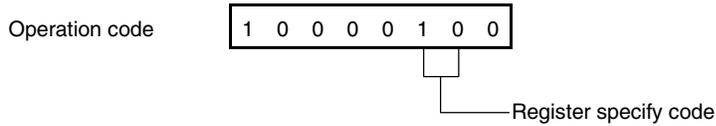
'r' and 'rp' can be described by absolute names (R0 to R7 and RP0 to RP3) as well as function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL).

[Description example]

MOV A, C; when selecting C register as r



INCW DE; when selecting DE register pair as rp



### 3.4.3 Direct addressing

**[Function]**

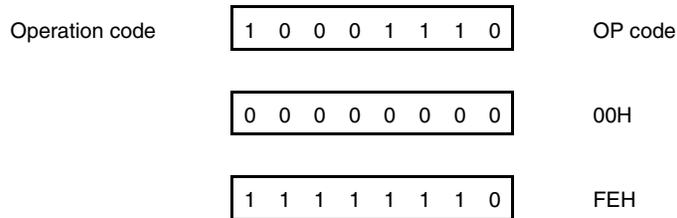
The memory to be manipulated is addressed with immediate data in an instruction word becoming an operand address.

**[Operand format]**

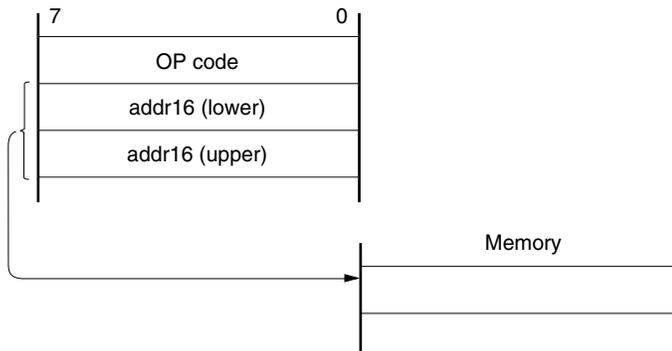
Identifier	Description
addr16	Label or 16-bit immediate data

**[Description example]**

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



**[Illustration]**



### 3.4.4 Short direct addressing

**[Function]**

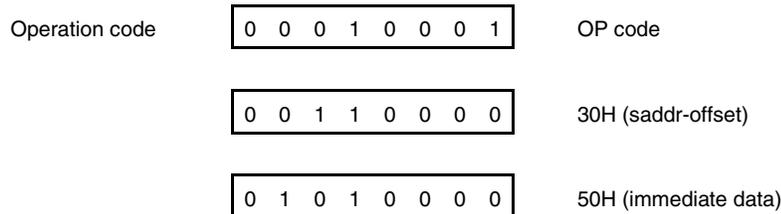
The memory to be manipulated in the fixed space is directly addressed with 8-bit data in an instruction word. This addressing is applied to the 256-byte space FE20H to FF1FH. Internal RAM and special function registers (SFR) are mapped at FE20H to FEFFH and FF00H to FF1FH, respectively. The SFR area (FF00H to FF1FH) where short direct addressing is applied is a part of the overall SFR area. Ports that are frequently accessed in a program and compare and capture registers of the timer/event counter are mapped in this area, allowing SFRs to be manipulated with a small number of bytes and clocks. When 8-bit immediate data is at 20H to FFH, bit 8 of an effective address is set to 0. When it is at 00H to 1FH, bit 8 is set to 1. Refer to the **[Illustration]** shown below.

**[Operand format]**

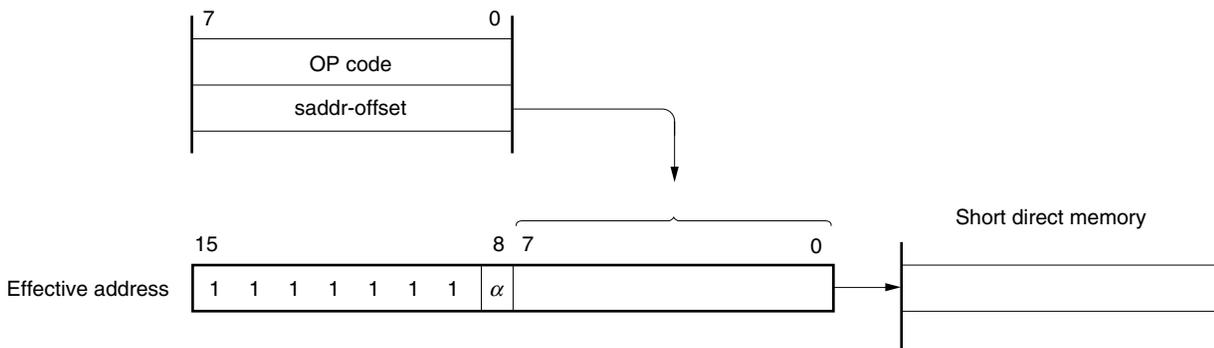
Identifier	Description
saddr	Label or FE20H to FF1FH immediate data
saddrp	Label or FE20H to FF1FH immediate data (even address only)

**[Description example]**

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



**[Illustration]**



When 8-bit immediate data is 20H to FFH,  $\alpha = 0$   
 When 8-bit immediate data is 00H to 1FH,  $\alpha = 1$



3.4.6 Register indirect addressing

[Function]

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

[Operand format]

Identifier	Description
-	[DE], [HL]

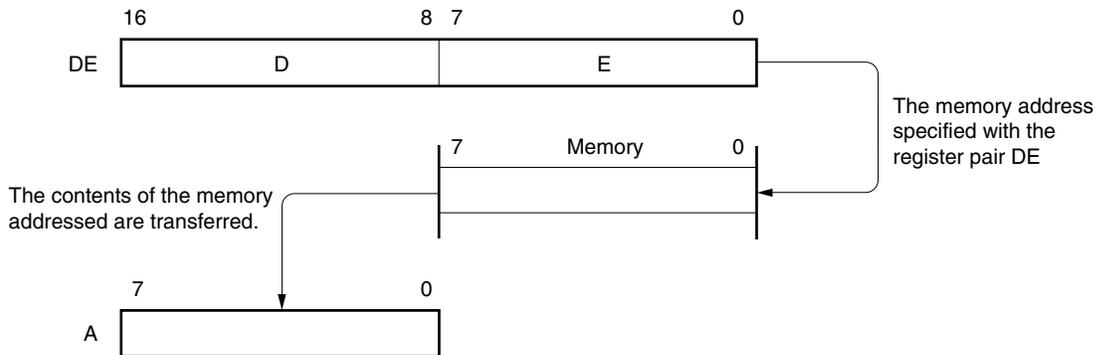
[Description example]

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

Operation code 

1	0	0	0	0	1	0	1
---	---	---	---	---	---	---	---

[Illustration]



### 3.4.7 Based addressing

#### [Function]

8-bit immediate data is added as offset data to the contents of the base register, that is, the HL register pair in an instruction word of the register bank specified with the register bank select flag (RBS0 and RBS1) and the sum is used to address the memory. Addition is performed by expanding the offset data as a positive number to 16 bits. A carry from the 16th bit is ignored. This addressing can be carried out for all the memory spaces.

#### [Operand format]

Identifier	Description
–	[HL + byte]

#### [Description example]

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

Operation code

1	0	1	0	1	1	1	0
---	---	---	---	---	---	---	---

0	0	0	1	0	0	0	0
---	---	---	---	---	---	---	---

### 3.4.8 Based indexed addressing

**[Function]**

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

**[Operand format]**

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

**[Description example]**

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

Operation code

1 0 1 0 1 0 1 1
-----------------

### 3.4.9 Stack addressing

**[Function]**

The stack area is indirectly addressed with the stack pointer (SP) contents. This addressing method is automatically employed when the PUSH, POP, subroutine call and return instructions are executed or the register is saved/reset upon generation of an interrupt request. With stack addressing, only the internal high-speed RAM area can be addressed.

**[Description example]**

In the case of PUSH DE

Operation code

1 0 1 1 0 1 0 1
-----------------

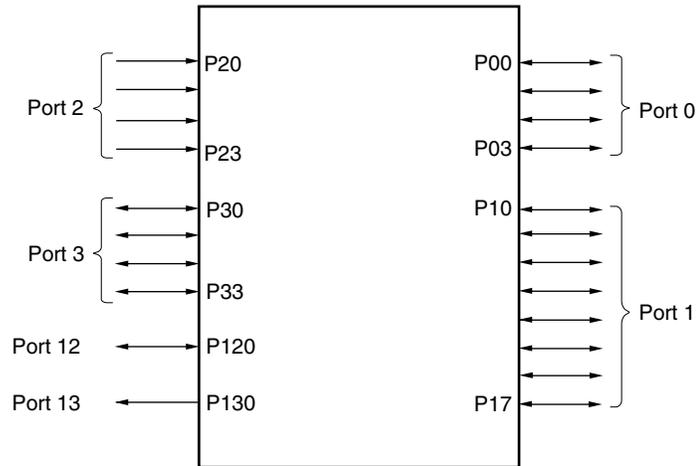
## CHAPTER 4 PORT FUNCTIONS

### 4.1 Port Functions

78K0/KB1 Series products are provided with the ports shown in Figure 4-1, which enable variety of control operations. The functions of each port are shown in Table 4-1.

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

Figure 4-1. Port Types



**Table 4-1. Port Functions**

Pin Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	TI000
P01				TI010/TO00
P02				–
P03				–
P10	I/O	Port 1. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	SCK10/TxD0 <sup>Note</sup>
P11				SI10/RxD0 <sup>Note</sup>
P12				SO10
P13				TxD6
P14				RxD6
P15				TOH0
P16				TOH1/INTP5
P17				TI50/TO50
P20 to P23	Input	Port 2. 4-bit input-only port.	Input	ANI0 to ANI3
P30 to P33	I/O	Port 3. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	INTP1 to INTP4
P120	I/O	Port 12. 1-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	INTP0
P130	Output	Port 13. 1-bit output-only port.	Output	–

**Note** TxD0 and RxD0 are available only in the  $\mu$ PD780102, 780103, and 78F0103.

## 4.2 Port Configuration

A port consists of the following hardware.

**Table 4-2. Port Configuration**

Item	Configuration
Control registers	Port mode register (PM0, PM1, PM3, PM12) Pull-up resistor option register (PU0, PU1, PU3, PU12) Input switch control register (ISC)
Port	Total: 22 (CMOS I/O: 17, CMOS input: 4, CMOS output: 1)
Pull-up resistor	Total: 22 (software control only)

4.2.1 Port 0

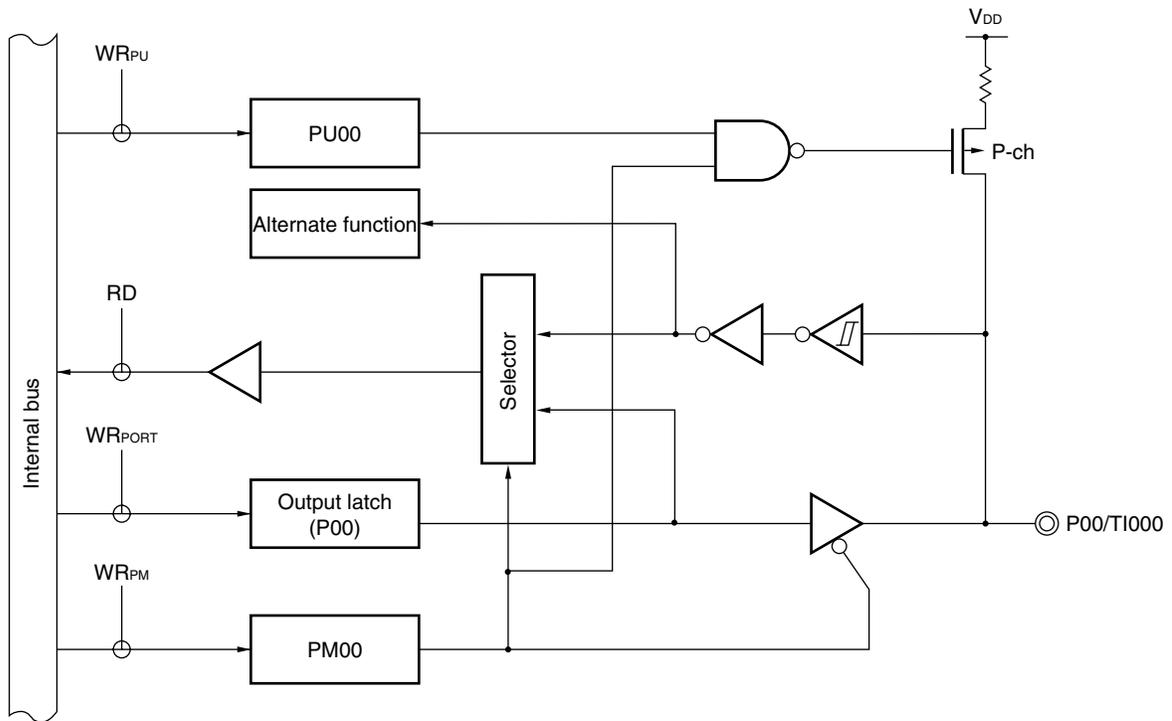
Port 0 is a 4-bit I/O port with an output latch. Port 0 can be set to the input mode or output mode in 1-bit units using port mode register 0 (PM0). When the P00 to P03 pins are used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 0 (PU0).

This port can also be used for timer I/O.

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

Figures 4-2 to 4-4 show block diagrams of port 0.

Figure 4-2. Block Diagram of P00



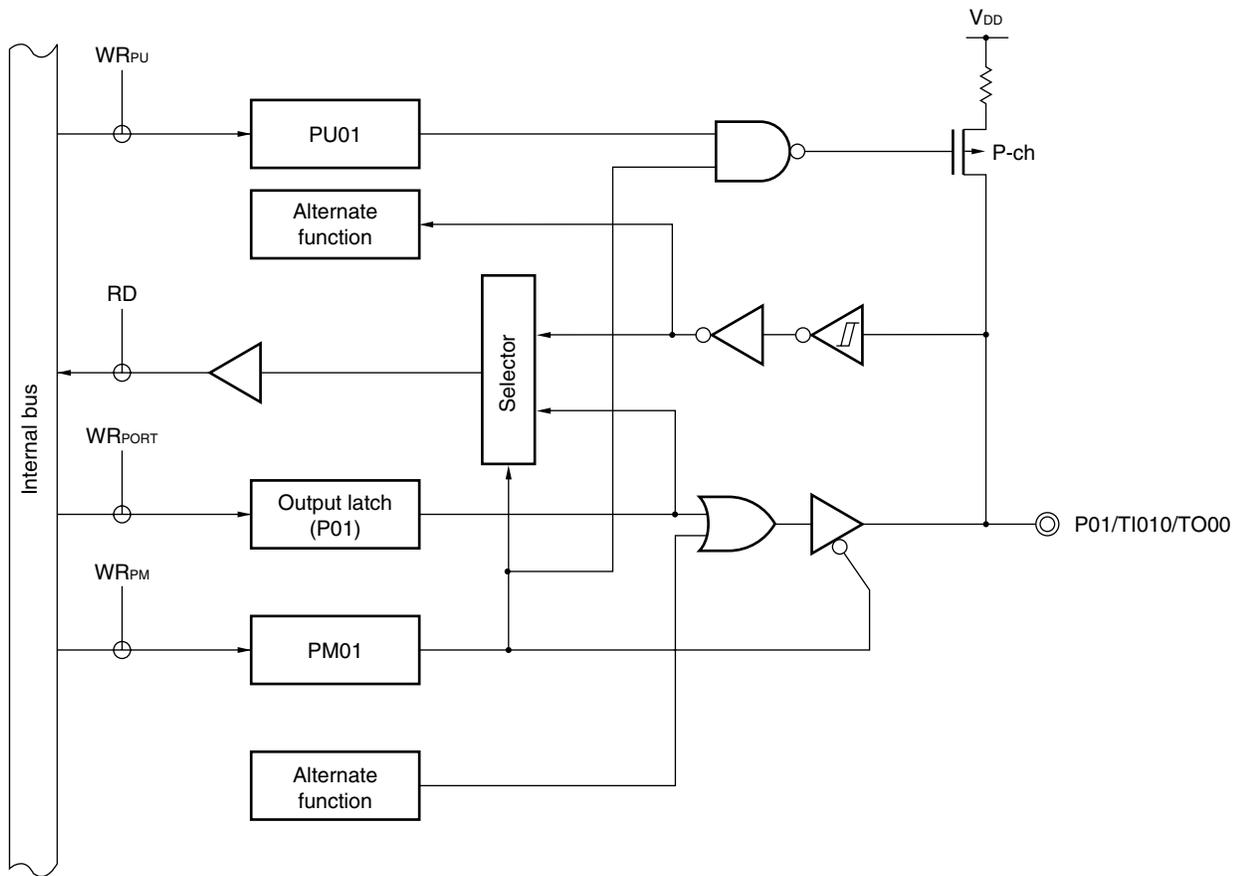
PU0: Pull-up resistor option register 0

PM: Port mode register

RD: Port 0 read signal

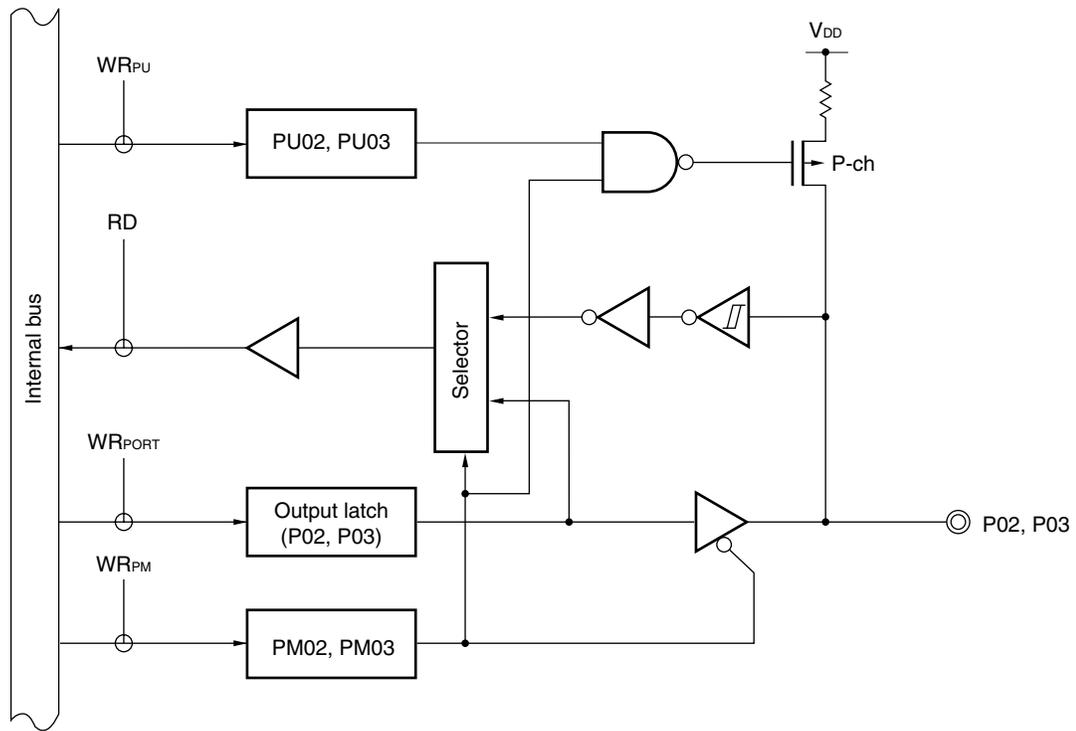
WR: Port 0 write signal

Figure 4-3. Block Diagram of P01



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

Figure 4-4. Block Diagram of P02 and P03



- 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

Port 1 is an 8-bit I/O port with an output latch. Port 1 can be set to the input mode or output mode in 1-bit units using port mode register 1 (PM1). When the P10 to P17 pins are used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 1 (PU1).

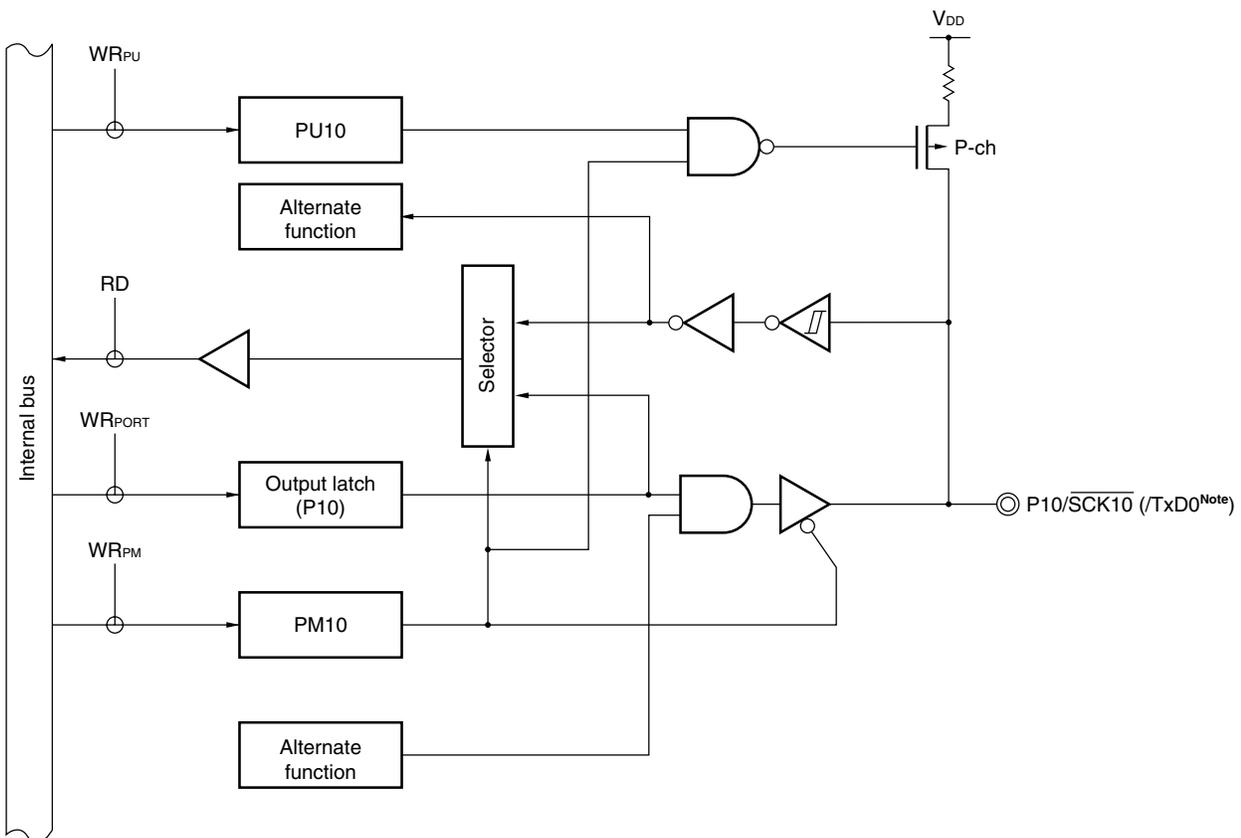
This port can also be used for external interrupt request input, serial interface data I/O, clock I/O, and timer I/O.

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

Figures 4-5 to 4-10 show block diagrams of port 1.

★ **Caution** When using P10/ $\overline{\text{SCK10}}$  ( $\text{TxD0}^{\text{Note}}$ ), P11/ $\overline{\text{SI10}}$  ( $\text{RxD0}^{\text{Note}}$ ), and P12/ $\overline{\text{SO10}}$  as general-purpose ports, do not write to serial clock selection register 10 (CSIC10).

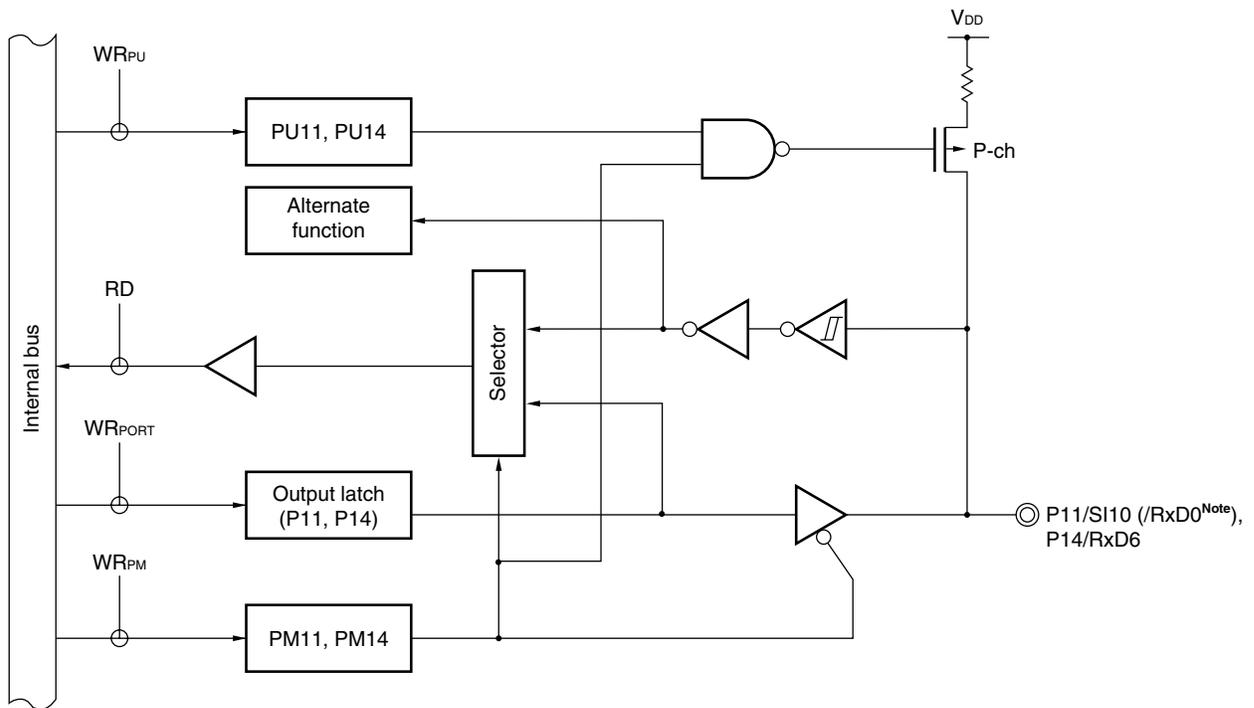
★ **Figure 4-5. Block Diagram of P10**



**Note** Available only in the  $\mu\text{PD780102}$ ,  $\text{780103}$ , and  $\text{78F0103}$ .

- PU1: Pull-up resistor option register 1
- PM: Port mode register
- RD: Port 1 read signal
- WR: Port 1 write signal

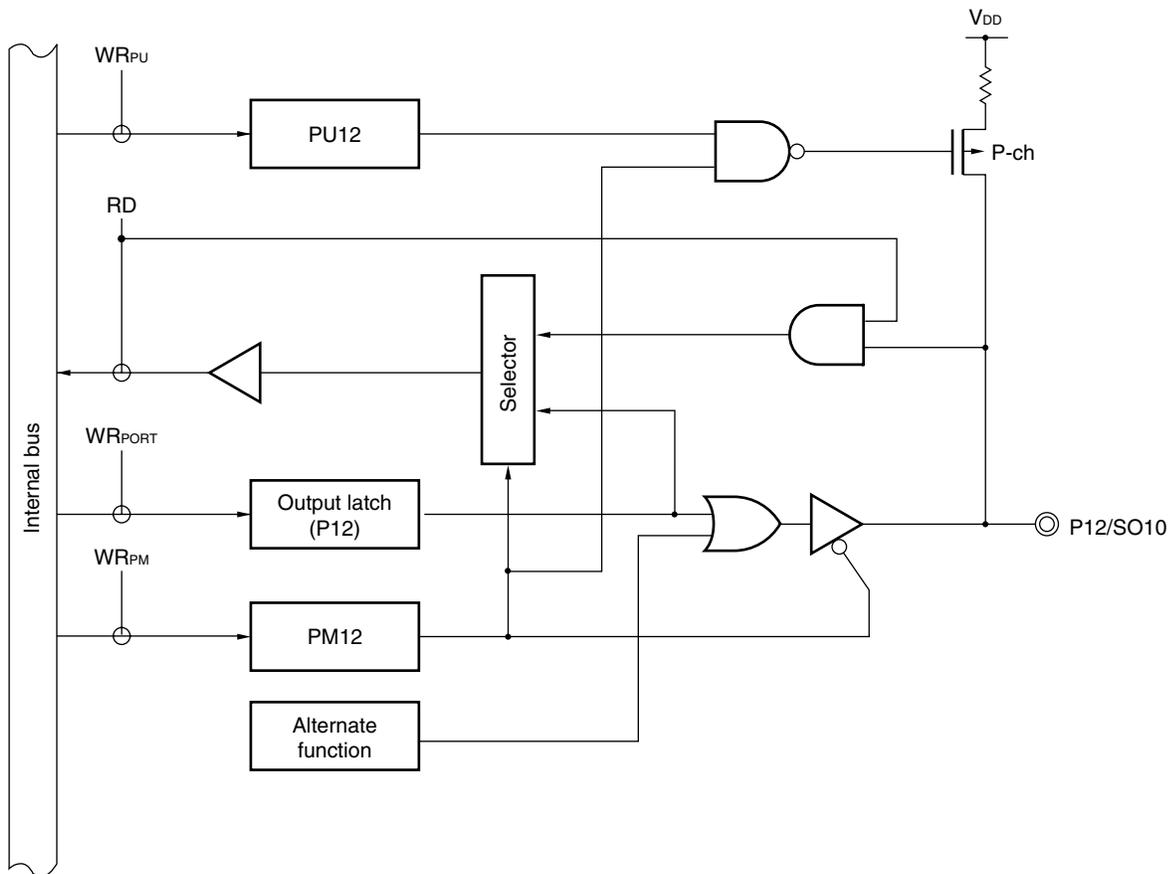
Figure 4-6. Block Diagram of P11 and P14



**Note** Available only in the  $\mu$ PD780102, 780103, and 78F0103.

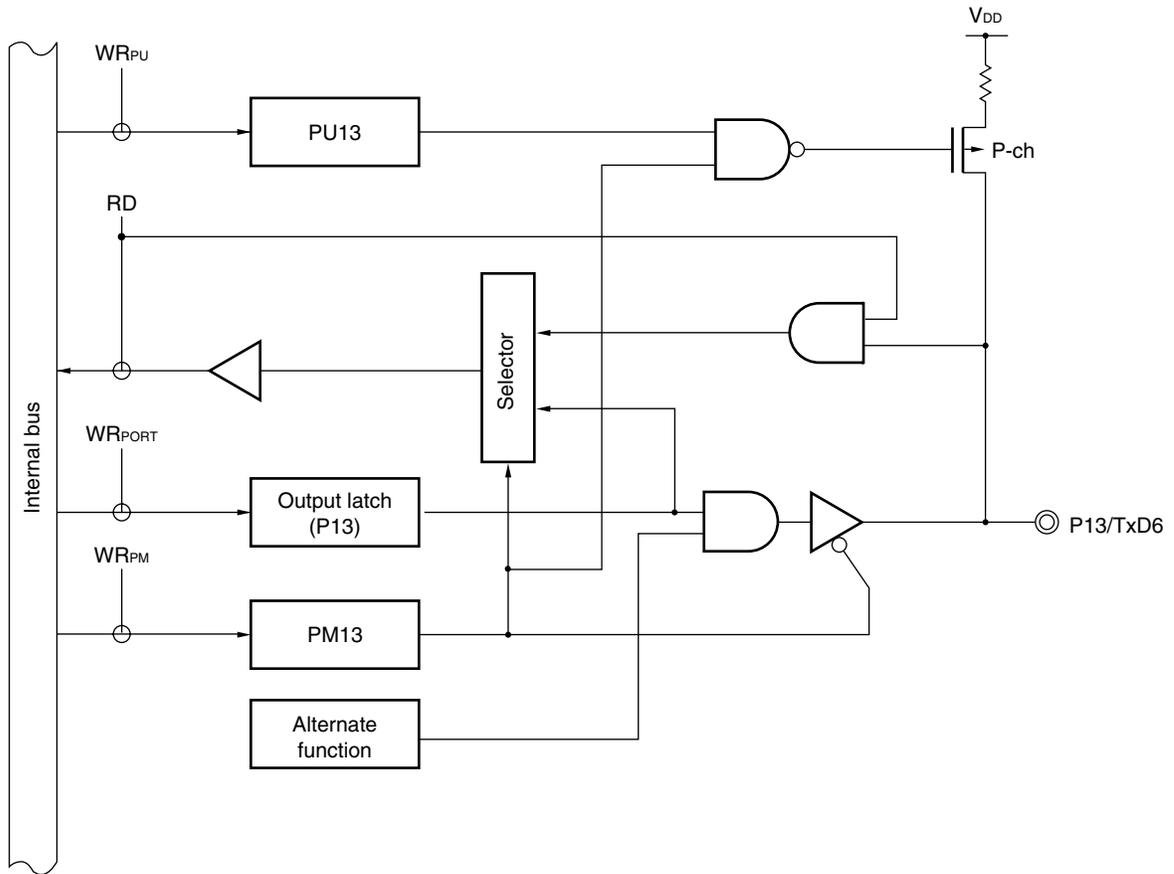
- PU1: Pull-up resistor option register 1
- PM: Port mode register
- RD: Port 1 read signal
- WR: Port 1 write signal

Figure 4-7. Block Diagram of P12



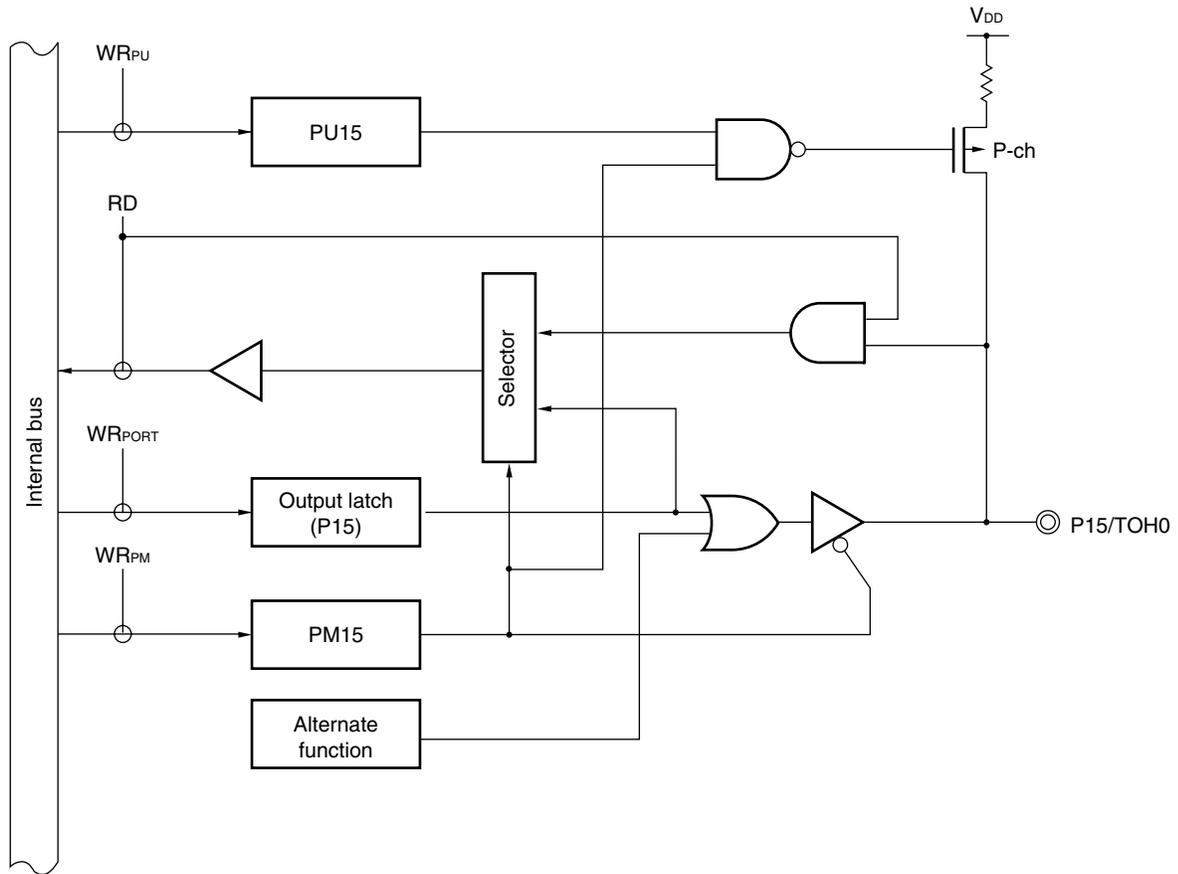
- PU1: Pull-up resistor option register 1
- PM: Port mode register
- RD: Port 1 read signal
- WR: Port 1 write signal

Figure 4-8. Block Diagram of P13



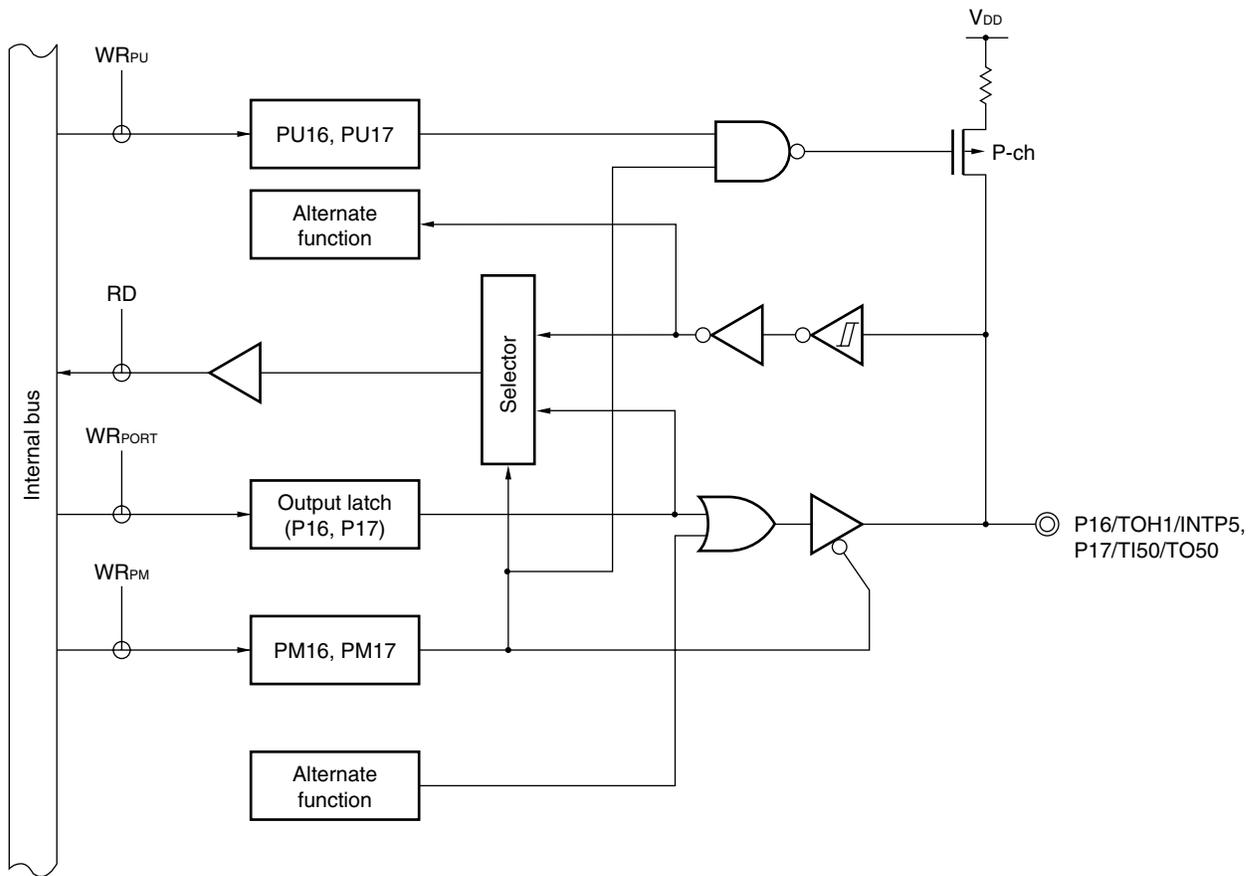
- PU1: Pull-up resistor option register 1
- PM: Port mode register
- RD: Port 1 read signal
- WR: Port 1 write signal

Figure 4-9. Block Diagram of P15



- PU1: Pull-up resistor option register 1
- PM: Port mode register
- RD: Port 1 read signal
- WR: Port 1 write signal

Figure 4-10. Block Diagram of P16 and P17



- PU1: Pull-up resistor option register 1
- PM: Port mode register
- RD: Port 1 read signal
- WR: Port 1 write signal

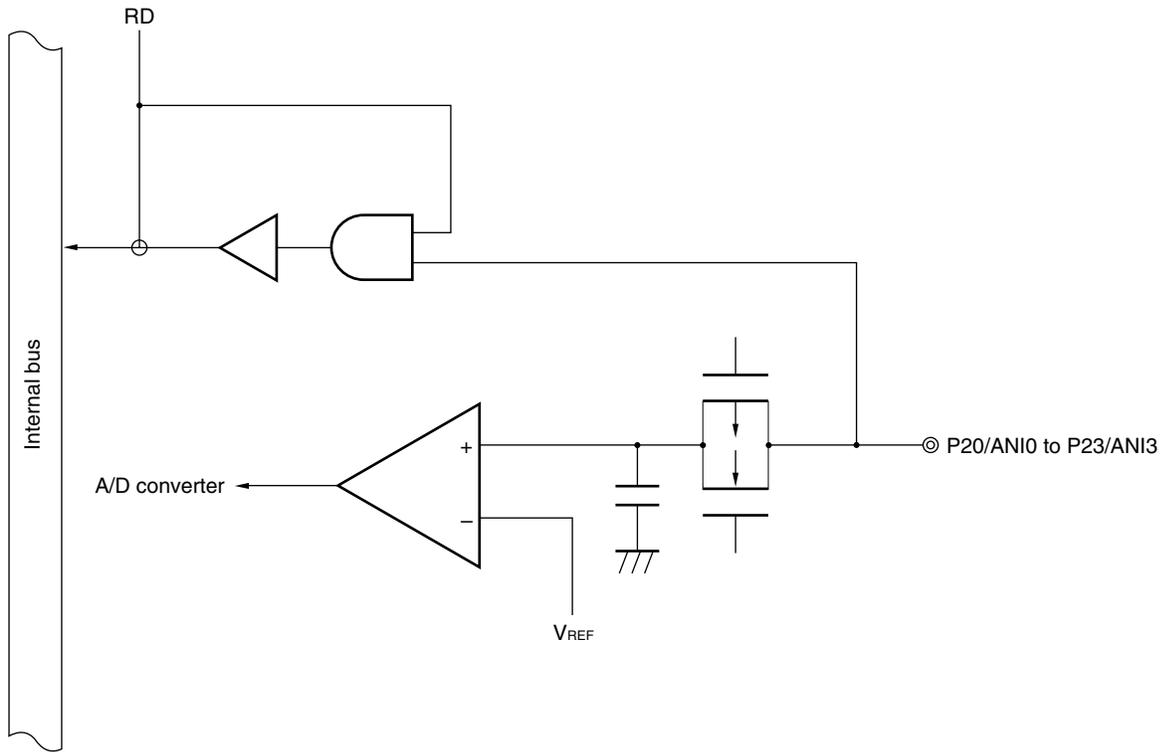
4.2.3 Port 2

Port 2 is a 4-bit input-only port.

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

Figure 4-11 shows a block diagram of port 2.

Figure 4-11. Block Diagram of P20 to P23



RD: Port 2 read signal

4.2.4 Port 3

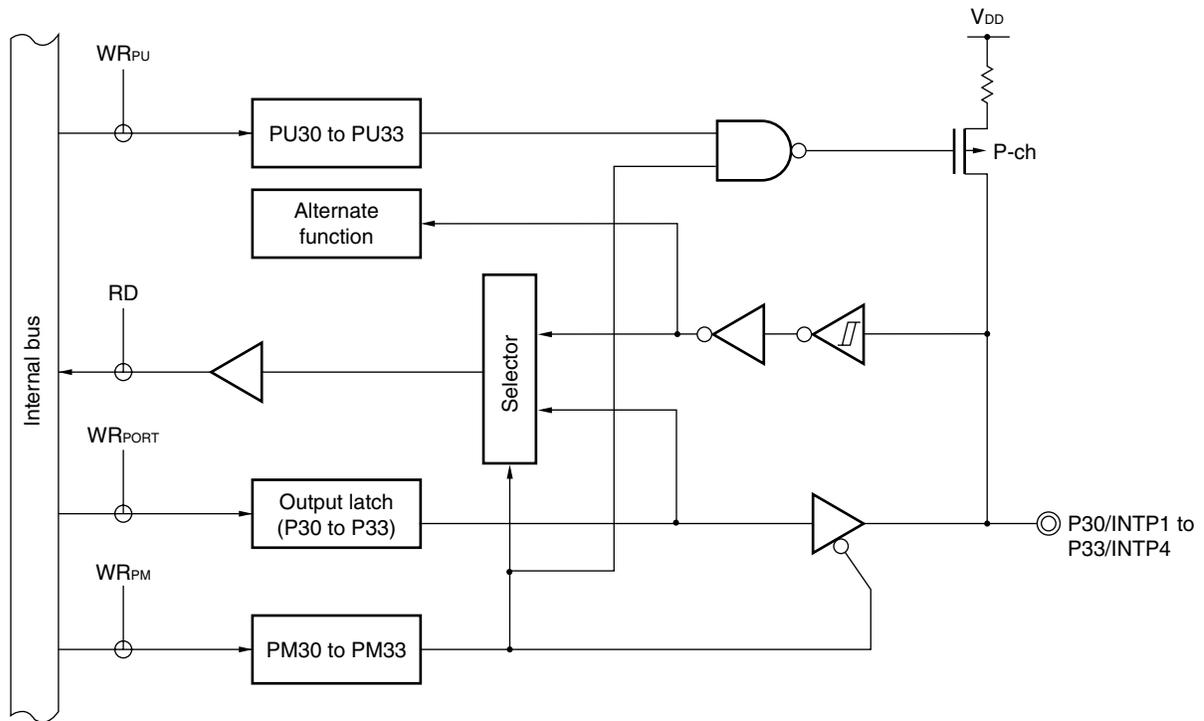
Port 3 is a 4-bit I/O port with an output latch. Port 3 can be set to the input mode or output mode in 1-bit units using port mode register 3 (PM3). When the P30 to P33 pins are used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 3 (PU3).

This port can also be used for external interrupt request input.

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

Figure 4-12 shows a block diagram of port 3.

Figure 4-12. Block Diagram of P30 to P33



PU3: Pull-up resistor option register 3

PM: Port mode register

RD: Port 3 read signal

WR: Port 3 write signal

4.2.5 Port 12

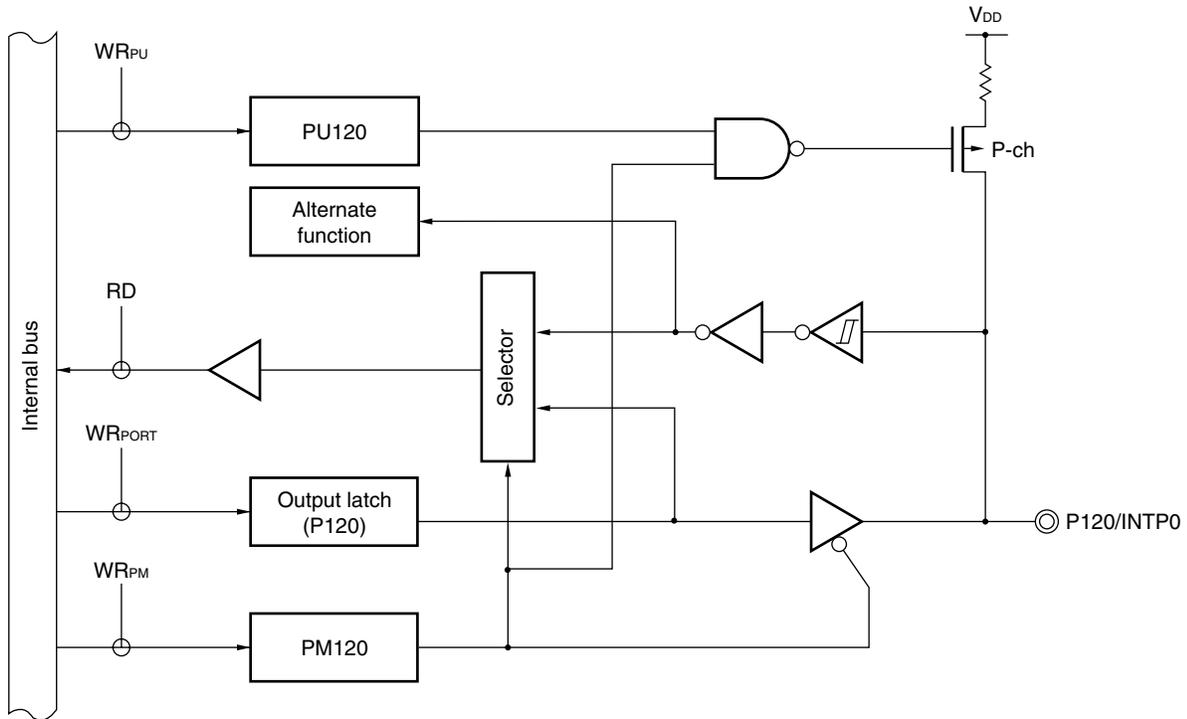
Port 12 is a 1-bit I/O port with an output latch. Port 12 can be set to the input mode or output mode in 1-bit units using port mode register 12 (PM12). When the P120 pin is used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 12 (PU12).

This port can also be used for external interrupt input.

$\overline{\text{RESET}}$  input sets port 12 to input mode.

Figure 4-13 shows a block diagram of port 12.

Figure 4-13. Block Diagram of P120



PU12: Pull-up resistor option register 12

PM: Port mode register

RD: Port 12 read signal

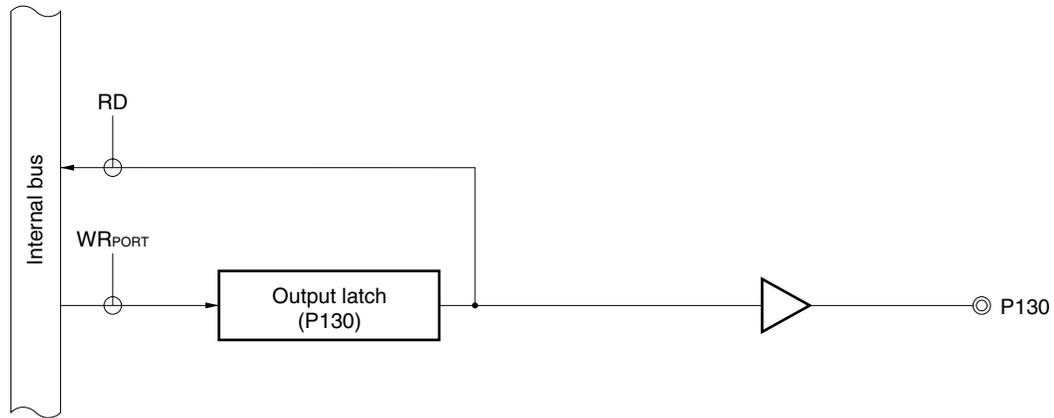
WR: Port 12 write signal

### 4.2.6 Port 13

Port 13 is a 1-bit output-only port.

Figure 4-14 shows a block diagram of port 13.

Figure 4-14. Block Diagram of P130



RD: Port 13 read signal

WR: Port 13 write signal

## 4.3 Registers Controlling Port Function

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

- Port mode registers (PM0, PM1, PM3, PM12)
- Pull-up resistor option registers (PU0, PU1, PU3, PU12)
- Input switch control register (ISC)

### (1) Port mode registers (PM0, PM1, PM3, and PM12)

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

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

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

When port pins are used as alternate-function pins, set the port mode register and output latch as shown in Table 4-3.

**Caution** Because P16, P30 to P33, and P120 can also be used as external interrupt input pins, when port function output mode is specified to change the output level, the interrupt request flag is set. Therefore, when these pins are used in output mode, preset the interrupt mask flags (PMK0 to PMK5) to 1.

Figure 4-15. Format of Port Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	1	1	1	PM03	PM02	PM01	PM00	FF20H	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FF21H	FFH	R/W
PM3	1	1	1	1	PM33	PM32	PM31	PM30	FF23H	FFH	R/W
PM12	1	1	1	1	1	1	1	PM120	FF2CH	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0, 1, 3, 12; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

**Table 4-3. Settings of Port Mode Register and Output Latch When Alternate-Function Is Used**

Pin Name	Alternate Function		PM <sub>xx</sub>	P <sub>xx</sub>
	Name	I/O		
P00	TI000	Input	1	×
P01	TI010	Input	1	×
	TO00	Output	0	0
P10	SCK10	Input	1	×
		Output	0	1
	TxD0 <sup>Note</sup>	Output	0	1
P11	SI10	Input	1	×
	RxD0 <sup>Note</sup>	Input	1	×
P12	SO10	Output	0	0
P13	TxD6	Output	0	1
P14	RxD6	Input	1	×
P15	TOH0	Output	0	0
P16	TOH1	Output	0	0
	INTP5	Input	1	×
P17	TI50	Input	1	×
	TO50	Output	0	0
P30 to P33	INTP1 to INTP4	Input	1	×
P120	INTP0	Input	1	×

**Note** TxD0 and RxD0 are available only in the  $\mu$ PD780102, 780103, and 78F0103.

**Remark** ×: Don't care  
 PM<sub>xx</sub>: Port mode register  
 P<sub>xx</sub>: Port output latch

**(2) Pull-up resistor option registers (PU0, PU1, PU3, and PU12)**

These registers specify whether the on-chip pull-up resistors of P00 to P03, P10 to P17, P30 to P33, or P120 is to be used or not. An on-chip pull-up resistors can be used in 1-bit units only for the bits set to input mode of the pins to which the use of an on-chip pull-up resistor has been specified. On-chip pull-up resistor cannot be used for bits set to output mode and bits used as alternate-function output pins, regardless of the settings of PU0, PU1, PU3 and PU12.

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

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

**Figure 4-16. Format of Pull-up Resistor Option Register**

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	0	0	0	PU03	PU02	PU01	PU00	FF30H	00H	R/W
PU1	7	6	5	4	3	2	1	0	FF31H	00H	R/W
PU3	7	6	5	4	3	2	1	0	FF33H	00H	R/W
PU12	7	6	5	4	3	2	1	0	FF3CH	00H	R/W

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

**(3) Input switch control register (ISC)**

This register is used to receive a status signal transmitted from the master during LIN (Local Interconnect Network) reception. The input signal is switched by setting ISC.

For the port configuration during LIN reception, refer to **Figure 12-3 Port Configuration for LIN Reception Operation** in **CHAPTER 12 SERIAL INTERFACE UART6**.

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

RESET input clears this register to 00H.

**Figure 4-17. Format of Input Switch Control Register (ISC)**

Address: FF4FH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	0	0	0	0	0	0	ISC1	ISC0

ISC1	Input signal selection
0	TI000/P00 input
1	RxD6/P14 input

ISC0	Input signal selection
0	INTP0/P120 input
1	RxD6/P14 input

## 4.4 Port Function Operations

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

### 4.4.1 Writing to I/O port

#### (1) Output mode

A value is written to the output latch by a transfer instruction, and the output latch contents are output from the pin.

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

#### (2) Input mode

A value is written to the output latch by a transfer instruction, but since the output buffer is off, the pin status does not change.

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

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

### 4.4.2 Reading from I/O port

#### (1) Output mode

The output latch contents are read by a transfer instruction. The output latch contents do not change.

#### (2) Input mode

The pin status is read by a transfer instruction. The output latch contents do not change.

### 4.4.3 Operations on I/O port

#### (1) Output mode

An operation is performed on the output latch contents, and the result is written to the output latch. The output latch contents are output from the pins.

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

#### (2) Input mode

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

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

## CHAPTER 5 CLOCK GENERATOR

### 5.1 Functions of Clock Generator

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

- X1 oscillator  
The X1 oscillator oscillates a clock of 2.0 to 10.0 MHz. Oscillation can be stopped by executing the STOP instruction or setting the main OSC control register (MOC).
- Ring-OSC oscillator  
The Ring-OSC oscillator oscillates a clock of 240 kHz (TYP.). Oscillation can be stopped by setting the Ring-OSC mode register (RCM) when “Can be stopped by software” is set by a mask option and the X1 input clock is used as the CPU clock.

### 5.2 Configuration of Clock Generator

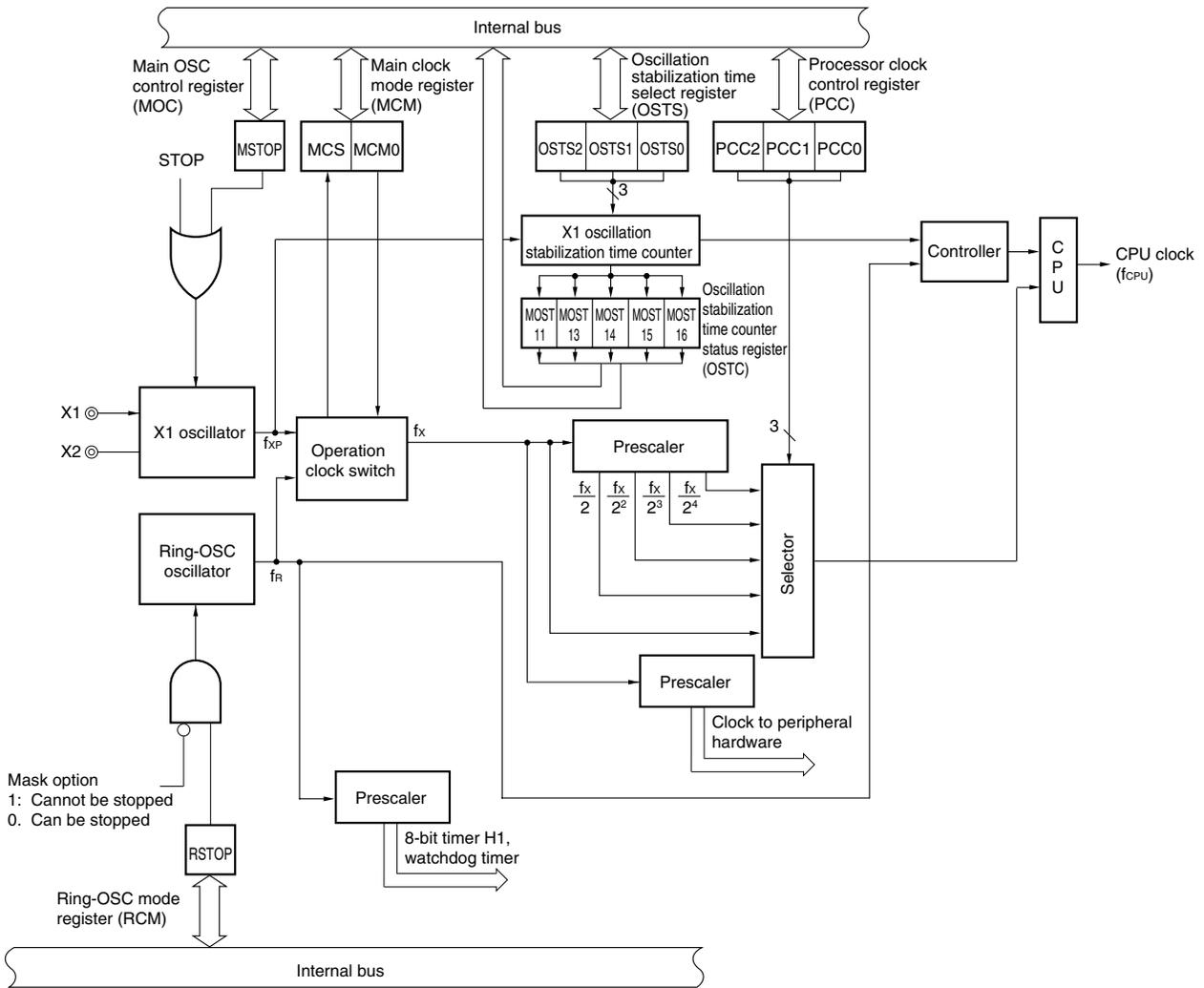
The clock generator consists of the following hardware.

**Table 5-1. Configuration of Clock Generator**

Item	Configuration
Control registers	Processor clock control register (PCC) Ring-OSC mode register (RCM) Main clock mode register (MCM) Main OSC control register (MOC) Oscillation stabilization time counter status register (OSTC) Oscillation stabilization time select register (OSTS)
Oscillator	X1 oscillator Ring-OSC oscillator

★

Figure 5-1. Block Diagram of Clock Generator



### 5.3 Registers Controlling Clock Generator

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

- Processor clock control register (PCC)
- Ring-OSC mode register (RCM)
- Main clock mode register (MCM)
- Main OSC control register (MOC)
- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)

#### (1) Processor clock control register (PCC)

This register sets the division ratio of the CPU clock.

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

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

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

Address: FFFBH After reset: 00H R/W

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

PCC2	PCC1	PCC0	CPU clock selection ( $f_{\text{CPU}}$ )
0	0	0	$f_x$
0	0	1	$f_x/2$
0	1	0	$f_x/2^2$
0	1	1	$f_x/2^3$
1	0	0	$f_x/2^4$
Other			Setting prohibited

The fastest instruction can be executed in 2 clocks of the CPU clock in the 78K0/KB1 Series. Therefore, the relationship between the CPU clock ( $f_{CPU}$ ) and minimum instruction execution time is as shown in the Table 5-2.

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

CPU Clock ( $f_{CPU}$ ) <sup>Note</sup>	Minimum Instruction Execution Time: $2/f_{CPU}$	
	X1 Input Clock (at 10 MHz Operation)	Ring-OSC Clock (at 240 kHz (TYP.) Operation)
$f_x$	0.2 $\mu s$	8.3 $\mu s$ (TYP.)
$f_x/2$	0.4 $\mu s$	16.6 $\mu s$ (TYP.)
$f_x/2^2$	0.8 $\mu s$	33.2 $\mu s$ (TYP.)
$f_x/2^3$	1.6 $\mu s$	66.4 $\mu s$ (TYP.)
$f_x/2^4$	3.2 $\mu s$	132.8 $\mu s$ (TYP.)

**Note** The main clock mode register (MCM) is used to set the CPU clock (X1 input clock/Ring-OSC clock) (see **Figure 5-4**).

**(2) Ring-OSC mode register (RCM)**

This register sets the operation mode of Ring-OSC.

This register is valid when “Can be stopped by software” is set for Ring-OSC by a mask option, and the X1 input clock is input to the CPU clock. If “Cannot be stopped” is selected for Ring-OSC by a mask option, settings for this register are invalid.

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

$\overline{RESET}$  input clears this register to 00H.

**Figure 5-3. Format of Ring-OSC Mode Register (RCM)**

Address: FFA0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<input type="checkbox"/>
RCM	0	0	0	0	0	0	0	RSTOP

RSTOP	Ring-OSC oscillating/stopped
0	Ring-OSC oscillating
1	Ring-OSC stopped

**Caution** Make sure that the bit 1 (MCS) of the main clock mode register (MCM) is 1 before setting RSTOP.

**(3) Main clock mode register (MCM)**

This register sets the CPU clock.

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

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

**Figure 5-4. Format of Main Clock Mode Register (MCM)**

Address: FFA1H After reset: 00H R/W

Symbol	7	6	5	4	3	2	<span style="border: 1px solid black; padding: 0 2px;">1</span>	<span style="border: 1px solid black; padding: 0 2px;">0</span>
MCM	0	0	0	0	0	0	MCS	MCM0

MCS	CPU clock status
0	Operates with Ring-OSC clock
1	Operates with X1 input clock

MCM0	Selection of clock supplied to CPU
0	Ring-OSC clock
1	X1 input clock

**Caution** When Ring-OSC clock is selected as the clock to be supplied to the CPU, the divided clock of the Ring-OSC clock oscillator output ( $f_x$ ) is supplied to the peripheral hardware ( $f_x = 240 \text{ kHz (TYP.)}$ ).

Operation of the peripheral hardware with Ring-OSC clock cannot be guaranteed. Therefore, when Ring-OSC clock is selected as the clock supplied to the CPU, do not use peripheral hardware. In addition, stop the peripheral hardware before switching the clock supplied to the CPU from the X1 input clock to the Ring-OSC clock. Note, however, that the following peripheral hardware can be used when the CPU operates on the Ring-OSC clock.

- Watchdog timer
- Clock monitor
- 8-bit timer H1 when  $f_R/2^7$  is selected as count clock
- Peripheral hardware selecting external clock as the clock source

**(4) Main OSC control register (MOC)**

This register selects the operation mode of the X1 input clock.

This register is used to stop the X1 input clock when the CPU is operating with the Ring-OSC clock. Therefore, this register is valid only when the CPU is operating with the Ring-OSC clock.

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

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

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

Address: FFA2H After reset: 00H R/W

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

MSTOP	Control of X1 input clock oscillation
0	X1 input clock oscillating
1	X1 input clock stopped

**Caution** Make sure that the bit 1 (MCS) of the main clock mode register (MCM) is 0 before setting MSTOP.

**(5) Oscillation stabilization time counter status register (OSTC)**

This is the status register of the X1 input clock oscillation stabilization time counter. If the Ring-OSC clock is used as the CPU clock, the X1 input clock oscillation stabilization time can be checked.

★ OSTC can be read by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$  input, STOP instruction, and MSTOP = 1 clear OSTC to 00H.

★ **Figure 5-6. Format of Oscillation Stabilization Time Counter Status Register (OSTC)**

Address: FFA3H After reset: 00H R

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

MOST11	MOST13	MOST14	MOST15	MOST16	Oscillation stabilization time status
1	0	0	0	0	$2^{11}/f_{XP}$ min. (204.8 $\mu$ s min.)
1	1	0	0	0	$2^{13}/f_{XP}$ min. (819.2 $\mu$ s min.)
1	1	1	0	0	$2^{14}/f_{XP}$ min. (1.64 ms min.)
1	1	1	1	0	$2^{15}/f_{XP}$ min. (3.27 ms min.)
1	1	1	1	1	$2^{16}/f_{XP}$ min. (6.55 ms min.)

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

**Remarks** 1. Values in parentheses are for operation with  $f_{XP} = 10$  MHz.  
 2.  $f_{XP}$ : X1 input clock oscillation frequency

**(6) Oscillation stabilization time select register (OSTS)**

This register is used to select the oscillation stabilization time of the X1 input clock when STOP mode is released. OSTS can be set by an 8-bit memory manipulation instruction. RESET input sets OSTS to 05H.

★

**Figure 5-7. Format of Oscillation Stabilization Time Select Register (OSTS)**

Address: FFA4H After reset: 05H R/W

Symbol	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

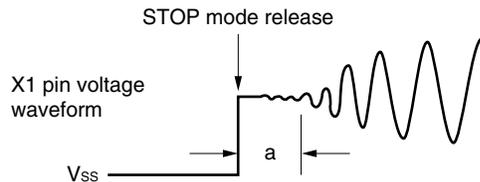
OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection
0	0	1	$2^{11}/f_{XP}$ (204.8 $\mu$ s)
0	1	0	$2^{13}/f_{XP}$ (819.2 $\mu$ s)
0	1	1	$2^{14}/f_{XP}$ (1.64 ms)
1	0	0	$2^{15}/f_{XP}$ (3.27 ms)
1	0	1	$2^{16}/f_{XP}$ (6.55 ms)
Other			Setting prohibited

**Cautions 1.** If the STOP mode is entered and then released while the Ring-OSC is being used as the CPU clock, set the oscillation stabilization time as follows.

- Desired OSTC oscillation stabilization time  $\leq$  Oscillation stabilization time set by OSTS

The setting value of OSTS can be initialized (to 05H) only by a reset. Therefore, note that the X1 oscillation stabilization time counter counts up to the set value of OSTS after STOP mode is released when the Ring-OSC is used as the CPU clock.

**2.** The wait time when STOP mode is released does not include the time after STOP mode release until clock oscillation starts (“a” below) regardless of whether STOP mode is released by RESET input or interrupt generation.



**Remarks 1.** Values in parentheses are for operation with  $f_{XP} = 10$  MHz.

**2.**  $f_{XP}$ : X1 input clock oscillation frequency

## 5.4 System Clock Oscillator

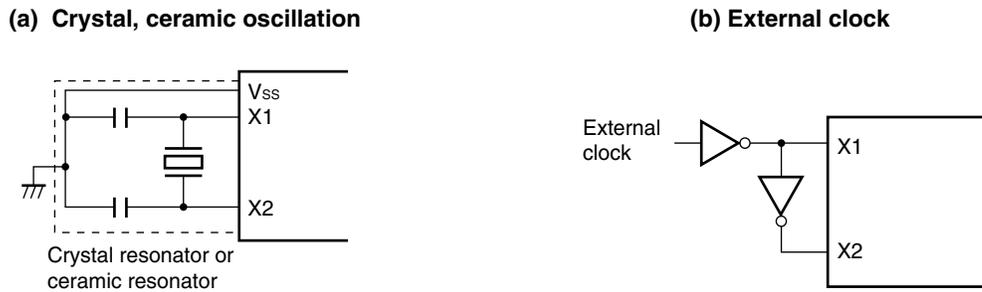
### 5.4.1 X1 oscillator

The X1 oscillator oscillates via a crystal resonator or ceramic resonator (standard: 10 MHz) connected to the X1 and X2 pins.

An external clock can be input to the X1 oscillator. In this case, input the clock signal to the X1 pin and input the inverse signal to the X2 pin.

Figure 5-8 shows the external circuit of the X1 oscillator.

**Figure 5-8. External Circuit of X1 Oscillator**



**Caution** When using the X1 oscillator, wire as follows in the area enclosed by the broken lines in the Figure 5-9 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.

Figure 5-9 shows examples of incorrect resonator connection.

Figure 5-9. Examples of Incorrect Resonator Connection (1/2)

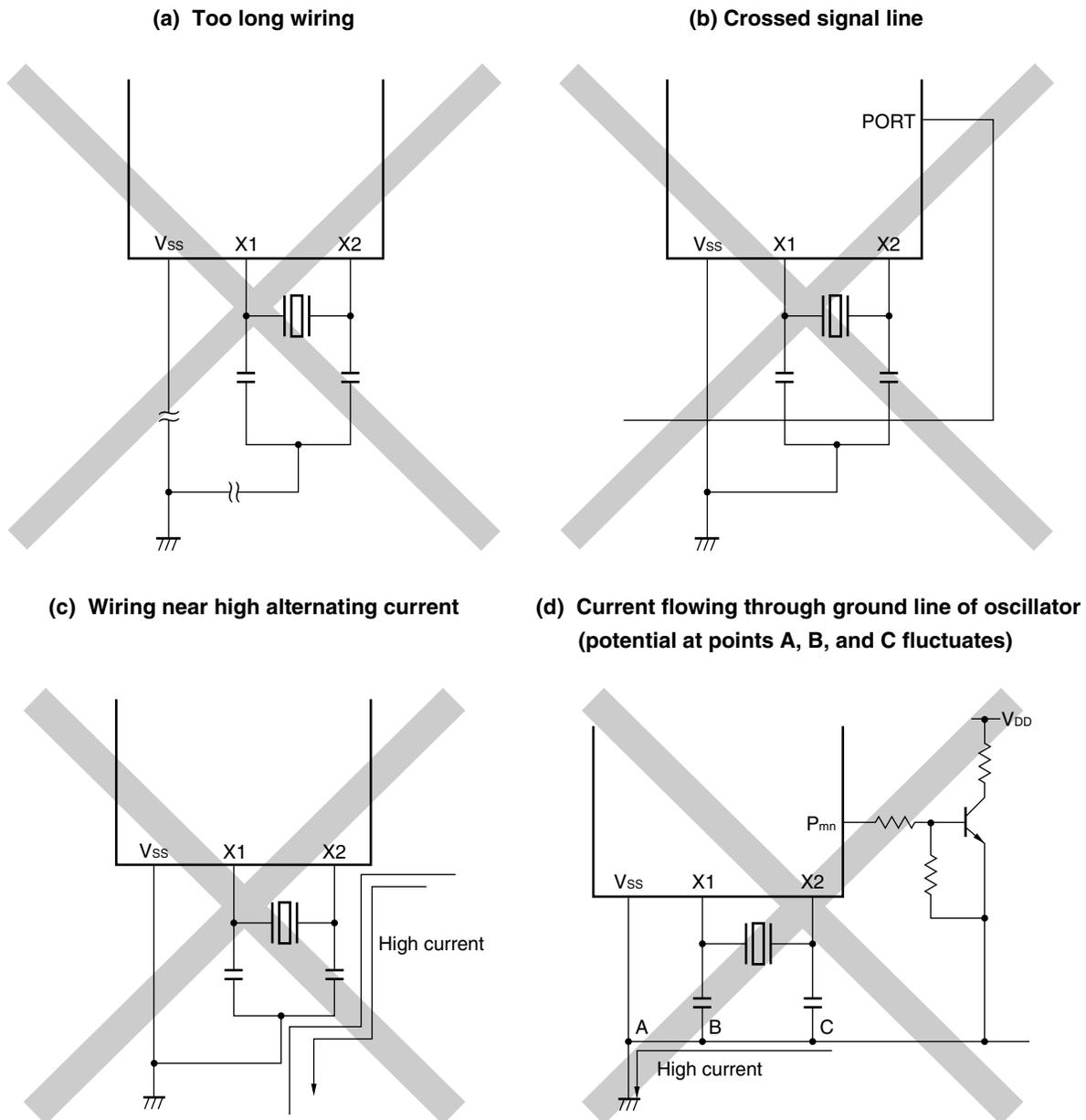
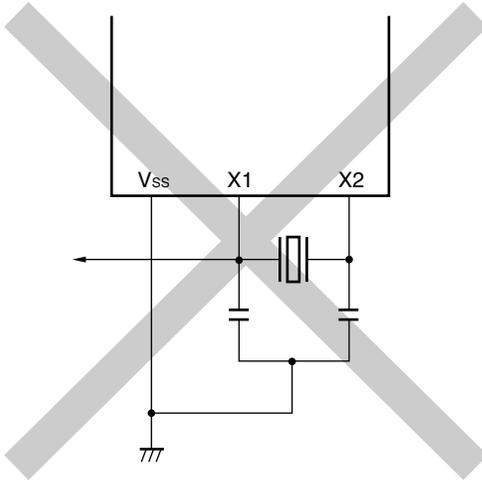


Figure 5-9. Examples of Incorrect Resonator Connection (2/2)

(e) Signals are fetched



#### 5.4.2 Ring-OSC oscillator

Ring-OSC is incorporated in the  $\mu$ PD780101, 780102, 780103, and 78F0103.

“Can be stopped by software” or “Cannot be stopped” can be selected by a mask option. The Ring-OSC clock always oscillates after  $\overline{\text{RESET}}$  release (240 kHz (TYP.)).

#### 5.4.3 Prescaler

The prescaler generates various clocks by dividing the X1 oscillator output ( $f_x$ ) when the X1 input clock is selected as the clock to be supplied to the CPU.

**Caution** When the Ring-OSC clock is selected as the clock supplied to the CPU, the prescaler generates various clocks by dividing the Ring-OSC oscillator ( $f_x$ ) ( $f_x = 240 \text{ kHz (TYP.)}$ ).

### 5.5 Clock Generator Operation

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

- X1 input clock  $f_{XP}$
- Ring-OSC clock  $f_R$
- CPU clock  $f_{CPU}$
- Clock to peripheral hardware

The Ring-OSC clock via the on-chip Ring-OSC oscillator is used as the CPU clock after reset release in the 78K0/KB1 Series, thus enabling the following.

**(1) Enhancement of security function**

When the X1 input clock is set as the CPU clock by the default setting, the device cannot operate if the X1 input clock is damaged or badly connected and therefore does not operate after reset is released. However, the start clock of the CPU is the on-chip Ring-OSC clock, so the device can be started by the Ring-OSC clock after reset release by the clock monitor (detection of X1 input clock stop). Consequently, the system can be safely shut down by performing a minimum operation, such as acknowledging a reset source by software or performing safety processing when there is a malfunction.

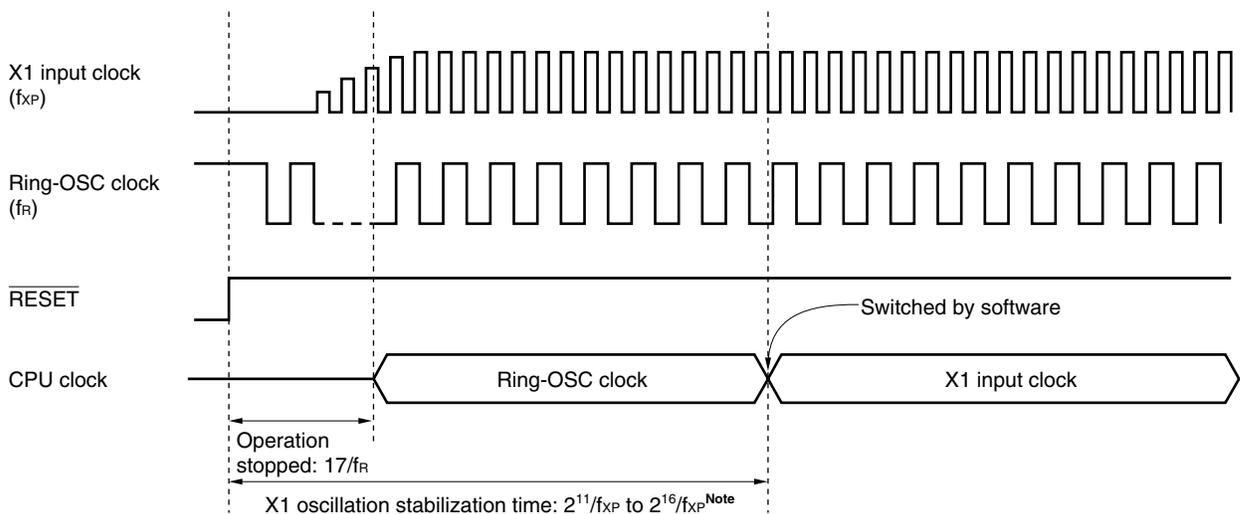
**(2) Improvement of performance**

Because the CPU can be started without waiting for the X1 input clock oscillation stabilization time, the total performance can be improved.

★ **Caution** When writing to the flash memory while it is mounted on the board, the X1 input clock is selected as the default CPU clock. In this case, the oscillation stabilization time is  $2^{16}/f_{XP}$  (s).

A timing diagram of the CPU default start using Ring-OSC is shown in Figure 5-10.

★ **Figure 5-10. Timing Diagram of CPU Default Start Using Ring-OSC**



**Note** Check using the oscillation stabilization time counter status register (OSTC).

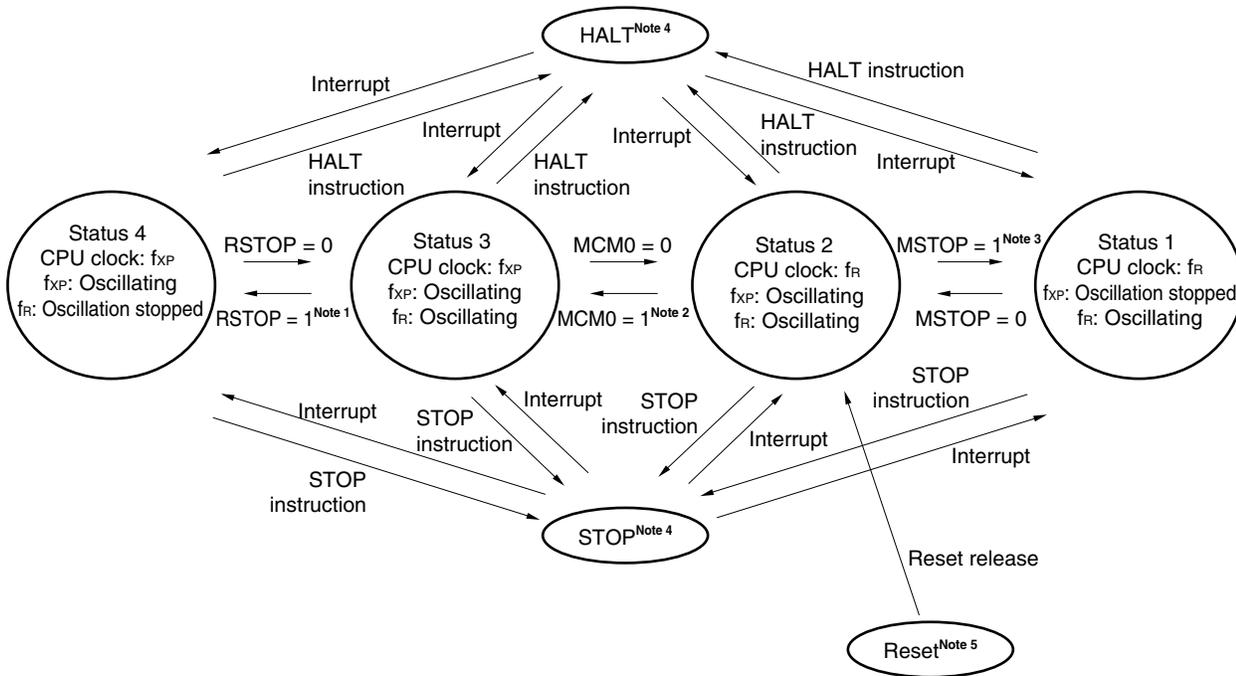
- (a) When the  $\overline{\text{RESET}}$  signal is generated, bit 0 of the main clock mode register (MCM) is set to 0 and the Ring-OSC clock is set as the CPU clock. However, the CPU clock is stopped for 17 clocks of the Ring-OSC clock after  $\overline{\text{RESET}}$  signal generation. While inputting a low level to the  $\overline{\text{RESET}}$  pin, oscillation of the X1 input clock and Ring-OSC clock is stopped.
- (b) After  $\overline{\text{RESET}}$  signal generation, the CPU clock can be switched from the Ring-OSC clock to the X1 input clock using bit 0 (MCM0) of the main clock mode register (MCM) after the X1 input clock oscillation stabilization time has elapsed. At this time, check the oscillation stabilization time using the oscillation stabilization time counter status register (OSTC) before switching the CPU clock. The CPU clock status can be checked using bit 1 (MCS) of MCM.
- (c) Ring-OSC can be set to stopped/oscillating using the Ring-OSC mode register (RCM) when “Can be stopped by software” is selected for the Ring-OSC by a mask option, if the X1 input is used as the CPU clock. Make sure that MCS is 1 at this time.

- (d) When Ring-OSC is used as the CPU clock, the X1 input clock can be set to stopped/oscillating using the main OSC control register (MOC). Make sure that MCS is 0 at this time.
- ★ (e) Select the X1 input clock oscillation stabilization time ( $2^{11}/f_{XP}$ ,  $2^{13}/f_{XP}$ ,  $2^{14}/f_{XP}$ ,  $2^{15}/f_{XP}$ ,  $2^{16}/f_{XP}$ ) using the oscillation stabilization time select register (OSTS) when releasing STOP mode while X1 input clock is being used as the CPU clock. In addition, when releasing STOP mode while  $\overline{\text{RESET}}$  is released and Ring-OSC clock is being used as the CPU clock, check the X1 input clock oscillation stabilization time using the oscillation stabilization time counter status register (OSTC).

A status transition diagram of this product is shown in Figure 5-11, and the relationship between the operation clocks in each operation status and between the oscillation control flag and oscillation status of each clock are shown in Tables 5-3 and 5-4, respectively.

Figure 5-11. Status Transition Diagram (1/2)

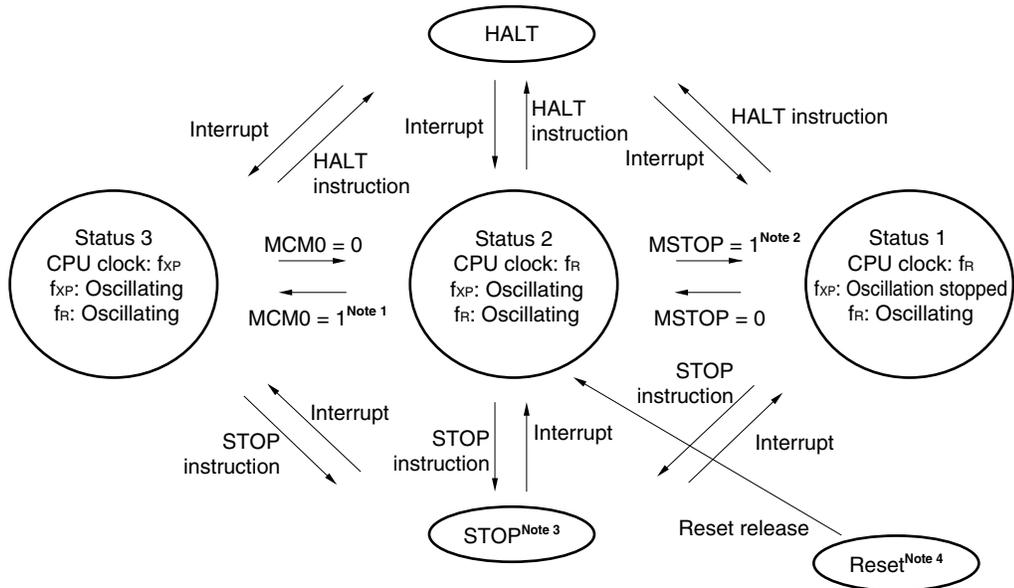
(1) When “Ring-OSC can be stopped by software” is selected by mask option



- Notes**
1. When shifting from status 3 to status 4, make sure that bit 1 (MCS) of the main clock mode register (MCM) is 1.
  2. Before shifting from status 2 to status 3 after reset and STOP are released, check the X1 input clock oscillation stabilization time status using the oscillation stabilization time counter status register (OSTC).
  3. When shifting from status 2 to status 1, make sure that MCS is 0.
  - ★ 4. When “Ring-OSC can be stopped by software” is selected by a mask option, the watchdog timer stops operating in the HALT and STOP modes, regardless of the source clock of the watchdog timer. However, oscillation of Ring-OSC does not stop even in the HALT and STOP modes if  $RSTOP = 0$ .
  5. All reset sources ( $\overline{\text{RESET}}$  input, POC, LVI, clock monitor, and WDT)

Figure 5-11. Status Transition Diagram (2/2)

(2) When “Ring-OSC cannot be stopped” is selected by mask option



- Notes**
1. Before shifting from status 2 to status 3 after reset and STOP are released, check the X1 input clock oscillation stabilization time status using the oscillation stabilization time counter status register (OSTC).
  2. When shifting from status 2 to status 1, make sure that MCS is 0.
  3. The watchdog timer operates using Ring-OSC even in STOP mode if “Ring-OSC cannot be stopped” is selected by a mask option. Ring-OSC division can be selected as the count source of 8-bit timer H1 (TMH1), so clear the watchdog timer using the TMH1 interrupt request before watchdog timer overflow. If this processing is not performed, an internal reset signal is generated at watchdog timer overflow after STOP instruction execution.
  4. All reset sources ( $\overline{RESET}$  input, POC, LVI, clock monitor, and WDT)

★ **Table 5-3. Relationship Between Operation Clocks in Each Operation Status**

Operation Mode \ Status	X1 Oscillator	Ring-OSC Oscillator			CPU Clock After Release	Prescaler Clock Supplied to Peripherals	
		Note 1	Note 2			MCM0 = 0	MCM0 = 1
			RSTOP = 0	RSTOP = 1			
Reset	Stopped	Stopped			Ring-OSC	Stopped	
STOP		Oscillating	Oscillating	Stopped	Note 3	Stopped	
HALT	Oscillating				Note 4	Ring-OSC	X1

**Caution** The RSTOP setting is valid only when “Can be stopped by software” is set for Ring-OSC by a mask option.

- Notes**
1. When “Cannot be stopped” is selected for Ring-OSC by a mask option.
  2. When “Can be stopped by software” is selected for Ring-OSC by a mask option.
  3. Operates using the CPU clock at STOP instruction execution.
  4. Operates using the CPU clock at HALT instruction execution.

**Remark** RSTOP: Bit 0 of the Ring-OSC mode register (RCM)  
MCM0: Bit 0 of the main clock mode register (MCM)

★ **Table 5-4. Oscillation Control Flags and Clock Oscillation Status**

		X1 Input Clock	Ring-OSC Clock
MSTOP = 1	RSTOP = 0	Stopped	Oscillating
	RSTOP = 1	Setting prohibited	
MSTOP = 0	RSTOP = 0	Oscillating	Oscillating
	RSTOP = 1		Stopped

**Caution** The RSTOP setting is valid only when “Can be stopped by software” is set for Ring-OSC by a mask option.

**Remark** MSTOP: Bit 7 of the main OSC control register (MOC)  
RSTOP: Bit 0 of the Ring-OSC mode register (RCM)

## 5.6 Time Required to Switch Between Ring-OSC Clock and X1 Input Clock

Bit 0 (MCM0) of the main clock mode register (MCM) is used to switch between the Ring-OSC clock and X1 input clock.

In the actual switching operation, switching does not occur immediately after MCM0 rewrite; several instructions are executed using the pre-switch clock after switching MCM0 (see **Table 5-5**).

Bit 1 (MCS) of MCM is used to judge that operation is performed using either the Ring-OSC clock or X1 input clock. To stop the clock, wait for the number of clocks shown in Table 5-5 before stopping.

**Table 5-5. Maximum Time Required to Switch Between Ring-OSC Clock and X1 Input Clock**

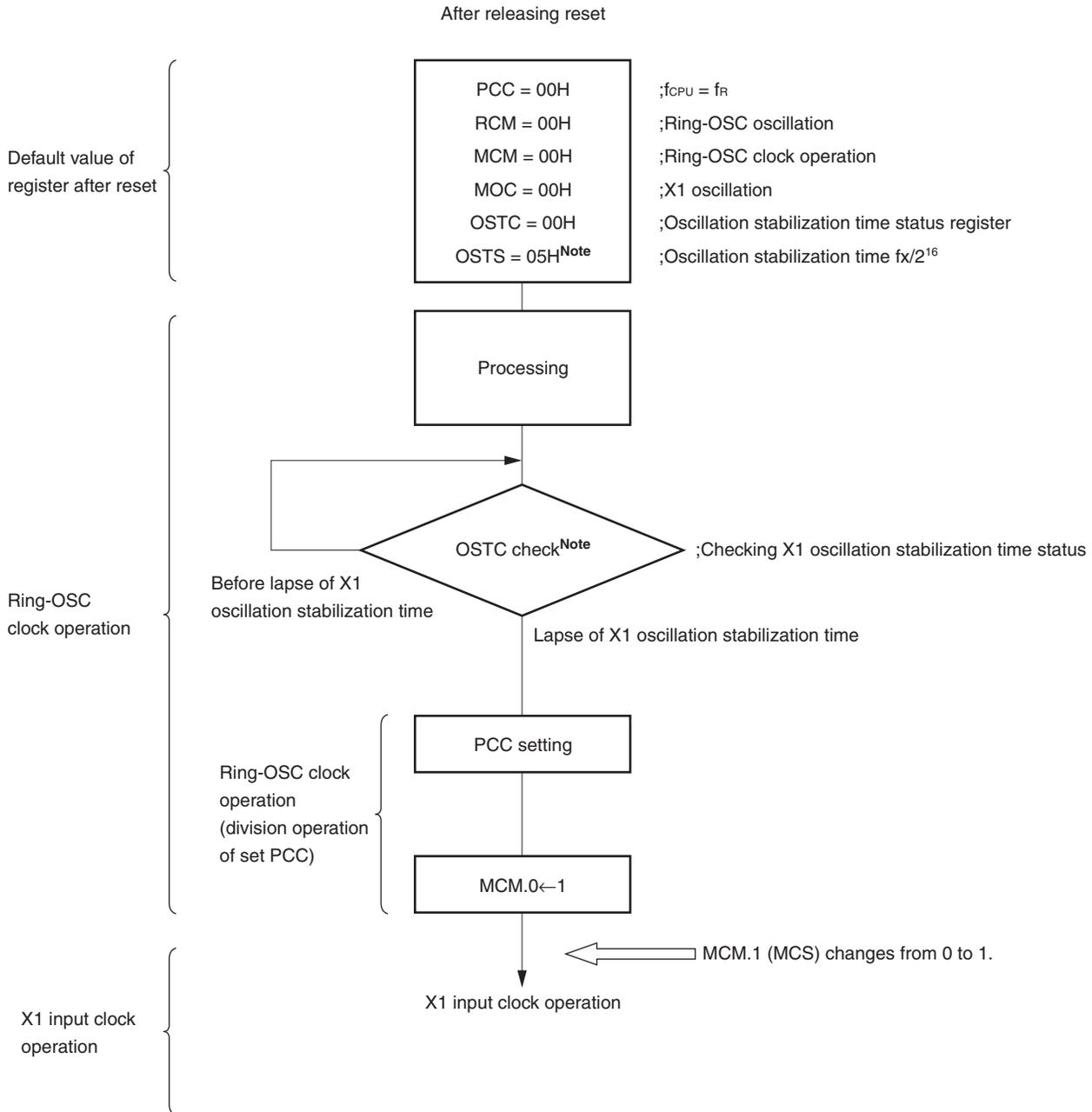
PCC			Maximum Time Required for Switching	
PCC2	PCC1	PCC0	X1→Ring-OSC	Ring-OSC→X1
0	0	0	$f_{XP}/120,000 + 1$ clock	2 clocks
0	0	1	$f_{XP}/240,000 + 1$ clock	
0	1	0	$f_{XP}/480,000 + 1$ clock	
0	1	1	$f_{XP}/960,000 + 1$ clock	
1	0	0	$f_{XP}/1,920,000 + 1$ clock	

- Remarks 1.** PCC: Processor clock control register
2.  $f_{XP}$ : X1 input clock oscillation frequency
  3.  $f_R$ : Ring-OSC clock oscillation frequency
  4. The maximum time is the number of clocks of the CPU clock before switching.

★ 5.7 Clock Selection Flowchart and Register Settings

5.7.1 Changing to X1 input clock from Ring-OSC clock

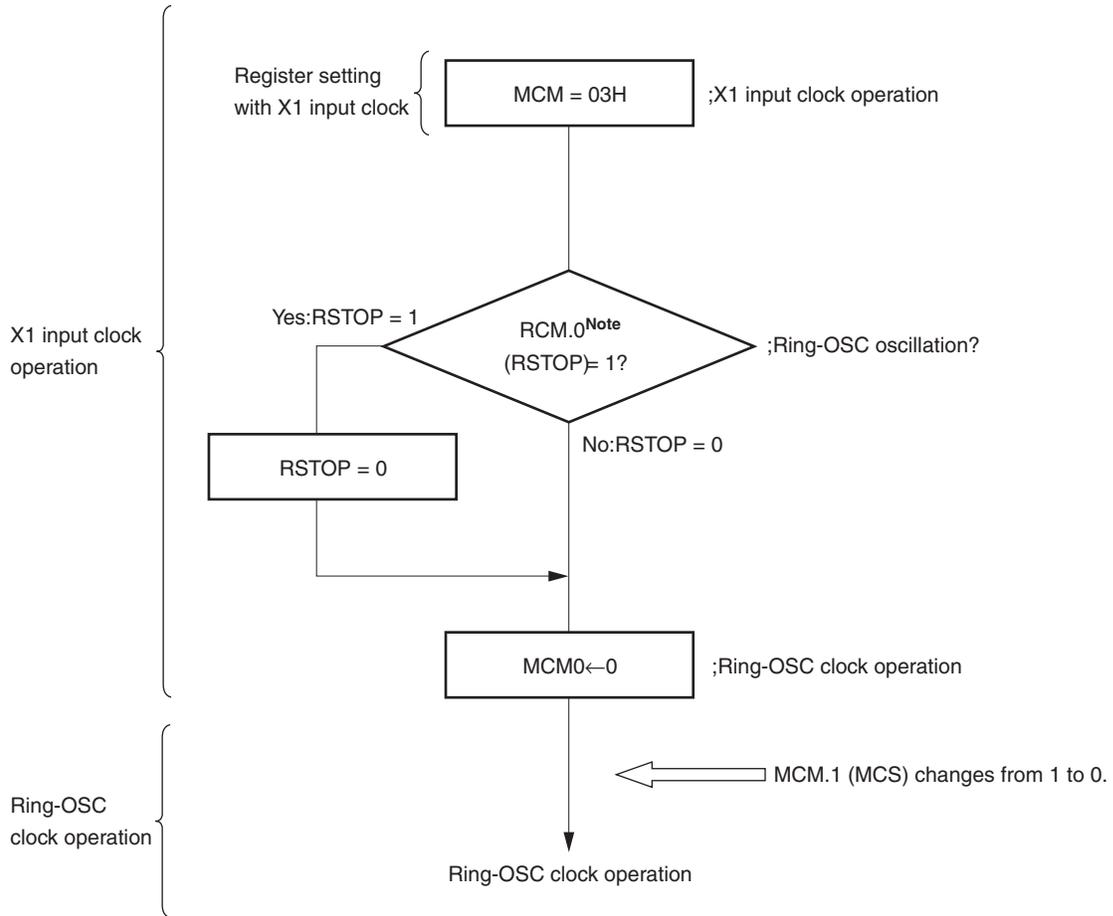
Figure 5-12. Changing to X1 Input Clock from Ring-OSC Clock (Flowchart)



**Note** Check the oscillation stabilization time of the X1 oscillation circuit using the OSTC register after the reset signal has been released and select the X1 input clock operation after the lapse of specified oscillation stabilization time. Setting the OSTS register is valid only when the STOP mode has been released with the system operating on the X1 input clock.

5.7.2 Changing from X1 input clock to Ring-OSC clock

Figure 5-13. Changing from X1 Input Clock to Ring-OSC Clock (Flowchart)



**Note** This is necessary only when “clock can be stopped by software” is selected for Ring-OSC by mask option.

5.7.3 Register settings

★

**Table 5-6. Clock and Register Settings**

f <sub>CPU</sub>	Mode	Setting Flag			Status Flag
		MCM Register	MOC Register	RCM Register	MCM Register
		MCM0	MSTOP	RSTOP <sup>Note 1</sup>	MCS
X1 input clock <sup>Note 2</sup>	Ring-OSC oscillating	1	0	0	1
	Ring-OSC stopped	1	0	1	1
Ring-OSC clock	X1 oscillating	0	0	0	0
	X1 stopped	0	1	0	0

**Notes** 1. This is valid only when “clock can be stopped by software” is selected for Ring-OSC by mask option.

★

2. Do not set MSTOP to 1 during X1 input clock operation (oscillation of X1 is not stopped even when MSTOP = 1).

### 6.1 Functions of 16-Bit Timer/Event Counter 00

16-bit timer/event counter 00 has the following functions.

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

**(1) Interval timer**

TM00 generates an interrupt request at the preset time interval.

**(2) PPG output**

TM00 can output a rectangular wave whose frequency and output pulse width can be set freely.

**(3) Pulse width measurement**

TM00 can measure the pulse width of an externally input signal.

**(4) External event counter**

TM00 can measure the number of pulses of an externally input signal.

**(5) Square-wave output**

TM00 can output a square wave with any selected frequency.

## 6.2 Configuration of 16-Bit Timer/Event Counter 00

16-bit timer/event counter 00 consists of the following hardware.

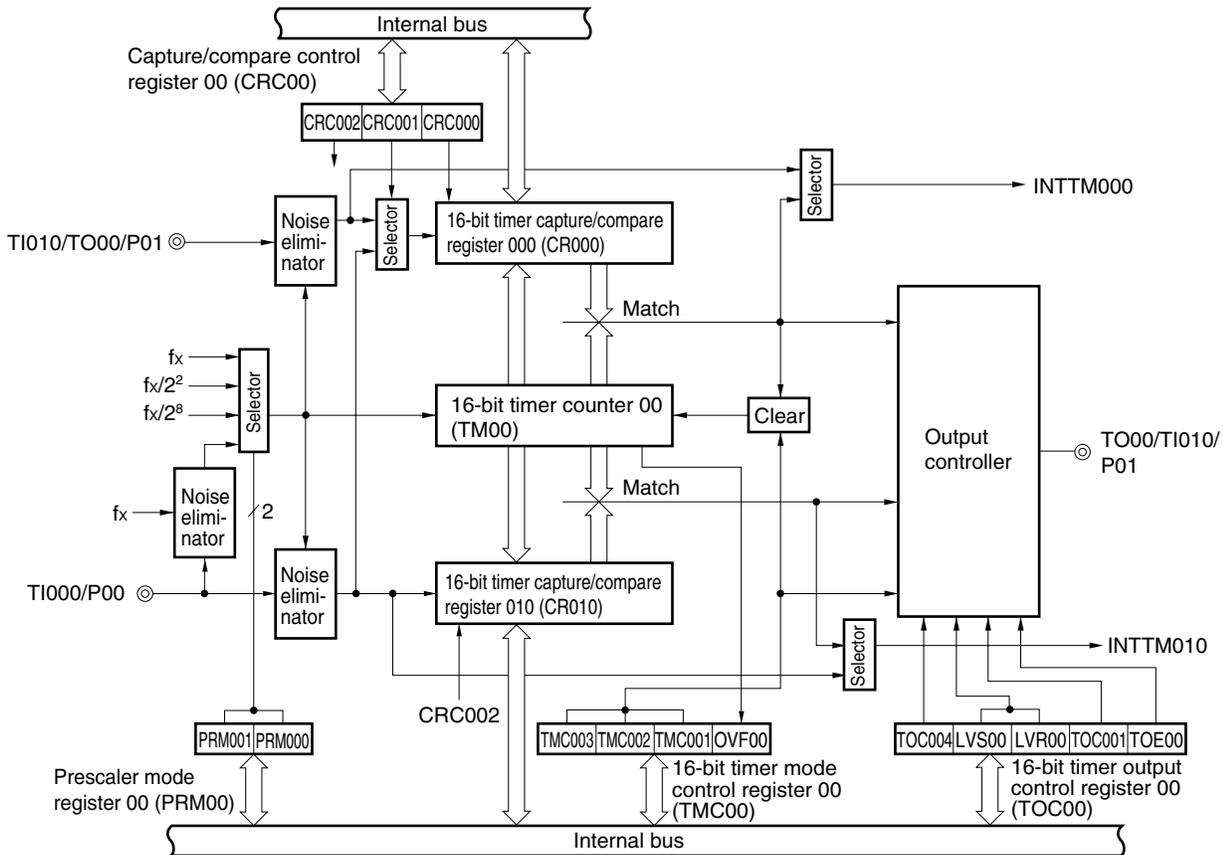
**Table 6-1. Configuration of 16-Bit Timer/Event Counter 00**

Item	Configuration
Timer counter	16 bits × 1 (TM00)
Register	16-bit timer capture/compare register: 16 bits × 2 (CR000, CR010)
Timer output	1 (TO00)
Control registers	16-bit timer mode control register 00 (TMC00) Capture/compare control register 00 (CRC00) 16-bit timer output control register 00 (TOC00) Prescaler mode register 00 (PRM00) Port mode register 0 (PM0) <sup>Note</sup>

**Note** See Figure 4-2 Block Diagram of P00 and Figure 4-3 Block Diagram of P01.

Figure 6-1 shows the block diagram.

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



**(1) 16-bit timer counter 00 (TM00)**

TM00 is a 16-bit read-only register that counts count pulses.

The counter is incremented in synchronization with the rising edge of the input clock. The count value is reset to 0000H in the following cases:

- <1> At  $\overline{\text{RESET}}$  input
- <2> If TMC003 and TMC002 are cleared
- <3> If the valid edge of TI000 is input in the clear & start mode entered by inputting the valid edge of TI000
- <4> If TM00 and CR000 match in the clear & start mode entered on a match of TM00 and CR000

**(2) 16-bit timer capture/compare register 000 (CR000)**

CR000 is a 16-bit register that has the functions of both a capture register and a compare register. Whether it is used as a capture register or as a compare register is set by bit 0 (CRC000) of capture/compare control register 00 (CRC000).

- **When CR000 is used as a compare register**

The value set in CR000 is constantly compared with the 16-bit timer counter 00 (TM00) count value, and an interrupt request (INTTM000) is generated if they match. It can also be used as the register that holds the interval time when TM00 is set to interval timer operation.

- **When CR000 is used as a capture register**

It is possible to select the valid edge of the TI000/P00 pin or the TI010/TO00/P01 pin as the capture trigger. The TI000 or TI010 valid edge is set using prescaler mode register 00 (PRM00).

If the capture trigger is specified to be the valid edge of the TI000/P00 pin, the situation is as shown in Table 6-2. On the other hand, when the capture trigger is specified to be the valid edge of the TI010/TO00/P01 pin, the situation is as shown in Table 6-3.

**Table 6-2. TI000/P00 Pin Valid Edge and CR000, CR010 Capture Trigger**

ES001	ES000	TI000/P00 Pin Valid Edge	CR000 Capture Trigger	CR010 Capture Trigger
0	0	Falling edge	Rising edge	Falling edge
0	1	Rising edge	Falling edge	Rising edge
1	0	Setting prohibited	Setting prohibited	Setting prohibited
1	1	Both rising and falling edges	No capture operation	Both rising and falling edges

**Table 6-3. TI010/TO00/P01 Pin Valid Edge and CR000 Capture Trigger**

ES101	ES100	TI010/TO00/P01 Pin Valid Edge	CR000 Capture Trigger
0	0	Falling edge	Falling edge
0	1	Rising edge	Rising edge
1	0	Setting prohibited	Setting prohibited
1	1	Both rising and falling edges	Both rising and falling edges

CR000 can be set by a 16-bit memory manipulation instruction.

- ★  $\overline{\text{RESET}}$  input clears this register to 0000H.

- Cautions**
1. Set a value other than 0000H in CR000 in the clear & start mode entered on a match of TM00 and CR000. However, in the free-running mode and in the clear mode using the valid edge of TI000, if CR000 is set to 0000H, an interrupt request (INTTM000) is generated following overflow (FFFFH).
  2. If the changed value of CR000 is smaller than the value of 16-bit timer counter 00 (TM00), TM00 continues counting and starts counting again from 0 after overflow. Therefore, if the value of CR000 after the change is smaller than before the change, the timer should be restarted after CR000 is changed.
  3. When P01 is used as the valid edge of TI010, it cannot be used as the timer output (TO00). Moreover, when P01 is used as TO00, it cannot be used as the valid edge of TI010.
  4. When CR000 is used as a capture register, read data is undefined if the register read time and capture trigger input conflict (the capture data itself is the correct value).  
If count stop input and capture trigger input conflict, the captured data is undefined.
  5. Do not rewrite CR000 during TM00 operation.

### (3) 16-bit timer capture/compare register 010 (CR010)

CR010 is a 16-bit register that has the functions of both a capture register and a compare register. Whether it is used as a capture register or a compare register is set by bit 2 (CRC002) of capture/compare control register 00 (CRC00).

- **When CR010 is used as a compare register**

The value set in the CR010 is constantly compared with the 16-bit timer counter 00 (TM00) count value, and an interrupt request (INTTM010) is generated if they match.

- **When CR010 is used as a capture register**

It is possible to select the valid edge of the TI000/P00 pin as the capture trigger. The TI000/P00 valid edge is set by prescaler mode register 00 (PRM00).

CR010 can be set by a 16-bit memory manipulation instruction.

- ★  $\overline{\text{RESET}}$  input clears this register to 0000H.

- Cautions**
1. Set CR010 to other than 0000H. This means a 1-pulse count operation cannot be performed when CR010 is used as the event counter.  
However, in the free-running mode and in the clear mode using the valid edge of TI000, if CR010 is set to 0000H, an interrupt request (INTTM010) is generated following overflow (FFFFH).
  2. When CR010 is used as a capture register, read data is undefined if the register read time and capture trigger input conflict (the capture data itself is the correct value).  
If count stop input and capture trigger input conflict, the captured data is undefined.
  3. CR010 can be rewritten during TM00 operation. For details, refer to Remark 2 in Figure 6-10.

- ★

### 6.3 Registers Controlling 16-Bit Timer/Event Counter 00

The following five registers are used to control the 16-bit timer/event counter.

- 16-bit timer mode control register 00 (TMC00)
- Capture/compare control register 00 (CRC00)
- 16-bit timer output control register 00 (TOC00)
- Prescaler mode register 00 (PRM00)
- Port mode register 0 (PM0)

#### (1) 16-bit timer mode control register 00 (TMC00)

This register sets the 16-bit timer operating mode, the 16-bit timer counter 00 (TM00) clear mode, and output timing, and detects an overflow.

TMC00 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$  input clears TMC00 to 00H.

**Caution** 16-bit timer counter 00 (TM00) starts operation at the moment TMC002 and TMC003 are set to values other than 0, 0 (operation stop mode), respectively. Set TMC002 and TMC003 to 0, 0 to stop the operation.

Figure 6-2. Format of 16-Bit Timer Mode Control Register 00 (TMC00)

Address FFBAH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TMC00	0	0	0	0	TMC003	TMC002	TMC001	OVF00

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

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

- Cautions**
1. Timer operation must be stopped before writing to bits other than the OVF00 flag.
  2. Set the valid edge of the TI000/P00 pin using prescaler mode register 00 (PRM00).
  3. If clear & start mode entered on match between TM00 and CR000 is selected, when the set value of CR000 is FFFFH and the TM00 value changes from FFFFH to 0000H, the OVF00 flag is set to 1.

**Remark**

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

**(2) Capture/compare control register 00 (CRC00)**

This register controls the operation of the 16-bit capture/compare registers (CR000, CR010).

CRC00 can be set by a 1-bit or 8-bit memory manipulation instruction.

RESET input clears CRC0 to 00H.

**Figure 6-3. Format of Capture/Compare Control Register 00 (CRC00)**

Address: FFBC<sub>H</sub> After reset: 00<sub>H</sub> R/W

Symbol	7	6	5	4	3	2	1	0
CRC00	0	0	0	0	0	CRC002	CRC001	CRC000

CRC002	CR010 operating mode selection
0	Operates as compare register
1	Operates as capture register

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

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

- Cautions**
1. Timer operation must be stopped before setting CRC00.
  2. When clear & start mode entered on a match between TM00 and CR000 is selected with 16-bit timer mode control register 00 (TMC00), CR000 should not be specified as a capture register.
  3. If both the rising and falling edges have been selected as the valid edges of TI000, capture is not performed.
  4. To ensure that the capture operation is performed properly, the capture trigger requires a pulse two times longer than the count clock selected by prescaler mode register 00 (PRM00).

**(3) 16-bit timer output control register 00 (TOC00)**

This register controls the operation of the 16-bit timer/event counter 00 output controller. It sets/resets the R-S type flip-flop (LV00), and enables/disables output inversion and 16-bit timer/event counter 00 timer output.

TOC00 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$  input clears TOC00 to 00H.

Figure 6-4 shows the format of TOC00.

**Figure 6-4. Format of 16-Bit Timer Output Control Register 00 (TOC00)**

Address: FFBDH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TOC00	0	0	0	TOC004	LVS00	LVR00	TOC001	TOE00

TOC004	Timer output F/F control by match of CR010 and TM00	
0	Disables inversion operation	
1	Enables inversion operation	

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

TOC001	Timer output F/F control by match of CR000 and TM00	
0	Disables inversion operation	
1	Enables inversion operation	

TOE00	16-bit timer/event counter output control	
0	Disables output (output set to level 0)	
1	Enables output	

- ★ **Cautions**
1. Timer operation must be stopped before setting other than TOC004.
  2. If LVS00 and LVR00 are read after data is set, 0 is read.
  3. Be sure to set bits 5 and 6 to 0.

**(4) Prescaler mode register 00 (PRM00)**

This register is used to set the 16-bit timer counter 00 (TM00) count clock and TI000 and TI010 input valid edges.

PRM00 can be set by a 1-bit or 8-bit memory manipulation instruction.

RESET input clears PRM00 to 00H.

**Figure 6-5. Format of Prescaler Mode Register 00 (PRM00)**

Address: FFBBH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PRM00	ES101	ES100	ES001	ES000	0	0	PRM001	PRM000

ES101	ES100	TI010 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES001	ES000	TI000 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

PRM001	PRM000	Count clock selection
0	0	$f_x$ (10 MHz)
0	1	$f_x/2^2$ (2.5 MHz)
1	0	$f_x/2^3$ (39.06 kHz)
1	1	TI000 valid edge <sup>Note</sup>

**Note** The external clock requires a pulse two times longer than internal clock ( $f_x$ ).

- Cautions**
1. If the valid edge of TI000 is to be set for the count clock, do not set the clear & start mode and the capture trigger using the valid edge of TI000.
  2. Always set data to PRM00 after stopping the timer operation.
  3. If the TI000 or TI010 pin is high level immediately after system reset, the rising edge is immediately detected after the rising edge or both the rising and falling edges are set as the valid edge(s) of the TI000 pin or TI010 pin to enable the operation of 16-bit timer counter 00 (TM00). Care is therefore required when pulling up the TI000 or TI010 pin. However, when re-enabling operation after the operation has been stopped once, the rising edge is not detected.
  4. When P01 is used as the TI010 valid edge, it cannot be used as the timer output (TO00), and when used as TO00, it cannot be used as the TI010 valid edge.

- Remarks**
1.  $f_x$ : X1 input clock oscillation frequency
  2. TI000, TI010: 16-bit timer/event counter input pin
  3. Figures in parentheses are for operation with  $f_x = 10$  MHz.

**(5) Port mode register 0 (PM0)**

This register sets port 0 input/output in 1-bit units.

When using the P01/TO00/TI010 pin for timer output, set PM01 and the output latch of P01 to 0.

PM0 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$  input sets PM0 to FFH.

**Figure 6-6. Format of Port Mode Register 0 (PM0)**

Address: FF20H After reset: FFH R/W

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

PM0n	P0n pin I/O mode selection (n = 0 to 3)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

## 6.4 Operation of 16-Bit Timer/Event Counter 00

### 6.4.1 Interval timer operation

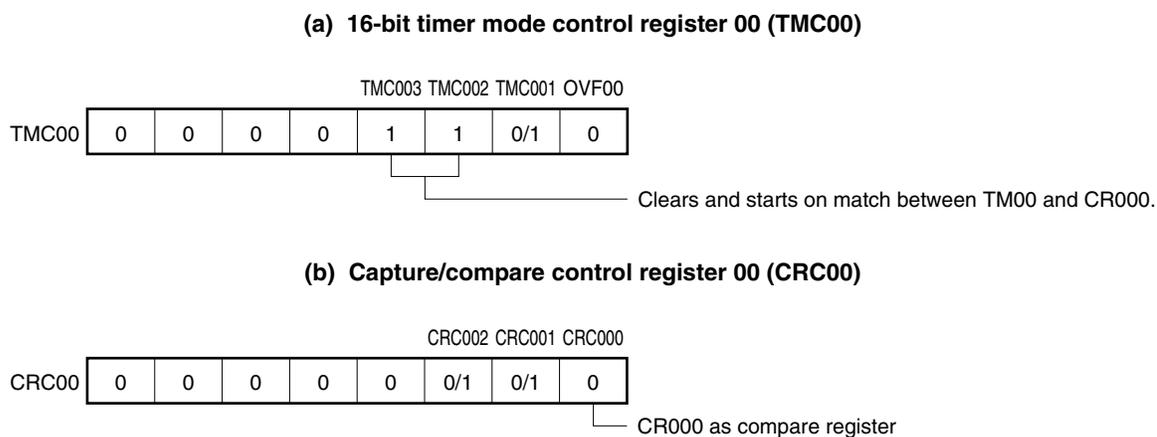
Setting 16-bit timer mode control register 00 (TMC00) and capture/compare control register 00 (CRC00) as shown in Figure 6-7 allows operation as an interval timer. Interrupt requests are generated repeatedly using the count value preset in 16-bit timer capture/compare register 000 (CR000) as the interval.

When the count value of 16-bit timer counter 00 (TM00) matches the value set in CR000, counting continues with the TM00 value cleared to 0 and the interrupt request signal (INTTM000) is generated.

The count clock of the 16-bit timer/event counter can be selected with bits 0 and 1 (PRM000, PRM001) of prescaler mode register 00 (PRM00).

See **6.5 Cautions on 16-Bit Timer/Event Counter 0 (2) 16-bit compare register setting** for details of the operation when the compare register value is changed during timer count operation.

**Figure 6-7. Control Register Settings for Interval Timer Operation**



**Remark** 0/1: Setting 0 or 1 allows another function to be used simultaneously with the interval timer. For details, refer to **Figures 6-2 and 6-3**.

Figure 6-8. Interval Timer Configuration Diagram

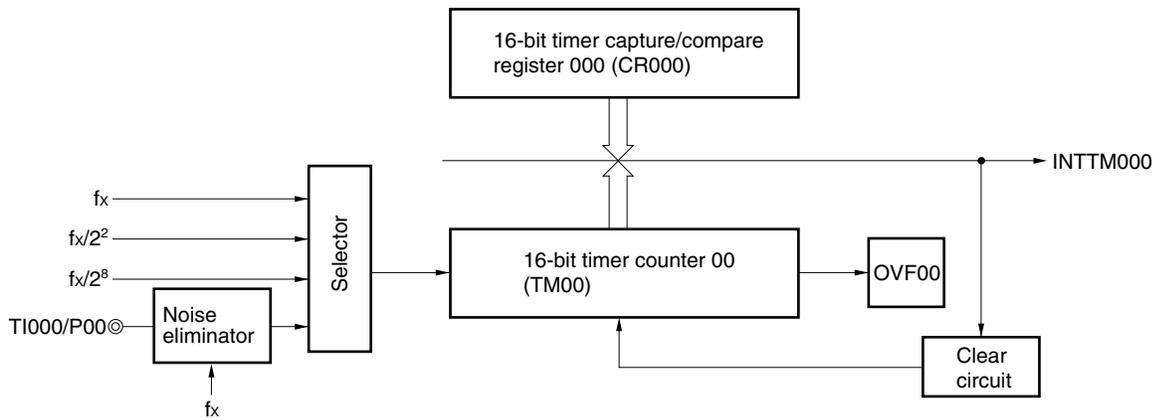
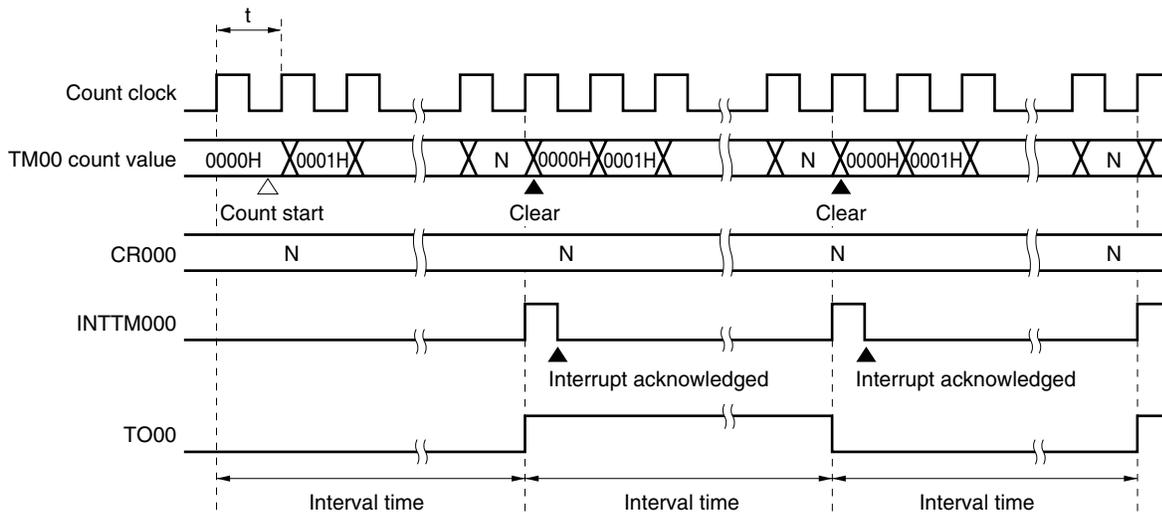


Figure 6-9. Timing of Interval Timer Operation



**Remark** Interval time =  $(N + 1) \times t$   
 $N = 0001H$  to  $FFFFH$

### 6.4.2 PPG output operations

Setting 16-bit timer mode control register 00 (TMC00) and capture/compare control register 00 (CRC00) as shown in Figure 6-10 allows operation as PPG (Programmable Pulse Generator) output.

In the PPG output operation, rectangular waves are output from the TO00/TI010/P01 pin with the pulse width and the cycle that correspond to the count values preset in 16-bit timer capture/compare register 010 (CR010) and in 16-bit timer capture/compare register 000 (CR000), respectively.



### 6.4.3 Pulse width measurement operations

It is possible to measure the pulse width of the signals input to the TI000/P00 pin and TI010/TO00/P01 pin using 16-bit timer counter 00 (TM00).

There are two measurement methods: measuring with TM00 used in free-running mode, and measuring by restarting the timer in synchronization with the edge of the signal input to the TI000/P00 pin.

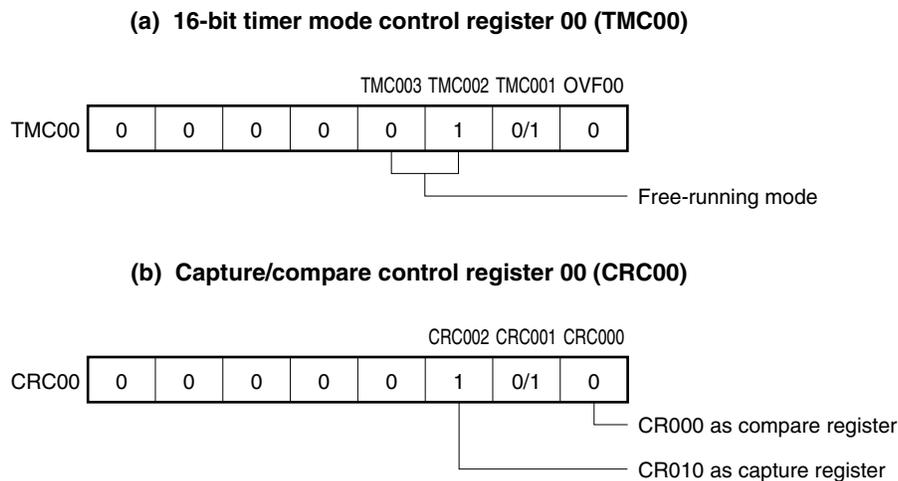
#### (1) Pulse width measurement with free-running counter and one capture register

When 16-bit timer counter 00 (TM00) is operated in free-running mode (see register settings in **Figure 6-11**), and the edge specified by prescaler mode register 00 (PRM00) is input to the TI000/P00 pin, the value of TM00 is taken into 16-bit timer capture/compare register 010 (CR010) and an external interrupt request signal (INTTM001) is set.

Any of three edges can be selected—rising, falling, or both edges—specified using bits 4 and 5 (ES000 and ES001) of PRM00.

For valid edge detection, sampling is performed using the count clock selected by PRM00, and a capture operation is only performed when a valid level is detected twice, thus eliminating noise with a short pulse width.

**Figure 6-11. Control Register Settings for Pulse Width Measurement with Free-Running Counter and One Capture Register**



**Remark** 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. See **Figures 6-2** and **6-3**.

Figure 6-12. Configuration Diagram for Pulse Width Measurement by Free-Running Counter

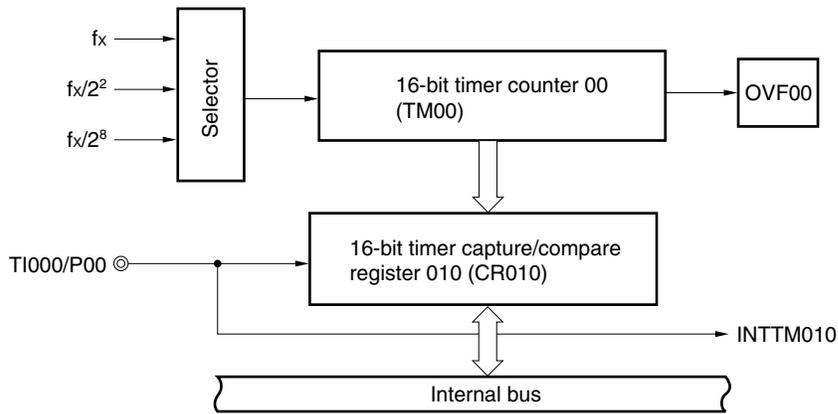
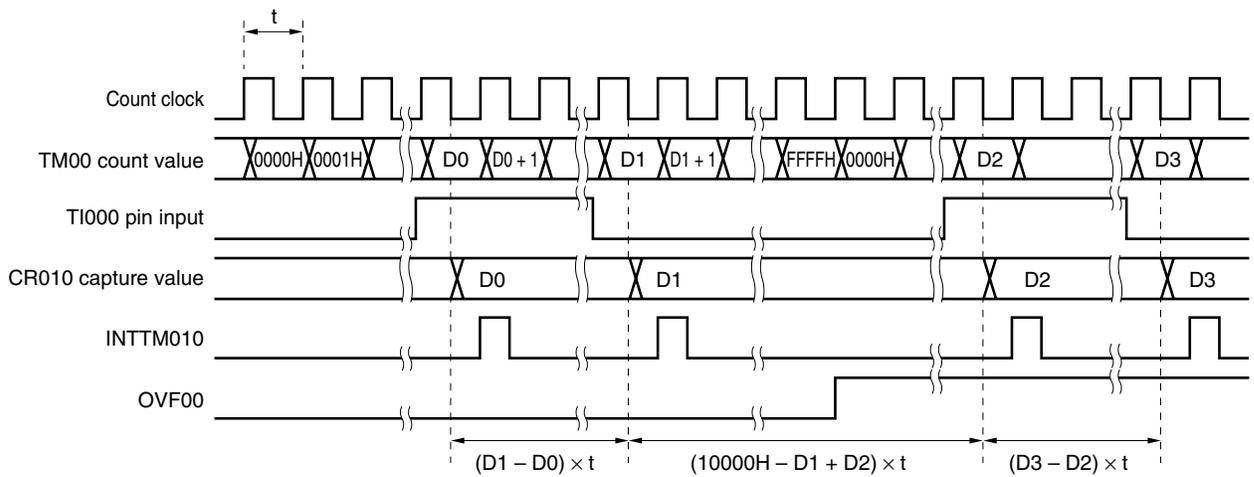


Figure 6-13. Timing of Pulse Width Measurement Operation by Free-Running Counter and One Capture Register (with Both Edges Specified)

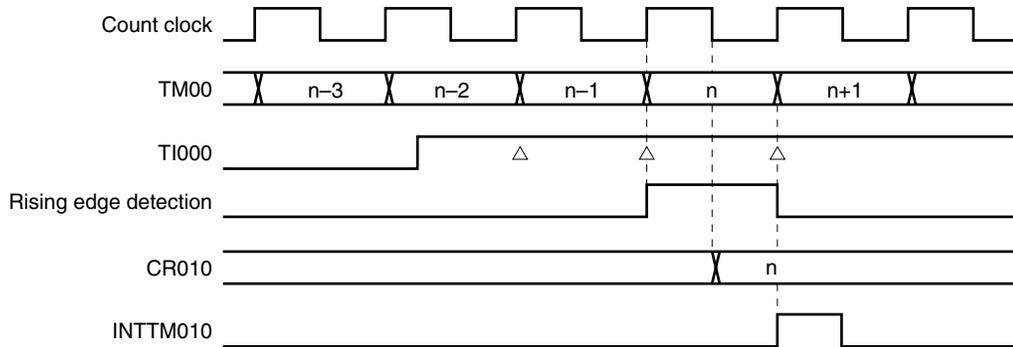




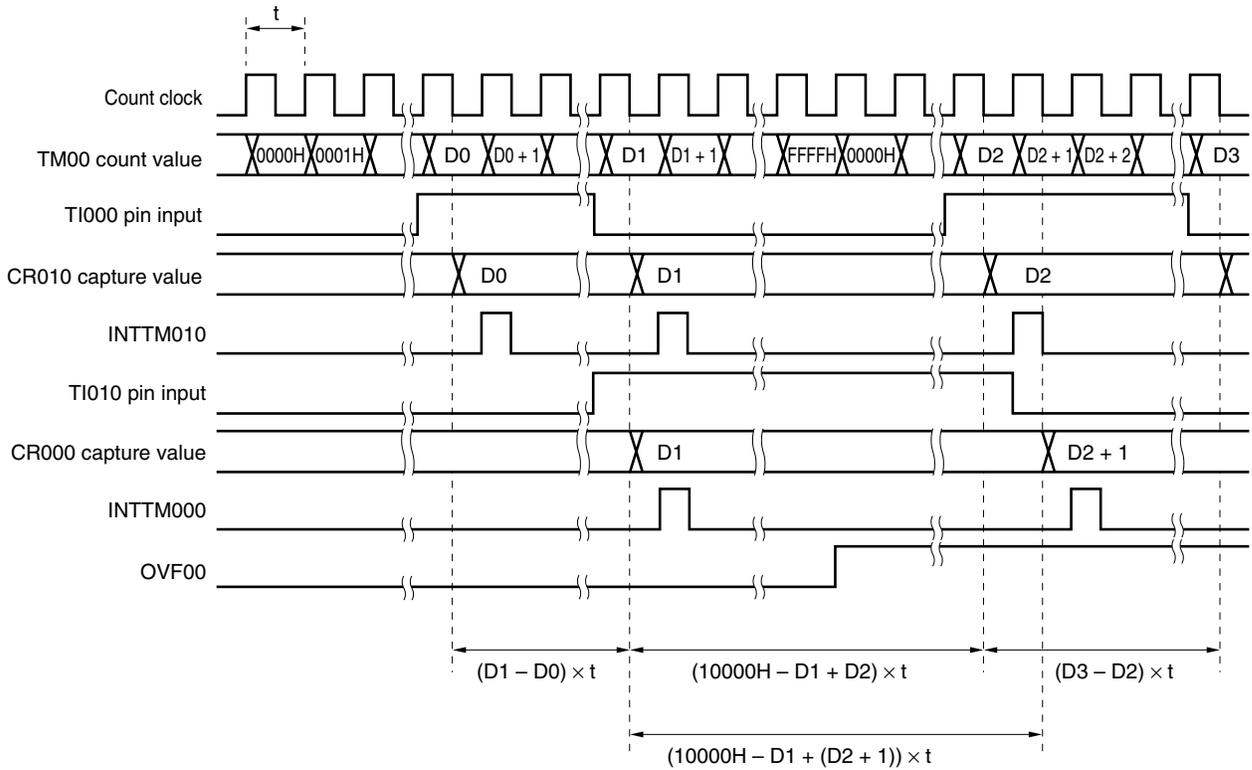
- **Capture operation (Free-running mode)**

Capture register operation in capture trigger input is shown below.

**Figure 6-15. CR010 Capture Operation with Rising Edge Specified**

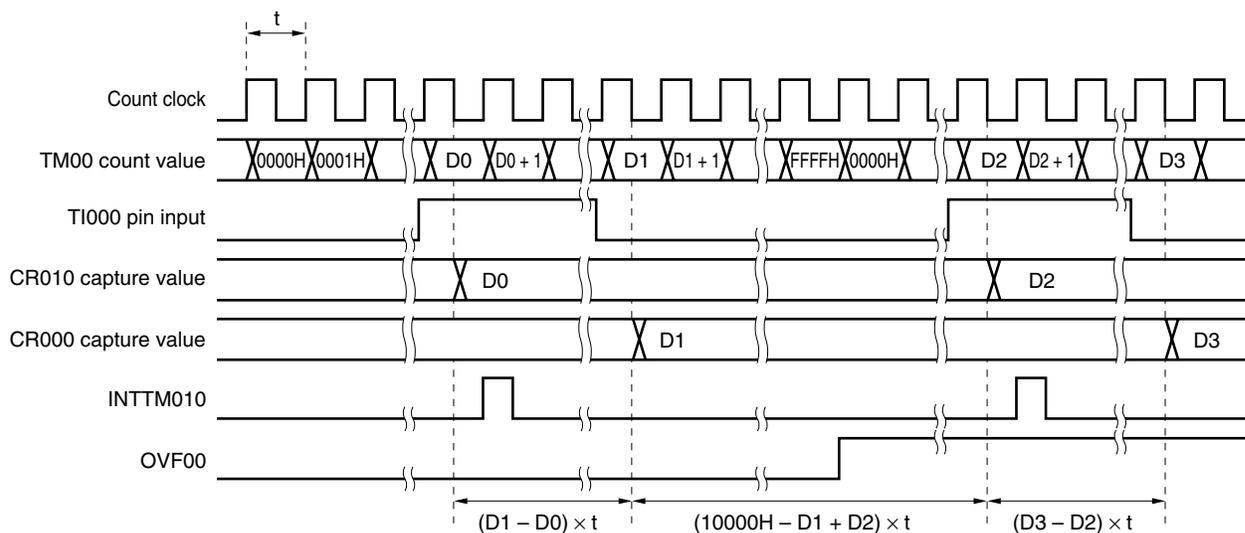


**Figure 6-16. Timing of Pulse Width Measurement Operation with Free-Running Counter (with Both Edges Specified)**





**Figure 6-18. Timing of Pulse Width Measurement Operation by Free-Running Counter and Two Capture Registers (with Rising Edge Specified)**



#### (4) Pulse width measurement by means of restart

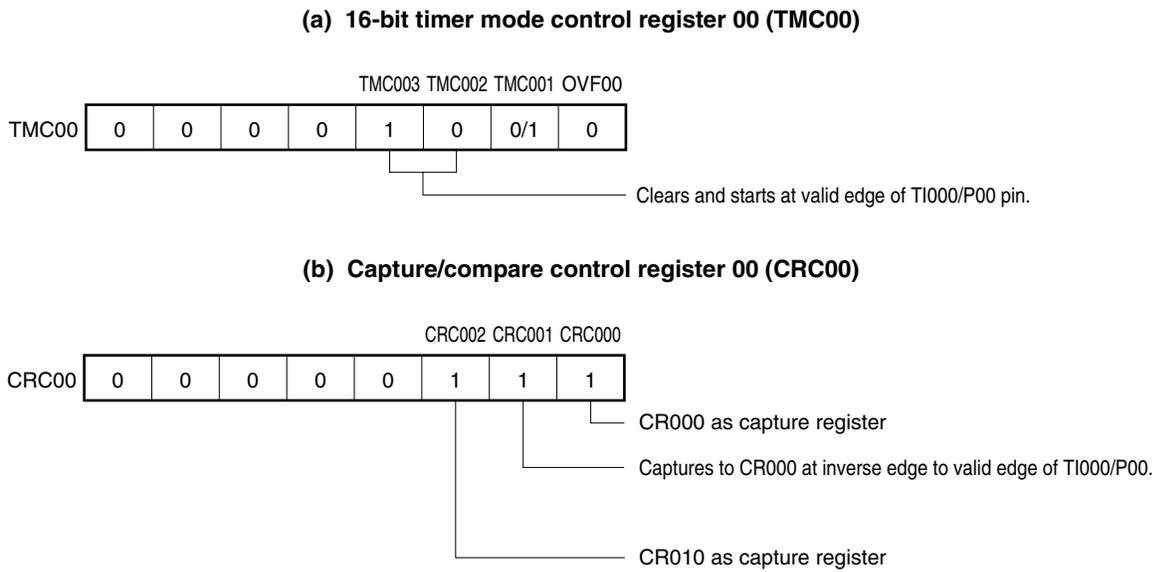
When input of a valid edge to the TI000/P00 pin is detected, the count value of 16-bit timer counter 00 (TM00) is taken into 16-bit timer capture/compare register 010 (CR010), and then the pulse width of the signal input to the TI000/P00 pin is measured by clearing TM00 and restarting the count (see **Figure 6-19**).

Either of two edges can be selected—rising or falling—specified using bits 4 and 5 (ES000 and ES001) of prescaler mode register 00 (PRM00).

In valid edge detection, sampling is performed using the count clock cycle selected by prescaler mode register 00 (PRM00) and a capture operation is only performed when a valid level is detected twice, thus eliminating noise with a short pulse width.

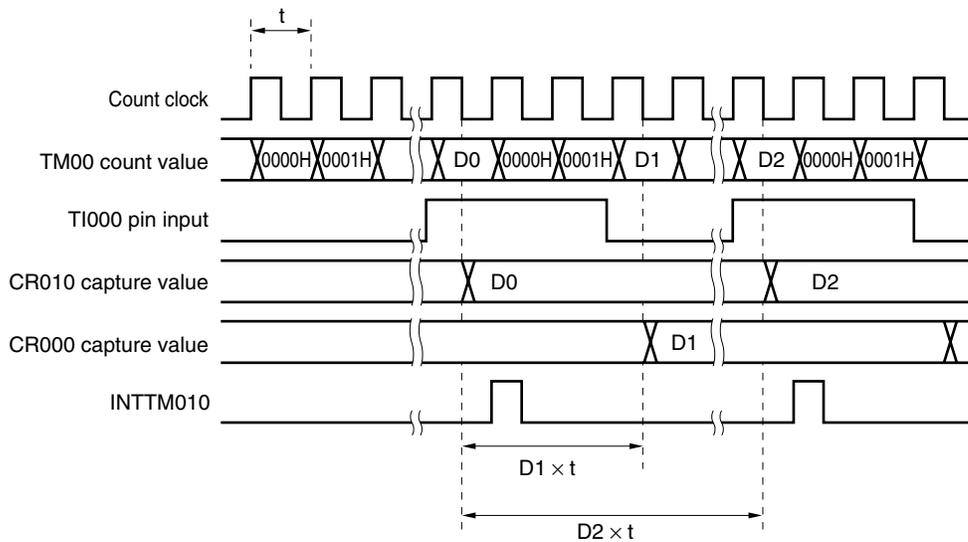
**Caution** If the valid edge of TI000/P00 pin is specified as both the rising and falling edges, 16-bit timer capture/compare register 000 (CR000) cannot perform the capture operation.

Figure 6-19. Control Register Settings for Pulse Width Measurement by Means of Restart



**Remark** 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. For details, see Figure 6-2.

Figure 6-20. Timing of Pulse Width Measurement Operation by Means of Restart (with Rising Edge Specified)



#### 6.4.4 External event counter operation

The external event counter counts the number of external clock pulses input to the TI000/P00 pin using 16-bit timer counter 00 (TM00).

TM00 is incremented each time the valid edge specified by prescaler mode register 00 (PRM00) is input.

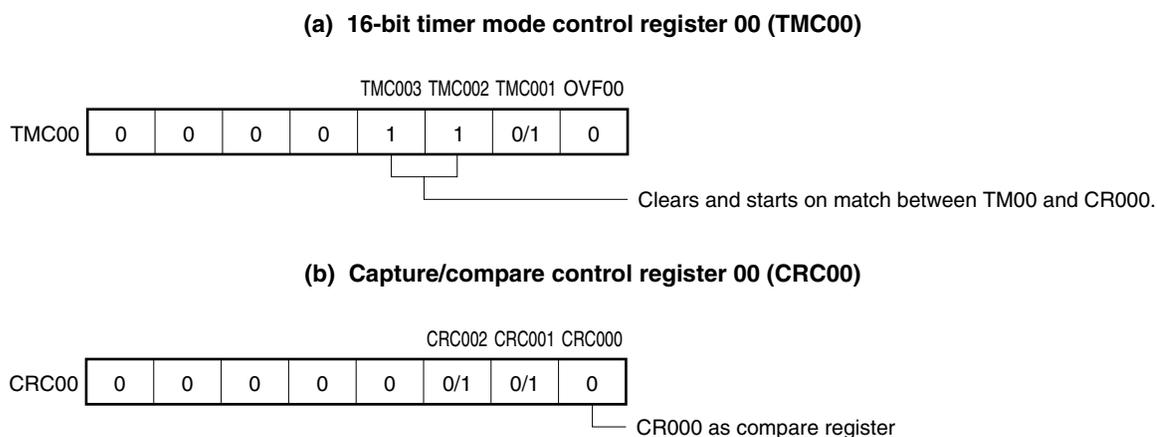
When the TM00 count value matches the 16-bit timer capture/compare register 000 (CR000) value, TM00 is cleared to 0 and the interrupt request signal (INTTM00) is generated.

Input a value other than 0000H to CR000 (a count operation with 1-bit pulse cannot be carried out).

Any of three edges can be selected—rising, falling, or both edges—specified using bits 4 and 5 (ES000 and ES001) of prescaler mode register 00 (PRM00).

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

**Figure 6-21. Control Register Settings in External Event Counter Mode**



**Remark** 0/1: Setting 0 or 1 allows another function to be used simultaneously with the external event counter. For details, see **Figures 6-2** and **6-3**.

Figure 6-22. Configuration Diagram of External Event Counter

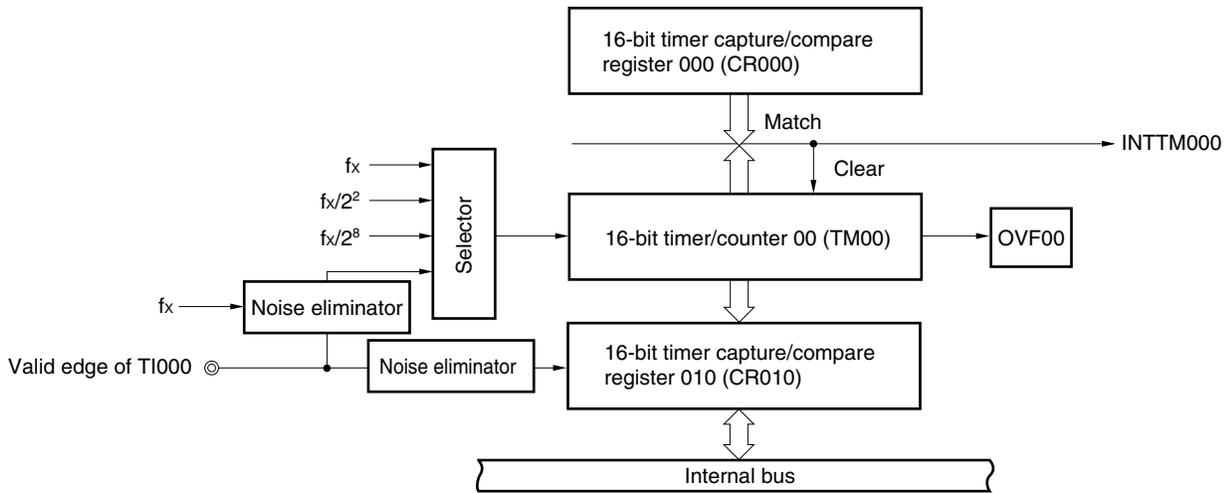
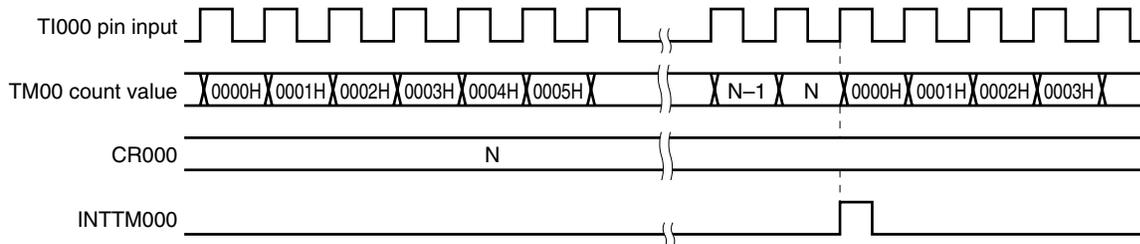


Figure 6-23. External Event Counter Operation Timing (with Rising Edge Specified)



**Caution** When reading the external event counter count value, TM00 should be read.

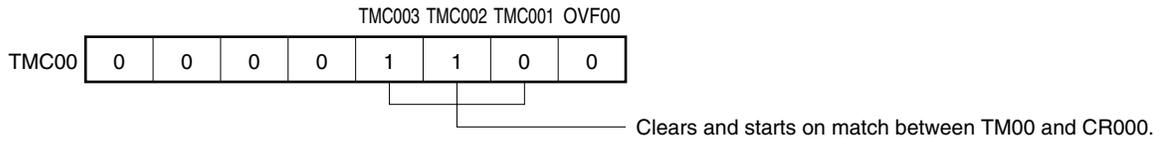
#### 6.4.5 Square-wave output operation

A square wave with any selected frequency can be output at intervals of the count value preset to 16-bit timer capture/compare register 000 (CR000).

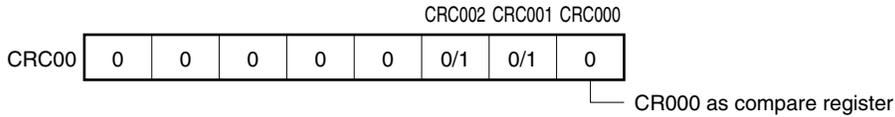
The TO00 pin output status is reversed at intervals of the count value preset to CR00 by setting bit 0 (TOE00) and bit 1 (TOC001) of 16-bit timer output control register 00 (TOC00) to 1. This enables a square wave with any selected frequency to be output.

Figure 6-24. Control Register Settings in Square-Wave Output Mode

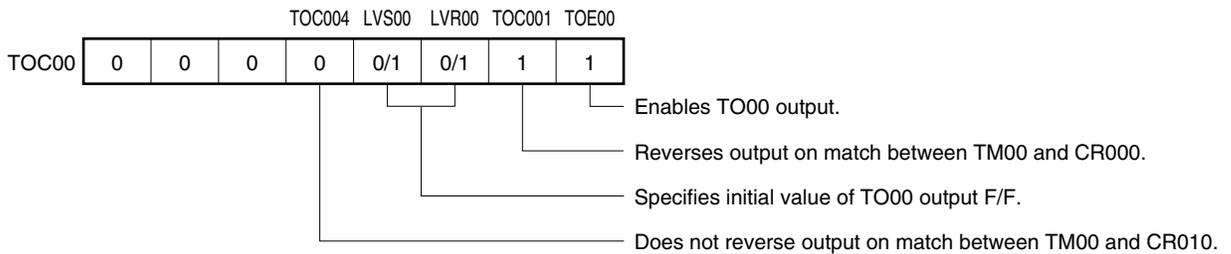
(a) 16-bit timer mode control register 00 (TMC00)



(b) Capture/compare control register 00 (CRC00)

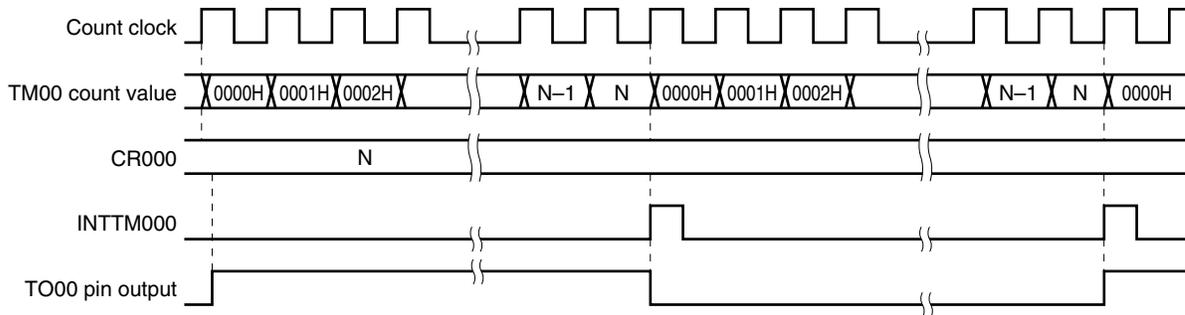


(c) 16-bit timer output control register 00 (TOC00)



**Remark** 0/1: Setting 0 or 1 allows another function to be used simultaneously with square-wave output. For details, see **Figures 6-3** and **6-4**.

Figure 6-25. Square-Wave Output Operation Timing

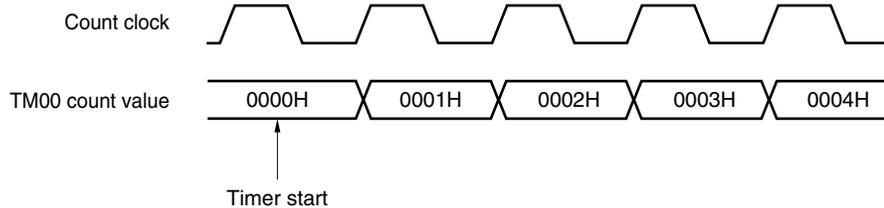


## 6.5 Cautions on 16-Bit Timer/Event Counter 00

### (1) Timer start errors

An error of up to one clock may occur in the time required for a match signal to be generated after timer start. This is because 16-bit timer counter 00 (TM00) is started asynchronously to the count clock.

**Figure 6-26. Start Timing of 16-Bit Timer Counter 00 (TM00)**



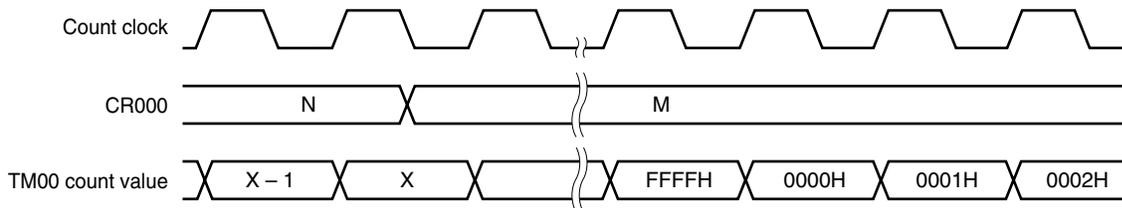
### (2) 16-bit timer capture compare register setting (in clear & start mode entered on match between TM00 and CR000)

Set 16-bit timer capture/compare registers 000, 010 (CR000, CR010) to other than 0000H. This means a 1-pulse count operation cannot be performed when 16-bit timer/event counter 00 is used as an event counter.

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

If the value after 16-bit timer capture/compare register 000 (CR000) is changed is smaller than that of 16-bit timer counter 00 (TM00), TM00 continues counting, overflows and then restarts counting from 0. Thus, if the value (M) after CR000 changes is smaller than that (N) before the change, it is necessary to restart the timer after changing CR000.

**Figure 6-27. Timings After Change of Compare Register During Timer Count Operation**



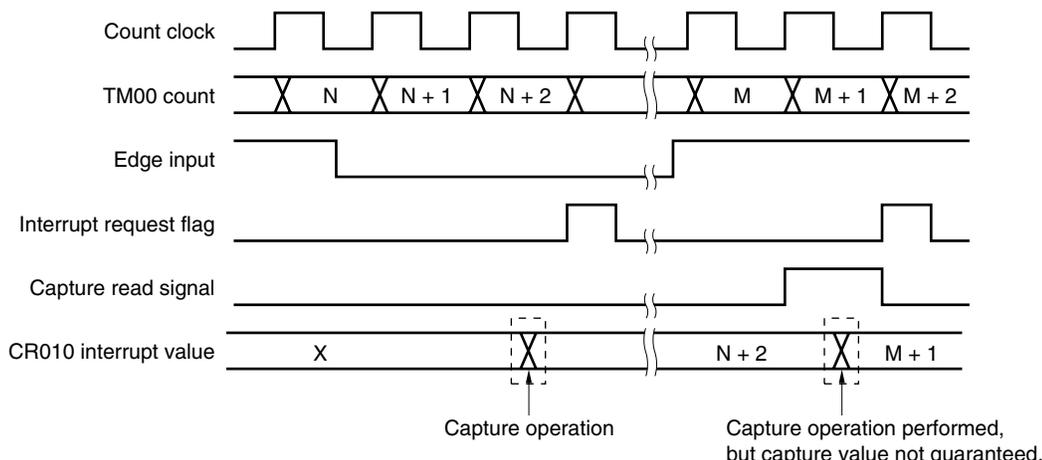
**Remark**  $N > X > M$

**(4) Capture register data retention timing**

If the valid edge of the TI000/P00 pin is input during 16-bit timer capture/compare register 010 (CR010) read, CR010 performs a capture operation. However, the capture value at this time is not guaranteed. The interrupt request flag (TMIF010) is set upon detection of the valid edge.

★

**Figure 6-28. Capture Register Data Retention Timing**



**(5) Valid edge setting**

Set the valid edge of the TI000/P00 pin after setting bits 2 and 3 (TMC002 and TMC003) of 16-bit timer mode control register 00 (TMC00) to 0, 0, respectively, and then stopping timer operation. The valid edge is set using bits 4 and 5 (ES000 and ES001) of prescaler mode register 00 (PRM00).

**(6) Operation of OVF00 flag**

<1> The OVF00 flag is set to 1 in the following case.

When the clear & start mode entered on a match between TM00 and CR000 or the free-running mode entered on a TI00 valid edge is selected

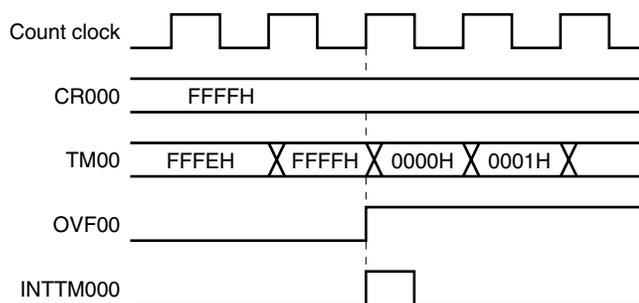
↓

CR000 is set to FFFFH

↓

TM00 is counted up from FFFFH to 0000H.

**Figure 6-29. Operation Timing of OVF00 Flag**



<2> Even if the OVF00 flag is cleared before the next count clock (before TM00 becomes 0001H) after the occurrence of TM00 overflow, the OVF00 flag is re-set newly and clear is disabled.

**(7) Conflicting operations**

Conflict between the read period of the 16-bit timer capture/compare register (CR000/CR010) and capture trigger input (CR000/CR010 used as capture register)

Capture trigger input has priority. The data read from CR000/CR010 is undefined.

**(8) Timer operation**

<1> Even if 16-bit timer counter 00 (TM00) is read, the value is not captured by 16-bit timer capture/compare register 010 (CR010).

<2> Regardless of the CPU's operation mode, when the timer stops, the input signals to the TI000/TI010 pins are not acknowledged.

**(9) Capture operation**

<1> If TI000 is specified as the valid edge of the count clock, a capture operation by the capture register specified as the trigger for TI000 is not possible.

<2> To ensure the reliability of the capture operation, the capture trigger requires a pulse two times longer than the count clock selected by prescaler mode register 00 (PRM00).

<3> The capture operation is performed at the fall of the count clock. An interrupt request input (INTTM000/INTTM010), however, is generated at the rise of the next count clock.

**(10) Compare operation**

<1> When the 16-bit timer capture/compare register (CR000/CR010) is overwritten during timer operation, a match interrupt may be generated or a clear operation may not be performed normally if that value is close to or larger than the timer value.

<2> A capture operation may not be performed for CR000/CR010 set in compare mode even if a capture trigger has been input.

**(11) Edge detection**

<1> If the TI000 or TI010 pin is high level immediately after system reset and the rising edge or both the rising and falling edges are specified as the valid edge of the TI000 or TI010 pin to enable the 16-bit timer counter 00 (TM00) operation, a rising edge is detected immediately after the operation is enabled. Be careful therefore when pulling up the TI000 or TI010 pin. However, the rising edge is not detected at restart after the operation has been stopped once.

<2> The sampling clock used to remove noise differs when the TI000 valid edge is used as the count clock and when it is used as a capture trigger. In the former case, the count clock is  $f_x$ , and in the latter case the count clock is selected by prescaler mode register 00 (PRM00). The capture operation is started only after a valid edge is detected twice by sampling, thus eliminating noise with a short pulse width.

## CHAPTER 7 8-BIT TIMER/EVENT COUNTER 50

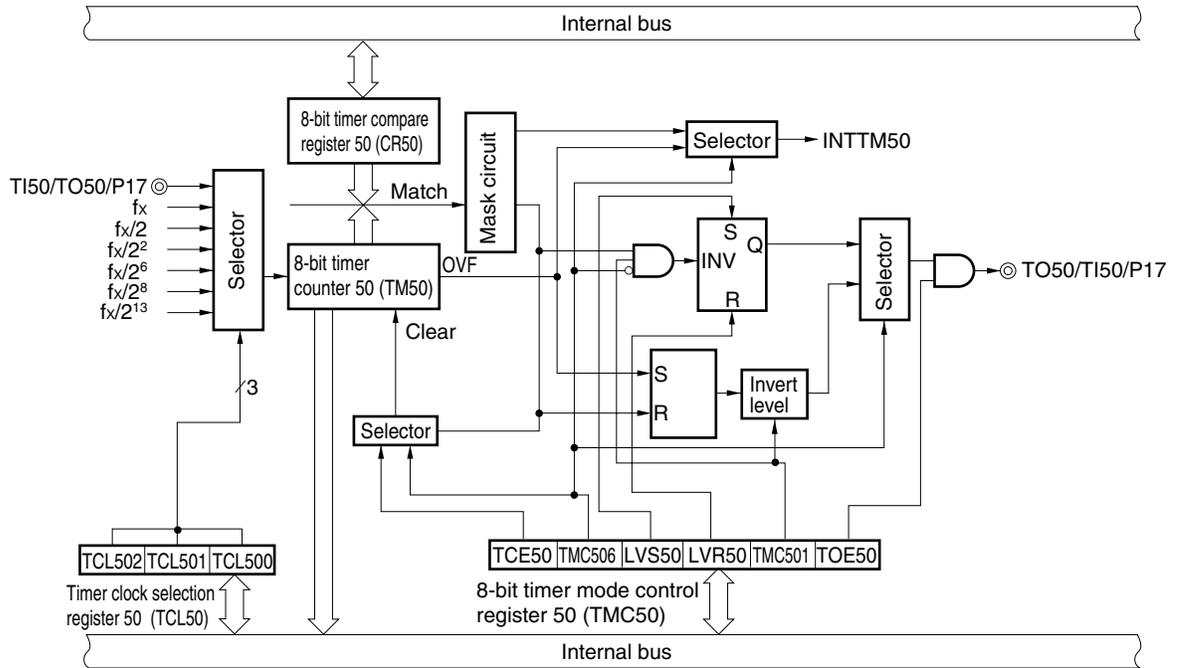
### 7.1 Functions of 8-Bit Timer/Event Counter 50

8-bit timer/event counter 50 (TM50) has the following functions.

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

Figure 7-1 shows the block diagram of 8-bit timer/event counter 50.

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



## 7.2 Configuration of 8-Bit Timer/Event Counter 50

8-bit timer/event counter 50 consists of the following hardware.

**Table 7-1. Configuration of 8-Bit Timer/Event Counter 50**

Item	Configuration
Timer register	8-bit timer counter 50 (TM50)
Register	8-bit timer compare register 50 (CR50)
Timer output	1 (TO50)
Control registers	Timer clock selection register 50 (TCL50) 8-bit timer mode control register 50 (TMC50) Port mode register 1 (PM1) <sup>Note</sup>

**Note** See Figure 4-1 Block Diagram of P10 to P17.

### (1) 8-bit timer counter 50 (TM50)

TM50 is an 8-bit register that counts the count pulses and is read-only.

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

When the count value is read during operation, count clock input is temporarily stopped, and then the count value is read. In the following situations, the count value is cleared to 00H.

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

<2> When TCE50 is cleared

<3> When TM50 and CR50 match in clear & start mode if this mode was entered upon a match of TM50 and CR50 values.

### (2) 8-bit timer compare register 50 (CR50)

CR50 can be read and written by an 8-bit memory manipulation instruction.

Except in PWM mode, the value set in CR50 is constantly compared with the 8-bit timer counter 50 (TM50) count value, and an interrupt request (INTTM50) is generated if they match.

In PWM mode, when the TO50 pin becomes high level due to a TM50 overflow and the values of TM50 and CR50 match, the TO50 pin becomes inactive.

The value of CR50 can be set within 00H to FFH.

**Cautions 1. In the clear & start mode entered on a match of TM50 and CR50 (TMC506 = 0), do not write other values to CR50 during operation.**

**2. In PWM mode, make the CR50 rewrite period 3 count clocks of the count clock (clock selected by TCL50) or more.**

### 7.3 Registers Controlling 8-Bit Timer/Event Counter 50

The following three registers are used to control 8-bit timer/event counter 50.

- Timer clock selection register 50 (TCL50)
- 8-bit timer mode control register 50 (TMC50)
- Port mode register 1 (PM1)

#### (1) Timer clock selection register 50 (TCL50)

This register sets the count clock of 8-bit timer/event counter 50 and the valid edge of TI50 input.

TCL50 can be set by an 8-bit memory manipulation instruction.

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

**Figure 7-2. Format of Timer Clock Selection Register 50 (TCL50)**

Address: FF6AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TCL50	0	0	0	0	0	TCL502	TCL501	TCL500

TCL502	TCL501	TCL500	Count clock selection
0	0	0	TI50 falling edge
0	0	1	TI50 rising edge
0	1	0	$f_x$ (10 MHz)
0	1	1	$f_x/2$ (5 MHz)
1	0	0	$f_x/2^2$ (2.5 MHz)
1	0	1	$f_x/2^3$ (156.25 kHz)
1	1	0	$f_x/2^8$ (39.06 kHz)
1	1	1	$f_x/2^{13}$ (1.22 kHz)

- Cautions**
1. When rewriting TCL50 to other data, stop the timer operation beforehand.
  2. Be sure to set bits 3 to 7 to 0.

- Remarks**
1.  $f_x$ : X1 input clock oscillation frequency
  2. Figures in parentheses apply to operation at  $f_x = 10$  MHz.

**(2) 8-bit timer mode control register 50 (TMC50)**

TMC50 is a register that performs the following five types of settings.

- <1> 8-bit timer counter 50 (TM50) count operation control
- <2> 8-bit timer counter 50 (TM50) operating mode selection
- <3> Timer output F/F (flip-flop) status setting
- <4> Active level selection in timer F/F control or PWM (free-running) mode
- <5> Timer output control

TMC50 can be set by a 1-bit or 8-bit memory manipulation instruction.

RESET input clears this register to 00H.

Figure 7-3 shows the TMC50 format.

**Figure 7-3. Format of 8-Bit Timer Mode Control Register 50 (TMC50)**

Address: FF6BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TMC50	TCE50	TMC506	0	0	LVS50	LVR50	TMC501	TOE50

TCE50	TM50 count operation control
0	After clearing to 0, count operation disabled (counter stopped)
1	Count operation start

TMC506	TM50 operating mode selection
0	Clear & start mode by match between TM50 and CR50
1	PWM (free-running) mode

LVS50	LVR50	Timer output F/F status setting
0	0	No change
0	1	Timer output F/F reset (0)
1	0	Timer output F/F set (1)
1	1	Setting prohibited

TMC501	In other modes (TMC506 = 0)	In PWM mode (TMC506 = 1)
	Timer F/F control	Active level selection
0	Inversion operation disabled	Active high
1	Inversion operation enabled	Active low

TOE50	Timer output control
0	Output disabled (TO50 pin outputs the low level)
1	Output enabled

**Cautions** 1. To clear TCE50 to 0, set the interrupt mask flag (TMMK50) to 1 beforehand. Otherwise, an interrupt may occur when TCE50 is cleared.

TCE50 is cleared to 0 as follows.

```
TMMK50 = 1;   Mask set
TCE50 = 0;   Timer clear
TMIF50 = 0;   Interrupt request flag clear
TMMK50 = 0;   Mask clear
⋮
TCE50 = 1;   Timer start
⋮
```

2. The settings of LVS50 and LVR50 are valid in other than PWM mode.
3. Do not rewrite TMC501 and TOE50 simultaneously.
4. When switching to the PWM mode, do not rewrite TM506 and LVS50 or LVR50 simultaneously.
5. To rewrite TMC506, stop operation beforehand.

**Remarks** 1. In PWM mode, PWM output is made inactive by setting TCE50 to 0.  
 2. If LVS50 and LVR50 are read after data is set, 0 is read.  
 3. The values of the TMC506, LVS50, LVR50, TMC501, and TOE50 bits are reflected at the TO50 pin regardless of the value of TCE50.

**(3) Port mode register 1 (PM1)**

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

When using the P17/TO50/TI50 pin for timer output, set PM17 and the output latches of P17 to 0.

PM1 can be set by a 1-bit or 8-bit memory manipulation instruction.

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

**Figure 7-4. Format of Port Mode Register 1 (PM1)**

Address: FF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	P1n pin input/output mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

## 7.4 Operations of 8-Bit Timer/Event Counter 50

### 7.4.1 Operation as interval timer

8-bit timer/event counter 50 operates as an interval timer that generates interrupt requests repeatedly at intervals of the count value preset to 8-bit timer compare register 50 (CR50).

When the count value of 8-bit timer counter 50 (TM50) matches the value set to CR50, counting continues with the TM50 value cleared to 0 and an interrupt request signal (INTTM50) is generated.

The count clock of TM50 can be selected with bits 0 to 2 (TCL500 to TCL502) of timer clock selection register 50 (TCL50).

#### [Setting]

<1> Set the registers.

- TCL50: Select the count clock.
- CR50: Compare value
- TMC50: Stop the count operation, select clear & start mode entered on a match of TM50 and CR50.  
(TMC50 = 0000xxx0B x = Don't care)

<2> After TCE50 = 1 is set, the count operation starts.

<3> If the values of TM50 and CR50 match, INTTM50 is generated (TM50 is cleared to 00H).

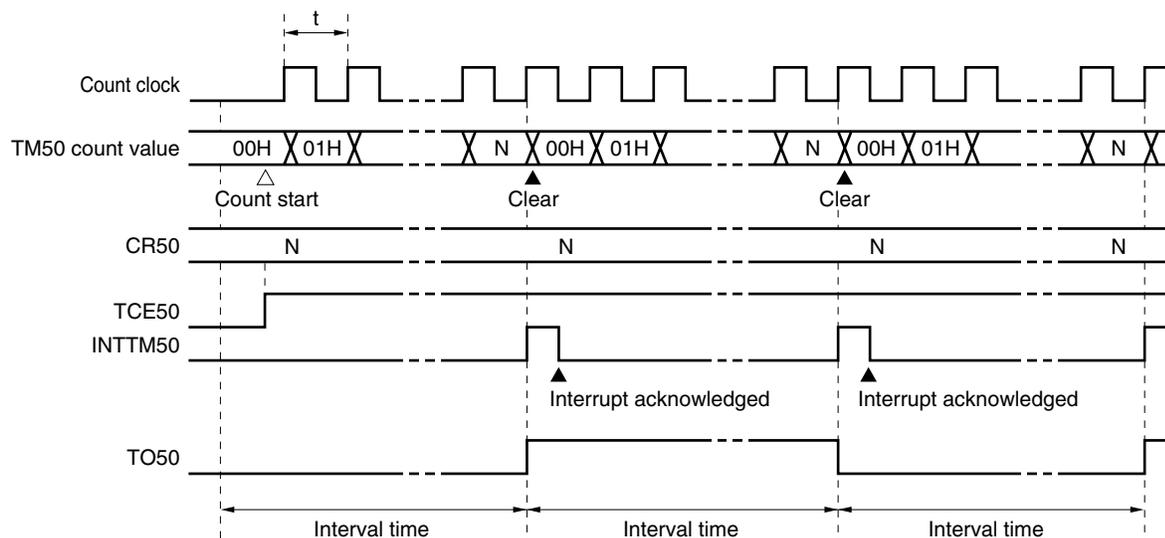
<4> INTTM50 is generated repeatedly at the same interval.

Set TCE50 to 0 to stop the count operation.

**Caution** Do not write other values to CR50 during operation.

Figure 7-5. Interval Timer Operation Timing (1/2)

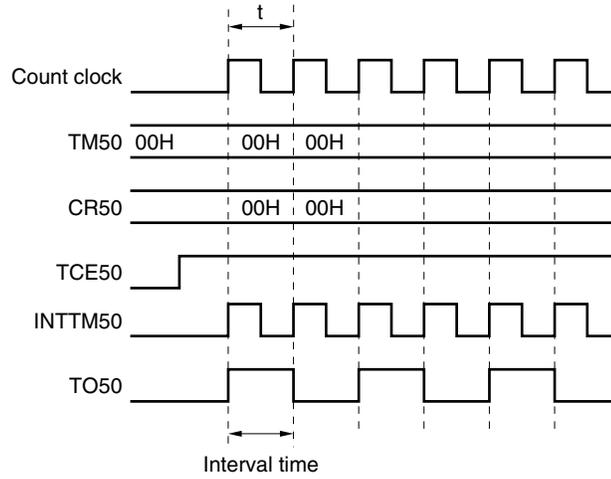
#### (a) Basic operation



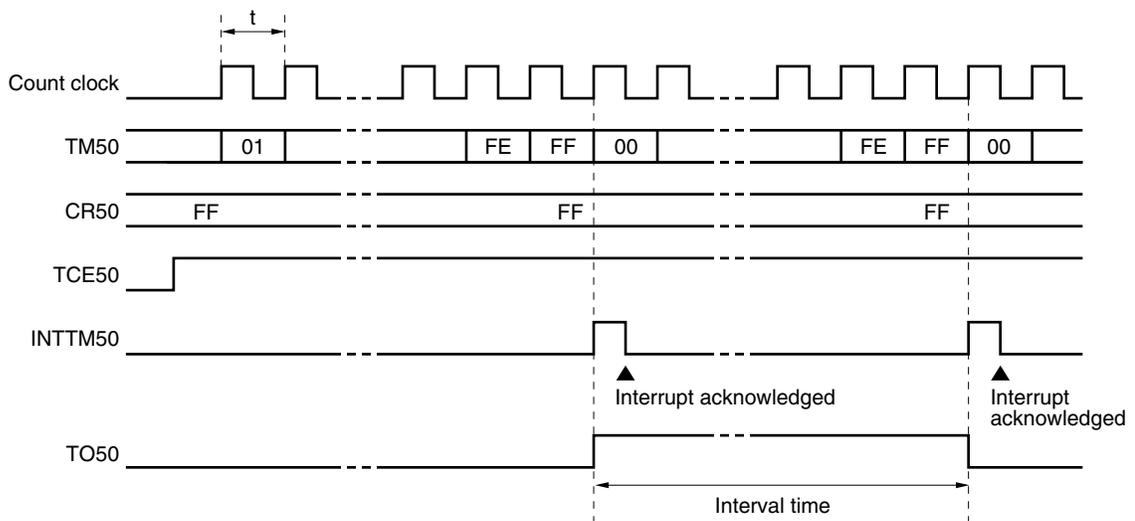
**Remark** Interval time =  $(N + 1) \times t$   
N = 00H to FFH

Figure 7-5. Interval Timer Operation Timing (2/2)

(b) When CR50 = 00H



(c) When CR50 = FFH



### 7.4.2 Operation as external event counter

The external event counter counts the number of external clock pulses to be input to TI50 by 8-bit timer counter 50 (TM50).

TM50 is incremented each time the valid edge specified by timer clock selection register 50 (TCL50) is input. Either the rising or falling edge can be selected.

When the TM50 count value matches the value of 8-bit timer compare register 50 (CR50), TM50 is cleared to 0 and an interrupt request signal (INTTM50) is generated.

Whenever the TM50 count value matches the value of CR50, INTTM50 is generated.

#### [Setting]

<1> Set each register.

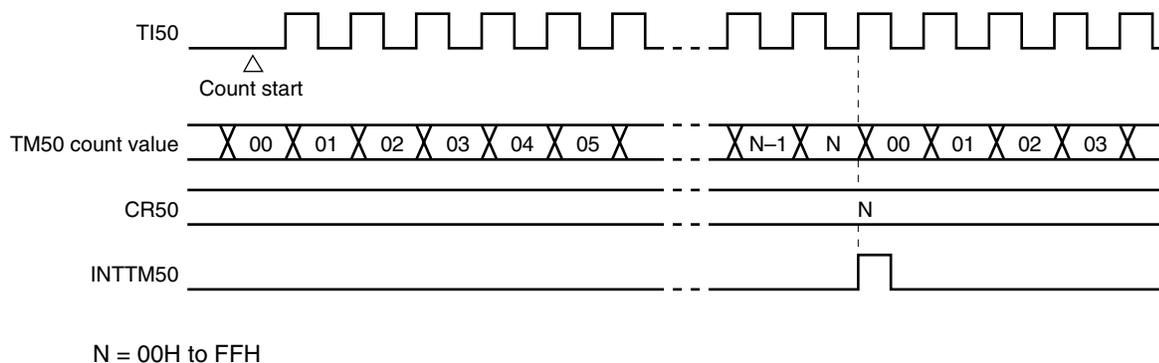
- TCL50: Select TI50 edge.  
TI50 falling edge → TCL50 = 00H  
TI50 rising edge → TCL50 = 01H
- CR50: Compare value
- TMC50: Stop the count operation, select clear & start mode entered on match of TM50 and CR50, disable the timer F/F inversion operation, disable timer output.  
(TMC50 = 0000××00B × = Don't care)

<2> When TCE50 = 1 is set, the number of pulses input from TI50 is counted.

<3> When the values of TM50 and CR50 match, INTTM50 is generated (TM50 is cleared to 00H).

<4> After these settings, INTTM50 is generated each time the values of TM50 and CR50 match.

**Figure 7-6. External Event Counter Operation Timing (with Rising Edge Specified)**



### 7.4.3 Operation as square-wave output

A square wave with any selected frequency is output at intervals of the value preset to 8-bit timer compare register 50 (CR50).

The TO50 pin output status is inverted at intervals of the count value preset to CR50 by setting bit 0 (TOE50) of 8-bit timer mode control register 50 (TMC50) to 1. This enables a square wave with any selected frequency to be output (duty = 50%).

**[Setting]**

<1> Set each register.

- Set the port latch (P17) and port mode register 1 (PM17) to 0.
- TCL50: Select the count clock.
- CR50: Compare value
- TMC50: Stop the count operation, select clear & start mode entered on a match of TM50 and CR50.

LVS50	LVR50	Timer Output F/F Status Setting
1	0	High-level output
0	1	Low-level output

Timer output F/F inversion enabled  
 Timer output enabled  
 (TMC50 = 00001011B or 00000111B)

<2> After TCE50 = 1 is set, the count operation starts.

<3> The timer output F/F is inverted by a match of TM50 and CR50. After INTTM50 is generated, TM50 is cleared to 00H.

<4> After these settings, the timer output F/F is inverted at the same interval and a square wave is output from TO50.

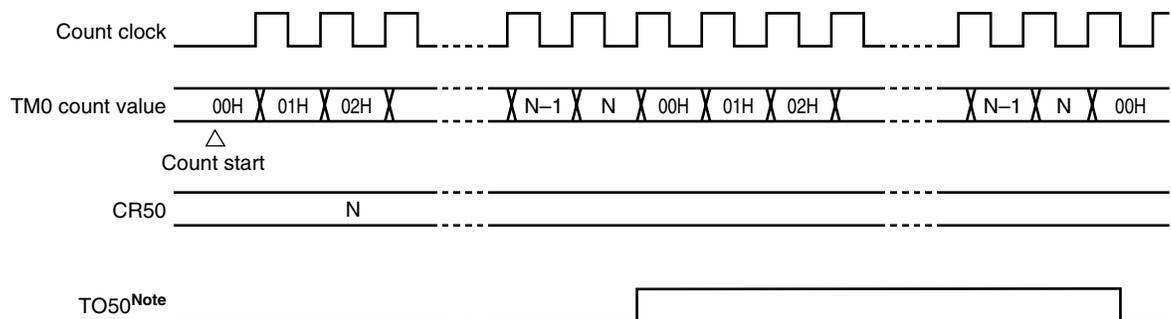
The frequency is as follows.

$$\text{Frequency} = f_{\text{CNT}}/2 (N + 1)$$

(N: 00H to FFH,  $f_{\text{CNT}}$ : Count clock)

**Caution** Do not write other values to CR50 during operation.

**Figure 7-7. Square-Wave Output Operation Timing**



**Note** The initial value of TO50 output can be set by bits 2 and 3 (LVR50, LVS50) of 8-bit timer mode control register 50 (TMC50).

#### 7.4.4 Operation as PWM output

8-bit timer/event counter 50 operates as a PWM output when bit 6 (TMC506) of 8-bit timer mode control register 50 (TMC50) is set to 1.

The duty ratio pulse is determined by the value set to 8-bit timer compare register 50 (CR50).

Set the active level width of the PWM pulse to CR50; the active level can be selected with bit 1 of TMC50 (TMC501).

The count clock can be selected with bits 0 to 2 (TCL500 to TCL502) of timer clock selection register 50 (TCL50).

PWM output can be enabled/disabled with bit 0 of TMC50 (TOE50).

**Caution** In PWM mode, make the CR50 rewrite period 3 count clocks of the count clock (clock selected by TCL50) or more.

#### (1) PWM output basic operation

##### [Setting]

<1> Set each register.

- Set the port latch (P17) and port mode register 1 (PM17) to 0.
- TCL50: Select the count clock.
- CR50: Compare value
- TMC50: Stop the count operation, select PWM mode.

The timer output F/F is not changed.

TMC501	Active Level Selection
0	Active-high
1	Active-low

Timer output enabled

(TMC50 = 01000001B or 01000011B)

<2> The count operation starts when TCE50 = 1.

Set TCE50 to 0 to stop the count operation.

##### [PWM output operation]

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

<2> When an overflow occurs, the active level set in <1> above is output.

The active level is output until CR50 matches the count value of 8-bit timer counter 50 (TM50).

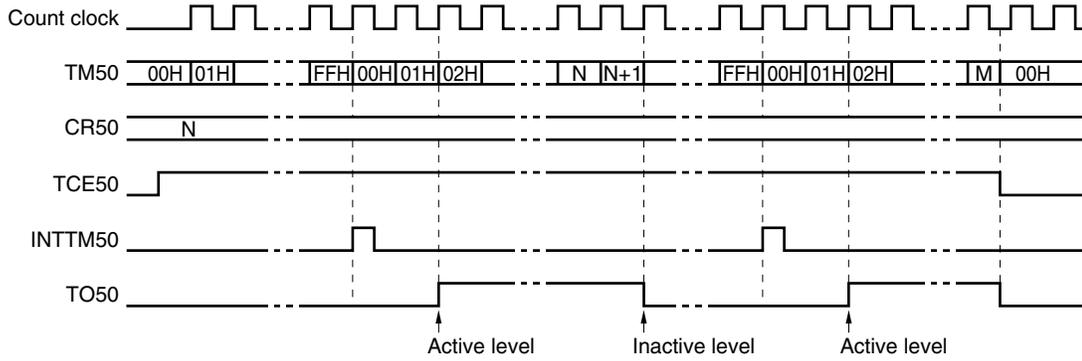
<3> After the CR50 matches the count value, the inactive level is output until an overflow occurs again.

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

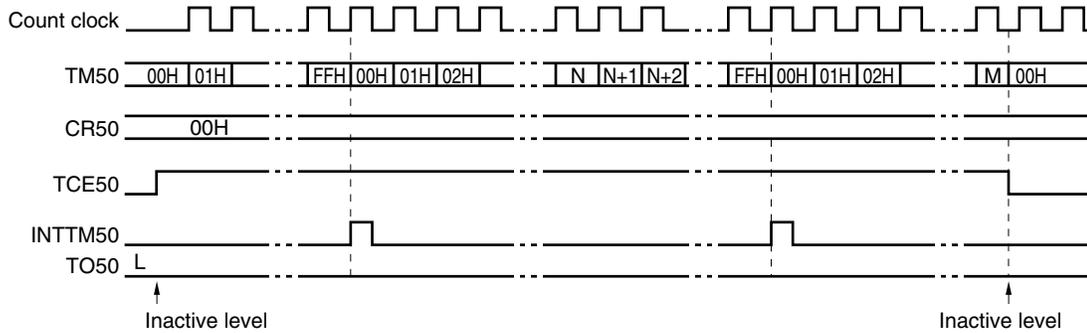
<5> When the count operation is stopped with TCE50 = 0, PWM output becomes inactive.

Figure 7-8. PWM Output Operation Timing

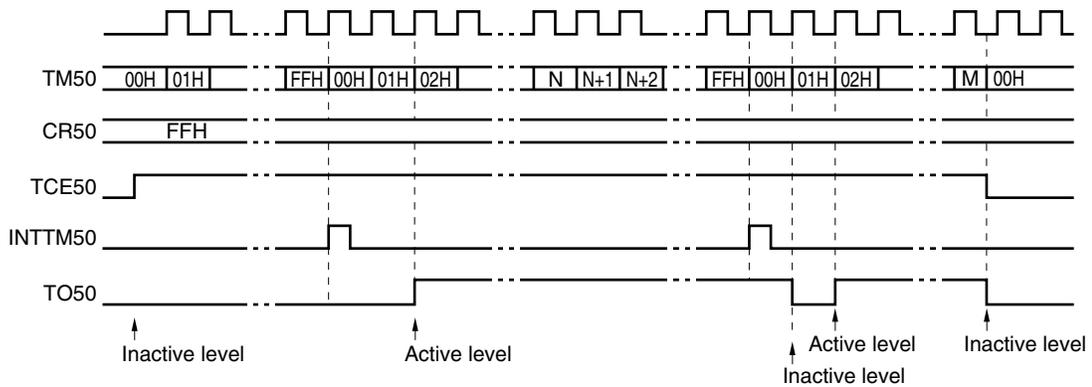
(a) Basic operation (active level = H)



(b) CR50 = 00H



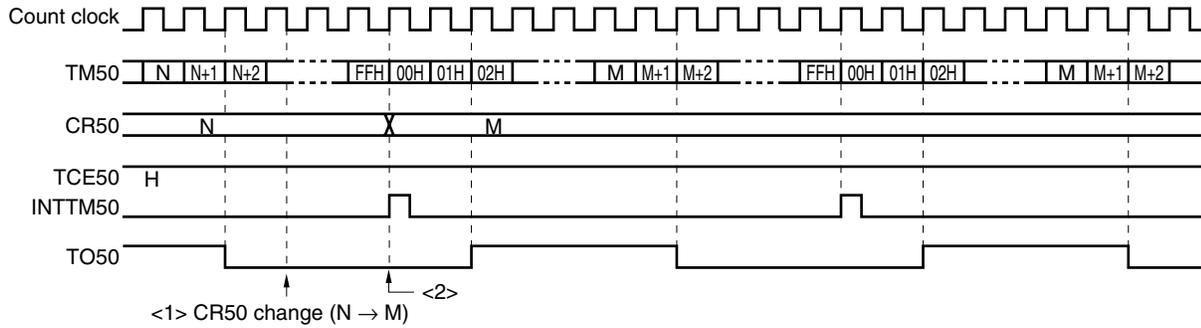
(c) CR50 = FFH



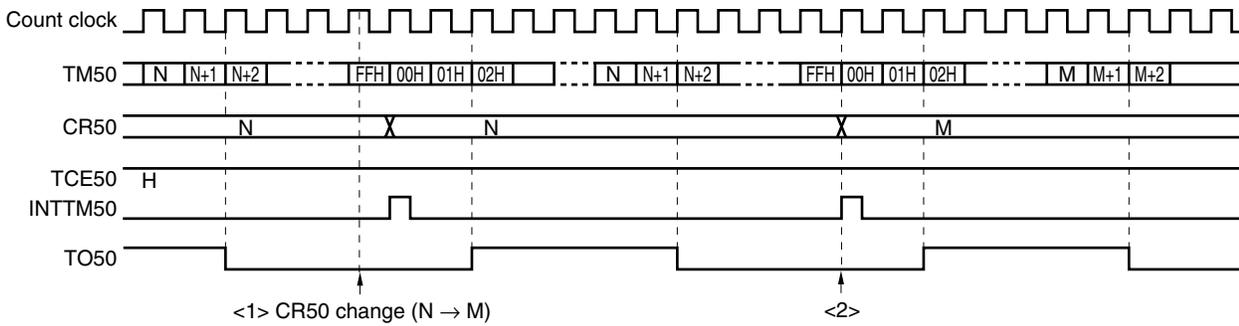
(2) Operation with CR50 changed

Figure 7-9. Timing of Operation with CR50 Changed

- (a) CR50 value is changed from N to M before clock rising edge of FFH  
 → Value is reloaded to CR50 at overflow immediately after change.



- (b) CR50 value is changed from N to M after clock rising edge of FFH  
 → Value is reloaded to CR50 at second overflow.



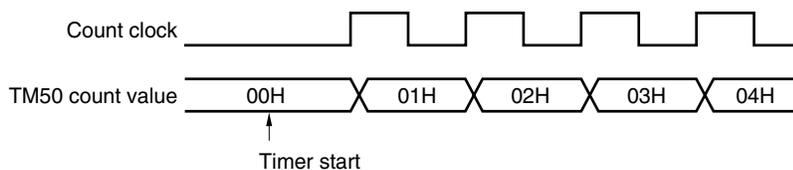
**Caution** When reading from CR50 between <1> and <2> in Figure 7-9, the value read differs from the actual value (read value: M, actual value of CR50: N).

7.5 Cautions on 8-Bit Timer/Event Counter 50

(1) Timer start errors

An error of up to one clock may occur in the time required for a match signal to be generated after timer start. This is because 8-bit timer counter 50 (TM50) is started asynchronously to the count clock.

Figure 7-10. 8-Bit Timer Counter 50 Start Timing



## CHAPTER 8 8-BIT TIMERS H0 AND H1

### 8.1 Functions of 8-Bit Timers H0 and H1

8-bit timers H0 and H1 have the following functions.

- 8-bit-accuracy interval timer
- 8-bit-accuracy PWM pulse generator mode

### 8.2 Configuration of 8-Bit Timers H0 and H1

8-bit timers H0 and H1 consist of the following hardware.

**Table 8-1. Configuration of 8-Bit Timers H0 and H1**

Item	Configuration
Timer register	8-bit timer counter Hn
Register	8-bit timer H compare register 0n (CMP0n) 8-bit timer H compare register 1n (CMP1n)
Timer output	Two outputs (TOHn)
Control register	8-bit timer H mode register n (TMHMDn)

**Remark** n = 0, 1

Figures 8-1 and 8-2 show the block diagrams.

**Figure 8-1. Block Diagram of 8-Bit Timer H0**

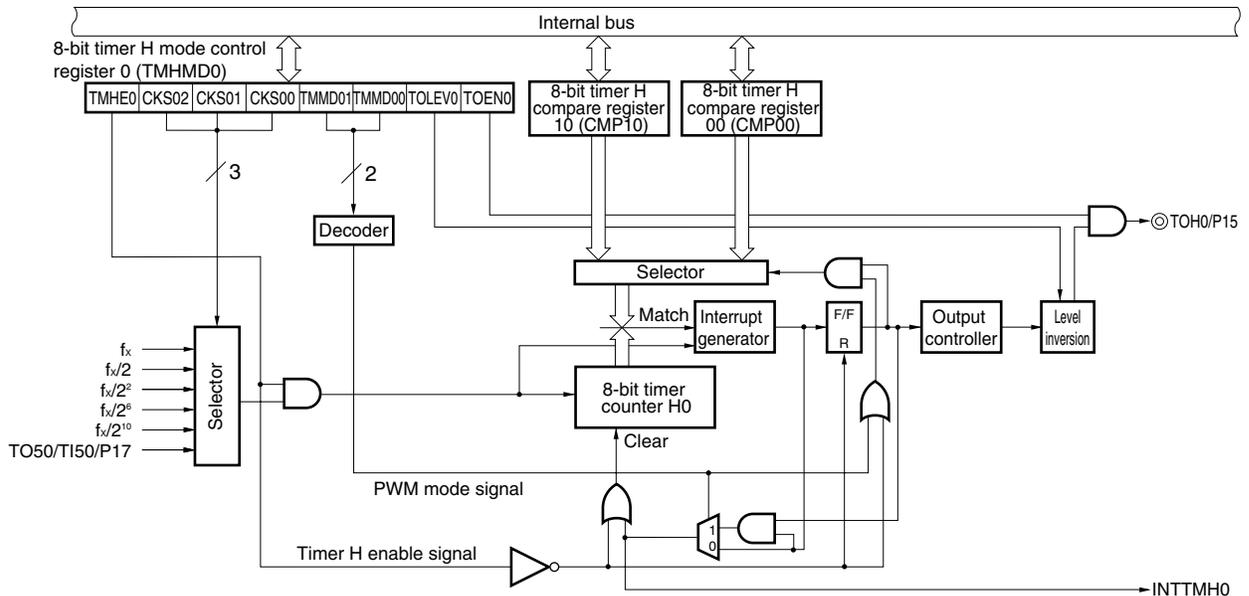
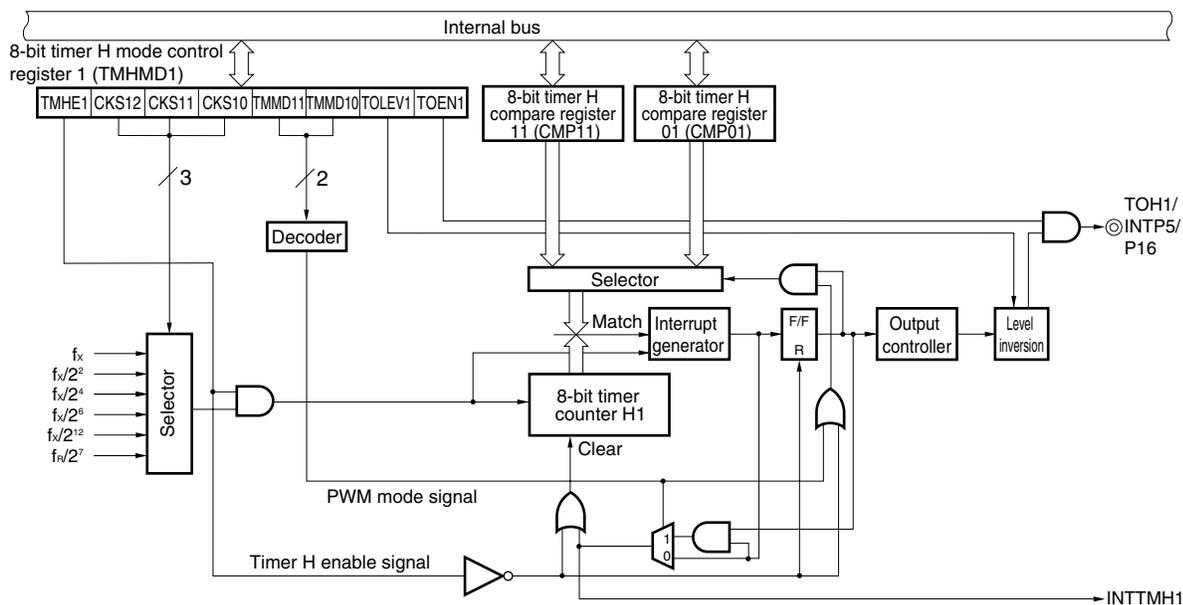


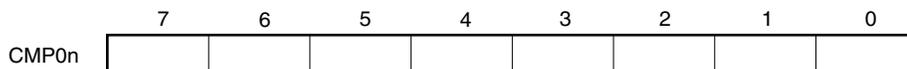
Figure 8-2. Block Diagram of 8-Bit Timer H1



(1) 8-bit timer H compare register 0n (CMP0n)

This register can be read/written by an 8-bit memory manipulation instruction.  $\overline{\text{RESET}}$  input clears this register to 00H.

After reset: 00H R/W Address: FF18H, FF1AH

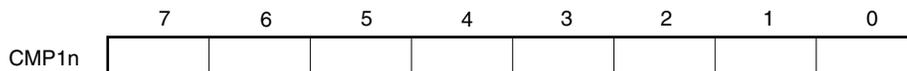


**Caution** This register cannot be rewritten during timer count operation.

(2) 8-bit timer H compare register 1n (CMP1n)

This register can be read/written by an 8-bit memory manipulation instruction.  $\overline{\text{RESET}}$  input clears this register to 00H.

After reset: 00H R/W Address: FF19H, FF1BH



CMP1n register can be rewritten during timer count operation.

If the CMP1n register value is rewritten during timer operation, reloading is performed at the timing at which the counter value and CMP1n register value match. If the transfer timing and writing from CPU to CMP1n register conflict, transfer is not performed.

**Caution** In the PWM pulse generator mode, be sure to set the CMP1n register when starting the timer count operation (TMHEn = 1) after the timer count operation was stopped (TMHEn = 0) (be sure to set again even if setting the same value to the CMP1n register).

**Remark** n = 0, 1

### 8.3 Registers Controlling 8-Bit Timers H0 and H1

8-bit timers H0 and H1 are controlled by 8-bit timer H mode registers 0 and 1 (TMHMD0, TMHMD1).

#### (1) 8-bit timer H mode registers 0 and 1 (TMHMD0, TMHMD1)

These registers control the mode of timer H.

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

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

Figure 8-3. Format of 8-Bit Timer H Mode Register 0 (TMHMD0)

Address: FF69H After reset: 00H R/W

	7	6	5	4	3	2	1	0
TMHMD0	TMHE0	CKS02	CKS01	CKS00	TMMD01	TMMD00	TOLEV0	TOEN0

TMHE0	Enables timer operation
0	Stops timer count operation
1	Enables timer count operation (count operation started by inputting clock)

CKS02	CKS01	CKS00	Count clock ( $f_{CNT}$ ) selection
0	0	0	$f_x$ (10 MHz)
0	0	1	$f_x/2$ (5 MHz)
0	1	0	$f_x/2^2$ (2.5 MHz)
0	1	1	$f_x/2^6$ (156.25 kHz)
1	0	0	$f_x/2^{10}$ (9.77 kHz)
1	0	1	TO50
Other			Setting prohibited

TMMD01	TMMD00	Timer operation mode
0	0	Interval timer mode
1	0	PWM pulse generator mode
Other		Setting prohibited

TOLEV0	Timer output level control (in default mode)
0	Low level
1	High level

TOEN0	Timer output control
0	Disables output
1	Enables output

- Cautions**
1. When **TMHE0 = 1**, setting the other bits of the **TMHMD0** register is disabled.
  2. In the **PWM pulse generator mode**, be sure to set **8-bit timer H compare register 10 (CMP10)** when starting the timer count operation (**TMHE0 = 1**) after the timer count operation was stopped (**TMHE0 = 0**) (be sure to set again even if setting the same value to the **CMP10** register).

- Remarks**
1.  $f_x$ : X1 input clock oscillation frequency
  2. Figures in parentheses apply to operation at  $f_x = 10$  MHz

Figure 8-4. Format of 8-Bit Timer H Mode Register 1 (TMHMD1)

Address: FF6CH After reset: 00H R/W

	7	6	5	4	3	2	1	0
TMHMD1	TMHE1	CKS12	CKS11	CKS10	TMMD11	TMMD10	TOLEV1	TOEN1

TMHE1	Enables timer operation
0	Stops timer count operation
1	Enables timer count operation (count operation started by inputting clock)

CKS12	CKS11	CKS10	Count clock ( $f_{CNT}$ ) selection
0	0	0	$f_x$ (10 MHz)
0	0	1	$f_x/2^2$ (2.5 MHz)
0	1	0	$f_x/2^4$ (625 kHz)
0	1	1	$f_x/2^6$ (156.25 kHz)
1	0	0	$f_x/2^{12}$ (2.44 kHz)
1	0	1	$f_R/2^7$ (1.88 kHz)
Other			Setting prohibited

TMMD11	TMMD10	Timer operation mode
0	0	Interval timer mode
1	0	PWM pulse generator mode
Other		Setting prohibited

TOLEV1	Timer output level control (in default mode)
0	Low level
1	High level

TOEN1	Timer output control
0	Disables output
1	Enables output

- Cautions**
1. When **TMHE1 = 1**, setting the other bits of the **TMHMD1** register is disabled.
  2. In the PWM pulse generator mode, be sure to set 8-bit timer H compare register 11 (**CMP11**) when starting the timer count operation (**TMHE1 = 1**) after the timer count operation was stopped (**TMHE1 = 0**) (be sure to set again even if setting the same value to the **CMP11** register).

- Remarks**
1.  $f_x$ : X1 input clock oscillation frequency
  2.  $f_R$ : Ring-OSC clock oscillation frequency
  3. Figures in parentheses apply to operation at  $f_x = 10$  MHz,  $f_R = 240$  kHz (TYP.).

## 8.4 Operation of 8-Bit Timers H0 and H1

### 8.4.1 Operation as interval timer

When 8-bit timer counter Hn and compare register 0n (CMP0n) match, an interrupt request signal (INTTMHn) is generated and 8-bit timer counter Hn is cleared to 00H.

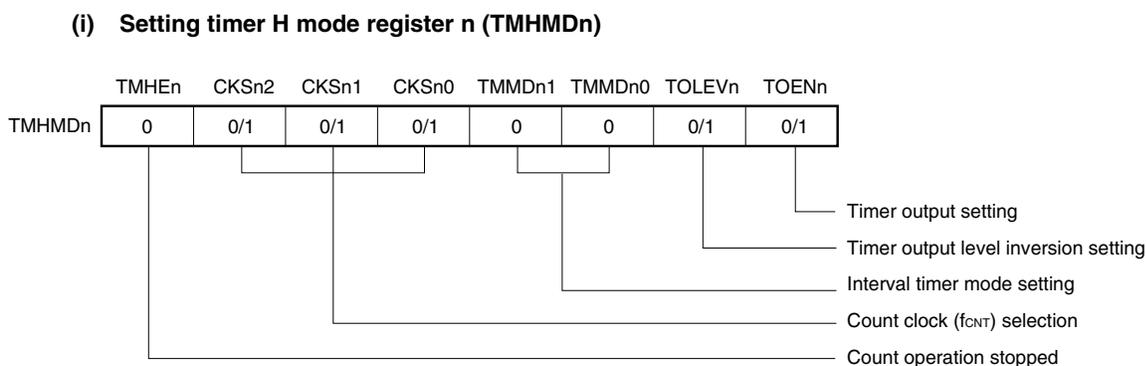
Compare register 1n (CMP1n) is not used in interval timer mode. Since a match of 8-bit timer counter Hn and the CMP1n register is not detected even if the CMP1n register is set, timer output is not affected.

#### (1) Usage

Generates the INTTMHn signal repeatedly at the same interval.

<1> Set each register.

Figure 8-5. Register Setting in Interval Timer Mode



#### (ii) CMP0n register setting

- Compare value (N)

<2> Count operation starts when TMHE<sub>n</sub> = 1.

<3> When the values of 8-bit timer counter Hn and CMP0n register match, the INTTMHn signal is generated and 8-bit timer counter Hn is cleared to 00H.

$$\text{Interval timer} = (N + 1)/f_{\text{CNT}}$$

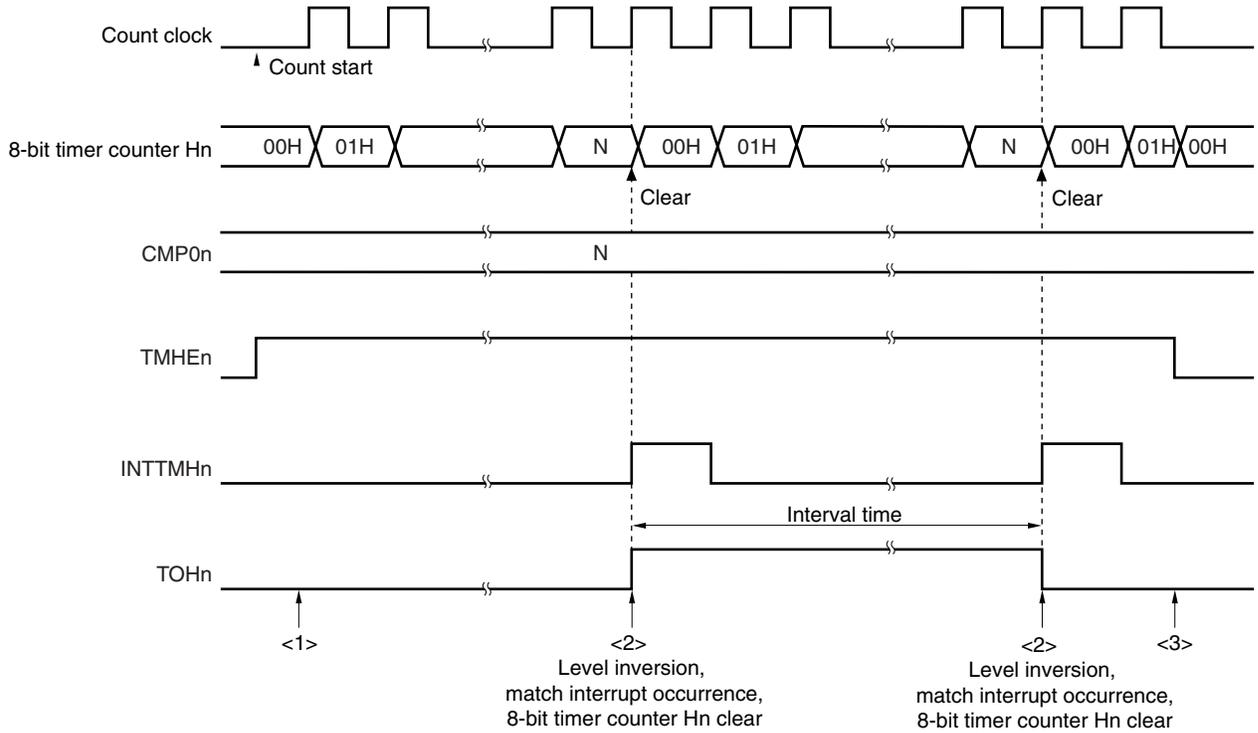
<4> Subsequently, the INTTMHn signal is generated at the same interval. To stop the count operation, set TMHE<sub>n</sub> to 0.

(2) Timing chart

The timing in interval timer mode is shown below.

Figure 8-6. Timing of Interval Timer Operation (1/2)

(a) Basic operation

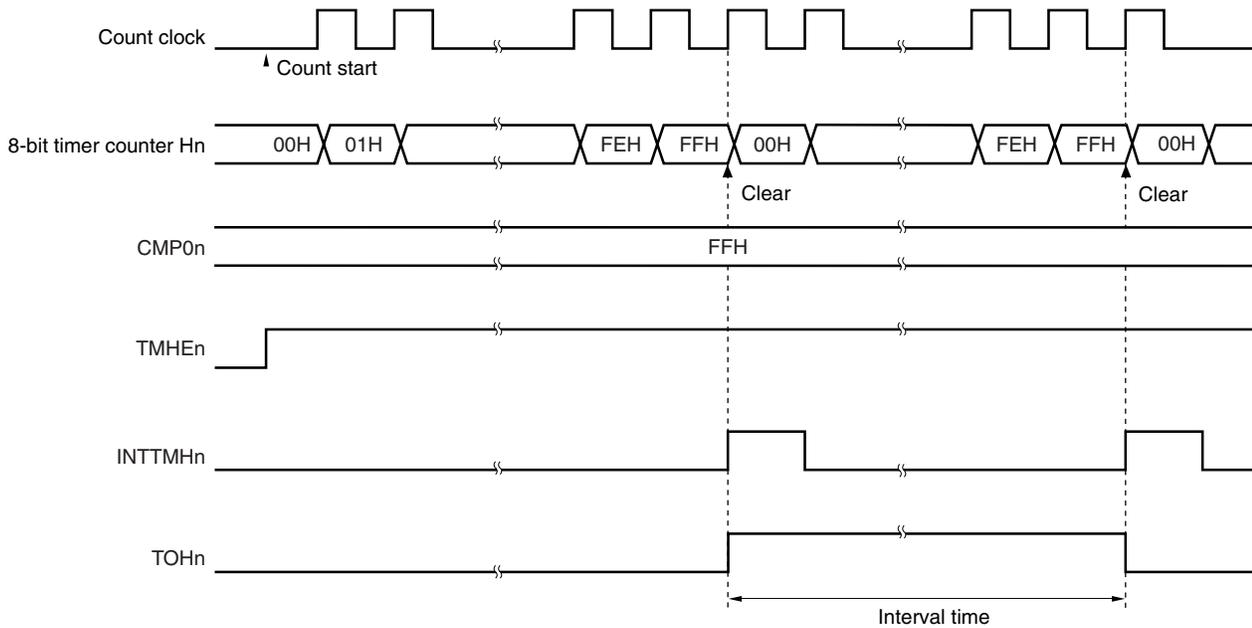


- <1> The count operation is enabled by setting the TMHEn bit to 1. The count clock starts counting no more than 1 clock after the operation is enabled.
- <2> When the values of 8-bit timer counter Hn and the CMP0n register match, the value of 8-bit timer counter Hn is cleared, the TOHn output level is inverted, and the INTTMHn signal is output.
- <3> The INTTMHn signal and TOHn output become inactive by setting the TMHEn bit to 0 during timer Hn operation. If these are inactive from the first, the level is retained.

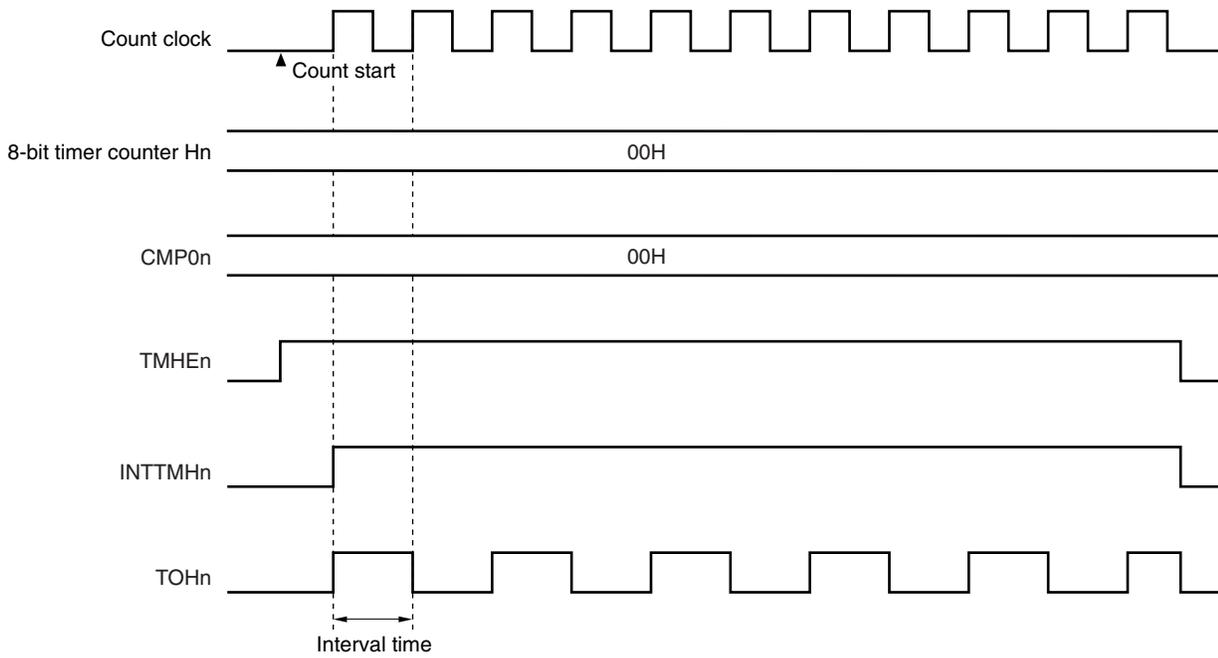
**Remark** n = 0, 1  
N = 00H to FFH

Figure 8-6. Timing of Interval Timer Operation (2/2)

(b) Operation when CMP0n = FFH



(c) Operation when CMP0n = 00H



**Remark** n = 0, 1

**8.4.2 Operation as PWM pulse generator**

In PWM mode, a pulse with arbitrary duty and arbitrary cycle can be output.

8-bit timer compare register 0n (CMP0n) controls the cycle of timer output (TOHn). Rewriting the CMP0n register during timer operation is prohibited.

8-bit timer compare register 1n (CMP1n) controls the duty of timer output (TOHn). Rewriting the CMP1n register during timer operation is possible.

The operation in PWM mode is as follows.

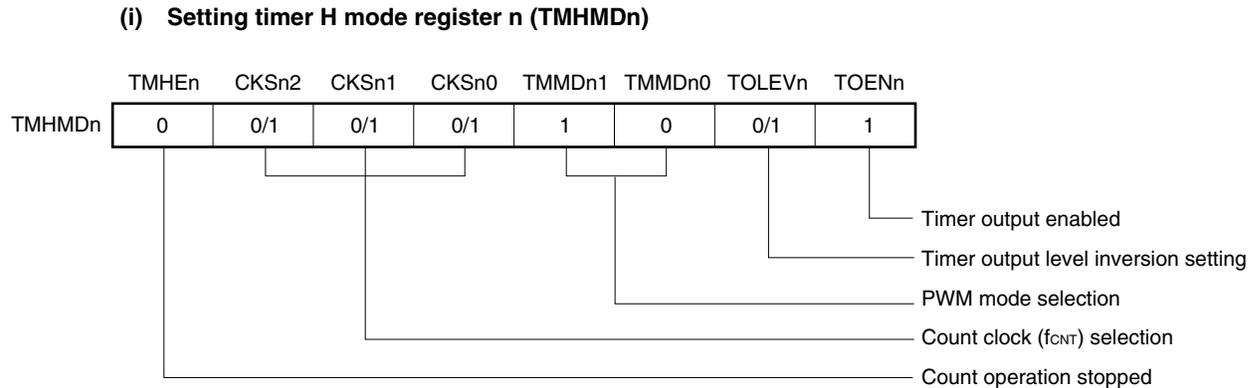
TOHn output becomes active and 8-bit timer counter Hn is cleared to 0 when 8-bit timer counter Hn and the CMP0n register match after the timer count is started. TOHn output becomes inactive when 8-bit timer counter Hn and the CMP1n register match.

**(1) Usage**

In PWM mode, a pulse, to which an arbitrary duty and arbitrary cycle can be set, is output.

<1> Set each register.

**Figure 8-7. Register Setting in PWM Pulse Generator Mode**



**(ii) Setting CMP0n register**

- Compare value (N): Cycle setting

**(iii) Setting CMP1n register**

- Compare value (M): Duty setting

**Remarks 1.** n = 0, 1

**2.** 00H ≤ CMP1n (M) < CMP0n (N) < FFH

<2> The count operation starts when TMHEn = 1.

<3> The CMP0n register is the compare register that is to be compared first after counter operation is enabled. When the values of 8-bit timer counter Hn and the CMP0n register match, 8-bit timer counter Hn is cleared, an interrupt request signal (INTTMHn) is generated, and TOHn output becomes active. At the same time, the compare register to be compared with 8-bit timer counter Hn is changed from the CMP0n register to the CMP1n register.

- <4> When 8-bit timer counter Hn and the CMP1n register match, TOHn output becomes inactive and the compare register to be compared with 8-bit timer counter Hn is changed from the CMP1n register to the CMP0n register. At this time, 8-bit timer counter Hn is not cleared and the INTTMHn signal is not generated.
- <5> By performing procedures <3> and <4> repeatedly, a pulse with an arbitrary duty ratio can be obtained.
- <6> To stop the count operation, set TMHEn = 0.

If the setting value of the CMP0n register is (N), the setting value of CMP1n register is (M), and the count clock frequency is  $f_{CNT}$ , the PWM pulse output cycle and duty ratio are as follows.

$$\begin{aligned} \text{PWM pulse output cycle} &= (N + 1)/f_{CNT} \\ \text{Duty ratio} &= \text{Inactive width: Active width} = (M + 1): (N - M) \end{aligned}$$

- Cautions**
1. In PWM mode, three operation clocks (signal selected using the CKSn2 to CKSn0 bits of the TMHMDn register) are required to transfer the CMP1n register value after rewriting the register.
  2. Be sure to set the CMP1n register when starting the timer count operation (TMHEn = 1) after the timer count operation was stopped (TMHEn = 0) (be sure to set again even if setting the same value to the CMP1n register).

## (2) Timing chart

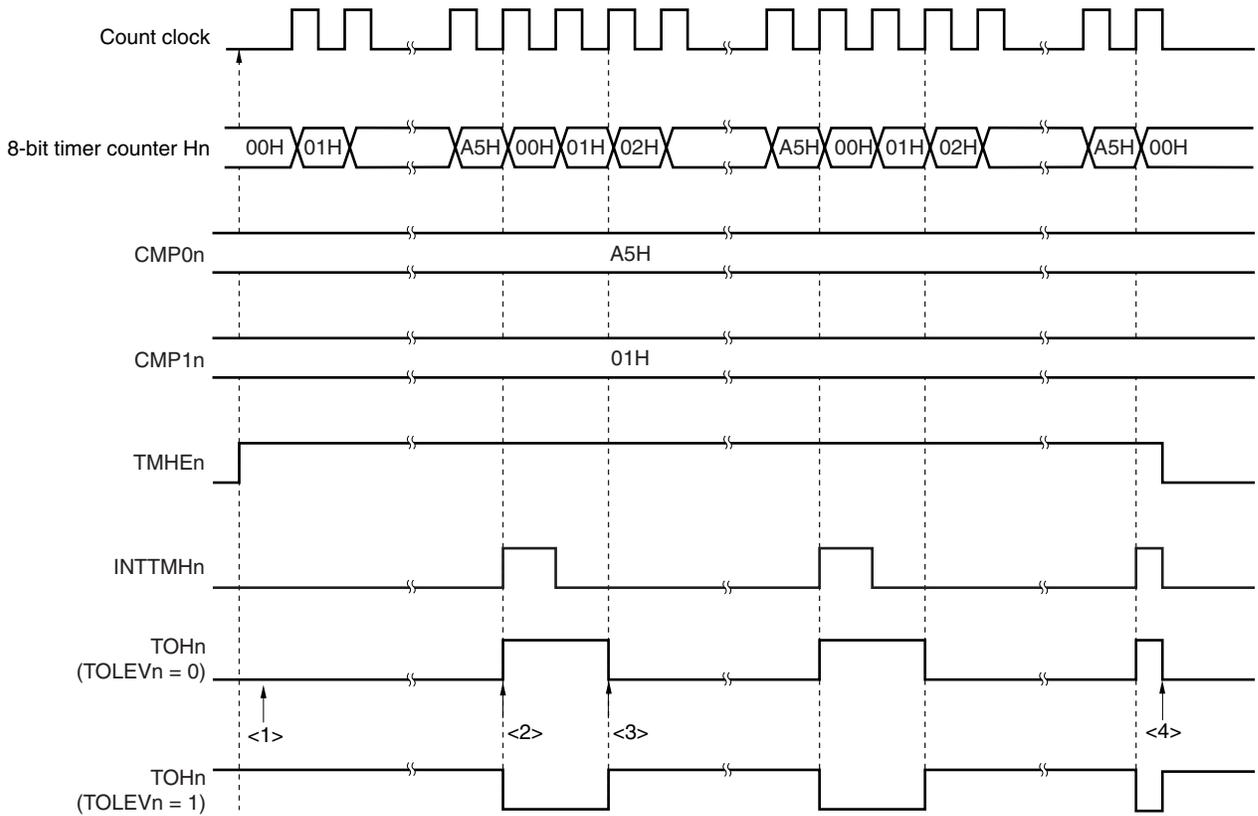
The operation timing in PWM mode is shown below.

**Caution** Make sure that the CMP1n register setting value (M) and CMP0n register setting value (N) are within the following range.

$$00H \leq \text{CMP1n (M)} < \text{CMP0n (N)} < FFH$$

Figure 8-8. Operation Timing in PWM Pulse Generator Mode (1/4)

(a) Basic operation

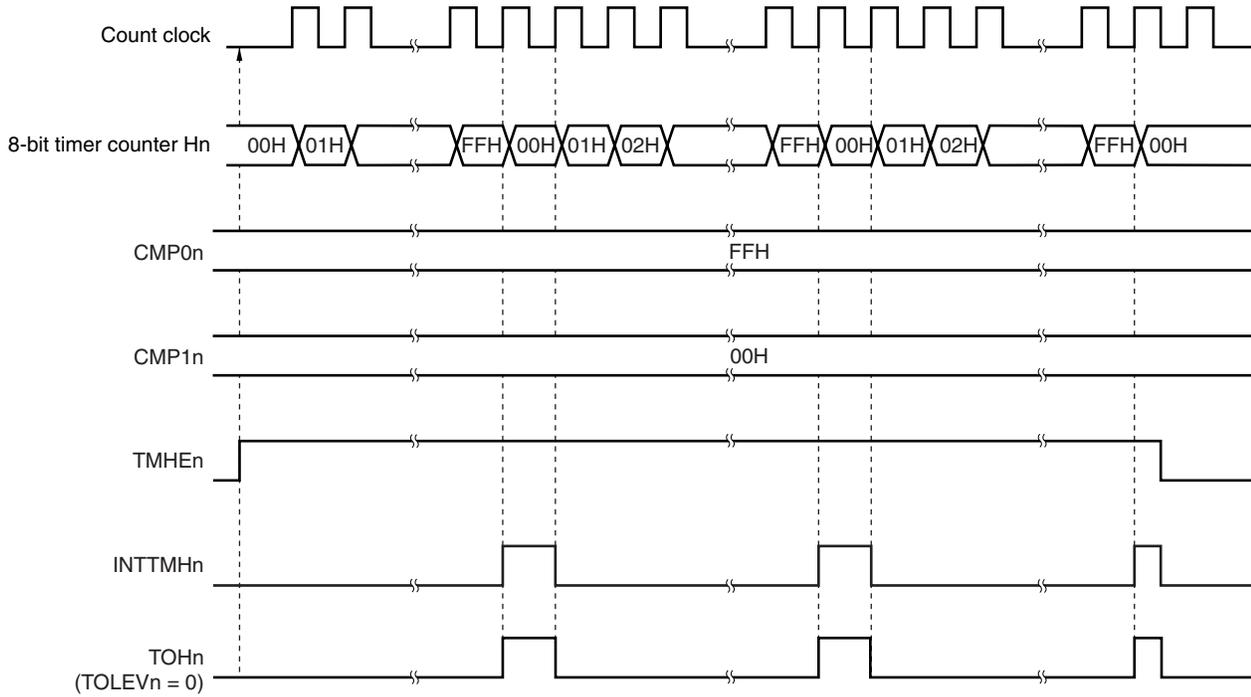


- <1> The count operation is enabled by setting the TMHEn bit to 1. Start 8-bit timer counter Hn by masking one count clock to count up. At this time, TOHn output remains inactive (when TOLEVn = 0).
- <2> When the values of 8-bit timer counter Hn and the CMP0n register match, TOHn output level is inverted, the value of 8-bit timer counter Hn is cleared, and INTTMHn signal is output.
- <3> When the values of 8-bit timer counter Hn and the CMP1n register match, the level of TOHn output is returned. At this time, the 8-bit timer counter value is not cleared and the INTTMHn signal is not output.
- <4> Setting the TMHEn bit to 0 during timer Hn operation makes the INTTMHn signal and TOHn output inactive.

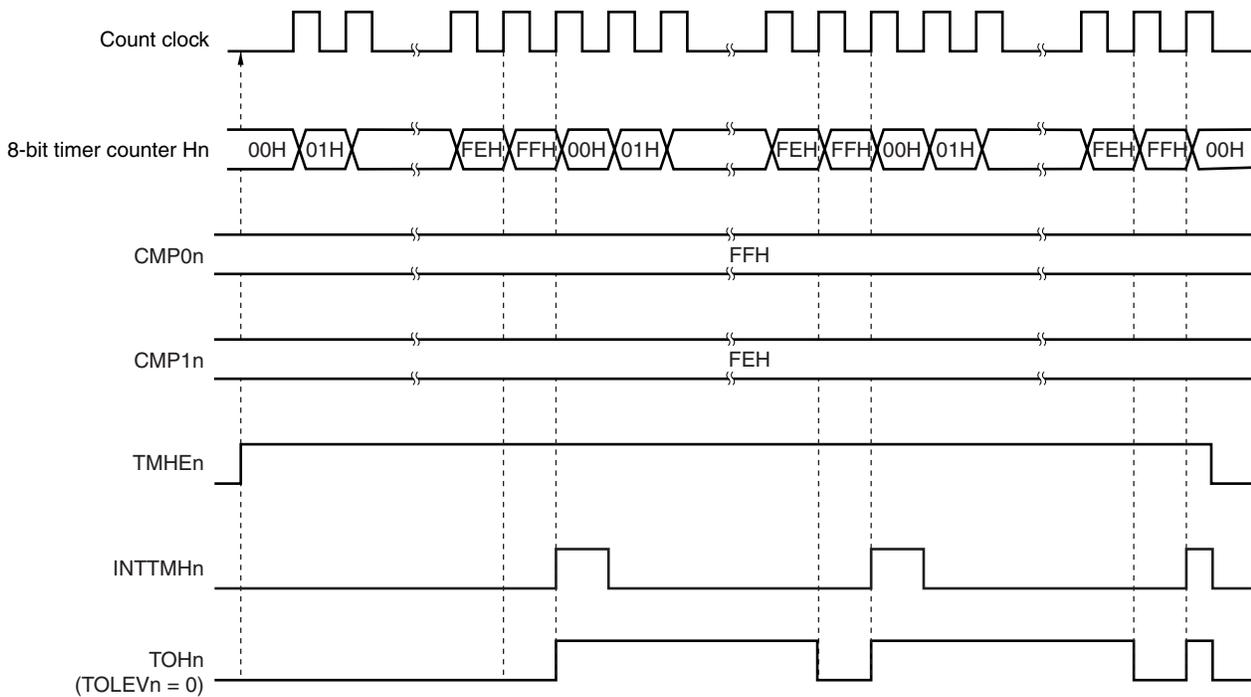
**Remark** n = 0, 1

Figure 8-8. Operation Timing in PWM Pulse Generator Mode (2/4)

(b) Operation when  $CMP0n = FFH$ ,  $CMP1n = 00H$



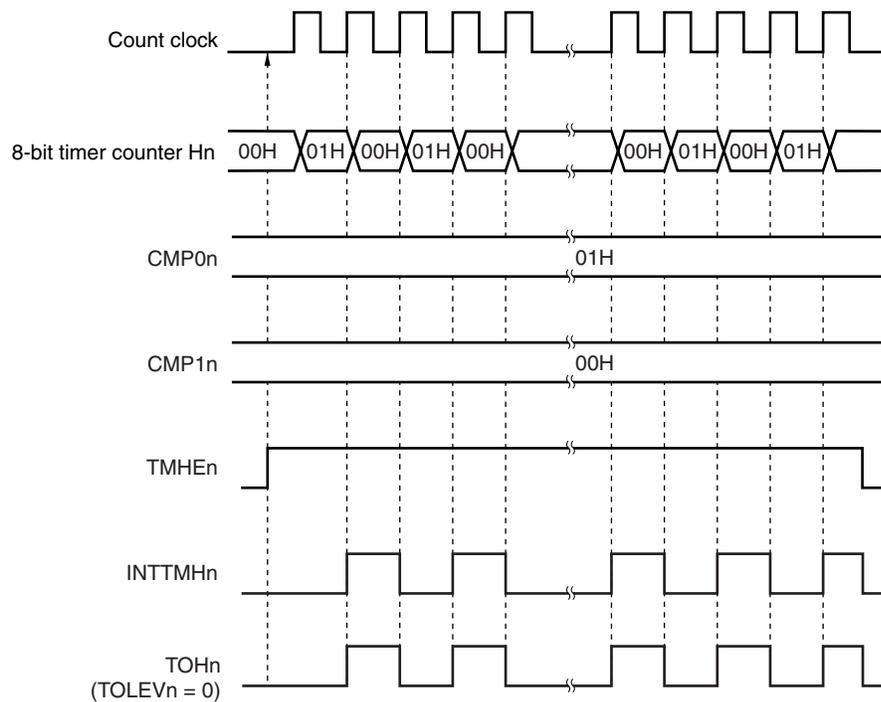
(c) Operation when  $CMP0n = FFH$ ,  $CMP1n = FEH$



**Remark** n = 0, 1

Figure 8-8. Operation Timing in PWM Pulse Generator Mode (3/4)

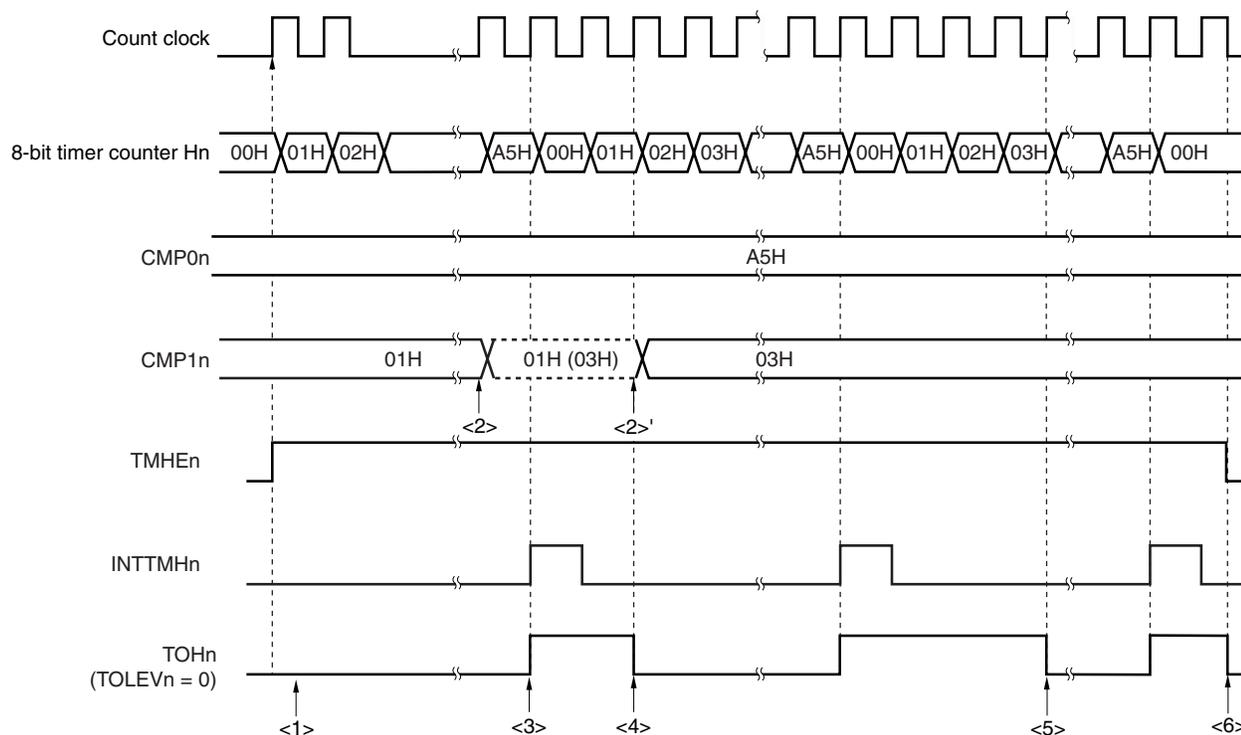
(d) Operation when  $CMP0n = 01H$ ,  $CMP1n = 00H$



**Remark** n = 0, 1

Figure 8-8. Operation Timing in PWM Pulse Generator Mode (4/4)

## (e) Operation by changing CMP1n (CMP1n = 01H → 03H, CMP0n = A5H)



- <1> The count operation is enabled by setting TMHEn = 1. Start 8-bit timer counter Hn by masking one count clock to count up. At this time, TOHn output remains inactive (when TOLEVn = 0).
- <2> The CMP1n register value can be changed during timer counter operation. This operation is asynchronous to the count clock.
- <3> When the values of 8-bit timer counter Hn and the CMP0n register match, the value of 8-bit timer counter Hn is cleared, TOHn output becomes active, and the INTTMHn signal is output.
- <4> If the CMP1n register value is changed, the value is latched and not transferred to the register. When the values of 8-bit timer counter Hn and the CMP1n register before the change match, the value is transferred to the CMP1n register and the CMP1n register value is changed (<2>').  
However, three count clocks or more are required from when the CMP1n register value is changed to when the value is transferred to the register. If a match signal is generated within three count clocks, the changed value cannot be transferred to the register.
- <5> When the values of 8-bit timer counter Hn and the CMP1n register after the change match, TOHn output becomes inactive. 8-bit timer counter Hn is not cleared and the INTTMHn signal is not generated.
- <6> Setting the TMHEn bit to 0 during timer Hn operation makes the INTTMHn signal and TOHn output inactive.

## CHAPTER 9 WATCHDOG TIMER

### 9.1 Functions of Watchdog Timer

The watchdog timer detects an inadvertent program loop. If a program loop is detected, an internal reset signal (WDTRES) is generated.

When a reset occurs due to the watchdog timer, bit 4 (WDTRF) of the reset control flag register (RESF) is set to 1. For details of RESF, refer to **CHAPTER 16 RESET FUNCTION**.

**Table 9-1. Loop Detection Time of Watchdog Timer**

Loop Detection Time	
During Ring-OSC Clock Operation	During X1 Input Clock Operation
$f_R/2^{11}$ (8.53 ms)	$f_{XP}/2^{13}$ (819.2 $\mu$ s)
$f_R/2^{12}$ (17.07 ms)	$f_{XP}/2^{14}$ (1.64 ms)
$f_R/2^{13}$ (34.13 ms)	$f_{XP}/2^{15}$ (3.28 ms)
$f_R/2^{14}$ (68.27 ms)	$f_{XP}/2^{16}$ (6.55 ms)
$f_R/2^{15}$ (136.53 ms)	$f_{XP}/2^{17}$ (13.11 ms)
$f_R/2^{16}$ (273.07 ms)	$f_{XP}/2^{18}$ (26.21 ms)
$f_R/2^{17}$ (546.13 ms)	$f_{XP}/2^{19}$ (52.43 ms)
$f_R/2^{18}$ (1.09 s)	$f_{XP}/2^{20}$ (104.86 ms)

- Remarks**
1.  $f_R$ : Ring-OSC clock oscillation frequency
  2.  $f_{XP}$ : X1 input clock oscillation frequency
  3. Figures in parentheses apply to operation at  $f_R = 240$  kHz (TYP.),  $f_{XP} = 10$  MHz

The operation mode of the watchdog timer (WDT) is switched according to the mask option setting of the on-chip Ring-OSC as shown in Table 9-2.

**Table 9-2. Mask Option Setting and Watchdog Timer Operation Mode**

	Mask Option	
	Ring-OSC Cannot Be Stopped	Ring-OSC Can Be Stopped by Software
Watchdog timer clock source	Fixed to $f_R$ <sup>Note 1</sup> .	<ul style="list-style-type: none"> <li>• Selectable by software (<math>f_{XP}</math>, <math>f_R</math> or stopped)</li> <li>• When reset is released: <math>f_R</math></li> </ul>
Operation after reset	Operation starts with the maximum interval.	Operation starts with maximum interval.
Operation mode selection	The interval can be changed only once.	The clock selection/interval can be changed only once.
Features	<ul style="list-style-type: none"> <li>• The watchdog timer cannot be stopped.</li> <li>• Current in STOP mode <math>\approx 10 \mu A</math></li> </ul>	The watchdog timer can be stopped in standby mode <sup>Note 2</sup> .

**Notes** 1. As long as power is being supplied, Ring-OSC oscillation absolutely cannot be stopped (except during reset).

2. Clock supply to the watchdog timer is stopped in accordance with the watchdog timer clock source as follows:

<1> When the clock source is  $f_{XP}$

Clock supply to the watchdog timer is stopped while  $f_{XP}$  is stopped and during the oscillation stabilization time.

<2> When the clock source is  $f_R$

Clock supply to the watchdog timer is stopped if  $f_R$  is stopped by software before STOP instruction execution when the CPU clock is  $f_{XP}$ .

**Remarks** 1.  $f_R$ : Ring-OSC clock oscillation frequency

2.  $f_{XP}$ : X1 input clock oscillation frequency

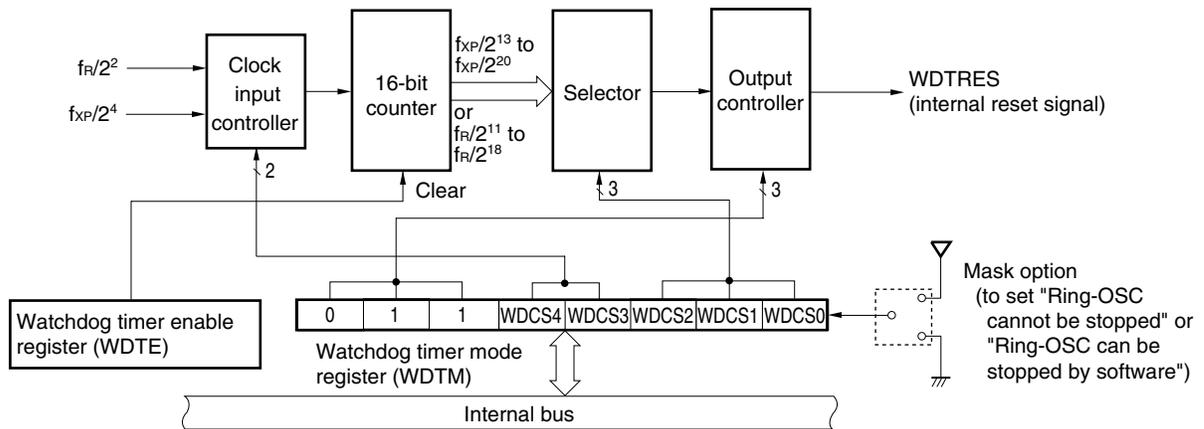
## 9.2 Configuration of Watchdog Timer

The watchdog timer consists of following hardware.

**Table 9-3. Configuration of Watchdog Timer**

Item	Configuration
Control registers	Watchdog timer mode register (WDTM) Watchdog timer enable register (WDTE)

Figure 9-1. Block Diagram of Watchdog Timer



### 9.3 Registers Controlling Watchdog Timer

The watchdog timer is controlled by the following two registers.

- Watchdog timer mode register (WDTM)
- Watchdog timer enable register (WDTE)

#### (1) Watchdog timer mode register (WDTM)

This register sets the overflow time and operation clock of the watchdog timer.

This register can be set by an 8-bit memory manipulation instruction and can be read many times, but can be written only once after reset is released.

$\overline{\text{RESET}}$  input sets this register to 67H.

Figure 9-2. Format of Watchdog Timer Mode Register (WDTM)

Address: FF98H After reset: 67H R/W

Symbol	7	6	5	4	3	2	1	0
WDTM	0	1	1	WDCS4	WDCS3	WDCS2	WDCS1	WDCS0

WDCS4 <sup>Note 1</sup>	WDCS3 <sup>Note 1</sup>	Operation clock selection
0	0	Ring-OSC clock ( $f_R$ )
0	1	X1 input clock ( $f_{XP}$ )
1	×	Watchdog timer operation stopped

WDCS2 <sup>Note 2</sup>	WDCS1 <sup>Note 2</sup>	WDCS0 <sup>Note 2</sup>	Overflow time setting	
			During Ring-OSC clock operation	During X1 input clock operation
0	0	0	$f_R/2^{11}$ (8.53 ms)	$f_{XP}/2^{13}$ (819.2 $\mu$ s)
0	0	1	$f_R/2^{12}$ (17.07 ms)	$f_{XP}/2^{14}$ (1.64 ms)
0	1	0	$f_R/2^{13}$ (34.13 ms)	$f_{XP}/2^{15}$ (3.28 ms)
0	1	1	$f_R/2^{14}$ (68.27 ms)	$f_{XP}/2^{16}$ (6.55 ms)
1	0	0	$f_R/2^{15}$ (136.53 ms)	$f_{XP}/2^{17}$ (13.11 ms)
1	0	1	$f_R/2^{16}$ (273.07 ms)	$f_{XP}/2^{18}$ (26.21 ms)
1	1	0	$f_R/2^{17}$ (546.13 ms)	$f_{XP}/2^{19}$ (52.43 ms)
1	1	1	$f_R/2^{18}$ (1.09 s)	$f_{XP}/2^{20}$ (104.86 ms)

- Notes**
1. If “Ring-OSC cannot be stopped” is specified by a mask option, this cannot be set. The Ring-OSC clock will be selected no matter what value is written.
  2. Reset is released at the maximum cycle (WDCS2, 1, 0 = 1, 1, 1).

- Cautions**
1. If data is written to WDTM, a wait cycle is generated. For details, refer to CHAPTER 25 CAUTIONS FOR WAIT.
  2. Set bits 7, 6, and 5 to 0, 1, and 1, respectively (when “Ring-OSC cannot be stopped” is selected by a mask option, other values are ignored).
  3. After reset is released, WDTM can be written only once by an 8-bit memory manipulation instruction. If writing attempted a second time, an internal reset signal is generated.
  4. WDTM cannot be set by a 1-bit memory manipulation instruction.

- Remarks**
1.  $f_R$ : Ring-OSC clock oscillation frequency
  2.  $f_{XP}$ : X1 input clock oscillation frequency
  3. ×: Don't care
  4. Figures in parentheses apply to operation at  $f_R = 240$  kHz (TYP.),  $f_{XP} = 10$  MHz

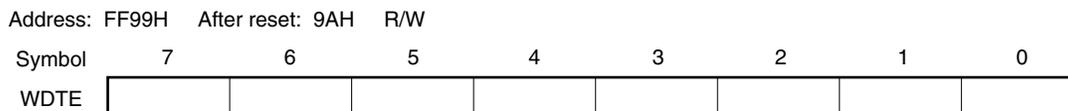
**(2) Watchdog timer enable register (WDTE)**

Writing ACH to WDTE clears the watchdog timer counter and starts counting again.

This register can be set by an 8-bit memory manipulation instruction.

RESET input sets this register to 9AH.

**Figure 9-3. Format of Watchdog Timer Enable Register (WDTE)**



- Cautions**
1. If a value other than ACH is written to WDTE, an internal reset signal is generated.
  2. If a 1-bit memory manipulation instruction is executed for WDTE, an internal reset signal is generated (an error occurs in the assembler).
  3. The value read from WDTE is 9AH (this differs from the written value (ACH)).

## 9.4 Operation of Watchdog Timer

### 9.4.1 Watchdog timer operation when “Ring-OSC cannot be stopped” is selected by mask option

The operation clock of watchdog timer is fixed to the Ring-OSC.

After reset is released, operation is started at the maximum cycle (bits 2, 1, and 0 (WDCS2, WDCS1, WDCS0) of the watchdog timer mode register (WDTM) = 1, 1, 1). The watchdog timer operation cannot be stopped.

The following shows the watchdog timer operation after reset release.

1. The status after reset release is as follows.
  - Operation clock: Ring-OSC clock
  - Cycle:  $f_R/2^{18}$  (1.09 seconds: At operation with  $f_R = 240$  kHz (TYP.))
  - Counting starts
2. The following should be set in the watchdog timer mode register (WDTM) by an 8-bit memory manipulation instruction<sup>Notes 1,2</sup>.
  - Cycle: Set using bits 2 to 0 (WDCS2 to WDCS0)
3. After the above procedures are executed, writing ACH to WDTE clears the count to 0, enabling recounting.

- Notes**
1. The operation clock (Ring-OSC clock) cannot be changed. If any value is written to bits 3 and 4 (WDCS3, WDCS4) of WDTM, it is ignored.
  2. As soon as WDTM is written, the counter of the watchdog timer is cleared.

**Caution** In this mode, operation of the watchdog timer absolutely cannot be stopped even in STOP mode. For 8-bit timer H1 (TMH1), a division of the Ring-OSC can be selected as the count source, so clear the watchdog timer using the interrupt request of TMH1 before the watchdog timer overflows. If this processing is not performed, an internal reset signal is generated when the watchdog timer overflows after STOP instruction execution.

### 9.4.2 Watchdog timer operation when “Ring-OSC can be stopped by software” is selected by mask option

The operation clock of the watchdog timer can be selected as either the Ring-OSC clock or the X1 input clock.

After reset is released, operation is started at the maximum cycle (bits 2, 1, and 0 (WDCS2, WDCS1, WDCS0) of the watchdog timer mode register (WDTM) = 1, 1, 1).

The following shows the watchdog timer operation after reset release.

1. The status after reset release is as follows.
  - Operation clock: Ring-OSC clock oscillation frequency ( $f_R$ )
  - Cycle:  $f_R/2^{18}$  (1.09 seconds: At operation with  $f_R = 240$  kHz (TYP.))
  - Counting starts
2. The following should be set in the watchdog timer mode register (WDTM) by an 8-bit memory manipulation instruction<sup>Notes 1, 2, 3</sup>.
  - Operation clock: Any of the following can be selected using bits 3 and 4 (WDCS3 and WDCS4).  
Ring-OSC clock ( $f_R$ )  
X1 input clock ( $f_{XP}$ )  
Watchdog timer operation stopped
  - Cycle: Set using bits 2 to 0 (WDCS2 to WDCS0)
3. After the above procedures are executed, writing ACH to WDTE clears the count to 0, enabling recounting.

- Notes**
1. As soon as WDTM is written, the counter of the watchdog timer is cleared.
  2. Set bits 7, 6, and 5 to 0, 1, 1, respectively. If other values are set, the watchdog timer cannot be operated (an error occurs in the assembler).
  3. If the watchdog timer is stopped by setting WDCS4 and WDCS3 to 1 and  $\times$ , respectively, an internal reset signal is not generated even if the following processing is performed.
    - WDTM is written a second time.
    - A 1-bit memory manipulation instruction is executed to WDTE.
    - A value other than ACH is written to WDTE.

- ★ **Caution** In this mode, watchdog timer operation is stopped during HALT/STOP instruction execution. After HALT/STOP mode is released, counting is started again using the operation clock of the watchdog timer set before HALT/STOP instruction execution by WDTM. At this time, the counter is not cleared to 0.

For the watchdog timer operation during STOP mode and HALT mode in each status, refer to 9.4.3 Watchdog timer operation in STOP mode and 9.4.4 Watchdog timer operation in HALT mode.

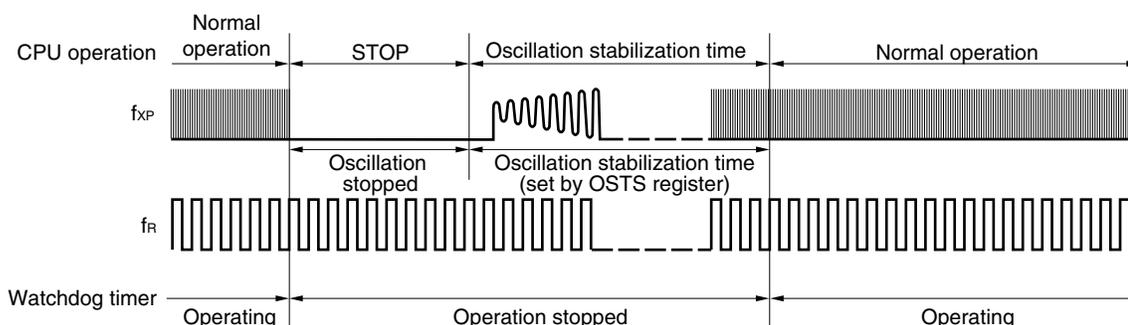
★ **9.4.3 Watchdog timer operation in STOP mode (when “Ring-OSC can be stopped by software” is selected by mask option)**

The watchdog timer stops counting during STOP instruction execution regardless of whether the X1 input clock or Ring-OSC clock is being used.

**(1) When the CPU clock and the watchdog timer operation clock are the X1 input clock ( $f_{XP}$ ) when the STOP instruction is executed**

When STOP instruction is executed, operation of the watchdog timer is stopped. After STOP mode is released, counting stops for the oscillation stabilization time set by the oscillation stabilization time select register (OSTS) and then counting is started again using the operation clock before the operation was stopped. At this time, the counter is not cleared to 0.

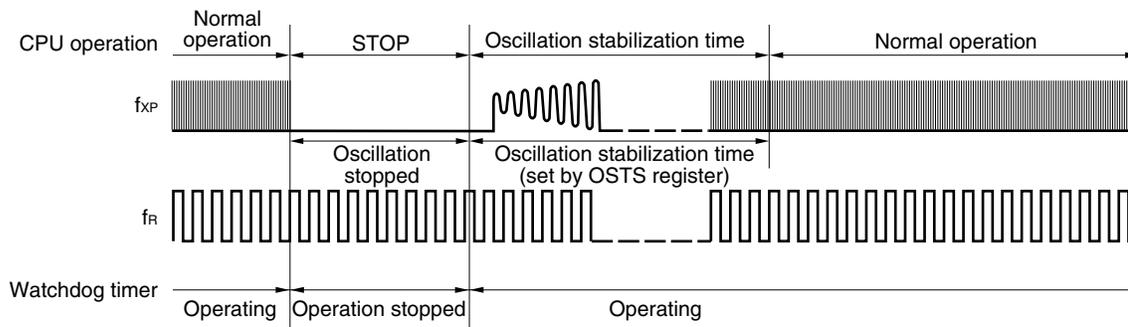
**Figure 9-4. Operation in STOP Mode (CPU Clock and WDT Operation Clock: X1 Input Clock)**



**(2) When the CPU clock is the X1 input clock ( $f_{XP}$ ) and the watchdog timer operation clock is the Ring-OSC clock ( $f_R$ ) when the STOP instruction is executed**

When the STOP instruction is executed, operation of the watchdog timer is stopped. After STOP mode is released, counting is started again using the operation clock before the operation was stopped. At this time, the counter is not cleared to 0.

**Figure 9-5. Operation in STOP Mode (CPU Clock: X1 Input Clock, WDT Operation Clock: Ring-OSC Clock)**



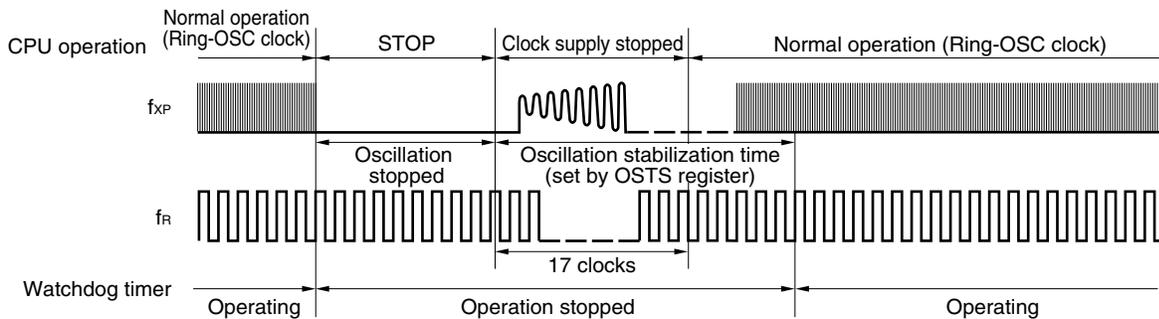
**(3) When the CPU clock is the Ring-OSC clock ( $f_R$ ) and the watchdog timer operation clock is the X1 input clock ( $f_{XP}$ ) when the STOP instruction is executed**

When the STOP instruction is executed, operation of the watchdog timer is stopped. After STOP mode is released, counting is stopped until the timing of <1> or <2>, whichever is earlier, and then counting is started using the operation clock before the operation was stopped. At this time, the counter is not cleared to 0.

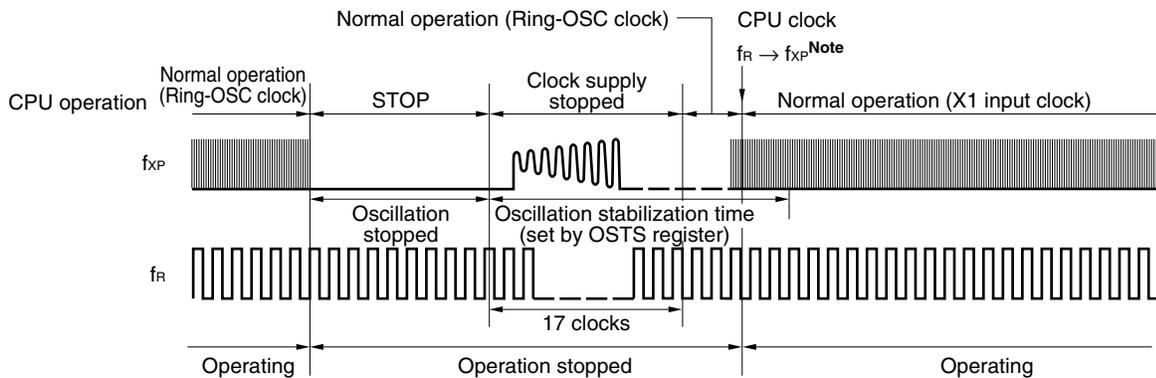
- <1> The oscillation stabilization time set by the oscillation stabilization time select register (OSTS) elapses.
- <2> The CPU clock is switched to the X1 input clock ( $f_{XP}$ ).

**Figure 9-6. Operation in STOP Mode  
(CPU Clock: Ring-OSC Clock, WDT Operation Clock: X1 Input Clock)**

<1> Timing when counting is started after the oscillation stabilization time set by the oscillation stabilization time select register (OSTS) has elapsed



<2> Timing when counting is started after the CPU clock is switched to the X1 input clock ( $f_{XP}$ )

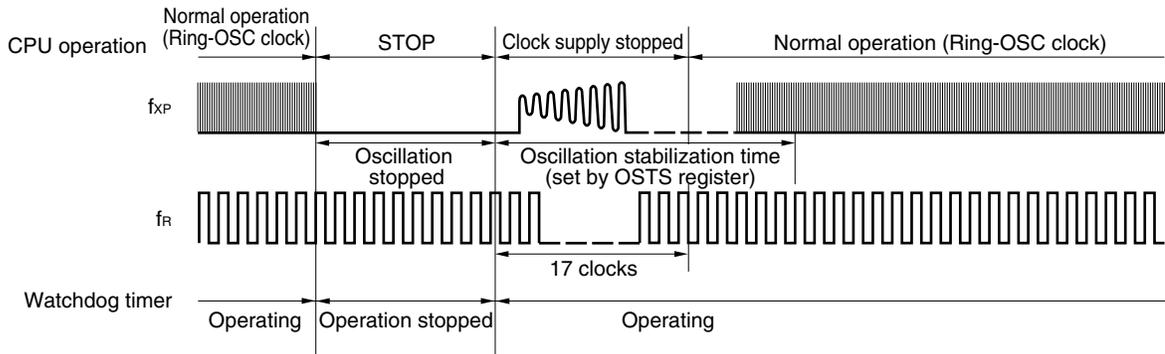


**Note** Confirm the oscillation stabilization time of  $f_{XP}$  using the oscillation stabilization time counter status register (OSTC).

**(4) When CPU clock and watchdog timer operation clock are the Ring-OSC clocks ( $f_R$ ) during STOP instruction execution**

When the STOP instruction is executed, operation of the watchdog timer is stopped. After STOP mode is released, counting is started again using the operation clock before the operation was stopped. At this time, the counter is not cleared to 0.

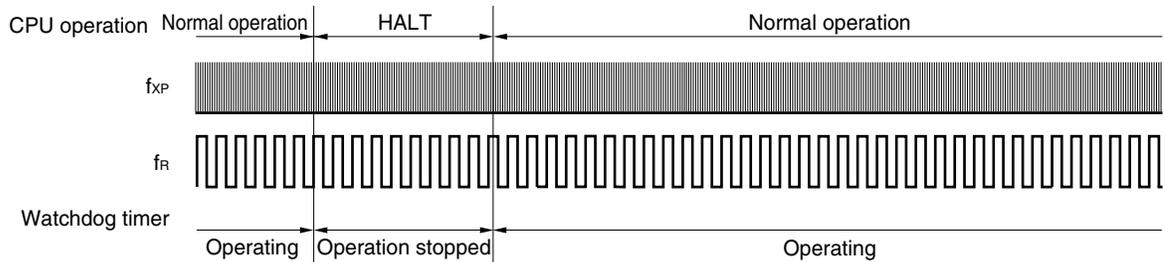
**Figure 9-7. Operation in STOP Mode (CPU Clock and WDT Operation Clock: Ring-OSC Clock)**



★ **9.4.4 Watchdog timer operation in HALT mode (when “Ring-OSC can be stopped by software” is selected by mask option)**

The watchdog timer stops counting during HALT instruction execution regardless of whether the CPU clock is the X1 input clock ( $f_{XP}$ ) or Ring-OSC clock ( $f_R$ ), or whether the operation clock of the watchdog timer is the X1 input clock ( $f_{XP}$ ) or Ring-OSC clock ( $f_R$ ). After HALT mode is released, counting is started again using the operation clock before the operation was stopped. At this time, the counter is not cleared to 0.

**Figure 9-8. Operation in HALT Mode**



## CHAPTER 10 A/D CONVERTER

### 10.1 Function of A/D Converter

The A/D converter converts an analog input signal into a digital value, and consists of up to four channels (ANI0 to ANI3) with a resolution of 10 bits.

The A/D converter has the following two functions.

#### (1) 10-bit resolution A/D conversion

10-bit resolution A/D conversion is carried out repeatedly for one channel selected from analog inputs ANI0 to ANI3. Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated.

#### (2) Power-fail detection function

This function is to detect a voltage drop in a battery. The values of the A/D conversion result (ADCR register value) and power-fail comparison threshold register (PFT) are compared. INTAD is generated only when a comparative condition has been matched.

Figure 10-1. Block Diagram of A/D Converter

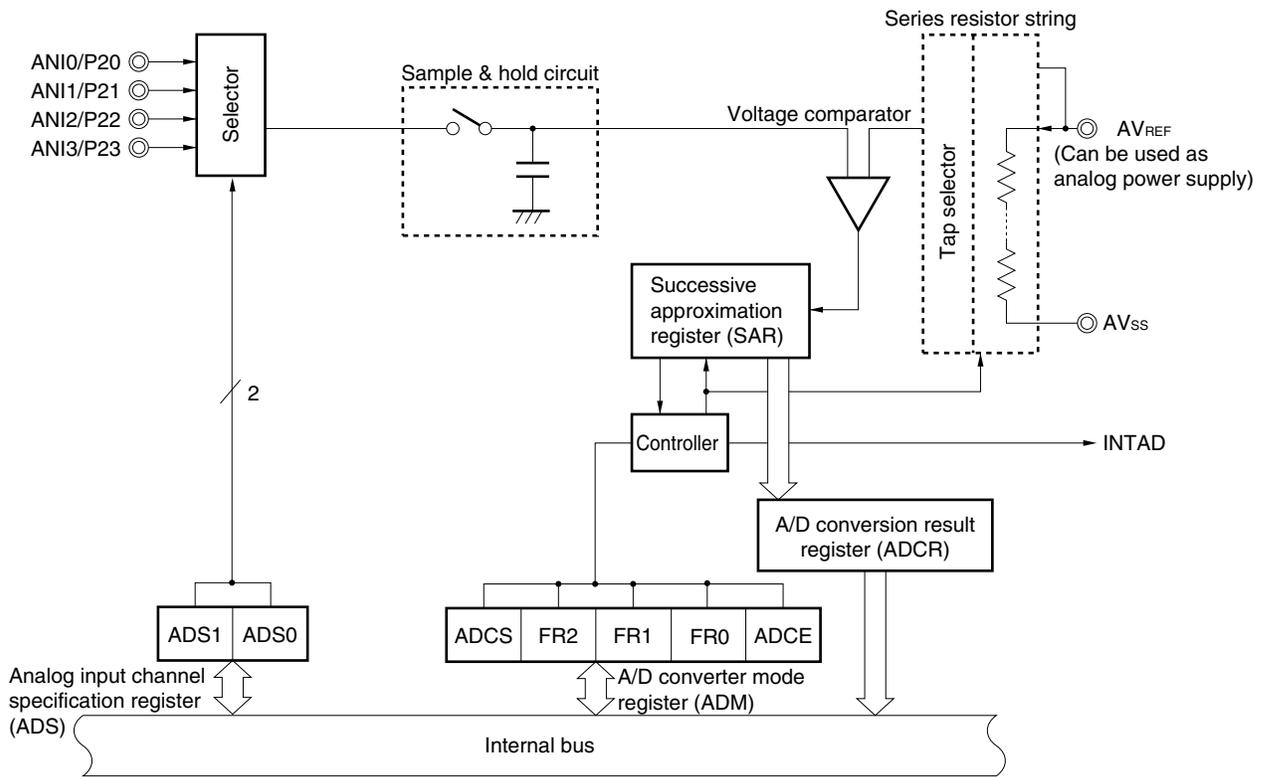
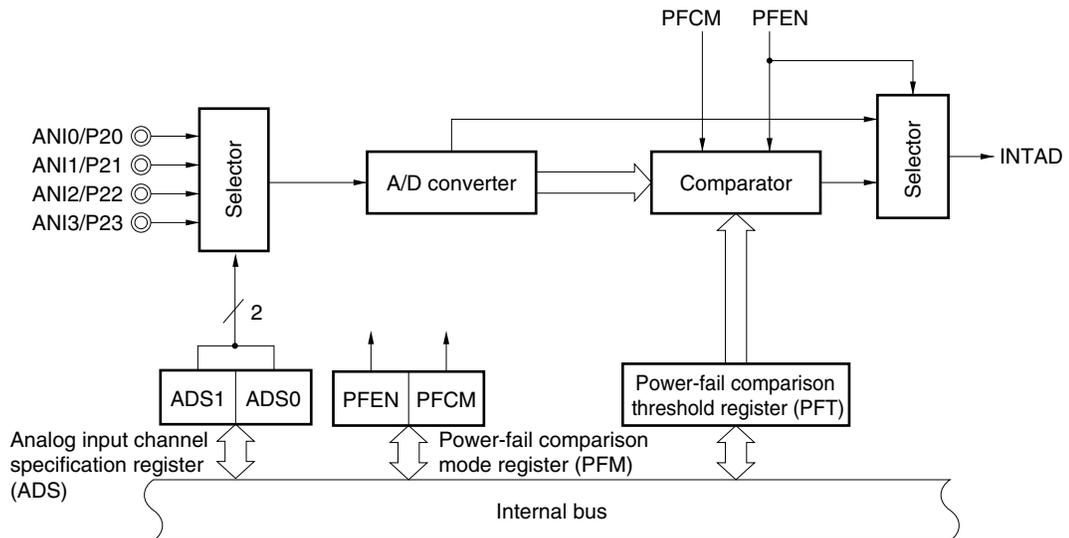


Figure 10-2. Block Diagram of Power-Fail Detection Function



## 10.2 A/D Converter Configuration

The A/D converter consists of the following hardware.

**Table 10-1. Configuration of A/D Converter**

Item	Configuration
Analog input	4 channels (ANI0 to ANI3)
Registers	Successive approximation register (SAR) A/D conversion result register (ADCR)
Control registers	A/D converter mode register (ADM) Analog input channel specification register (ADS) Power-fail comparison mode register (PFM) Power-fail comparison threshold register (PFT)

### (1) Successive approximation register (SAR)

This register compares the analog input voltage value with the voltage tap (compare voltage) value applied from the series resistor string, and holds the result starting from the most significant bit (MSB).

When the result up to the least significant bit (LSB) is held (end of A/D conversion), the SAR contents are transferred to the A/D conversion result register.

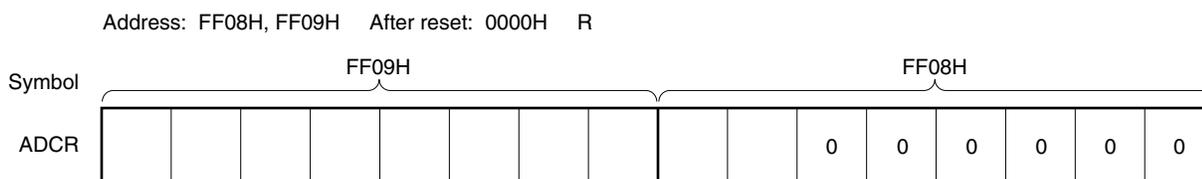
### (2) A/D conversion result register (ADCR)

The ADCR is 16-bit register that stores the A/D conversion result. The lower six bits are fixed to 0. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register, and is stored in ADCR in order starting from the most significant bit (MSB).

ADCR can be read by a 16-bit memory manipulation instruction.

RESET input sets ADCR to 0000H.

**Figure 10-3. Format of A/D Conversion Register (ADCR)**



**Cautions** 1. When writing to the A/D converter mode register (ADM) and analog input channel specification register (ADS), the contents of ADCR may become undefined. Read the conversion result following conversion completion before writing to ADM and ADS. Using timing other than the above may cause an incorrect conversion result to be read.

2. If data is read from ADCR, a wait cycle is generated. For details, refer to CHAPTER 25 CAUTIONS FOR WAIT.

**(3) Sample & hold circuit**

The sample & hold circuit samples each analog input signal sequentially applied from the input circuit, and sends it to the voltage comparator. This circuit holds the sampled analog input voltage value during A/D conversion.

**(4) Voltage comparator**

The voltage comparator compares the analog input with the series resistor string output voltage.

**(5) Series resistor string**

The series resistor string is connected between  $AV_{REF}$  and  $AV_{SS}$ , and generates a voltage to be compared with the analog input.

**(6) ANI0 to ANI3 pins**

These four-channel analog input pins input analog signals to undergo A/D conversion to the A/D converter. ANI0 to ANI3 are alternate-function pins that can also be used for digital input.

- Cautions**
- 1. Observe the rated range of the ANI0 to ANI3 input voltage. If a voltage of  $AV_{REF}$  or higher or a voltage of  $AV_{SS}$  or lower (even if within the range of absolute maximum ratings) is input to an analog input channel, the converted value of that channel becomes undefined. In addition, the converted values of the other channels may also be affected.**
  - 2. The analog input pins (ANI0 to ANI3) are also used as input port pins (P20 to P23). When A/D conversion is performed with any of ANI0 to ANI3 selected, do not execute the input instruction to port 2 while conversion is in progress; otherwise the conversion resolution may be degraded. If a digital pulse is applied to the pins adjacent to the pins currently used for A/D conversion, the expected value of the A/D conversion may not be obtained due to coupling noise. Therefore, do not apply a pulse to the pins adjacent to the pin undergoing A/D conversion.**

**(7)  $AV_{REF}$  pin**

The  $AV_{REF}$  pin inputs the A/D converter reference voltage.

It converts signals input to ANI0 to ANI3 into digital signals based on a voltage between  $AV_{REF}$  and  $AV_{SS}$ .

In a standby mode, the current flowing into series resistor strings can be reduced by changing the input voltage of the  $AV_{REF}$  pin to  $AV_{SS}$  level.

It can also be used as the analog power supply. When the A/D converter is used, be sure to use the  $AV_{REF}$  pin for the power supply.

**Caution** A series resistor string of several tens of  $k\Omega$  is connected between the  $AV_{REF}$  and  $AV_{SS}$  pins. Therefore, if the output impedance of the reference voltage source is high, this will result in series connection to the series resistor string between the  $AV_{REF}$  and  $AV_{SS}$  pins, resulting in a large reference voltage error.

**(8)  $AV_{SS}$  pin**

The  $AV_{SS}$  pin is the GND potential pin for the A/D converter. Always use the  $AV_{SS}$  pin at the same potential as the  $V_{SS0}$  pin, even when the A/D converter is not used.

### 10.3 Registers Controlling A/D Converter

The following four registers are used to control the A/D converter.

- A/D converter mode register (ADM)
- Analog input channel specification register (ADS)
- Power-fail comparison mode register (PFM)
- Power-fail comparison threshold register (PFT)

#### (1) A/D converter mode register (ADM)

This register sets the conversion time for analog input to be A/D converted, and starts/stops conversion.

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

RESET input clears this register to 00H.

**Figure 10-4. Format of A/D Converter Mode Register (ADM)**

Address: FF28H After reset: 00H R/W

Symbol 7 6 5 4 3 2 1 0

ADM	ADCS	0	FR2	FR1	FR0	0	0	ADCE
-----	------	---	-----	-----	-----	---	---	------

ADCS	A/D conversion operation control
0	Stops conversion operation
1	Enables conversion operation

FR2	FR1	FR0	Conversion time selection <sup>Note 1</sup>		
				f <sub>x</sub> = 8.38 MHz	f <sub>x</sub> = 10 MHz
0	0	0	288/f <sub>x</sub>	34.3 μs	28.8 μs
0	0	1	240/f <sub>x</sub>	28.6 μs	24.0 μs
0	1	0	192/f <sub>x</sub>	22.9 μs	19.2 μs
1	0	0	144/f <sub>x</sub>	17.2 μs	14.4 μs
1	0	1	120/f <sub>x</sub>	14.3 μs	12.0 μs <sup>Note 1</sup>
1	1	0	96/f <sub>x</sub>	11.5 μs <sup>Note 1</sup>	9.6 μs <sup>Note 1</sup>
Other			Setting prohibited		

ADCE	Boost reference voltage generator operation control <sup>Note 2</sup>
0	Stops operation of reference voltage generator
1	Enables operation of reference voltage generator

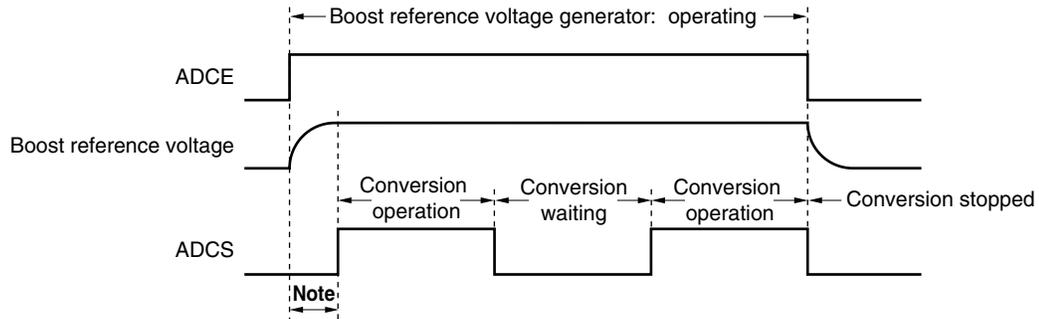
- Notes**
1. Set so that the A/D conversion time is 14 μs or longer.
  2. A booster circuit is incorporated to realize low-voltage operation. The operation of the circuit that generates the reference voltage for boosting is controlled by ADCE, and it takes 14 μs from operation start to operation stabilization. Therefore, when ADCS is set to 1 after 14 μs or more has elapsed from the time ADCE is set to 1, the conversion result at that time has priority over the first conversion result.

Table 10-2. Settings of ADCS and ADCE

ADCS	ADCE	A/D Conversion Operation
0	0	Stop status (DC power consumption path does not exist)
0	1	Conversion waiting mode (only reference voltage generator consumes power)
1	0	Conversion mode (reference voltage generator operation stopped <sup>Note</sup> )
1	1	Conversion mode (reference voltage generator operates)

**Note** Data of first conversion cannot be used.

Figure 10-5. Timing Chart When Boost Reference Voltage Generator Is Used



**Note** 14  $\mu$ s or more is required for reference voltage stabilization.

- Cautions**
1. A/D conversion must be stopped before rewriting bits FR0 to FR2 to values other than the identical data.
  2. For the A/D converter sampling time and A/D conversion start delay time, refer to 10.6 Cautions for A/D Converter (11).
  3. If data is written to ADM, a wait cycle is generated. For details, refer to CHAPTER 25 CAUTIONS FOR WAIT.

**Remark** fx: X1 input clock oscillation frequency

**(2) Analog input channel specification register (ADS)**

This register specifies the analog voltage input port to be A/D converted.

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

RESET input clears this register to 00H.

**Figure 10-6. Format of Analog Input Channel Specification Register (ADS)**

Address: FF29H    After reset: 00H    R/W

Symbol	7	6	5	4	3	2	1	0
ADS	0	0	0	0	0	ADS2	ADS1	ADS0

ADS2	ADS1	ADS0	Analog input channel specification
0	0	0	ANI0
0	0	1	ANI1
0	1	0	ANI2
0	1	1	ANI3

**Cautions 1. Be sure to set bits 2 to 7 of ADS to 0.**

**2. If data is written to ADS, a wait cycle is generated. For details, refer to CHAPTER 25 CAUTIONS FOR WAIT.**

**(3) Power-fail comparison mode register (PFM)**

Power-fail comparison mode register (PFM) is a register that controls the comparison operation.

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

RESET input clears this register to 00H.

**Figure 10-7. Format of Power-Fail Comparison Mode Register (PFM)**

Address: FF2AH    After reset: 00H    R/W

Symbol	<span style="border: 1px solid black; padding: 0 2px;">7</span>	<span style="border: 1px solid black; padding: 0 2px;">6</span>	5	4	3	2	1	0
PFM	PFEN	PFCM	0	0	0	0	0	0

PFEN	Power-fail comparison enable
0	Stops power-fail comparison (used as a normal A/D converter)
1	Enables power-fail comparison (used for power-fail detection)

PFCM		Power-fail comparison mode selection
0	ADCR3 ≥ PFT3	Interrupt request signal (INTAD) generation
	ADCR3 < PFT3	No INTAD generation
1	ADCR3 ≥ PFT3	No INTAD generation
	ADCR3 < PFT3	INTAD generation

**Caution If data is written to PFM, a wait cycle is generated. For details, refer to CHAPTER 25 CAUTIONS FOR WAIT.**

**(4) Power-fail comparison threshold register (PFT)**

The power-fail comparison threshold register (PFT) is a register that sets the threshold value when comparing the values with the A/D conversion result.

8-bit data in PFT is compared to the higher 8 bits (FF09H) of the 10-bit A/D conversion result.

PFT can be set by an 8-bit memory manipulation instruction.

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

**Figure 10-8. Format of Power-Fail Comparison Threshold Register (PFT)**

Address: FF2BH	After reset: 00H	R/W						
Symbol	7	6	5	4	3	2	1	0
PFT	PFT7	PFT6	PFT5	PFT4	PFT3	PFT2	PFT1	PFT0

**Caution** If data is written to PFT, a wait cycle is generated. For details, refer to CHAPTER 25 CAUTIONS FOR WAIT.

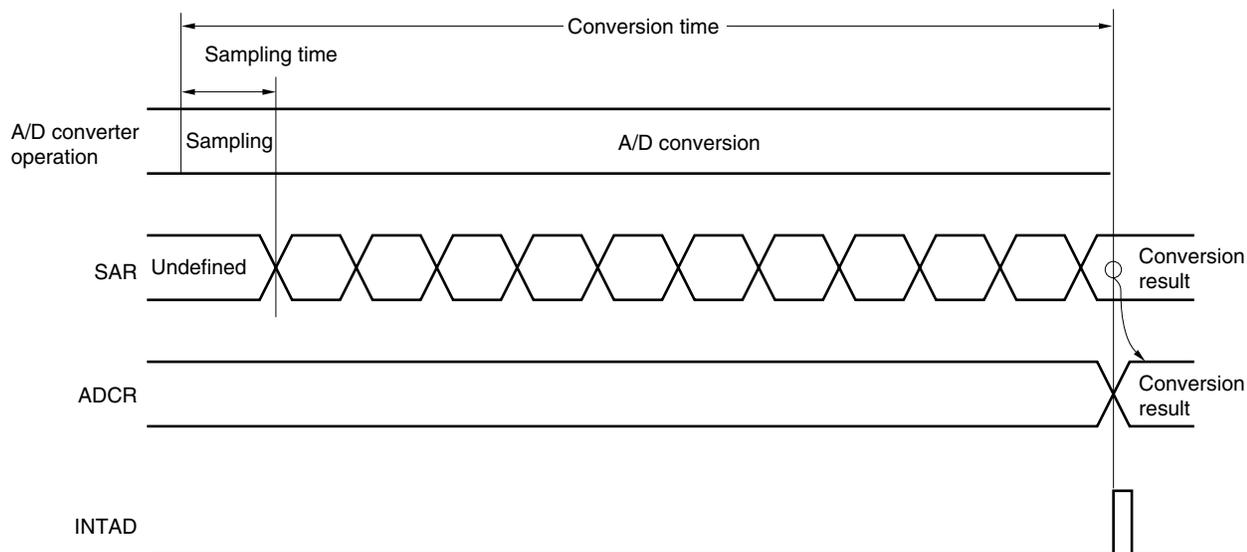
## 10.4 A/D Converter Operations

### 10.4.1 Basic operations of A/D converter

- <1> Select one channel for A/D conversion with analog input channel specification register (ADS).
- <2> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.
- <3> When sampling has been done for a certain time, the sample & hold circuit is placed in the hold state and the input analog voltage is held until the A/D conversion operation is ended.
- <4> Bit 9 of the successive approximation register (SAR) is set. The series resistor string voltage tap is set to (1/2)  $AV_{REF}$  by the tap selector.
- <5> The voltage difference between the series resistor string voltage tap and analog input is compared by the voltage comparator. If the analog input is greater than (1/2)  $AV_{REF}$ , the MSB of SAR remains set to 1. If the analog input is smaller than (1/2)  $AV_{REF}$ , the MSB is reset to 0.
- <6> Next, bit 8 of SAR is automatically set to 1, and the operation proceeds to the next comparison. The series resistor string voltage tap is selected according to the preset value of bit 9, as described below.
  - Bit 9 = 1: (3/4)  $V_{DD}$
  - Bit 9 = 0: (1/4)  $V_{DD}$
 The voltage tap and analog input voltage are compared and bit 8 of SAR is manipulated as follows.
  - Analog input voltage  $\geq$  Voltage tap: Bit 8 = 1
  - Analog input voltage < Voltage tap: Bit 8 = 0
- <7> Comparison is continued in this way up to bit 0 of SAR.
- <8> Upon completion of the comparison of 10 bits, an effective digital result value remains in SAR, and the result value is transferred to the A/D conversion result register (ADCR) and then latched.  
At the same time, the A/D conversion end interrupt request (INTAD) can also be generated.

**Caution** The first A/D conversion value immediately after A/D conversion operations start may not fall within the rating.

Figure 10-9. Basic Operation of A/D Converter



A/D conversion operations are performed continuously until bit 7 (ADCS) of the A/D converter mode register (ADM) is reset (0) by software.

If a write operation is performed to one of the ADM, analog input channel specification register (ADS), power-fail comparison mode register (PFM), or power-fail comparison threshold register (PFT) during an A/D conversion operation, the conversion operation is initialized, and if the ADCS bit is set (1), conversion starts again from the beginning.

$\overline{\text{RESET}}$  input sets the A/D conversion result register (ADCR) to 0000H.

**10.4.2 Input voltage and conversion results**

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI3) and the A/D conversion result (stored in the A/D conversion result register (ADCR)) is shown by the following expression.

$$ADCR = \text{INT} \left( \frac{V_{IN}}{AV_{REF}} \times 1024 + 0.5 \right)$$

or

$$(ADCR - 0.5) \times \frac{AV_{REF}}{1024} - V_{IN} < (ADCR + 0.5) \times \frac{AV_{REF}}{1024}$$

where, INT( ): Function which returns integer part of value in parentheses

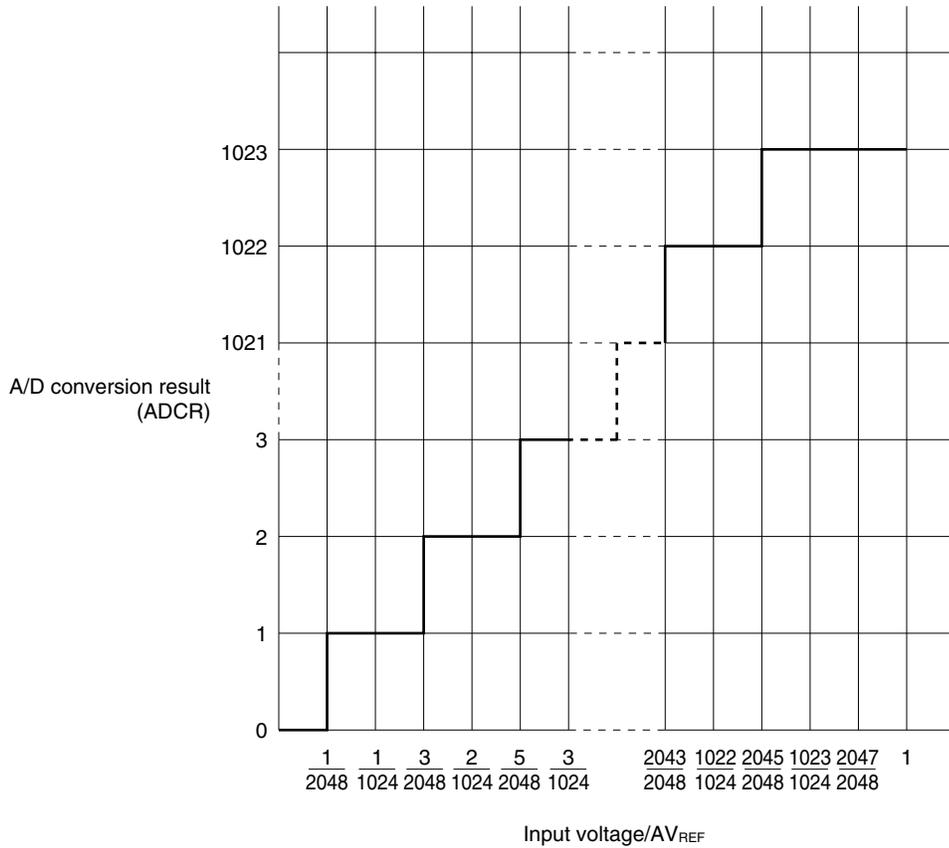
$V_{IN}$ : Analog input voltage

$AV_{REF}$ :  $AV_{REF}$  pin voltage

ADCR: A/D conversion result register (ADCR) value

Figure 10-11 shows the relationship between the analog input voltage and the A/D conversion result.

**Figure 10-10. Relationship Between Analog Input Voltage and A/D Conversion Result**



### 10.4.3 A/D converter operation mode

The operation mode of the A/D converter is the select mode. One channel of analog input is selected from ANI0 to ANI3 by the analog input channel specification register (ADS) and A/D conversion is executed.

In addition, the following two functions can be selected by setting of bit 7 (PFEN) of the power-fail comparison mode register (PFM).

- Normal 10-bit A/D converter (PFEN = 0)
- Power-fail detection function (PFEN = 1)

#### (1) A/D conversion operation (when PFEN = 0)

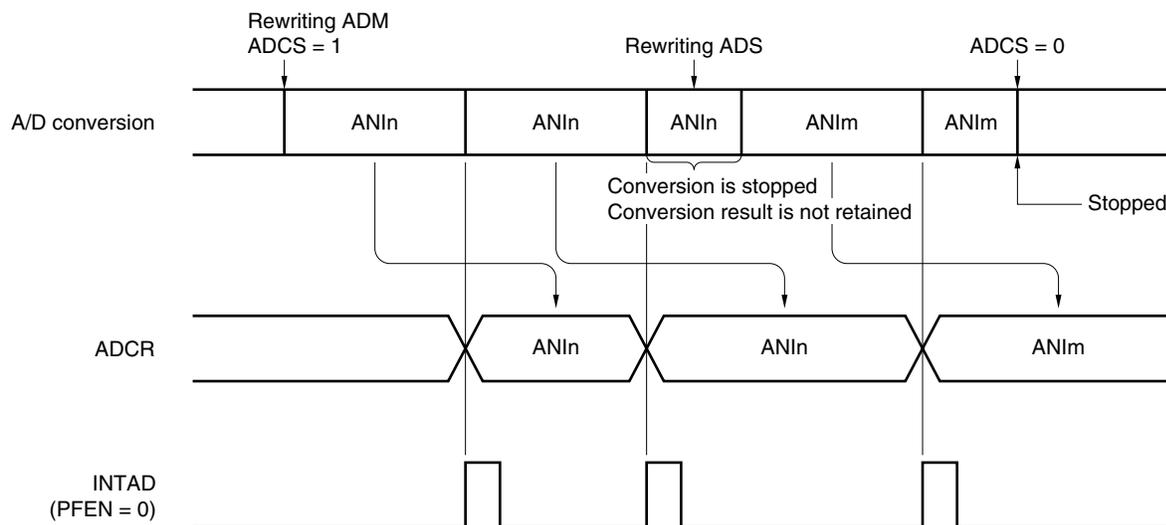
By setting bit 7 (ADCS) of the A/D converter mode register (ADM) to 1 and bit 7 (PFEN) of the power-fail comparison mode register (PFM) to 0, the A/D conversion operation of the voltage, which is applied to the analog input pin specified by the analog input channel specification register (ADS), is started.

When A/D conversion has been completed, the result of the A/D conversion is stored in the A/D conversion result register (ADCR), and an interrupt request signal (INTAD) is generated. Once the A/D conversion has started and when one A/D conversion has been completed, the next A/D conversion operation is immediately started. The A/D conversion operations are repeated until new data is written to ADS.

If ADS is rewritten during A/D conversion, the A/D conversion under execution is suspended, and the A/D conversion of the newly selected analog input channel is started.

If 0 is written to ADCS of ADM during A/D conversion, the conversion operation is immediately stopped.

Figure 10-11. A/D Conversion Operation



- Remarks 1.** n = 0 to 3  
**2.** m = 0 to 3

**(2) Power-fail detection function (when PFEN = 1)**

By setting bit 7 (ADCS) of the A/D converter mode register (ADM) to 1 and bit 7 (PFEN) of the power-fail comparison mode register (PFM) to 1, the A/D conversion operation of the voltage, which applied to the analog input pin specified by the analog input channel specification register (ADS), is started.

When the A/D conversion has been completed, the result of the A/D conversion is stored in the A/D conversion result register (ADCR), the values are compared with power-fail comparison threshold register (PFT), and an interrupt request signal (INTAD) is generated under the condition specified by bit 6 (PFCM) of PFM.

<1> When PFEN = 0

INTAD is generated at the end of each A/D conversion.

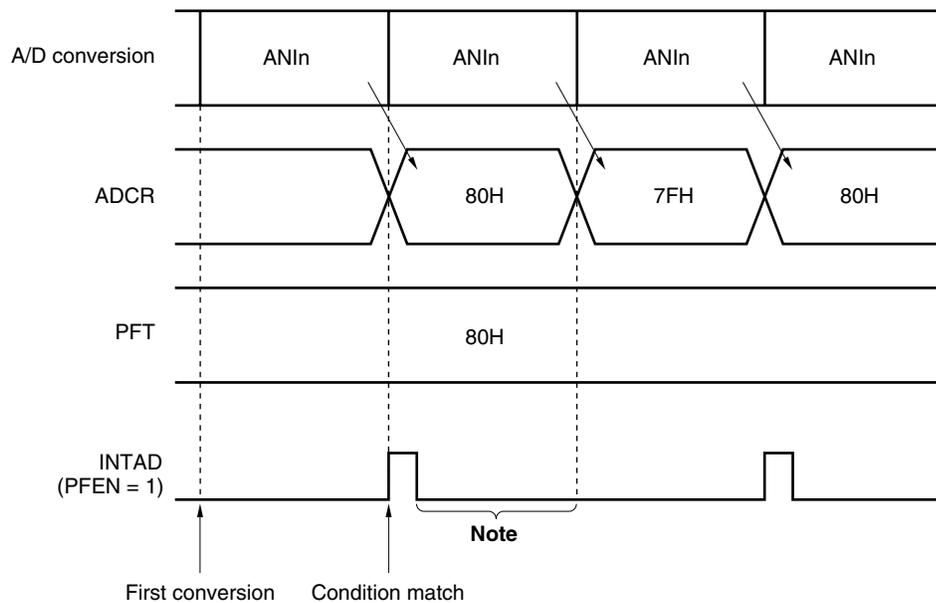
<2> When PFEN = 1 and PFCM = 0

The ADCR and PFT values are compared when A/D conversion ends and INTAD is only generated when  $ADCR \geq PFT$ .

<3> When PFEN = 1 and PFCM = 1

The ADCR and PFT values are compared when A/D conversion ends and INTAD is only generated when  $ADCR < PFT$ .

**Figure 10-12. Power-Fail Detection (When PFEN = 1 and PFCM = 0)**



**Note** If the conversion result is not read before the end of the next conversion after INTAD is output, the result is replaced by the next conversion result.

**Remark** n = 0 to 3

The setting methods are described below.

- When used as A/D conversion operation
  - <1> Set bit 0 (ADCE) of the A/D converter mode register (ADM) to 1.
  - <2> Select the channel and conversion time using bits 1 and 0 (ADS1, ADS0) of the analog input channel specification register (ADS) and bits 5 to 3 (FR2 to FR0) of ADM.
  - <3> Set bit 7 (ADCS) of ADM to 1.
  - <4> An interrupt request signal (INTAD) is generated.
  - <5> Transfer the A/D conversion data to the A/D conversion result register (ADCR).
- <Change the channel>
  - <6> Change the channel using bits 1 and 0 (ADS1, ADS0) of ADS.
  - <7> An interrupt request signal (INTAD) is generated.
  - <8> Transfer the A/D conversion data to the A/D conversion result register (ADCR).
- <Complete A/D conversion>
  - <9> Clear ADCS to 0.
  - <10> Clear ADCE to 0.

- Cautions**
1. Make sure the period of <1> to <3> is 14  $\mu$ s or more.
  2. It is no problem if the order of <1> and <2> is reversed.
  3. <1> can be omitted. However, do not use the first conversion result after <3> in this case.
  4. The period from <4> to <7> differs from the conversion time set using bits 5 to 3 (FR2 to FR0) of ADM. The period from <6> to <7> is the conversion time set using FR2 to FR0.

- When used as power-fail function
  - <1> Set the bit 7 (PFEN) of the power-fail comparison mode register (PFM).
  - <2> Set power-fail comparison condition using bit 6 (PFCM) of PFM.
  - <3> Set bit 0 (ADCE) of the A/D converter mode register (ADM) to 1.
  - <4> Select the channel and conversion time using bits 1 and 0 (ADS1, ADS0) of the analog input channel specification register (ADS) and bits 5 to 3 (FR2 to FR0) of ADM.
  - <5> Set a threshold value to the power-fail comparison threshold register (PFT).
  - <6> Set bit 7 (ADCS) of ADM to 1.
  - <7> Transfer the A/D conversion data to the A/D conversion result register (ADCR).
  - <8> ADCR and PFT are compared and an interrupt request signal (INTAD) is generated if the conditions match.
- <Change the channel>
  - <9> Change the channel using bits 1 and 0 (ADS1, ADS0) of ADS.
  - <10> Transfer the A/D conversion data to the A/D conversion result register (ADCR).
  - <11> ADCR and the power-fail comparison threshold register (PFT) are compared and an interrupt request signal (INTAD) is generated if the conditions match.
- <Complete A/D conversion>
  - <12> Clear ADCS to 0.
  - <13> Clear ADCE to 0.

- Cautions**
1. Make sure the period of <3> to <6> is 14  $\mu$ s or more.
  2. It is no problem if order of <3>, <4>, and <5> is changed.
  3. <3> can be omitted. However, do not use the first conversion result after <6> in this case.
  4. The period from <7> to <11> differs from the conversion time set using bits 5 to 3 (FR2 to FR0) of ADM. The period from <9> to <11> is the conversion time set using FR2 to FR0.

## 10.5 How to Read A/D Converter Characteristics Table

Here, special terms unique to the A/D converter are explained.

### (1) Resolution

This is the minimum analog input voltage that can be identified. That is, the percentage of the analog input voltage per bit of digital output is called 1LSB (Least Significant Bit). The percentage of 1LSB with respect to the full scale is expressed by %FSR (Full Scale Range).

1LSB is as follows when the resolution is 10 bits.

$$\begin{aligned} 1\text{LSB} &= 1/2^{10} = 1/1024 \\ &= 0.098\% \text{FSR} \end{aligned}$$

Accuracy has no relation to resolution, but is determined by overall error.

### (2) Overall error

This shows the maximum error value between the actual measured value and the theoretical value.

Zero-scale error, full-scale error, integral linearity error, and differential linearity errors that are combinations of these express the overall error.

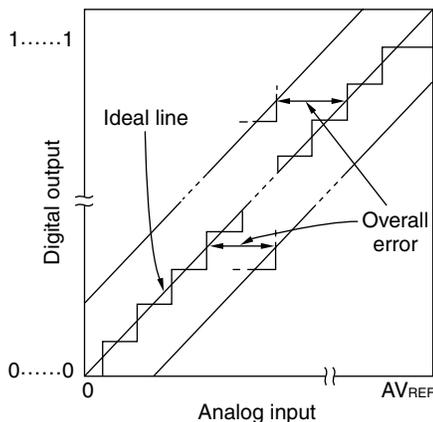
Note that the quantization error is not included in the overall error in the characteristics table.

### (3) Quantization error

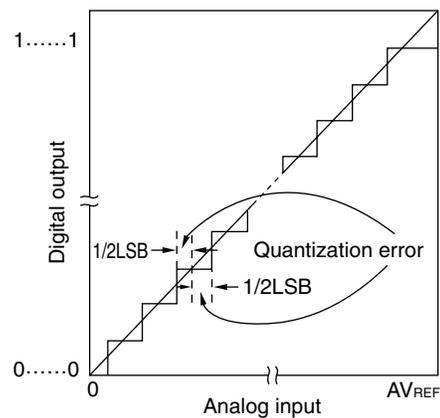
When analog values are converted to digital values, a  $\pm 1/2\text{LSB}$  error naturally occurs. In an A/D converter, an analog input voltage in a range of  $\pm 1/2\text{LSB}$  is converted to the same digital code, so a quantization error cannot be avoided.

Note that the quantization error is not included in the overall error, zero-scale error, full-scale error, integral linearity error, and differential linearity error in the characteristics table.

**Figure 10-13. Overall Error**



**Figure 10-14. Quantization Error**



**(4) Zero-scale error**

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (1/2LSB) when the digital output changes from 0.....000 to 0.....001.

If the actual measurement value is greater than the theoretical value, it shows the difference between the actual measurement value of the analog input voltage and the theoretical value (3/2LSB) when the digital output changes from 0.....000 to 0.....010.

**(5) Full-scale error**

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (Full-scale – 3/2LSB) when the digital output changes from 1.....110 to 1.....111.

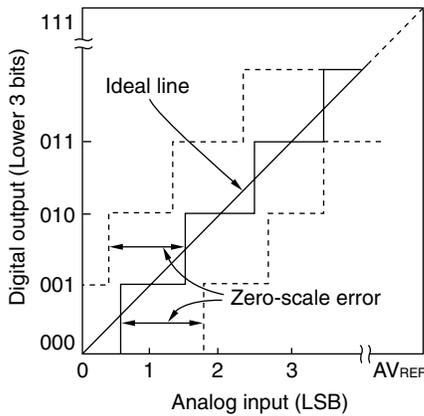
**(6) Integral linearity error**

This shows the degree to which the conversion characteristics deviate from the ideal linear relationship. It expresses the maximum value of the difference between the actual measurement value and the ideal straight line when the zero-scale error and full-scale error are 0.

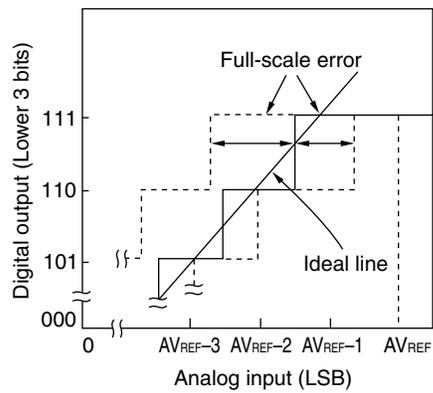
**(7) Differential linearity error**

While the ideal width of code output is 1LSB, this indicates the difference between the actual measurement value and the ideal value.

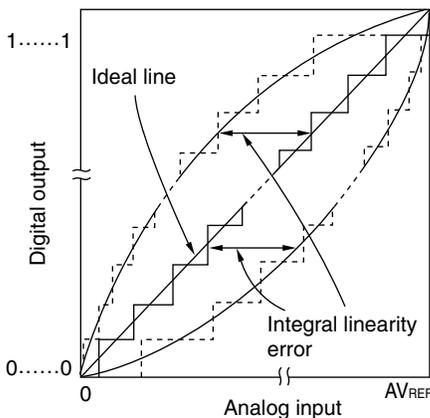
**Figure 10-15. Zero-Scale Error**



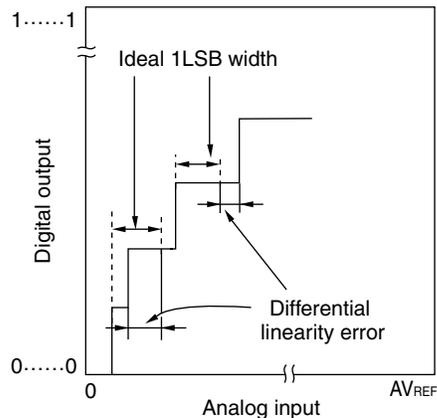
**Figure 10-16. Full-Scale Error**



**Figure 10-17. Integral Linearity Error**



**Figure 10-18. Differential Linearity Error**



**(8) Conversion time**

This expresses the time from when the analog input voltage was applied to the time when the digital output was obtained.

The sampling time is included in the conversion time in the characteristics table.

**(9) Sampling time**

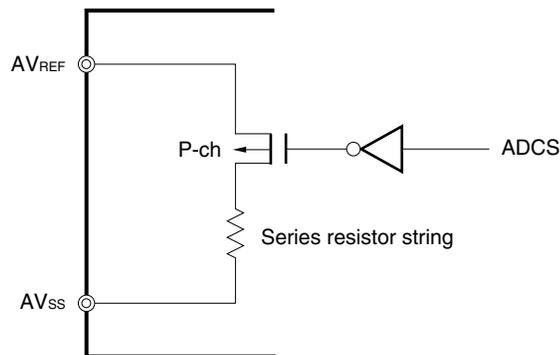
This is the time the analog switch is turned on for the analog voltage to be sampled by the sample & hold circuit.

**10.6 Cautions for A/D Converter****(1) Current consumption in standby mode**

The A/D converter stops operating in the standby mode. At this time, the current consumption can be reduced by stopping the conversion operation (by setting bit 7 (ADCS) of the A/D converter mode register (ADM) to 0).

Figure 10-20 shows how to reduce the current consumption in the standby mode.

**Figure 10-19. Example of Method of Reducing Current Consumption in Standby Mode**

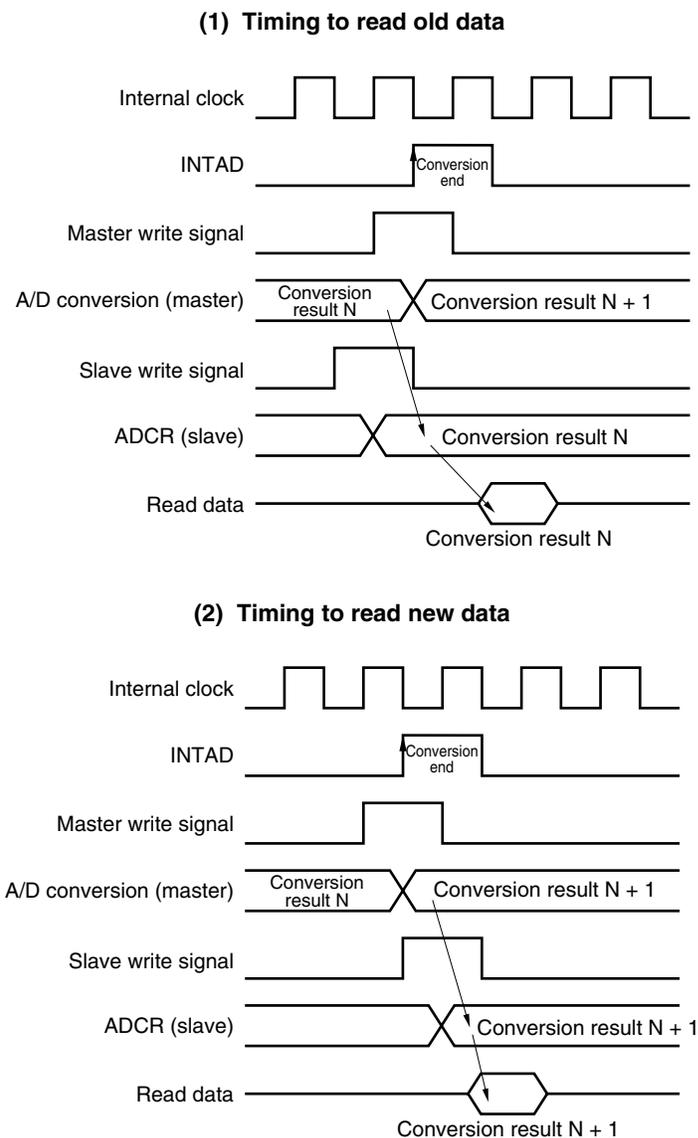
**(2) Input range of ANI0 to ANI3**

Observe the rated range of the ANI0 to ANI3 input voltage. If a voltage of  $AV_{REF}$  or higher and  $AV_{SS}$  or lower (even in the range of absolute maximum ratings) is input to an analog input channel, the converted value of that channel becomes undefined. In addition, the converted values of the other channels may also be affected.

**(3) Conflicting operations**

- <1> Conflict between A/D conversion result register (ADCR) write and ADCR read by instruction upon the end of conversion  
 ADCR read has priority. After the read operation, the new conversion result is written to ADCR.  
 Old data can be read from ADCR at the timing of (1) and new data can be read from ADCR at the timing of (2) as shown in Figure 10-21. A master-slave configuration is employed for transferring the A/D conversion result to ADCR.

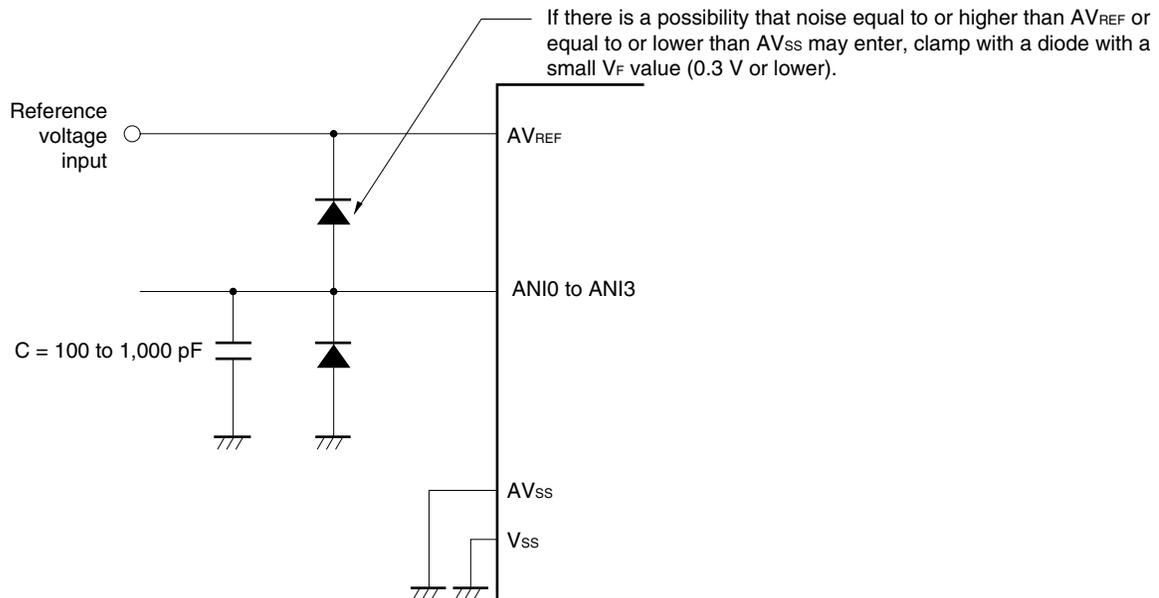
**Figure 10-20. Storing Conversion Result in ADCR and Timing of Data Read from ADCR**



- <2> Contention between ADCR write and A/D converter mode register (ADM) write or analog input channel specification register (ADS) write  
 ADM or ADS write has priority. ADCR write is not performed, nor is the conversion end interrupt signal (INTAD) generated.

**(4) Noise countermeasures**

To maintain the 10-bit resolution, attention must be paid to noise input to the  $AV_{REF}$  pin and pins ANI0 to ANI3. Because the effect increases in proportion to the output impedance of the analog input source, it is recommended that a capacitor be connected externally, as shown in Figure 10-21, to reduce noise.

**Figure 10-21. Analog Input Pin Connection****(5) ANI0/P20 to ANI3/P23**

The analog input pins (ANI0 to ANI3) are also used as input port pins (P20 to P23).

When A/D conversion is performed with any of ANI0 to ANI3 selected, do not execute the input instruction to port 2 while conversion is in progress; otherwise the conversion resolution may be degraded.

If a digital pulse is applied to the pins adjacent to the pins currently used for A/D conversion, the expected value of the A/D conversion may not be obtained due to coupling noise. Therefore, do not apply a pulse to the pins adjacent to the pin undergoing A/D conversion.

**(6) Input impedance of ANI0 to ANI3 pins**

In this A/D converter, the internal sampling capacitor is charged and sampling is performed for approx. one tenth of the conversion time.

Since only the leakage current flows other than during sampling and the current for charging the capacitor also flows during sampling, the input impedance fluctuates and has no meaning.

To perform sufficient sampling, however, it is recommended to make the output impedance of the analog input source 10 k $\Omega$  or lower, or attach a capacitor of around 100 pF to the ANI0 to ANI3 pins (see **Figure 10-21**).

**(7)  $AV_{REF}$  pin input impedance**

A series resistor string of several tens of 10 k $\Omega$  is connected between the  $AV_{REF}$  and  $AV_{SS}$  pins.

Therefore, if the output impedance of the reference voltage source is high, this will result in a series connection to the series resistor string between the  $AV_{REF}$  and  $AV_{SS}$  pins, resulting in a large reference voltage error.

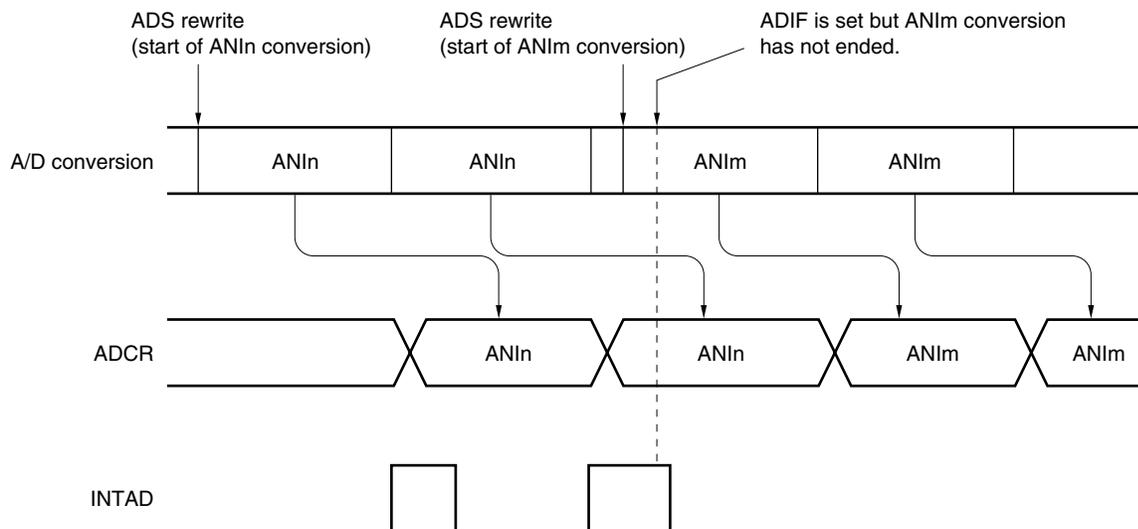
**(8) Interrupt request flag (ADIF)**

The interrupt request flag (ADIF) is not cleared even if the analog input channel specification register (ADS) is changed.

Therefore, if an analog input pin is changed during A/D conversion, the A/D conversion result and ADIF for the pre-change analog input may be set just before the ADS rewrite. Caution is therefore required since, at this time, when ADIF is read immediately after the ADS rewrite, ADIF is set despite the fact A/D conversion for the post-change analog input has not ended.

When A/D conversion is stopped and then resumed, clear ADIF before the A/D conversion operation is resumed.

**Figure 10-22. Timing of A/D Conversion End Interrupt Request Generation**



**Remarks 1.**  $n = 0$  to 3

**2.**  $m = 0$  to 3

**(9) Conversion results just after A/D conversion start**

The first A/D conversion value immediately after A/D conversion starts may not fall within the rating. Poll the A/D conversion end interrupt request (INTAD) and take measures such as removing the first conversion result.

**(10) A/D conversion result register (ADCR) read operation**

When a write operation is performed to the A/D converter mode register (ADM) and analog input channel specification register (ADS), the contents of ADCR may become undefined. Read the conversion result following conversion completion before writing to ADM and ADS. Using timing other than the above may cause an incorrect conversion result to be read.

★ (11) A/D converter sampling time and A/D conversion start delay time

The A/D converter sampling time differs depending on the set value of the A/D converter mode register (ADM). The delay time exists until actual sampling is started after A/D converter operation is enabled. When using a set in which the A/D conversion time must be strictly observed, care is required for the contents shown in Figure 10-23 and Table 10-3.

Figure 10-23. Timing of A/D Converter Sampling and A/D Conversion Start Delay

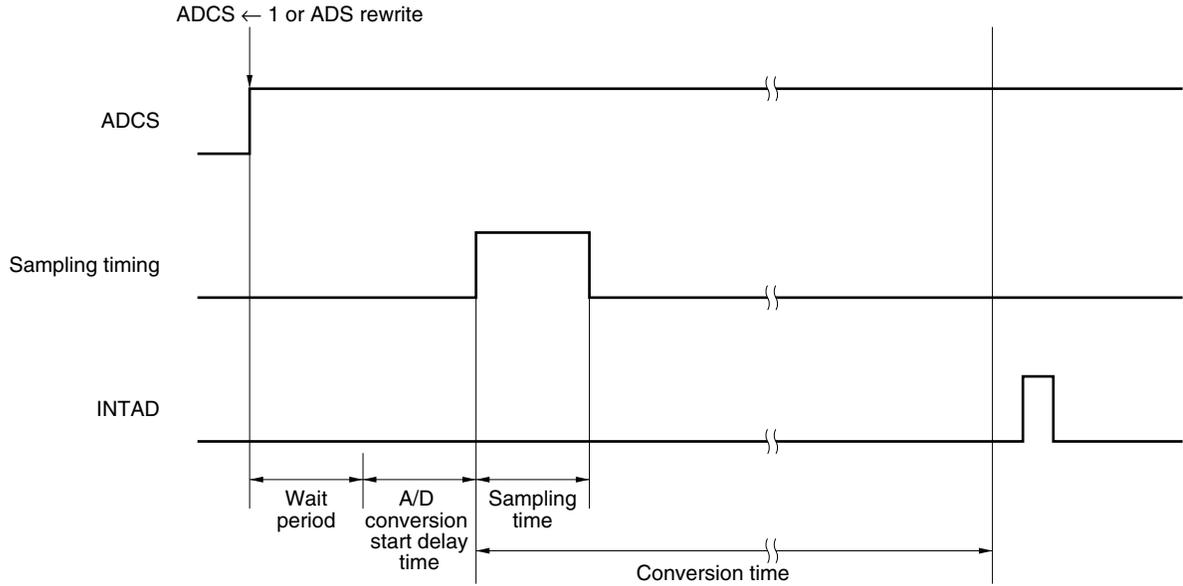


Table 10-3. A/D Converter Sampling Time and A/D Conversion Start Delay Time (ADM Set Value)

FR2	FR1	FR0	Conversion Time	Sampling Time	A/D Conversion Start Delay Time <sup>Note</sup>	
					MIN.	MAX.
0	0	0	288/f <sub>x</sub>	40/f <sub>x</sub>	32/f <sub>x</sub>	36/f <sub>x</sub>
0	0	1	240/f <sub>x</sub>	32/f <sub>x</sub>	28/f <sub>x</sub>	32/f <sub>x</sub>
0	1	0	192/f <sub>x</sub>	24/f <sub>x</sub>	24/f <sub>x</sub>	28/f <sub>x</sub>
1	0	0	144/f <sub>x</sub>	20/f <sub>x</sub>	16/f <sub>x</sub>	18/f <sub>x</sub>
1	0	1	120/f <sub>x</sub>	16/f <sub>x</sub>	14/f <sub>x</sub>	16/f <sub>x</sub>
1	1	0	96/f <sub>x</sub>	12/f <sub>x</sub>	12/f <sub>x</sub>	14/f <sub>x</sub>
Other than above			Setting prohibited	–	–	–

**Note** The A/D conversion start delay time is the time after wait period. For the wait function, refer to **CHAPTER 25 CAUTIONS FOR WAIT**.

**Remark** f<sub>x</sub>: X1 clock oscillation frequency

## CHAPTER 11 SERIAL INTERFACE UART0 ( $\mu$ PD780102, 780103, AND 78F0103 ONLY)

### 11.1 Functions of Serial Interface UART0

Serial interface UART0 has the following two modes.

#### (1) Operation stop mode

This mode is used when serial transfer is not executed and can enable a reduction in the power consumption. For details, refer to **11.4.1 Operation stop mode**.

#### (2) Asynchronous serial interface (UART) mode

The functions of this mode are outlined below.

- Two-pin configuration TxD0: Transmit data output pin  
RxB0: Receive data input pin
- Length of transfer data can be selected from 7 or 8 bits.
- Dedicated internal 5-bit baud rate generator allowing any baud rate to be set
- Transmission and reception can be performed independently.
- Four operating clock inputs selectable
- Fixed to LSB-first transfer

- Cautions**
1. The default value of the TxD0 pin is high level. Exercise care when using the TxD0 pin as a port pin.
  2. If clock supply to serial interface UART0 is not stopped (e.g., in the HALT mode), normal operation continues. If clock supply to serial interface UART0 is stopped (e.g., in the STOP mode), each register stops operating, and holds the value immediately before clock supply was stopped. The TxD0 pin also holds the value immediately before clock supply was stopped and outputs it. However, the operation is not guaranteed after clock supply is resumed. Therefore, reset the circuit so that POWER0 = 0, RXE0 = 0, and TXE0 = 0.
  3. Set POWER0 = 1 and then set TXE0 = 1 (transmission) or RXE0 = 1 (reception) to start communication.
  4. TXE0 and RXE0 are synchronized with the base clock (f<sub>CLK</sub>) set by BRGC0. Therefore, the transmission unit may not be initialized if TXE0 = 1 is not set again 2 clocks after TXE0 = 0 is set. Similarly, the reception unit may not be initialized if RXE0 = 1 is not set again 2 clocks after RXE0 = 0 is set.

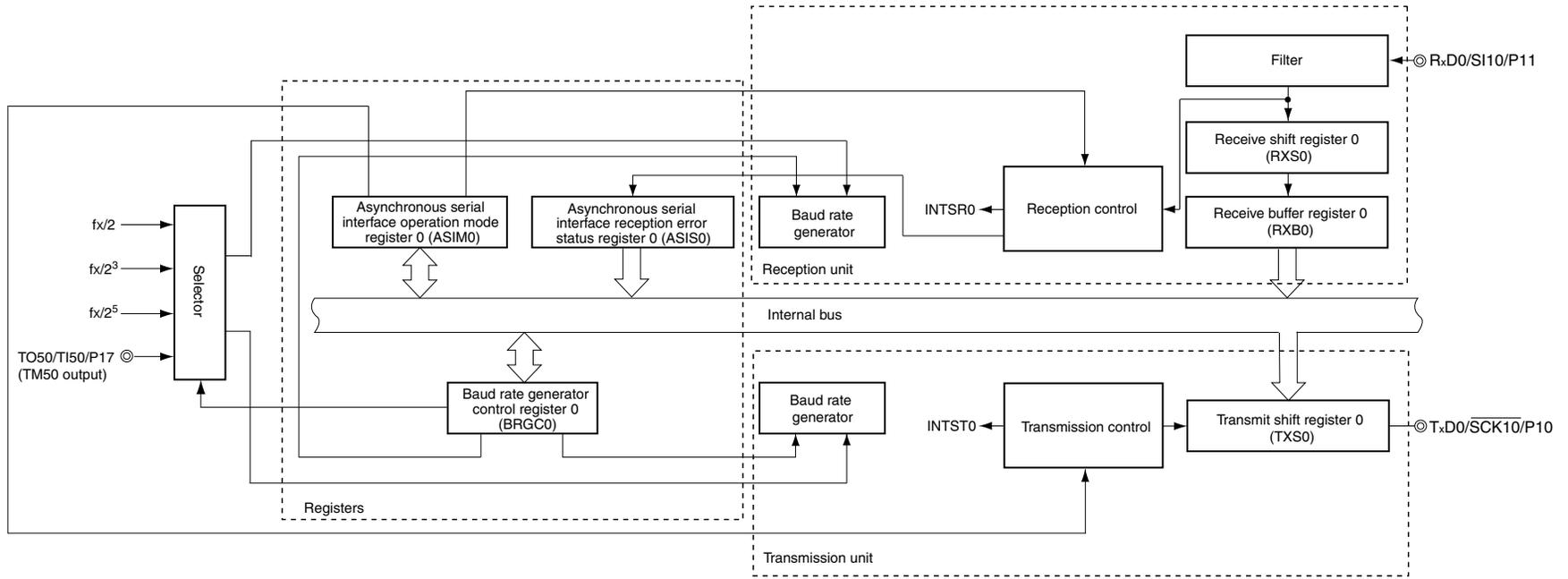
## 11.2 Configuration of Serial Interface UART0

Serial interface UART0 consists of the following hardware.

**Table 11-1. Configuration of Serial Interface UART0**

Item	Configuration
Registers	Receive buffer register 0 (RXB0) Receive shift register 0 (RXS0) Transmit shift register 0 (TXS0)
Control registers	Asynchronous serial interface operation mode register 0 (ASIM0) Asynchronous serial interface reception error status register 0 (ASIS0) Baud rate generator control register 0 (BRGC0)

Figure 11-1. Block Diagram of Serial Interface UART0



**(1) Receive buffer register 0 (RXB0)**

This 8-bit register stores parallel data converted by receive shift register 0 (RXS0).

Each time 1 byte of data has been received, new receive data is transferred to this register from receive shift register 0 (RXS0).

If the data length is set to 7 bits the receive data is transferred to bits 0 to 6 of RXB0 and the MSB of RXB0 is always 0.

If an overrun error (OVE0) occurs, the receive data is not transferred to RXB0.

$\overline{\text{RESET}}$  input or  $\text{POWER0} = 0$  sets this register to FFH.

RXB0 can be read by an 8-bit memory manipulation instruction. No data can be written to this register.

**(2) Receive shift register 0 (RXS0)**

This register converts the serial data input to the RxD0 pin into parallel data.

RXS0 cannot be directly manipulated by a program.

**(3) Transmit shift register 0 (TXS0)**

This register is used to set transmit data. Transmission is started when data is written to TXS0, and serial data is transmitted from the TxD0 pins.

$\overline{\text{RESET}}$  input,  $\text{POWER0} = 0$ , or  $\text{TXE0} = 0$  sets this register to FFH.

TXS0 can be written by an 8-bit memory manipulation instruction. This register cannot be read.

**Caution** Do not write the next transmit data to TXS0 before the transmission completion interrupt signal (INTST0) is generated.

### 11.3 Registers Controlling Serial Interface UART0

Serial interface UART0 is controlled by the following three registers.

- Asynchronous serial interface operation mode register 0 (ASIM0)
- Asynchronous serial interface reception error status register 0 (ASIS0)
- Baud rate generator control register 0 (BRGC0)

#### (1) Asynchronous serial interface operation mode register 0 (ASIM0)

This 8-bit register controls the serial transfer operations of serial interface UART0.

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

$\overline{\text{RESET}}$  input sets this register to 01H.

**Figure 11-2. Format of Asynchronous Serial Interface Operation Mode Register 0 (ASIM0) (1/2)**

Address: FF70H After reset: 01H R/W

Symbol	<span style="border: 1px solid black; padding: 0 2px;">7</span>	<span style="border: 1px solid black; padding: 0 2px;">6</span>	<span style="border: 1px solid black; padding: 0 2px;">5</span>	4	3	2	1	0
ASIM0	POWER0	TXE0	RXE0	PS01	PS00	CL0	SL0	1

POWER0	Enables/disables operation of internal operation clock
0 <sup>Note</sup>	Disables operation of the internal operation clock (fixes the clock to low level) and asynchronously resets the internal circuit.
1	Enables operation of the internal operation clock.

TXE0	Enables/disables transmission
0	Disables transmission (synchronously resets the transmission circuit).
1	Enables transmission.

RXE0	Enables/disables reception
0	Disables reception (synchronously resets the reception circuit).
1	Enables reception.

**Note** The input from the RxD0 pin is fixed to high level when POWER0 = 0.

- Cautions**
1. At startup, set POWER0 to 1 and then set TXE0 to 1. Clear TXE0 to 0 first, and then clear POWER0 to 0.
  2. At startup, set POWER0 to 1 and then set RXE0 to 1. Clear RXE0 to 0 first, and then clear POWER0 to 0.
  3. TXE0 and RXE0 are synchronized with the base clock (f<sub>CLK</sub>) set by BRGC0. Therefore, the transmission unit may not be initialized if TXE0 = 1 is not set again 2 clocks after TXE0 = 0 is set. Similarly, the reception unit may not be initialized if RXE0 = 1 is not set again 2 clocks after RXE0 = 0 is set.
  4. Be sure to set bit 0 to 1.

Figure 11-2. Format of Asynchronous Serial Interface Operation Mode Register 0 (ASIM0) (2/2)

PS01	PS00	Transmission operation	Reception operation
0	0	Does not output parity bit.	Reception without parity
0	1	Outputs 0 parity.	Reception as 0 parity <sup>Note</sup>
1	0	Outputs odd parity.	Judges as odd parity.
1	1	Outputs even parity.	Judges as even parity.

CL0	Specifies character length of transmit/receive data
0	Character length of data = 7 bits
1	Character length of data = 8 bits

SL0	Specifies number of stop bits of transmit data
0	Number of stop bits = 1
1	Number of stop bits = 2

**Note** If “reception as 0 parity” is selected, the parity is not judged. Therefore, bit 2 (PE0) of asynchronous serial interface status register 0 (ASIS0) is not set and the error interrupt does not occur.

- Cautions**
1. Clear the TXE0 and RXE0 bits to 0 before rewriting the PS01, PS00, and CL0 bits.
  2. Make sure that TXE0 = 0 when rewriting the SL0 bit. Reception is always performed with “the number of stop bits = 1”, and therefore, is not affected by the set value of the SL0 bit.

**(2) Asynchronous serial interface reception error status register 0 (ASIS0)**

This register indicates an error status on completion of reception by serial interface UART0. It includes three error flag bits (PE0, FE0, OVE0).

This register can be set by an 8-bit memory manipulation instruction and is read-only.

$\overline{\text{RESET}}$  input clears this register to 00H if bit 7 (POWER0) and bit 5 (RXE0) of ASIM0 = 0. 00H is read when this register is read.

**Figure 11-3. Format of Asynchronous Serial Interface Reception Error Status Register 0 (ASIS0)**

Address: FF73H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS0	0	0	0	0	0	PE0	FE0	OVE0

PE0	Status flag indicating parity error
0	If POWER0 = 0 and RXE0 = 0, or if ASIS0 register is read.
1	If the parity of transmit data does not match the parity bit on completion of reception.

FE0	Status flag indicating framing error
0	If POWER0 = 0 and RXE0 = 0, or if ASIS0 register is read.
1	If the stop bit is not detected on completion of reception.

OVE0	Status flag indicating overrun error
0	If POWER0 = 0 and RXE0 = 0, or if ASIS0 register is read.
1	If receive data is set to the RXB register and the next reception operation is completed before the data is read.

- Cautions**
1. The operation of the PE0 bit differs depending on the set values of the PS01 and PS00 bits of asynchronous serial interface mode register 0 (ASIM0).
  2. Only the first bit of the receive data is checked as the stop bit, regardless of the number of stop bits.
  3. If an overrun error occurs, the next receive data is not written to receive buffer register 0 (RXB0) but discarded.
  4. If data is read from ASIS0, a wait cycle is generated. For details, refer to CHAPTER 25 CAUTIONS FOR WAIT.

**(3) Baud rate generator control register 0 (BRGC0)**

This register selects the base clock of serial interface UART0 and controls the baud rate.

BRGC0 can be set by an 8-bit memory manipulation instruction.

RESET input sets this register to 1FH.

**Figure 11-4. Format of Baud Rate Generator Control Register 0 (BRGC0)**

Address: FF71H After reset: 1FH R/W

Symbol	7	6	5	4	3	2	1	0
BRGC0	TPS01	TPS00	0	MDL04	MDL03	MDL02	MDL01	MDL00

TPS01	TPS00	Base clock (f <sub>CLK</sub> ) selection
0	0	TM50 output (TO50)
0	1	f <sub>x</sub> /2 (5 MHz)
1	0	f <sub>x</sub> /2 <sup>5</sup> (1.25 MHz)
1	1	f <sub>x</sub> /2 <sup>5</sup> (312.5 kHz)

MDL04	MDL03	MDL02	MDL01	MDL00	k	Selection of 5-bit counter output clock
0	0	×	×	×	×	Setting prohibited
0	1	0	0	0	8	f <sub>CLK</sub> /8
0	1	0	0	1	9	f <sub>CLK</sub> /9
0	1	0	1	0	10	f <sub>CLK</sub> /10
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
1	1	0	1	0	26	f <sub>CLK</sub> /26
1	1	0	1	1	27	f <sub>CLK</sub> /27
1	1	1	0	0	28	f <sub>CLK</sub> /28
1	1	1	1	0	30	f <sub>CLK</sub> /30
1	1	1	1	1	31	f <sub>CLK</sub> /31

- Cautions**
1. Make sure that bit 6 (TXE0) and bit 5 (RXE0) of the ASIM0 register = 0 when rewriting the MDL04 to MDL00 bits.
  2. The baud rate is the output clock of the 5-bit counter divided by 2.

- Remarks**
1. f<sub>CLK</sub>: Frequency of base clock (Clock) selected by the TPS01 and TPS00 bits
  2. f<sub>x</sub>: X1 input clock oscillation frequency
  3. k: Value set by the MDL04 to MDL00 bits (k = 8, 9, 10, ..., 31)
  4. ×: Don't care
  5. Figures in parentheses apply to operation at f<sub>x</sub> = 10 MHz

## 11.4 Operation of Serial Interface UART0

This section explains the two modes of serial interface UART0.

### 11.4.1 Operation stop mode

In this mode, serial transfer cannot be executed; therefore, the power consumption can be reduced. In addition, the pins can be used as ordinary port pins in this mode.

#### (1) Register setting

The operation stop mode is set by asynchronous serial interface operation mode register 0 (ASIM0).

ASIM0 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$  input sets this register to 01H.

Address: FF70H After reset: 01H R/W

Symbol	<span style="border: 1px solid black; padding: 0 2px;">7</span>	<span style="border: 1px solid black; padding: 0 2px;">6</span>	<span style="border: 1px solid black; padding: 0 2px;">5</span>	4	3	2	1	0
ASIM0	POWER0	TXE0	RXE0	PS01	PS00	CL0	SL0	1

POWER0	Enables/disables operation of internal operation clock
0 <sup>Note</sup>	Disables operation of the internal operation clock (fixes the clock to low level) and asynchronously resets the internal circuit.
1	Enables operation of the internal operation clock.

TXE0	Enables/disables transmission
0	Disables transmission (synchronously resets the transmission circuit).
1	Enables transmission.

RXE0	Enables/disables reception
0	Disables reception (synchronously resets the reception circuit).
1	Enables reception.

**Note** The input from the RxD0 pin is fixed to high level when POWER0 = 0.

- Cautions**
1. At startup, set POWER0 to 1 and then set TXE0 to 1. Clear TXE0 to 0 first, and then clear POWER0 to 0.
  2. At startup, set POWER0 to 1 and then set RXE0 to 1. Clear RXE0 to 0 first, and then clear POWER0 to 0.
  3. TXE0 and RXE0 are synchronized with the base clock (f<sub>CLK</sub>) set by BRGC0. Therefore, the transmission unit may not be initialized if TXE0 = 1 is not set again 2 clocks after TXE0 = 0 is set. Similarly, the reception unit may not be initialized if RXE0 = 1 is not set again 2 clocks after RXE0 = 0 is set.

### 11.4.2 Asynchronous serial interface (UART) mode

In this mode, 1-byte data is transmitted/received following a start bit, and a full-duplex operation can be performed.

A dedicated UART baud rate generator is incorporated, so that communication can be executed at a wide range of baud rates.

#### (1) Register setting

The UART mode is set by asynchronous serial interface operation mode register 0 (ASIM0), asynchronous serial interface reception error status register 0 (ASIS0), and baud rate generator control register 0 (BRGC0).

##### (a) Asynchronous serial interface operation mode register 0 (ASIM0)

This 8-bit register controls the serial transfer operations of serial interface UART0.

ASIM0 can be set by a 1-bit or 8-bit memory manipulation instruction.

RESET input sets this register to 01H.

Address: FF70H After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
ASIM0	POWER0	TXE0	RXE0	PS01	PS00	CL0	SL0	1

POWER0	Enables/disables operation of internal operation clock
0 <sup>Note</sup>	Disables operation of the internal operation clock (fixes the clock to low level) and asynchronously resets the internal circuit.
1	Enables operation of the internal operation clock.

TXE0	Enables/disables transmission
0	Disables transmission (synchronously resets the transmission circuit).
1	Enables transmission.

RXE0	Enables/disables reception
0	Disables reception (synchronously resets the reception circuit).
1	Enables reception

**Note** The input from the RxD0 pin is fixed to high level when POWER0 = 0.

- Cautions**
1. At startup, set POWER0 to 1 and then set TXE0 to 1. Clear TXE0 to 0 first, and then clear POWER0 to 0.
  2. At startup, set POWER0 to 1 and then set RXE0 to 1. Clear RXE0 to 0 first, and then clear POWER0 to 0.
  3. TXE0 and RXE0 are synchronized with the base clock (f<sub>CLK</sub>) set by BRGC0. Therefore, the transmission unit may not be initialized if TXE0 = 1 is not set again 2 clocks after TXE0 = 0 is set. Similarly, the reception unit may not be initialized if RXE0 = 1 is not set again 2 clocks after RXE0 = 0 is set.
  4. Be sure to set bit 0 to 1.

PS01	PS00	Transmission operation	Reception operation
0	0	Does not output parity bit.	Reception without parity
0	1	Outputs 0 parity.	Reception as 0 parity <sup>Note</sup>
1	0	Outputs odd parity.	Judges as odd parity.
1	1	Outputs even parity.	Judges as even parity.

CL0	Specifies character length of transmit/receive data
0	Character length of data = 7 bits
1	Character length of data = 8 bits

SL0	Specifies number of stop bits of transmit data
0	Number of stop bits = 1
1	Number of stop bits = 2

**Note** If “reception as 0 parity” is selected, the parity is not judged. Therefore, bit 2 (PE0) of asynchronous serial interface status register 0 (ASIS0) is not set and the error interrupt does not occur.

- Cautions**
1. Clear the TXE0 and RXE0 bits to 0 before rewriting the PS01, PS00, and CL0 bits.
  2. Make sure that TXE0 = 0 when rewriting the SL0 bit. Reception is always performed with "the number of stop bits = 1", and therefore, is not affected by the set value of the SL0 bit.

**(b) Asynchronous serial interface reception error status register 0 (ASIS0)**

This register indicates an error status on completion of reception by serial interface UART0. It includes three error flag bits (PE0, FE0, OVE0).

This register can be set by an 8-bit memory manipulation instruction and is read-only.

$\overline{\text{RESET}}$  input clears this register to 00H if bit 7 (POWER0) and bit 5 (RXE0) of ASIM0 = 0. 00H is read when this register is read.

Address: FF73H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS0	0	0	0	0	0	PE0	FE0	OVE0

PE0	Status flag indicating parity error
0	If POWER0 = 0 and RXE0 = 0, or if ASIS0 register is read.
1	If the parity of transmit data does not match the parity bit on completion of reception.

FE0	Status flag indicating framing error
0	If POWER0 = 0 and RXE0 = 0, or if ASIS0 register is read.
1	If the stop bit is not detected on completion of reception.

OVE0	Status flag indicating overrun error
0	If POWER0 = 0 and RXE0 = 0, or if ASIS0 register is read.
1	If receive data is set to the RXB register and the next reception operation is completed before the data is read.

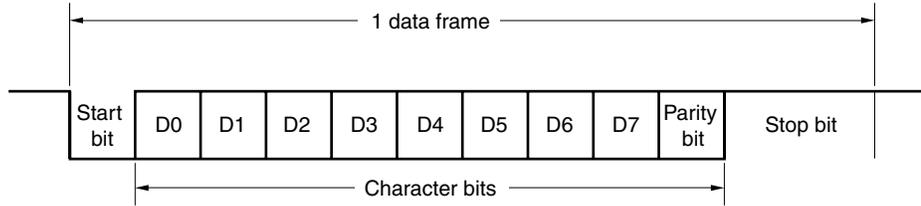
- Cautions**
1. The operation of the PE0 bit differs depending on the set values of the PS01 and PS00 bits of asynchronous serial interface mode register 0 (ASIM0).
  2. Only the first bit of the receive data is checked as the stop bit, regardless of the number of stop bits.
  3. If an overrun error occurs, the next receive data is not written to receive buffer register 0 (RXB0) but discarded.
  4. If data is read from ASIS0, a wait cycle is generated. For details, refer to CHAPTER 25 CAUTIONS FOR WAIT.

(2) Communication operation

(a) Normal transmit/receive data format

Figure 11-5 shows the format of the transmit/receive data.

Figure 11-5. Format of Normal UART Transmit/Receive Data



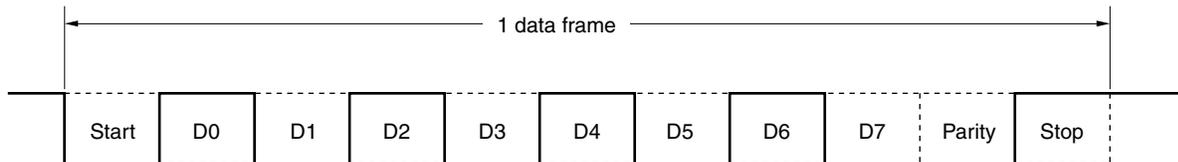
One data frame consists of the following bits.

- Start bit ... 1 bit
- Character bits ... 7 or 8 bits (LSB first)
- Parity bit ... Even parity, odd parity, 0 parity, or no parity
- Stop bit ... 1 or 2 bits

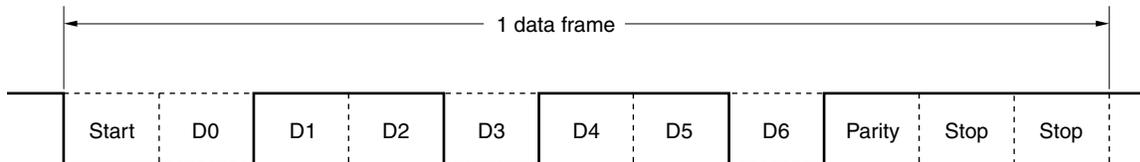
The character bit length, parity, and stop bit length in one data frame are specified by asynchronous serial interface mode register 0 (ASIM0).

Figure 11-6. Example of Normal UART Transmit/Receive Data Format

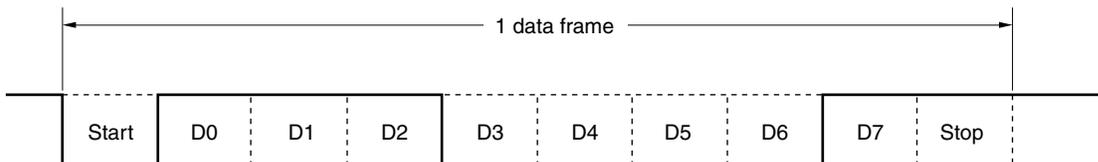
1. Data length: 8 bits, Parity: Even parity, Stop bit: 1 bit, Transfer data: 55H



2. Data length: 7 bits, Parity: Odd parity, Stop bit: 2 bits, Transfer data: 36H



3. Data length: 8 bits, Parity: None, Stop bit: 1 bit, Transfer data: 87H



**(b) Parity types and operation**

The parity bit is used to detect a bit error in communication data. Usually, the same type of parity bit is used on both the transmission and reception sides. With even parity and odd parity, a 1-bit (odd number) error can be detected. With zero parity and no parity, an error cannot be detected.

**(i) Even parity****• Transmission**

Transmit data, including the parity bit, is controlled so that the number of bits that are "1" is even.

The value of the parity bit is as follows.

If transmit data has an odd number of bits that are "1": 1

If transmit data has an even number of bits that are "1": 0

**• Reception**

The number of bits that are "1" in the receive data, including the parity bit, is counted. If it is odd, a parity error occurs.

**(ii) Odd parity****• Transmission**

Unlike even parity, transmit data, including the parity bit, is controlled so that the number of bits that are "1" is odd.

If transmit data has an odd number of bits that are "1": 0

If transmit data has an even number of bits that are "1": 1

**• Reception**

The number of bits that are "1" in the receive data, including the parity bit, is counted. If it is even, a parity error occurs.

**(iii) 0 parity**

The parity bit is cleared to 0 when data is transmitted, regardless of the transmit data.

The parity bit is not detected when the data is received. Therefore, a parity error does not occur regardless of whether the parity bit is "0" or "1".

**(iv) No parity**

No parity bit is appended to the transmit data.

Reception is performed assuming that there is no parity bit when data is received. Because there is no parity bit, a parity error does not occur.

**(c) Transmission**

The TxD0 pin outputs a high level when bit 7 (POWER0) of asynchronous serial interface mode register 0 (ASIM0) is set to 1. If bit 6 (TXE0) of ASIM0 is then set to 1, transmission is enabled. Transmission can be started by writing transmit data to transmit shift register 0 (TXS0). The start bit, parity bit, and stop bit are automatically appended to the data.

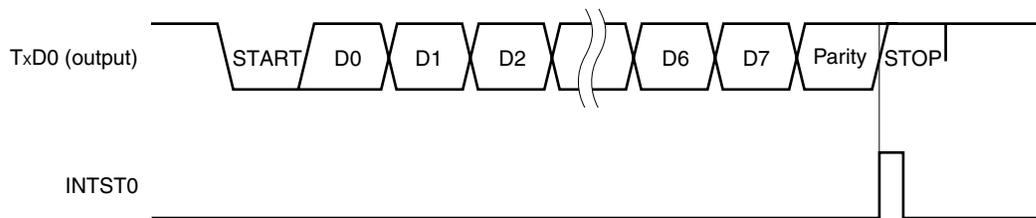
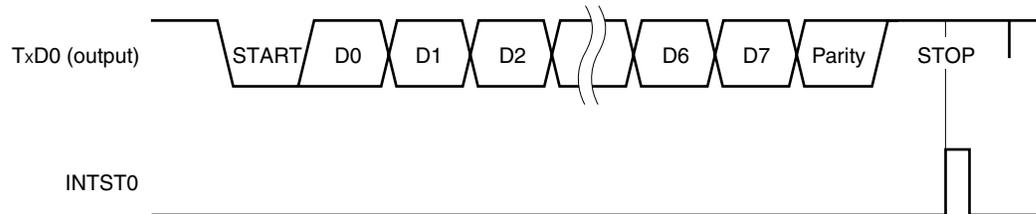
When transmission is started, the start bit is output from the TxD0 pin, followed by the rest of the data in order starting from the LSB. When transmission is completed, the parity and stop bits set by ASIM0 are appended and a transmission completion interrupt request (INTST0) is generated.

Transmission is stopped until the data to be transmitted next is written to TXS0.

Figure 11-7 shows the timing of the transmission completion interrupt request (INTST0). This interrupt occurs as soon as the last stop bit has been output.

**Caution** After transmit data is written to TXS0, do not write the next transmit data before the transmission completion interrupt signal (INTST0) is generated.

**Figure 11-7. Normal Transmission Completion Interrupt Request Timing**

**1. Stop bit length: 1****2. Stop bit length: 2**

**(d) Reception**

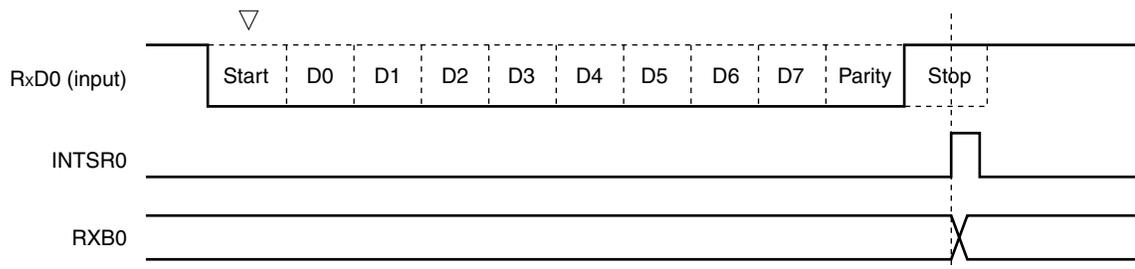
Reception is enabled and the RxD0 pin input is sampled when bit 7 (POWER0) of asynchronous serial interface mode register 0 (ASIM0) is set to 1 and then bit 5 (RXE0) of ASIM0 is set to 1.

The 5-bit counter of the baud rate generator starts counting when the falling edge of the RxD0 pin input is detected. When the set value of baud rate generator control register 0 (BRGC0) has been counted, the RxD0 pin input is sampled again (▽ in Figure 11-8). If the RxD0 pin is low level at this time, it is recognized as a start bit.

When the start bit is detected, reception is started, and serial data is sequentially stored in receive shift register 0 (RXS0) at the set baud rate. When the stop bit has been received, the reception completion interrupt (INTSR0) is generated and the data of RXS0 is written to receive buffer register 0 (RXB0). If an overrun error (OVE0) occurs, however, the receive data is not written to RXB0.

Even if a parity error (PE0) or a framing error (FE0) occurs while reception is in progress, reception continues to the reception position of the stop bit, and an error interrupt (INTSR0) is generated after completion of reception.

**Figure 11-8. Reception Completion Interrupt Request Timing**



- Cautions**
1. Be sure to read receive buffer register 0 (RXB0) even if a reception error occurs. Otherwise, an overrun error will occur when the next data is received, and the reception error status will persist.
  2. Reception is always performed with the “number of stop bits = 1”. The second stop bit is ignored.
  3. Be sure to read asynchronous serial interface reception error status register 0 (ASIS0) before reading RXB0.

**(e) Reception error**

Three types of errors may occur during reception: a parity error, framing error, or overrun error. If the error flag of asynchronous serial interface reception error status register 0 (ASIS0) is set as a result of data reception, a reception error interrupt request (INTSR0) is generated.

Which error has occurred during reception can be identified by reading the contents of ASIS0 in the reception error interrupt servicing (INTSR0) (refer to **Table 11-2**).

The contents of ASIS0 are reset to 0 when ASIS0 is read.

**Table 11-2. Cause of Reception Error**

Reception Error	Cause	Value of ASIS0
Parity error	The parity specified for transmission does not match the parity of the receive data.	04H
Framing error	Stop bit is not detected.	02H
Overrun error	Reception of the next data is completed before data is read from receive buffer register 0 (RXB0).	01H

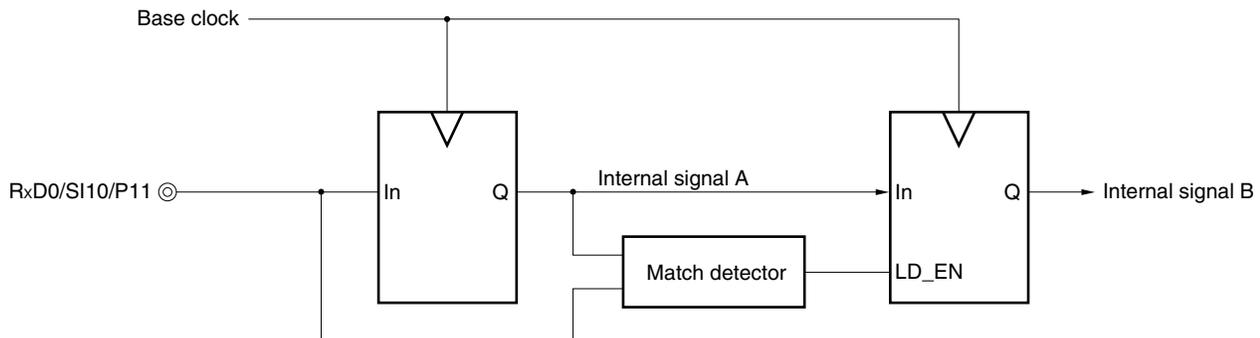
**(f) Noise filter of receive data**

The RxD0 signal is sampled using the base clock output by the prescaler block.

If two sampled values are the same, the output of the match detector changes, and the data is sampled as input data.

Because the circuit is configured as shown in Figure 11-9, the internal processing of the reception operation is delayed by two clocks from the external signal status.

**Figure 11-9. Noise Filter Circuit**



### 11.4.3 Dedicated baud rate generator

The dedicated baud rate generator consists of a source clock selector and an 5-bit programmable counter, and generates a serial clock for transmission/reception of UART0.

Separate 5-bit counters are provided for transmission and reception.

#### (1) Configuration of baud rate generator

- Base clock (Clock)

The clock selected by bits 7 and 6 (TPS01 and TPS00) of baud rate generator control register 0 (BRGC0) is supplied to each module when bit 7 (POWER0) of asynchronous serial interface mode register 0 (ASIM0) is 1. This clock is called the base clock “Clock” and its frequency is called  $f_{CLK}$ . “Clock” is fixed to low level when  $POWER0 = 0$ .

- Transmission counter

This counter stops, cleared to 0, when bit 7 (POWER0) or bit 6 (TXE0) of asynchronous serial interface mode register 0 (ASIM0) is 0.

It starts counting when  $POWER0 = 1$  and  $TXE0 = 1$ .

The counter is cleared to 0 when the first data transmitted is written to transmit shift register 0 (TXS0).

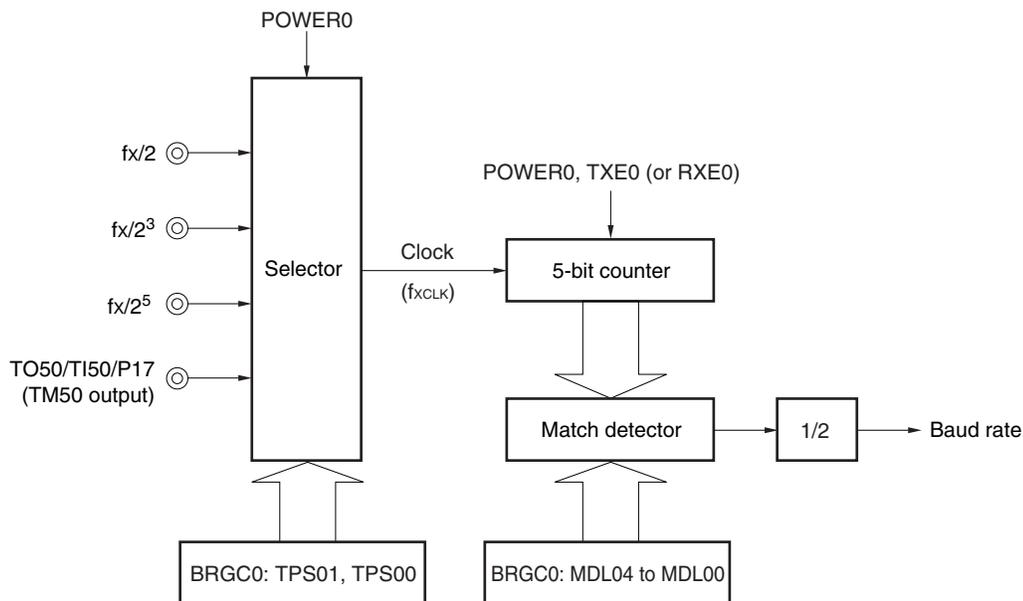
- Reception counter

This counter stops operation, cleared to 0, when bit 7 (POWER0) or bit 5 (RXE0) of asynchronous serial interface mode register 0 (ASIM0) is 0.

It starts counting when the start bit has been detected.

The counter stops operation after one frame has been received, until the next start bit is detected.

Figure 11-10. Configuration of Baud Rate Generator



**Remark** POWER0: Bit 7 of asynchronous serial interface mode register 0 (ASIM0)  
 TXE0: Bit 6 of ASIM0  
 RXE0: Bit 5 of ASIM0  
 BRGC0: Baud rate generator control register 0

**(2) Generation of serial clock**

A serial clock can be generated by using baud rate generator control register 0 (BRGC0).

Select the clock to be input to the 5-bit counter by using bits 7 and 6 (TPS01 and TPS00) of BRGC0.

Bits 4 to 0 (MDL04 to MDL00) of BRGC0 can be used to select the division value of the 5-bit counter.

**(a) Baud rate generator control register 0 (BRGC0)**

This register selects the base clock of serial interface UART0 and controls the baud rate.

BRGC0 can be set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$  input sets this register to 1FH.

Address: FF71H After reset: 1FH R/W

Symbol	7	6	5	4	3	2	1	0
BRGC0	TPS01	TPS00	0	MDL04	MDL03	MDL02	MDL01	MDL00

TPS01	TPS00	Base clock ( $f_{\text{CLK}}$ ) selection
0	0	TM50 output (TO50)
0	1	$f_x/2$ (5 MHz)
1	0	$f_x/2^3$ (1.25 MHz)
1	1	$f_x/2^5$ (312.5 kHz)

MDL04	MDL03	MDL02	MDL01	MDL00	k	Selection of 5-bit counter output clock
0	0	×	×	×	×	Setting prohibited
0	1	0	0	0	8	$f_{\text{CLK}}/8$
0	1	0	0	1	9	$f_{\text{CLK}}/9$
0	1	0	1	0	10	$f_{\text{CLK}}/10$
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
1	1	0	1	0	26	$f_{\text{CLK}}/26$
1	1	0	1	1	27	$f_{\text{CLK}}/27$
1	1	1	0	0	28	$f_{\text{CLK}}/28$
1	1	1	1	0	30	$f_{\text{CLK}}/30$
1	1	1	1	1	31	$f_{\text{CLK}}/31$

**Cautions** 1. Make sure that bit 6 (TXE0) and bit 5 (RXE0) of the ASIM0 register = 0 when rewriting the MDL04 to MDL00 bits.

2. The baud rate value is the output clock of the 5-bit counter divided by 2.

**Remarks** 1.  $f_{\text{CLK}}$ : Frequency of base clock (Clock) selected by the TPS01 and TPS00 bits

2.  $f_x$ : X1 input clock oscillation frequency

3. k: Value set by the MDL04 to MDL00 bits (k = 8, 9, 10, ..., 31)

4. ×: Don't care

5. Figures in parentheses apply to operation with  $f_x = 10$  MHz

**(b) Baud rate**

The baud rate can be calculated by the following expression.

- Baud rate =  $\frac{f_{\text{CLK}}}{2 \times k}$  [bps]

$f_{\text{CLK}}$ : Frequency of base clock (Clock) selected by the TPS01 and TPS00 bits of the BRGC0 register

k: Value set by the MDL04 to MDL00 bits of the BRGC0 register (k = 8, 9, 10, ..., 31)

**(c) Error of baud rate**

The baud rate error can be calculated by the following expression.

- Error (%) =  $\left( \frac{\text{Actual baud rate (baud rate with error)}}{\text{Desired baud rate (correct baud rate)}} - 1 \right) \times 100$  [%]

**Cautions 1. Keep the baud rate error during transmission to within the permissible error range at the reception destination.**

**2. Make sure that the baud rate error during reception satisfies the range shown in (4) Permissible baud rate range during reception.**

**Example:** Frequency of base clock (Clock) = 2.5 MHz = 2,500,000 Hz  
 Set value of MDL04 to MDL00 bits of BRGC0 register = 10000B (k = 16)  
 Target baud rate = 76,800 bps

$$\begin{aligned} \text{Baud rate} &= 2.5 \text{ M}/(2 \times 16) \\ &= 2,500,000/(2 \times 16) = 78,125 \text{ [bps]} \end{aligned}$$

$$\begin{aligned} \text{Error} &= (78,125/76,800 - 1) \times 100 \\ &= 1.725 \text{ [%]} \end{aligned}$$

## (3) Example of setting baud rate

Table 11-3. Set Data of Baud Rate Generator

Baud Rate [bps]	fx = 10.0 MHz				fx = 8.38 MHz				fx = 4.19 MHz			
	TPS01, TPS00	k	Calculated value	ERR[%]	TPS01, TPS00	k	Calculated value	ERR[%]	TPS01, TPS00	k	Calculated value	ERR[%]
2400	–	–	–	–	–	–	–	–	3	27	2425	1.03
4800	–	–	–	–	3	16	4883	1.73	3	14	4676	–2.58
9600	3	16	9766	1.73	3	8	9766	1.73	2	27	9699	1.03
10400	3	15	10417	0.16	2	30	10417	0.16	2	25	10475	0.72
19200	3	8	19531	1.73	2	16	19531	1.73	2	14	18705	–2.58
31250	2	20	31250	0	2	10	31250	0	–	–	–	–
38400	2	16	39063	1.73	2	8	39063	1.73	1	27	38796	1.03
76800	2	8	78125	1.73	1	16	78125	1.73	1	14	74821	–2.58
115200	1	22	113636	–1.36	1	11	113636	–1.36	1	9	116389	1.03
153600	1	16	156250	1.73	1	8	156250	1.73	–	–	–	–
230400	1	11	227273	–1.36	–	–	–	–	–	–	–	–

**Remark** TPS01, TPS00: Bits 7 and 6 of baud rate generator control register 0 (BRGC0) (setting of base clock (fxCLK))

k: Value set by the MDL04 to MDL00 bits of BRGC0 (k = 8, 9, 10, ..., 31)

fx: X1 input clock oscillation frequency

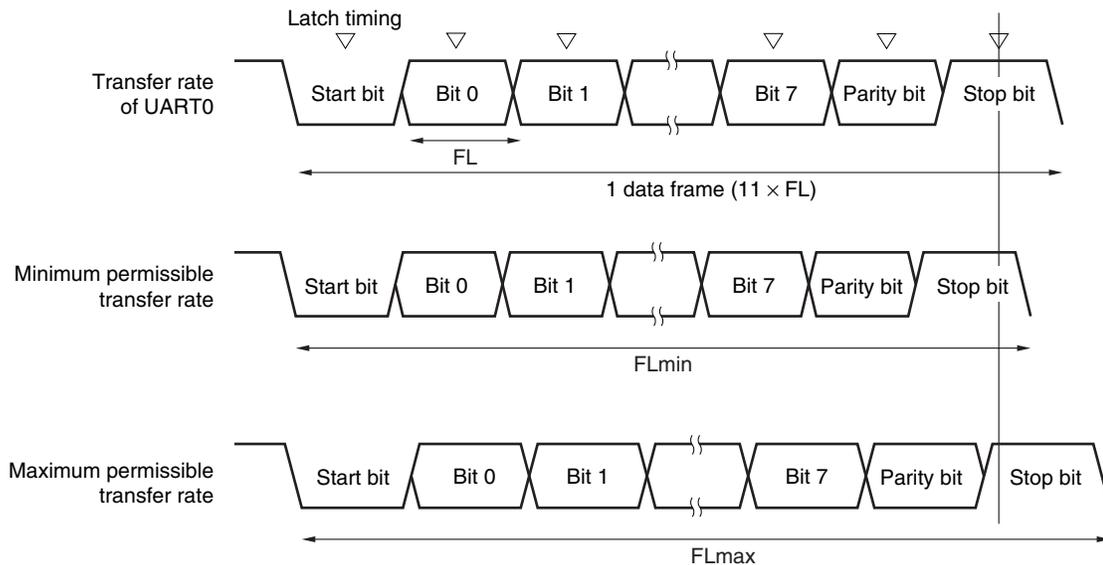
ERR: Baud rate error

**(4) Permissible baud rate range during reception**

The permissible error from the baud rate at the transmission destination during reception is shown below.

**Caution** Make sure that the baud rate error during reception is within the permissible error range, by using the calculation expression shown below.

**Figure 11-11. Permissible Baud Rate Range During Reception**



As shown in Figure 11-11, the latch timing of the receive data is determined by the counter set by baud rate generator control register 0 (BRGC0) after the start bit has been detected. If the last data (stop bit) meets this latch timing, the data can be correctly received.

Assuming that 11-bit data is received, the theoretical values can be calculated as follows.

$$FL = (\text{Brate})^{-1}$$

Brate: Baud rate of UART0

k: Set value of BRGC0

FL: 1-bit data length

Margin of latch timing: 2 clocks

$$\text{Minimum permissible transfer rate: } FL_{\min} = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$$

Therefore, the maximum receivable baud rate at the transmission destination is as follows.

$$BR_{\max} = (FL_{\min}/11)^{-1} = \frac{22k}{21k+2} \text{ Brate}$$

Similarly, the maximum permissible transfer rate can be calculated as follows.

$$\frac{10}{11} \times FL_{\max} = 11 \times FL - \frac{k+2}{2 \times k} \times FL = \frac{21k-2}{2 \times k} FL$$

$$FL_{\max} = \frac{21k \times 2}{20k} FL \times 11$$

Therefore, the minimum receivable baud rate at the transmission destination is as follows.

$$BR_{\min} = (FL_{\max}/11)^{-1} = \frac{20k}{21k-2} \text{ Brate}$$

The permissible baud rate error between UART0 and the transmission destination can be calculated from the above minimum and maximum baud rate expressions, as follows.

**Table 11-4. Maximum/Minimum Permissible Baud Rate Error**

Division Ratio (k)	Maximum Permissible Baud Rate Error	Minimum Permissible Baud Rate Error
8	+3.53%	-3.61%
16	+4.14%	-4.19%
24	+4.34%	-4.38%
31	+4.44%	-4.47%

- Remarks**
1. The accuracy of reception depends on the number of bits in one frame, input clock frequency, and division ratio (k). The higher the input clock frequency and the higher the division ratio (k), the higher the accuracy.
  2. k: Set value of BRGC0

## CHAPTER 12 SERIAL INTERFACE UART6

### 12.1 Functions of Serial Interface UART6

Serial interface UART6 has the following two modes.

#### (1) Operation stop mode

This mode is used when serial transfer is not executed and can enable a reduction in the power consumption. For details, refer to **12.4.1 Operation stop mode**.

#### (2) Asynchronous serial interface (UART) mode

This mode supports the LIN (Local Interconnect Network) bus. The functions of this mode are outlined below.

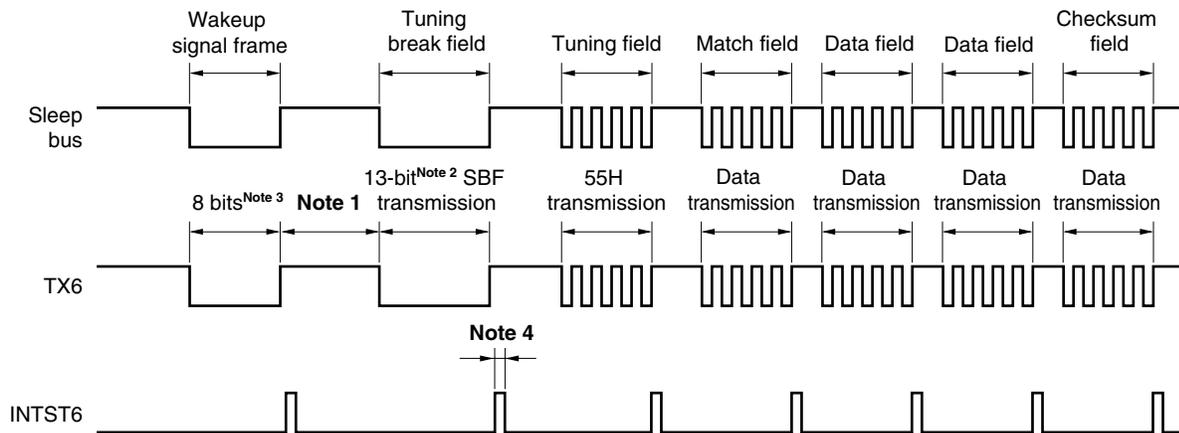
- Two-pin configuration    TxD6: Transmit data output pin  
                                  Rx6B: Receive data input pin
- Data length of transfer data can be selected from 7 or 8 bits.
- Dedicated internal 8-bit baud rate generator allowing any baud rate to be set
- Transmission and reception can be performed independently.
- Twelve operating clock inputs selectable
- MSB- or LSB-first transfer selectable
- Inverted transmission operation
- Tuning break field transmission from 13 to 20 bits
- More than 11 bits can be identified for tuning break field reception (SBF reception flag provided).

- Cautions**
1. **The default value of the TxD6 pin is the high level. Exercise care when using the TxD6 pin as a port pin.**
  2. **The TxD6 output inversion function inverts only the transmission side and not the reception side. To use this function, the reception side must be ready for reception of inverted data (it must be able to recognize a low-level start bit).**
  3. **If clock supply to serial interface UART6 is not stopped (e.g., in the HALT mode), normal operation continues. If clock supply to serial interface UART6 is stopped (e.g., in the STOP mode), each register stops operating, and holds the value immediately before clock supply was stopped. The TxD6 pin also holds the value immediately before clock supply was stopped and outputs it. However, the operation is not guaranteed after clock supply is resumed. Therefore, reset the circuit so that POWER6 = 0, RXE6 = 0, and TXE6 = 0.**
  4. **If data is continuously transmitted, the transfer rate from the stop bit to the next start bit is extended two clocks. However, this does not affect the result of transfer because the reception side initializes the timing when it has detected a start bit. Do not use the continuous transmission function if the interface is incorporated in LIN.**

- ★ **Remark** LIN stands for Local Interconnect Network and is a low-speed (1 to 20 kbps) serial communication protocol designed to reduce the cost of an automobile network. LIN uses single-master communication, and up to 15 slaves can be connected to one master. A LIN slave is used to control switches, actuators, and sensors, which are connected to the LIN master via the LIN. The LIN master is usually connected to a network such as CAN (Controller Area Network). The LIN bus is a single-wire type and each node is connected to the bus via a transceiver conforming to ISO9141. The LIN protocol defines that the master transmits frames that include baud rate information, and a slave receives this information and corrects the baud rate error to that of the master. Therefore, communication is enabled if the baud rate error of the slave is within  $\pm 15\%$ .

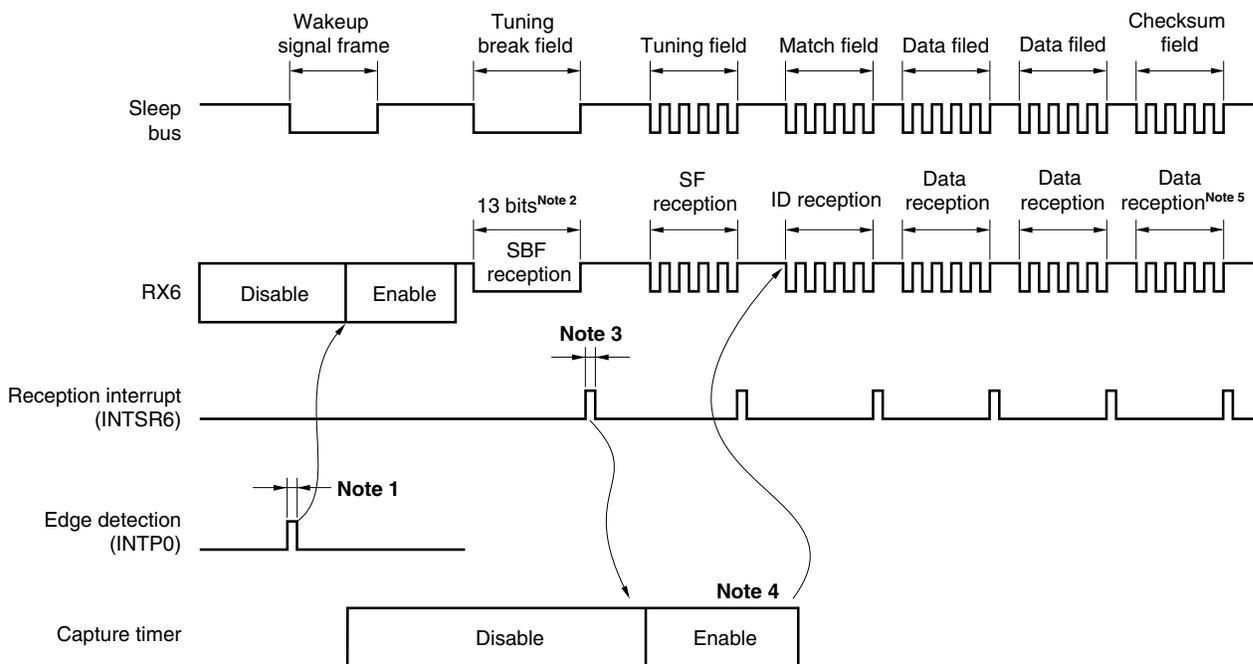
Figures 12-1 and 12-2 outline the transmission and reception operations of LIN.

**Figure 12-1. LIN Transmission Operation**



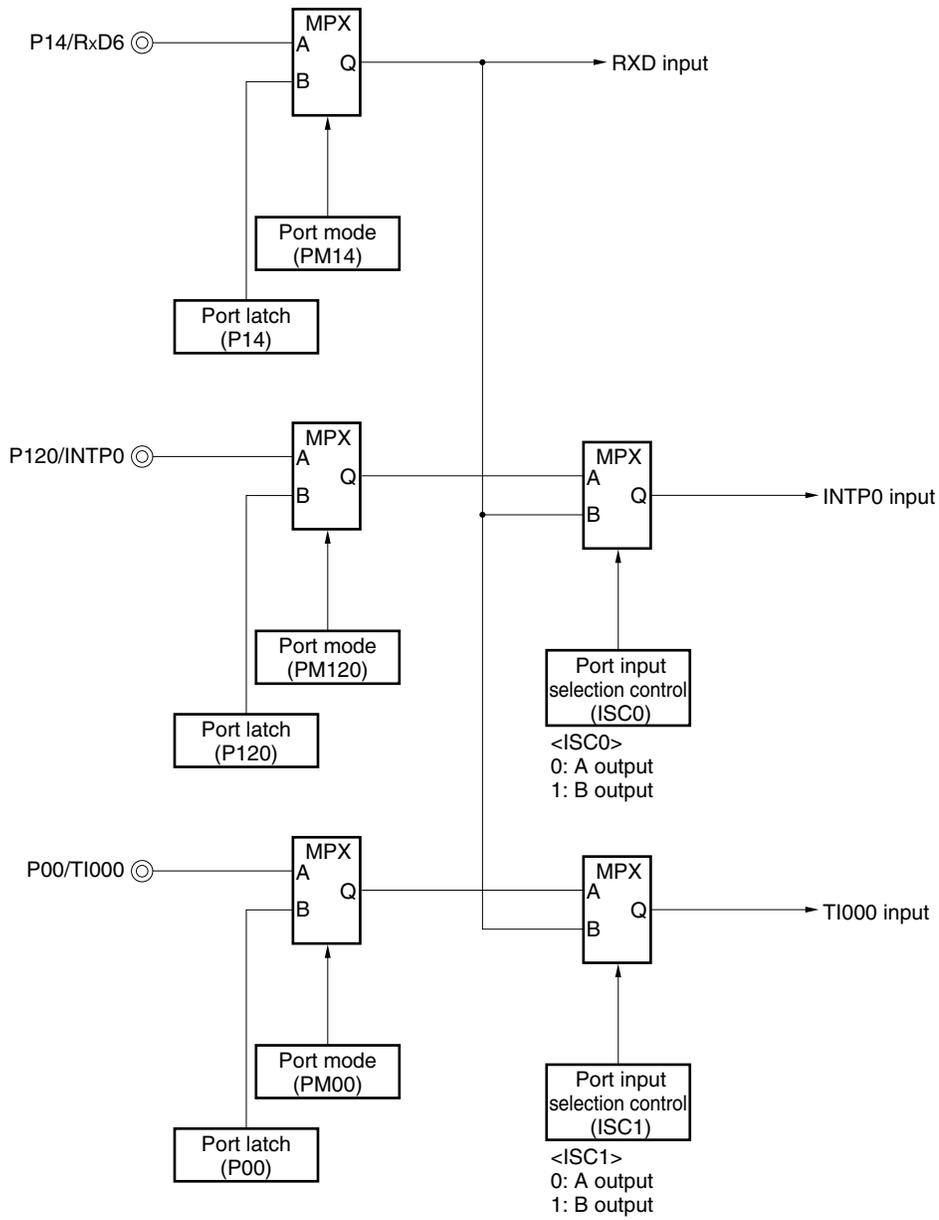
- Notes**
1. The interval between each field is controlled by software.
  2. The tuning break field is output by hardware. The output width is equal to the bit length set by bits 5 to 3 (SBL62 to SBL60) of the asynchronous serial interface control register 6 (ACICL6). If the output width needs to be adjusted more accurately, use baud rate generator control register 6 (BRGC6).
  3. The wakeup signal frame is substituted by 80H transfer in the 8-bit mode.
  4. INTST6 is output on completion of each transmission. It is also output when SBF is transmitted.

Figure 12-2. LIN Reception Operation



- Notes**
1. The wakeup signal is detected at the edge of the pin, and enables UART6 and sets the SBF reception mode.
  2. Reception continues until the STOP bit is detected. When 11 bits or more of SBF have been detected, it is assumed that SBF reception has been completed correctly, and an interrupt signal is output. If less than 11 bits of SBF have been detected, it is assumed that an SBF reception error has occurred. The interrupt signal is not output and the SBF reception mode is restored.
  3. If SBF reception has been completed correctly, an interrupt signal is output. This SBF reception completion interrupt enables the capture timer. Detection of errors OVE6, PE6, and FE6 is suppressed, and error detection processing of UART communication and data transfer of the shift register and RXB6 is not performed. The shift register holds the reset value FFH.
  4. Calculate the baud rate error from the value obtained from the capture timer, disable UART6 after SF reception, and then re-set baud rate generator control register 6 (BRGC6).
  5. Distinguish the checksum field by software. Also perform processing by software to initialize UART6 after reception of the checksum field and to set the SBF reception mode again.

Figure 12-3. Port Configuration for LIN Reception Operation



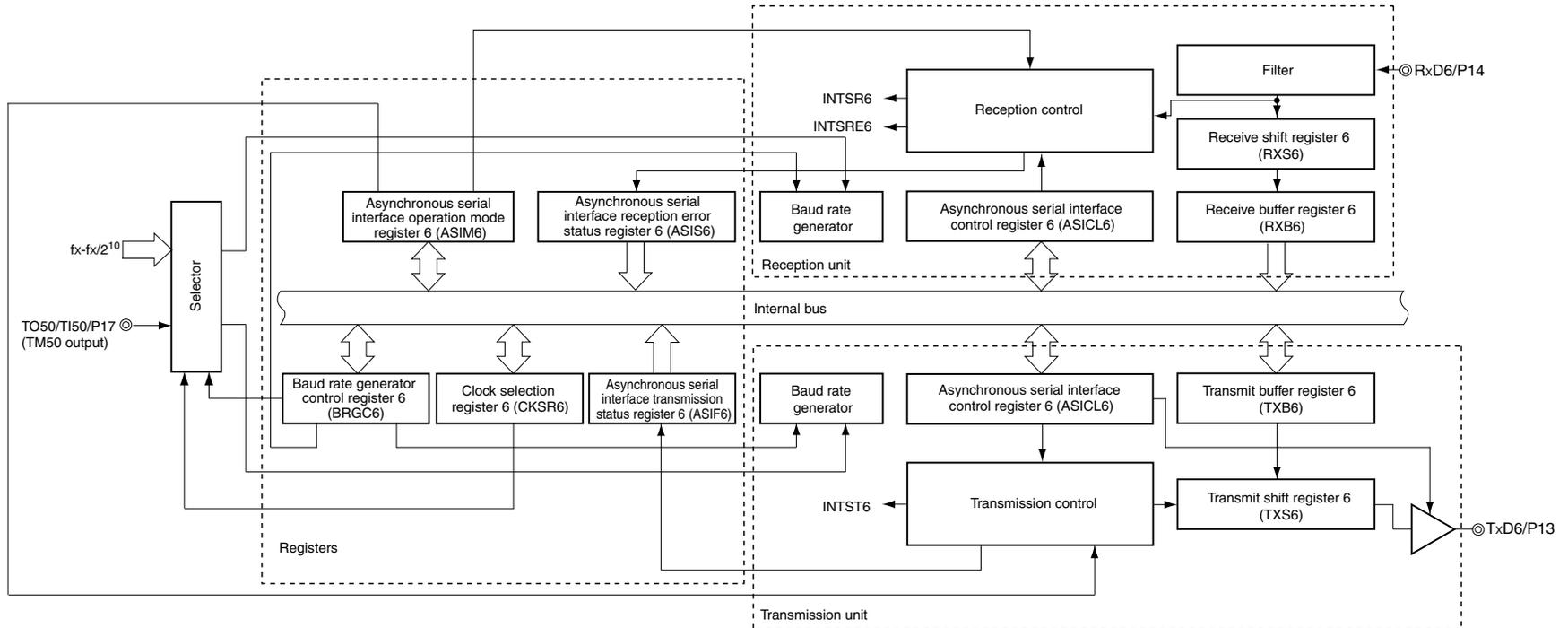
## 12.2 Configuration of Serial Interface UART6

Serial interface UART6 consists of the following hardware.

**Table 12-1. Configuration of Serial Interface UART6**

Item	Configuration
Registers	Receive buffer register 6 (RXB6) Receive shift register 6 (RXS6) Transmit buffer register 6 (TXB6) Transmit shift register 6 (TXS6)
Control registers	Asynchronous serial interface operation mode register 6 (ASIM6) Asynchronous serial interface reception error status register 6 (ASIS6) Asynchronous serial interface transmission status register 6 (ASIF6) Clock selection register 6 (CKSR6) Baud rate generator control register 6 (BRGC6) Asynchronous serial interface control register 6 (ASICL6)

Figure 12-4. Block Diagram of Serial Interface UART6



**(1) Receive buffer register 6 (RXB6)**

This 8-bit register stores parallel data converted by the receive shift register.

Each time 1 byte of data has been received, new receive data is transferred to this register from receive shift register 6 (RXS6). If the data length is set to 7 bits, data is transferred as follows.

- In LSB-first reception, the receive data is transferred to bits 0 to 6 of RXB6 and the MSB of RXB6 is always 0.
- In MSB-first reception, the receive data is transferred to bits 1 to 7 of RXB6 and the LSB of RXB6 is always 0.

If an overrun error (OVE6) occurs, the receive data is not transferred to RXB6.

RXB6 can be read by an 8-bit memory manipulation instruction. No data can be written to this register.

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

**(2) Receive shift register 6 (RXS6)**

This register converts the serial data input to the RxD6 pin into parallel data.

RXS6 cannot be directly manipulated by a program.

**(3) Transmit buffer register 6 (TXB6)**

This buffer register is used to set transmit data. Transmission is started when data is written to TXB6.

This register can be read or written by an 8-bit memory manipulation instruction.

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

- Cautions**
1. Do not write data to TXB6 when bit 1 (TXBF6) of asynchronous serial interface transmission status register 6 (ASIF6) is 1.
  2. Do not refresh (write the same value to) TXB6 by software during a communication operation (when bit 7 (POWER6) and bit 6 (TXE6) of the asynchronous serial interface operation mode register 6 (ASIM6) are 1 or when bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 are 1). However, if the same value is continuously transmitted in the transmission mode (POWER6 = 1 and TXE6 = 1), the same value can be written.

**(4) Transmit shift register 6 (TXS6)**

This register transmits the data transferred from TXB6 from the TxD6 pin as serial data. Data is transferred from TXB6 immediately after TXB6 is written for the first transmission, or immediately before INTST6 occurs after one frame was transmitted for continuous transmission. Data is transferred from TXB6 and transmitted from the TxD6 pin at the falling edge of the internal clock.

TXS6 cannot be directly manipulated by a program.

### 12.3 Registers Controlling Serial Interface UART6

Serial interface UART6 is controlled by the following six registers.

- Asynchronous serial interface operation mode register 6 (ASIM6)
- Asynchronous serial interface reception error status register 6 (ASIS6)
- Asynchronous serial interface transmission status register 6 (ASIF6)
- Clock selection register 6 (CKSR6)
- Baud rate generator control register 6 (BRGC6)
- Asynchronous serial interface control register 6 (ASICL6)

#### (1) Asynchronous serial interface operation mode register 6 (ASIM6)

This 8-bit register controls the serial transfer operations of serial interface UART6.

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

$\overline{\text{RESET}}$  input sets this register to 01H.

**Remark** ASIM6 can be refreshed (the same value is written) by software during a communication operation (when bit 7 (POWER6) and bit 6 (TXE6) of ASIM6 = 1 or bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 = 1).

**Figure 12-5. Format of Asynchronous Serial Interface Operation Mode Register 6 (ASIM6) (1/2)**

Address: FF50H After reset: 01H R/W

Symbol	[7]	[6]	[5]	4	3	2	1	0
ASIM6	POWER6	TXE6	RXE6	PS61	PS60	CL6	SL6	ISRM6

POWER6	Enables/disables operation of internal operation clock
0 <sup>Note 1</sup>	Disables operation of the internal operation clock (fixes the clock to low level) and asynchronously resets the internal circuit.
1 <sup>Note 2</sup>	Enables operation of the internal operation clock

TXE6	Enables/disables transmission
0	Disables transmission (synchronously resets the transmission circuit).
1	Enables transmission

- Notes**
1. The output of the TxD6 pin goes high and the input from the RxD6 pin is fixed to the high level when POWER6 = 0.
  2. Operation of the internal operation clock is enabled at the second input clock after 1 is written to the POWER6 bit.

**Caution** At startup, set POWER6 to 1 and then set TXE6 to 1. Clear TXE6 to 0 first, and then clear POWER6 to 0.

Figure 12-5. Format of Asynchronous Serial Interface Operation Mode Register 6 (ASIM6) (2/2)

RXE6	Enables/disables reception
0	Disables reception (synchronously resets the reception circuit).
1	Enables reception

PS61	PS60	Transmission operation	Reception operation
0	0	Does not output parity bit.	Reception without parity
0	1	Outputs 0 parity.	Reception as 0 parity <sup>Note</sup>
1	0	Outputs odd parity.	Judges as odd parity.
1	1	Outputs even parity.	Judges as even parity.

CL6	Specifies character length of transmit/receive data
0	Character length of data = 7 bits
1	Character length of data = 8 bits

SL6	Specifies number of stop bits of transmit data
0	Number of stop bits = 1
1	Number of stop bits = 2

ISRM6	Enables/disables occurrence of reception completion interrupt in case of error
0	"INTSRE6" occurs in case of error (at this time, INTSR6 does not occur).
1	"INTSR6" occurs in case of error (at this time, INTSRE6 does not occur).

**Note** If "reception as 0 parity" is selected, the parity is not judged. Therefore, bit 2 (PE6) of asynchronous serial interface status register 6 (ASIS6) is not set and the error interrupt does not occur.

- Cautions**
1. At startup, set POWER6 to 1 and then set RXE6 to 1. Clear RXE6 to 0 first, and then clear POWER6 to 0.
  2. Clear the TXE6 and RXE6 bits to 0 before rewriting the PS61, PS60, and CL6 bits.
  3. Fix the PS61 and PS60 bits to 0 when mounting the device on LIN.
  4. Make sure that TXE6 = 0 when rewriting the SL6 bit. Reception is always performed with "the number of stop bits = 1", and therefore, is not affected by the set value of the SL6 bit.
  5. Make sure that RXE6 = 0 when rewriting the ISRM6 bit.

**(2) Asynchronous serial interface reception error status register 6 (ASIS6)**

This register indicates an error status on completion of reception by serial interface UART6. It includes three error flag bits (PE6, FE6, OVE6).

This register can be set by an 8-bit memory manipulation instruction and is read-only.

$\overline{\text{RESET}}$  input clears this register to 00H if bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 = 0. 00H is read when this register is read.

**Figure 12-6. Format of Asynchronous Serial Interface Reception Error Status Register 6 (ASIS6)**

Address: FF53H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS6	0	0	0	0	0	PE6	FE6	OVE6

PE6	Status flag indicating parity error
0	If POWER6 = 0 and RXE6 = 0, or if ASIS6 register is read
1	If the parity of transmit data does not match the parity bit on completion of reception

FE6	Status flag indicating framing error
0	If POWER6 = 0 and RXE6 = 0, or if ASIS6 register is read
1	If the stop bit is not detected on completion of reception

OVE6	Status flag indicating overrun error
0	If POWER6 = 0 and RXE6 = 0, or if ASIS6 register is read
1	If receive data is set to the RXB register and the next reception operation is completed before the data is read.

- Cautions**
1. The operation of the PE6 bit differs depending on the set values of the PS61 and PS60 bits of asynchronous serial interface mode register 6 (ASIM6).
  2. The first bit of the receive data is checked as the stop bit, regardless of the number of stop bits.
  3. If an overrun error occurs, the next receive data is not written to receive buffer register 6 (RXB6) but discarded.
  4. If data is read from ASIS6, a wait cycle is generated. For details, refer to CHAPTER 25 CAUTIONS FOR WAIT.

**(3) Asynchronous serial interface transmission status register 6 (ASIF6)**

This register indicates the status of transmission by serial interface UART6. It includes two status flag bits (TXBF6 and TXSF6).

Transmission can be continued without disruption even during an interrupt period, by writing the next data to the TXB6 register after data has been transferred from the TXB6 register to the TXS6 register.

This register can be set by an 8-bit memory manipulation instruction, and is read-only.

RESET input clears this register to 00H if bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 = 0.

**Figure 12-7. Format of Asynchronous Serial Interface Transmission Status Register 6 (ASIF6)**

Address: FF55H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIF6	0	0	0	0	0	0	TXBF6	TXSF6

TXBF6	Transmit buffer data flag
0	If POWER6 = 0 or TXE6 = 0, or if data is transferred to transmit shift register 6 (TXS6)
1	If data is written to transmit buffer register 6 (TXB6) (if data exists in TXB6)

TXSF6	Transmit shift register data flag
0	If POWER6 = 0 or TXE6 = 0, or if the next data is not transferred from transmit buffer register 6 (TXB6) after completion of transfer
1	If data is transferred from transmit buffer register 6 (TXB6) (if data transmission is in progress)

- Cautions**
- To continuously transmit data, write the data of the first byte to TXB6, check that the value of the TXBF6 flag is 0, and then write the data of the second byte to TXB6. The operation is not guaranteed if data is written to TXB6 while the TXBF6 flag is 1.**
  - While continuous transmission is being executed, check the value of the TXSF6 flag after the transmission completion interrupt to determine the subsequent write processing to TXB6.**
    - If TXSF6 is 1: Continuous transmission is in progress. Data of 1 byte can be written.**
    - If TXSF6 is 0: Continuous transmission is complete. Data of 2 bytes can be written. When doing so, observe Caution 1 above.**
  - While continuous transmission is in progress, check that TXSF6 is 0 after the transmission completion interrupt, and then execute clearing (POWER6 = 0 or TXE6 = 0). If clearing is executed while the TXSF6 flag is 1, the transmit data cannot be guaranteed.**

**(4) Clock selection register 6 (CKSR6)**

This register selects the base clock of serial interface UART6.

CKSR6 can be set by an 8-bit memory manipulation instruction.

RESET input clears this register to 00H.

**Remark** CKSR6 can be refreshed (the same value is written) by software during a communication operation (when bit 7 (POWER6) and bit 6 (TXE6) of ASIM6 = 1 or bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 = 1).

**Figure 12-8. Format of Clock Selection Register 6 (CKSR6)**

Address: FF56H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CKSR6	0	0	0	0	TPS63	TPS62	TPS61	TPS60

TPS63	TPS62	TPS61	TPS60	Base clock (f <sub>CLK</sub> )
0	0	0	0	f <sub>x</sub> (10 MHz)
0	0	0	1	f <sub>x</sub> /2 (5 MHz)
0	0	1	0	f <sub>x</sub> /2 <sup>2</sup> (2.5 MHz)
0	0	1	1	f <sub>x</sub> /2 <sup>3</sup> (1.25 MHz)
0	1	0	0	f <sub>x</sub> /2 <sup>4</sup> (625 kHz)
0	1	0	1	f <sub>x</sub> /2 <sup>5</sup> (312.5 kHz)
0	1	1	0	f <sub>x</sub> /2 <sup>6</sup> (156.25 kHz)
0	1	1	1	f <sub>x</sub> /2 <sup>7</sup> (78.13 kHz)
1	0	0	0	f <sub>x</sub> /2 <sup>8</sup> (39.06 kHz)
1	0	0	1	f <sub>x</sub> /2 <sup>9</sup> (19.53 kHz)
1	0	1	0	f <sub>x</sub> /2 <sup>10</sup> (9.77 kHz)
1	0	1	1	TM50 output
Other				Setting prohibited

**Caution** Make sure POWER6 = 0 when rewriting TPS63 to TPS60.

- Remarks**
1. Figures in parentheses are for operation with f<sub>x</sub> = 10 MHz
  2. f<sub>x</sub>: X1 input clock oscillation frequency

**(5) Baud rate generator control register 6 (BRGC6)**

This register selects the base clock of serial interface UART6.  
 BRGC6 can be set by an 8-bit memory manipulation instruction.  
 RESET input sets this register to FFH.

**Remark** BRGC6 can be refreshed (the same value is written) by software during a communication operation (when bit 7 (POWER6) and bit 6 (TXE6) of ASIM6 = 1 or bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 = 1).

**Figure 12-9. Format of Baud Rate Generator Control Register 6 (BRGC6)**

Address: FF57H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
BRGC6	MDL67	MDL66	MDL65	MDL64	MDL63	MDL62	MDL61	MDL60

MDL67	MDL66	MDL65	MDL64	MDL63	MDL62	MDL61	MDL60	k	Output clock selection of 8-bit counter
0	0	0	0	0	×	×	×	×	Setting prohibited
0	0	0	0	1	0	0	0	8	$f_{CLK}/8$
0	0	0	0	1	0	0	1	9	$f_{CLK}/9$
0	0	0	0	1	0	1	0	10	$f_{CLK}/10$
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
1	1	1	1	1	1	0	0	252	$f_{CLK}/252$
1	1	1	1	1	1	0	1	253	$f_{CLK}/253$
1	1	1	1	1	1	1	0	254	$f_{CLK}/254$
1	1	1	1	1	1	1	1	255	$f_{CLK}/255$

- Cautions**
1. Make sure that bit 6 (TXE6) and bit 5 (RXE6) of the ASIM6 register = 0 when rewriting the MDL67 to MDL60 bits.
  2. The baud rate is the output clock of the 8-bit counter divided by 2.

- Remarks**
1.  $f_{CLK}$ : Frequency of base clock (Clock) selected by the TPS63 to TPS60 bits of CKSR6 register
  2. k: Value set by MDL67 to MDL60 bits (k = 8, 9, 10, ..., 255)
  3. ×: Don't care

**(6) Asynchronous serial interface control register 6 (ASICL6)**

This register controls the serial transfer operations of serial interface UART6.

ASICL6 can be set by a 1-bit transfer instruction or an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$  input sets this register to 16H.

**Remark** ASICL6 can be refreshed (the same value is written) by software during a communication operation (when bit 7 (POWER6) and bit 6 (TXE6) of ASIM6 = 1 or bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 = 1). However, transfer is started by refresh because bit 6 (SBRT6) and bit 5 (SBTT6) of ASICL6 are cleared to 0 when communication is complete (when an interrupt signal is generated).

**Figure 12-10. Format of Asynchronous Serial Interface Control Register 6 (ASICL6) (1/2)**

Address: FF58H After reset: 16H R/W

Symbol	<span style="border: 1px solid black; padding: 0 2px;">7</span>	6	5	4	3	2	1	0
ASICL6	SBRF6	SBRT6	SBTT6	SBL62	SBL61	SBL60	DIR6	TXDLV6

SBRF6	SBF reception status flag
0	If POWER6 = 0 and RXE6 = 0 or if SBF reception has been completed correctly
1	SBF reception in progress

SBRT6	SBF reception trigger
0	–
1	SBF reception trigger

SBTT6	SBF transmission trigger
0	–
1	SBF transmission trigger

- Cautions**
1. In the case of an SBF reception error, return the mode to the SBF reception mode and hold the status of the SBRF6 flag.
  2. Before setting the SBRT6 bit, make sure that bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 = 1.
  3. The read value of the SBRT6 bit is always 0. SBRT6 is automatically cleared to 0 after SBF reception has been correctly completed.
  4. Before setting the SBTT6 bit to 1, make sure that bit 7 (POWER6) and bit 6 (TXE6) of ASIM6 = 1.
  5. The read value of the SBTT6 bit is always 0. SBTT6 is automatically cleared to 0 at the end of SBF transmission.

**Figure 12-10. Format of Asynchronous Serial Interface Control Register 6 (ASICL6) (2/2)**

SBL62	SBL61	SBL60	SBF transmission output width control
1	0	1	SBF is output with 13-bit length.
1	1	0	SBF is output with 14-bit length.
1	1	1	SBF is output with 15-bit length.
0	0	0	SBF is output with 16-bit length.
0	0	1	SBF is output with 17-bit length.
0	1	0	SBF is output with 18-bit length.
0	1	1	SBF is output with 19-bit length.
1	0	0	SBF is output with 20-bit length.

DIR6	MSB/LSB-first transfer
0	MSB-first transfer
1	LSB-first transfer

TXDLV6	Enables/disables inverting TxD6 output
0	Normal output of TxD6
1	Inverted output of TxD6

**Caution** Before rewriting the DIR6 and TXDLV6 bits, clear the TXE6 and RXE6 bits to 0.

## 12.4 Operation of Serial Interface UART6

This section explains the two modes of serial interface UART6.

### 12.4.1 Operation stop mode

In this mode, serial transfer cannot be executed; therefore, the power consumption can be reduced. In addition, the pins can be used as ordinary port pins in this mode.

#### (1) Register setting

The operation stop mode is set by asynchronous serial interface operation mode register 6 (ASIM6).

ASIM6 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$  input sets this register to 01H.

**Remark** ASIM6 can be refreshed (the same value is written) by software during a communication operation (when bit 7 (POWER6) and bit 6 (TXE6) of ASIM6 = 1 or bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 = 1).

Address: FF50H After reset: 01H R/W

Symbol	<span style="border: 1px solid black; padding: 0 2px;">7</span>	<span style="border: 1px solid black; padding: 0 2px;">6</span>	<span style="border: 1px solid black; padding: 0 2px;">5</span>	4	3	2	1	0
ASIM6	POWER6	TXE6	RXE6	PS61	PS60	CL	SL6	ISRM6

POWER6	Enables/disables operation of internal operation clock
0 <sup>Note 1</sup>	Disables operation of the internal operation clock (fixes the clock to low level) and asynchronously resets the internal circuit.
1 <sup>Note 2</sup>	Enables operation of the internal operation clock.

TXE6	Enables/disables transmission
0	Disables transmission operation (synchronously resets the transmission circuit).
1	Enables transmission

RXE6	Enables/disables reception
0	Disables reception (synchronously resets the reception circuit).
1	Enables reception

- Notes**
1. The output of the TxD6 pin goes high and the input from the RxD6 pin is fixed to the high level when POWER6 = 0.
  2. Operation of the internal operation clock is enabled at the second input clock after 1 is written to the POWER6 bit.

- Cautions**
1. At startup, set POWER6 to 1 and then set TXE6 to 1. Clear TXE6 to 0 first, and then clear POWER6 to 0.
  2. At startup, set POWER6 to 1 and then set RXE6 to 1. Clear RXE6 to 0 first, and then clear POWER6 to 0.

**12.4.2 Asynchronous serial interface (UART) mode**

In this mode, data of 1 byte is transmitted/received following a start bit, and a full-duplex operation can be performed.

A dedicated UART baud rate generator is incorporated, so that communication can be executed at a wide range of baud rates.

**(1) Register setting**

The UART mode is set by asynchronous serial interface operation mode register 6 (ASIM6), asynchronous serial interface reception error status register 6 (ASIS6), asynchronous serial interface transmission status register 6 (ASIF6), clock selection register 6 (CKSR6), baud rate generator control register 6 (BRGC6), and asynchronous serial interface control register 6 (ASICL6).

**(a) Asynchronous serial interface operation mode register 6 (ASIM6)**

This 8-bit register controls the serial transfer operations of serial interface UART6.

ASIM6 can be set by a 1-bit or 8-bit memory manipulation instruction.

RESET input sets this register to 01H.

**Remark** ASIM6 can be refreshed (the same value is written) by software during a communication operation (when bit 7 (POWER6) and bit 6 (TXE6) of ASIM6 = 1 or bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 = 1).

Address: FF50H After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
ASIM6	POWER6	TXE6	RXE6	PS61	PS60	CL6	SL6	ISRM6

POWER6	Enables/disables operation of internal operation clock
0 <sup>Note 1</sup>	Disables operation of the internal operation clock (fixes the clock to low level) and asynchronously resets the internal circuit.
1 <sup>Note 2</sup>	Enables operation of the internal operation clock.

TXE6	Enables/disables transmission
0	Disables transmission (synchronously resets the transmission circuit).
1	Enables transmission

- Notes**
1. The output of the TxD6 pin goes high and the input from the RxD6 pin is fixed to the high level when POWER6 = 0.
  2. Operation of the internal operation clock is enabled at the second input clock after 1 is written to the POWER6 bit.

**Caution** At startup, set POWER6 to 1 and then set TXE6 to 1. Clear TXE6 to 0 first, and then clear POWER6 to 0.

RXE6	Enables/disables reception
0	Disables reception (synchronously resets the reception circuit).
1	Enables reception

PS61	PS60	Transmission operation	Reception operation
0	0	Does not output parity bit.	Reception without parity
0	1	Outputs 0 parity.	Reception as 0 parity <sup>Note</sup>
1	0	Outputs odd parity.	Judges as odd parity.
1	1	Outputs even parity.	Judges as even parity.

CL6	Specifies character length of transmit/receive data
0	Character length of data = 7 bits
1	Character length of data = 8 bits

SL6	Specifies number of stop bits of transmit data
0	Number of stop bits = 1
1	Number of stop bits = 2

ISRM6	Enables/disables occurrence of reception completion interrupt in case of error
0	"INTSRE6" occurs in case of error (at this time, INTSR6 does not occur).
1	"INTSR6" occurs in case of error (at this time, INTSRE6 does not occur).

**Note** If "reception as 0 parity" is selected, the parity is not judged. Therefore, bit 2 (PE6) of asynchronous serial interface status register 6 (ASIS6) is not set and the error interrupt does not occur.

- Cautions**
1. At startup, set POWER6 to 1 and then set RXE6 to 1. Clear RXE6 to 0 first, and then clear POWER6 to 0.
  2. Clear the TXE6 and RXE6 bits to 0 before rewriting the PS61, PS60, and CL6 bits.
  3. Fix the PS61 and PS60 bits to 0 when mounting the device on LIN.
  4. Make sure that TXE6 = 0 when rewriting the SL6 bit. Reception is always performed with "the number of stop bits = 1", and therefore, is not affected by the set value of the SL6 bit.
  5. Make sure that RXE6 = 0 when rewriting the ISRM6 bit.

**(b) Asynchronous serial interface reception error status register 6 (ASIS6)**

This register indicates an error status on completion of reception by serial interface UART6. It includes three error flag bits (PE6, FE6, OVE6).

This register can be set by an 8-bit memory manipulation instruction and is read-only.

$\overline{\text{RESET}}$  input clears this register to 00H if bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 = 0. 00H is read when this register is read.

Address: FF53H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS6	0	0	0	0	0	PE6	FE6	OVE6

PE6	Status flag indicating parity error
0	If POWER6 = 0 and RXE6 = 0, or if ASIS6 register is read
1	If the parity of transmit data does not match the parity bit on completion of reception

FE6	Status flag indicating framing error
0	If POWER6 = 0 and RXE6 = 0, or if ASIS6 register is read
1	If the stop bit is not detected on completion of reception

OVE6	Status flag indicating overrun error
0	If POWER6 = 0 and RXE6 = 0, or if ASIS6 register is read
1	If receive data is set to the RXB register and the next reception operation is completed before the data is read.

- Cautions**
1. The operation of the PE6 bit differs depending on the set values of the PS61 and PS60 bits of asynchronous serial interface mode register 6 (ASIM6).
  2. The first bit of the receive data is checked as the stop bit, regardless of the number of stop bits.
  3. If an overrun error occurs, the next receive data is not written to receive buffer register 6 (RXB6) but discarded.
  4. If data is read from ASIS6, a wait cycle is generated. For details, refer to CHAPTER 25 CAUTIONS FOR WAIT.

**(c) Asynchronous serial interface transmission status register 6 (ASIF6)**

This register indicates the status of transmission by serial interface UART6. It includes two status flag bits (TXBF6 and TXSF6).

Transmission can be continued without disruption even during an interrupt period, by writing the next data to the TXB6 register after data has been transferred from the TXB6 register to the TXS6 register.

This register can be set by an 8-bit memory manipulation instruction, and is read-only.

RESET input clears this register to 00H if bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 = 0.

Address: FF55H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIF6	0	0	0	0	0	0	TXBF6	TXSF6

TXBF6	Transmit buffer data flag
0	If POWER6 = 0 or TXE6 = 0, or if data is transferred to transmit shift register 6 (TXS6)
1	If data is written to transmit buffer register 6 (TXB6) (if data exists in TXB6)

TXSF6	Transmit shift register data flag
0	If POWER6 = 0 or TXE6 = 0, or if the next data is not transferred from transmit buffer register 6 (TXB6) after completion of transfer
1	If data is transferred from transmit buffer register 6 (TXB6) (if data transmission is in progress)

- Cautions**
- 1. To continuously transmit data, write the data of the first byte to TXB6, check that the value of the TXBF6 flag is 0, and then write the data of the second byte to TXB6. The operation is not guaranteed if data is written to TXB6 while the TXBF6 flag is 1.**
  - 2. While continuous transmission is being executed, check the value of the TXSF6 flag after the transmission completion interrupt to determine the subsequent write processing to TXB6.**
    - If TXSF6 is 1: Continuous transmission is in progress. Data of 1 byte can be written.**
    - If TXSF6 is 0: Continuous transmission is complete. Data of 2 bytes can be written. When doing so, observe Caution 1 above.**
  - 3. While continuous transmission is in progress, check that TXSF6 is 0 after the transmission completion interrupt, and then execute clearing (POWER6 = 0 or TXE6 = 0). If clearing is executed while the TXSF6 flag is 1, the transmit data cannot be guaranteed.**

**(d) Asynchronous serial interface control register 6 (ASICL6)**

This register controls the serial transfer operations of serial interface UART6.

ASICL6 can be set by a 1-bit transfer instruction or an 8-bit memory manipulation instruction.

RESET input sets this register to 16H.

**Remark** ASICL6 can be refreshed (the same value is written) by software during a communication operation (when bit 7 (POWER6) and bit 6 (TXE6) of ASIM6 = 1 or bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 = 1). However, transfer is started by refresh because bit 6 (SBRT6) and bit 5 (SBTT6) of ASICL6 are cleared to 0 when communication is complete (when an interrupt signal is generated).

Address: FF58H After reset: 16H R/W

Symbol	<b>7</b>	6	5	4	3	2	1	0
ASICL6	SBRF6	SBRT6	SBTT6	SBL62	SBL61	SBL60	DIR6	TXDLV6

SBRF6	SBF reception status flag
0	If POWER6 = 0 and RXE6 = 0 or if SBF reception has been completed correctly
1	SBF reception in progress

SBRT6	SBF reception trigger
0	–
1	SBF reception trigger

SBTT6	SBF transmission trigger
0	–
1	SBF transmission trigger

- Cautions**
1. In the case of an SBF reception error, return the mode to the SBF reception mode and hold the status of the SBRF6 flag.
  2. Before setting the SBRT6 bit, make sure that bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 = 1.
  3. The read value of the SBRT6 bit is always 0. SBRT6 is automatically cleared to 0 after SBF reception has been correctly completed.
  4. Before setting the SBTT6 bit to 1, make sure that bit 7 (POWER6) and bit 6 (TXE6) of ASIM6 = 1.
  5. The read value of the SBTT6 bit is always 0. SBTT6 is automatically cleared to 0 at the end of SBF transmission.

SBL62	SBL61	SBL60	SBF transmission output width control
1	0	1	SBF is output with 13-bit length.
1	1	0	SBF is output with 14-bit length.
1	1	1	SBF is output with 15-bit length.
0	0	0	SBF is output with 16-bit length.
0	0	1	SBF is output with 17-bit length.
0	1	0	SBF is output with 18-bit length.
0	1	1	SBF is output with 19-bit length.
1	0	0	SBF is output with 20-bit length.

DIR6	MSB/LSB-first transfer
0	MSB-first transfer
1	LSB-first transfer

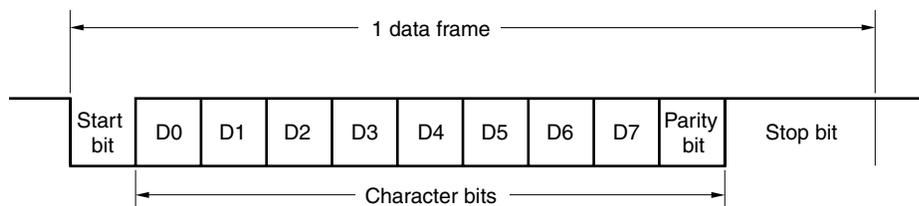
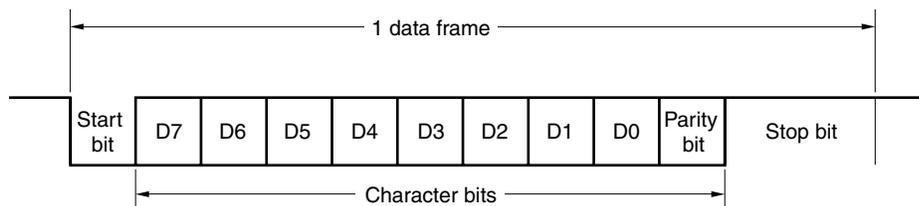
TXDLV6	Enables/disables inverting TxD6 output
0	Normal output of TxD6
1	Inverted output of TxD6

**Caution** Before rewriting the DIR6 and TXDLV6 bits, clear the TXE6 and RXE6 bits to 0.

**(2) Communication operation****(a) Normal transmit/receive data format**

Figure 12-11 shows the format of the transmit/receive data.

**Figure 12-11. Format of Normal UART Transmit/Receive Data**

**1. LSB-first transmission/reception****2. MSB-first transmission/reception**

One data frame consists of the following bits.

- Start bit ... 1 bit
- Character bits ... 7 or 8 bits
- Parity bit ... Even parity, odd parity, 0 parity, or no parity
- Stop bit ... 1 or 2 bits

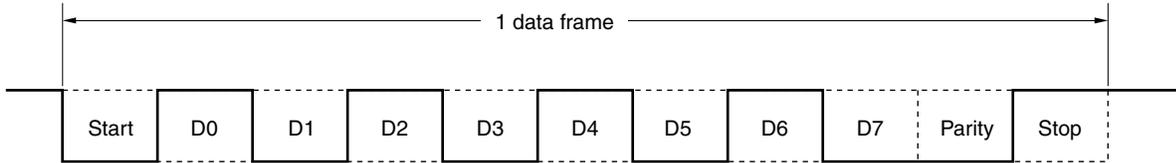
The character bit length, parity, and stop bit length in one data frame are specified by asynchronous serial interface mode register 6 (ASIM6).

Whether data is transferred with the LSB or MSB first is specified by bit 1 (DIR6) of asynchronous serial interface control register 6 (ASICL6).

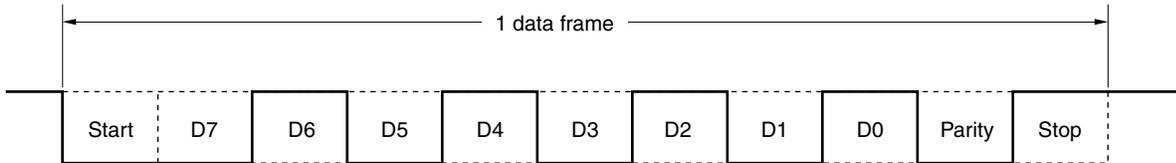
Whether the TxD6 pin outputs normal or inverted data is specified by bit 0 (TXDLV6) of ASICL6.

Figure 12-12. Example of Normal UART Transmit/Receive Data Format

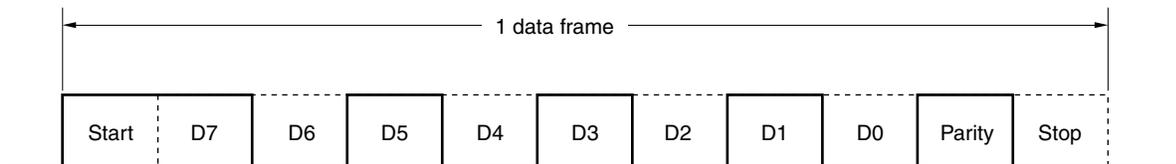
1. Data length: 8 bits, LSB first, Parity: Even parity, Stop bit: 1 bit, Transfer data: 55H



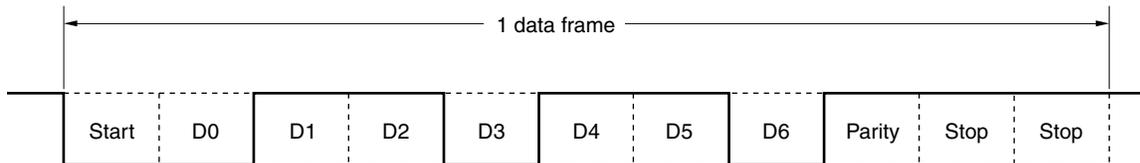
2. Data length: 8 bits, MSB first, Parity: Even parity, Stop bit: 1 bit, Transfer data: 55H



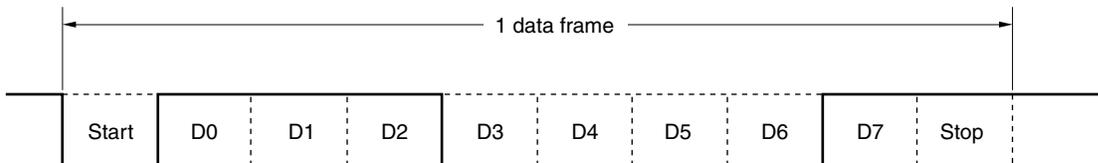
3. Data length: 8 bits, MSB first, Parity: Even parity, Stop bit: 1 bit, Transfer data: 55H, TxD6 pin inverted output



4. Data length: 7 bits, LSB first, Parity: Odd parity, Stop bit: 2 bits, Transfer data: 36H



5. Data length: 8 bits, LSB first, Parity: None, Stop bit: 1 bit, Transfer data: 87H



**(b) Parity types and operation**

The parity bit is used to detect a bit error in communication data. Usually, the same type of parity bit is used on both the transmission and reception sides. With even parity and odd parity, a 1-bit (odd number) error can be detected. With zero parity and no parity, an error cannot be detected.

**Caution** Fix the PS61 and PS60 bits to 0 when the device is incorporated in LIN.

**(i) Even parity**

## • Transmission

Transmit data, including the parity bit, is controlled so that the number of bits that are "1" is even. The value of the parity bit is as follows.

If transmit data has an odd number of bits that are "1": 1

If transmit data has an even number of bits that are "1": 0

## • Reception

The number of bits that are "1" in the receive data, including the parity bit, is counted. If it is odd, a parity error occurs.

**(ii) Odd parity**

## • Transmission

Unlike even parity, transmit data, including the parity bit, is controlled so that the number of bits that are "1" is odd.

If transmit data has an odd number of bits that are "1": 0

If transmit data has an even number of bits that are "1": 1

## • Reception

The number of bits that are "1" in the receive data, including the parity bit, is counted. If it is even, a parity error occurs.

**(iii) 0 parity**

The parity bit is cleared to 0 when data is transmitted, regardless of the transmit data.

The parity bit is not detected when the data is received. Therefore, a parity error does not occur regardless of whether the parity bit is "0" or "1".

**(iv) No parity**

No parity bit is appended to the transmit data.

Reception is performed assuming that there is no parity bit when data is received. Because there is no parity bit, a parity error does not occur.

**(c) Normal transmission**

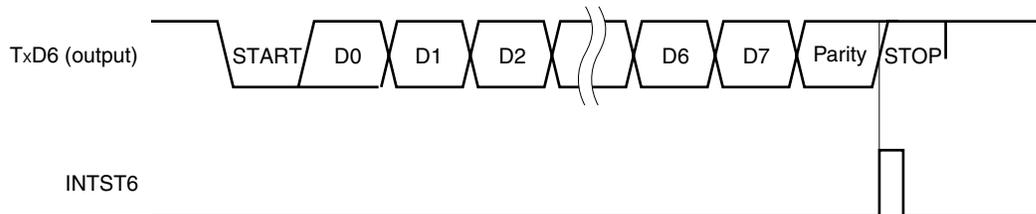
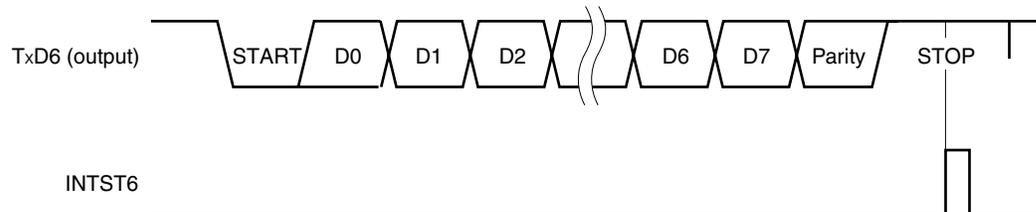
The TxD6 pin outputs a high level when bit 7 (POWER6) of asynchronous serial interface mode register 6 (ASIM6) is set to 1. If bit 6 (TXE6) of ASIM6 is then set to 1, transmission is enabled. Transmission can be started by writing transmit data to transmit buffer register 6 (TXB6). The start bit, parity bit, and stop bit are automatically appended to the data.

When transmission is started, the data in TXB6 is transferred to transmit shift register 6 (TXS6). After that, the data is sequentially output from TXS6 to the TxD6 pin, starting from the LSB. When transmission is completed, a transmission completion interrupt request (INTST6) is generated.

Transmission is stopped until the data to be transmitted next is written to TXB6.

Figure 12-13 shows the timing of the transmission completion interrupt request (INTST6). This interrupt occurs as soon as the last stop bit has been output.

**Figure 12-13. Normal Transmission Completion Interrupt Request Timing**

**1. Stop bit length: 1****2. Stop bit length: 2**

**(d) Continuous transmission**

When transmit shift register 6 (TXS6) has started the shift operation, the next transmit data can be written to transmit buffer register 6 (TXB6). As a result, data can be transmitted without intermission even while an interrupt that has occurred after transmission of one data frame is being serviced, thus an efficient communication rate is realized. To transmit data continuously, however, transmission processing must be executed while referencing bits 1 (TXBF6) and 0 (TXSF6) of asynchronous serial interface transmission status register 6 (ASIF6).

**Caution** When the device is incorporated in to LIN, the continuous transmission function cannot be used. Make sure that asynchronous serial interface transmission status register 6 (ASIF6) is 00H before writing transmit data to transmit buffer register 6 (TXB6).

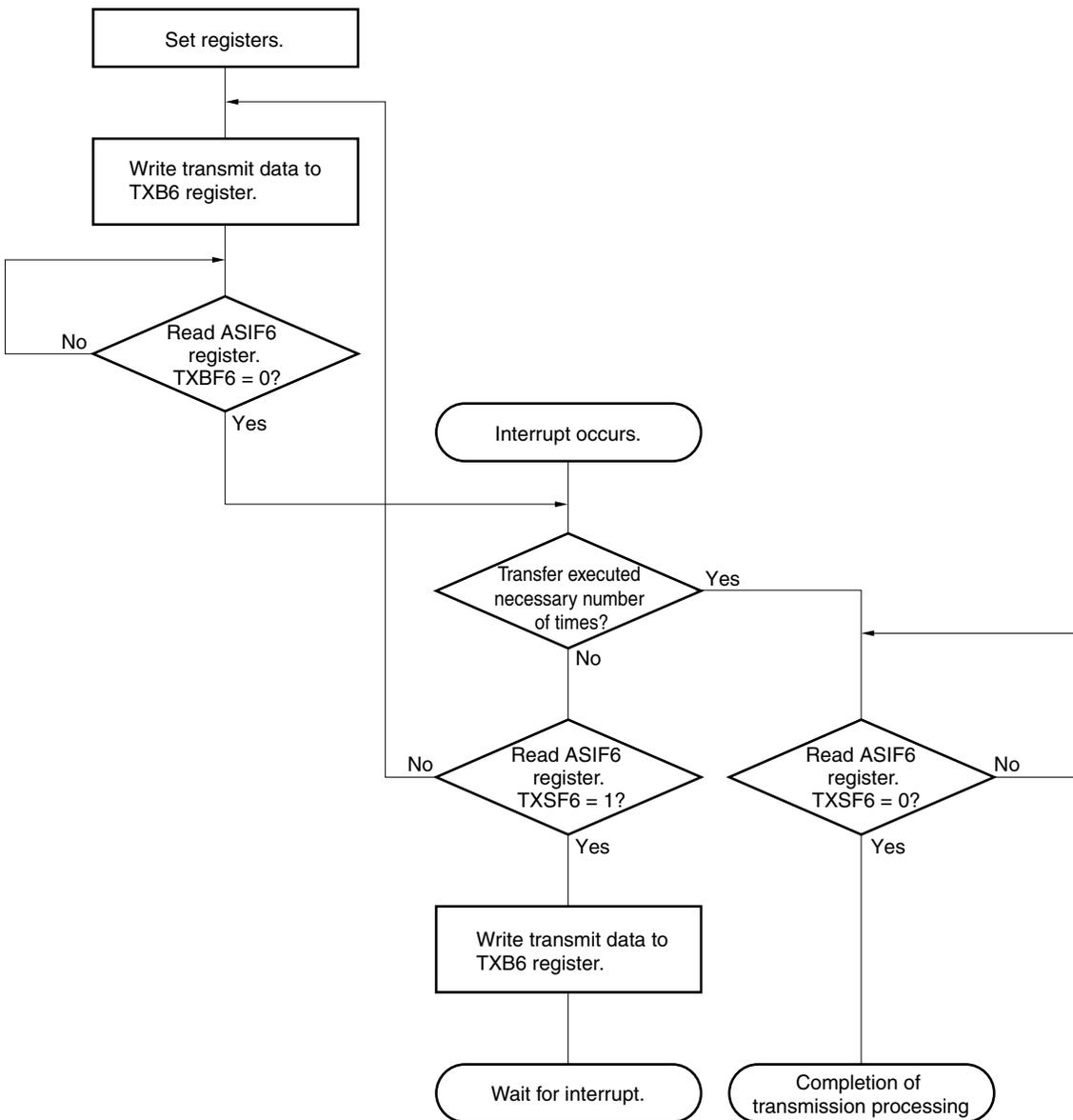
**Table 12-2. Write Processing and Writing to TXB6 During Execution of Continuous Transmission**

TXBF6	TXSF6	Write Processing During Execution of Continuous Transmission	Writing to TXB6 During Execution of Continuous Transmission
0	0	Enables writing 2 bytes or transmission completion processing	Enables writing
0	1	Enables writing 1 byte	Enables writing
1	0	Enables writing 2 bytes or transmission completion processing	Disables writing
1	1	Enables writing 1 byte	Disables writing

- Cautions**
- To continuously transmit data, write the data of the first byte to TXB6, check that the value of the TXBF6 flag is 0, and then write the data of the second byte to TXB6. The operation is not guaranteed if data is written to TXB6 while the TXBF6 flag is 1.
  - While continuous transmission is being executed, check the value of the TXSF6 flag after the transmission completion interrupt to determine the subsequent write processing to TXB6.
    - If TXSF6 is 1: Continuous transmission is in progress. Data of 1 byte can be written.
    - If TXSF6 is 0: Continuous transmission is completed. Data of 2 bytes can be written. To do so, observe Caution 1 above.
  - While continuous transmission is in progress, check that TXSF6 is 0 after the transmission completion interrupt, and then execute clearing (POWER6 = 0 or TXE6 = 0). If clearing is executed while the TXSF6 flag is 1, the transmit data cannot be guaranteed.

Figure 12-14 shows the processing flow of continuous transmission.

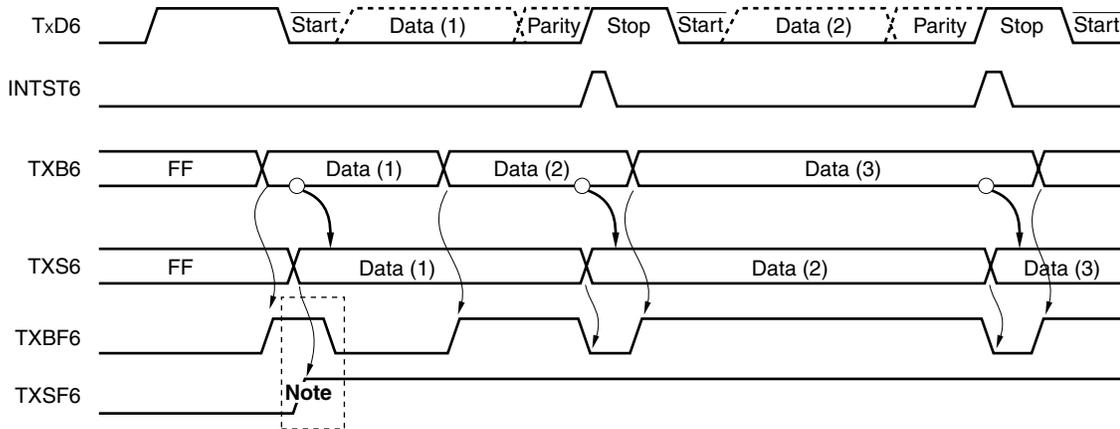
**Figure 12-14. Processing Flow of Continuous Transmission**



**Remark** TXB6: Transmit buffer register 6  
 ASIF6: Asynchronous serial interface transmission status register 6  
 TXBF6: Bit 1 of ASIF6 (transmit buffer data flag)  
 TXSF6: Bit 0 of ASIF6 (transmit shift register data flag)

Figure 12-15 shows the timing of starting continuous transmission, and Figure 12-16 shows the timing of ending continuous transmission.

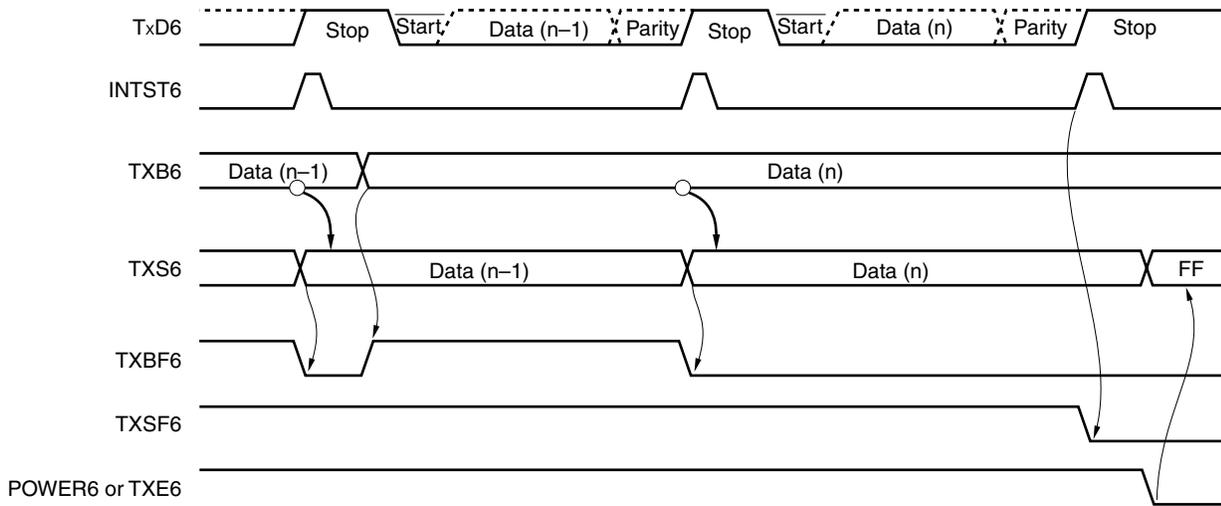
**Figure 12-15. Timing of Starting Continuous Transmission**



**Note** When ASIF6 is read, there is a period in which TXBF6 and TXSF6 = 1, 1. Therefore, judge whether writing is enabled using only the TXBF6 bit.

- Remark**
- TxD6: TxD6 pin (output)
  - INTST6: Interrupt request signal
  - TXB6: Transmit buffer register 6
  - TXS6: Transmit shift register 6
  - ASIF6: Asynchronous serial interface transmission status register 6
  - TXBF6: Bit 1 of ASIF6
  - TXSF6: Bit 0 of ASIF6

**Figure 12-16. Timing of Ending Continuous Transmission**



- Remark**
- TxD6: TxD6 pin (output)
  - INTST6: Interrupt request signal
  - TXB6: Transmit buffer register 6
  - TXS6: Transmit shift register 6
  - ASIF6: Asynchronous serial interface transmission status register 6
  - TXBF6: Bit 1 of ASIF6
  - TXSF6: Bit 0 of ASIF6
  - POWER6: Bit 7 of asynchronous serial interface mode register (ASIM6)
  - TXE6: Bit 6 of asynchronous serial interface mode register (ASIM6)

**(e) Normal reception**

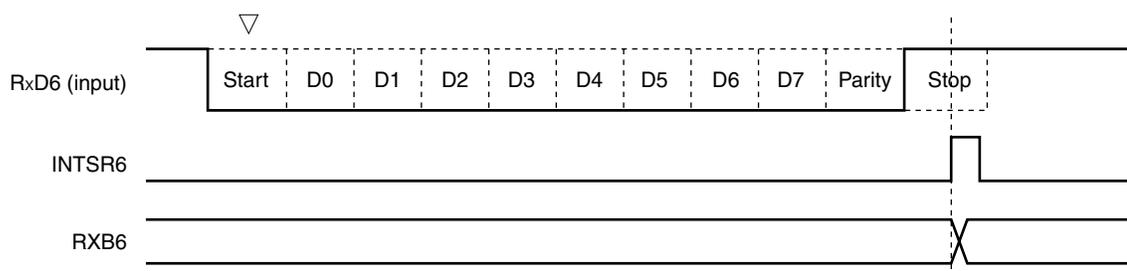
Reception is enabled and the RXD6 pin input is sampled when bit 7 (POWER6) of asynchronous serial interface mode register 6 (ASIM6) is set to 1 and then bit 5 (RXE6) of ASIM6 is set to 1.

The 8-bit counter of the baud rate generator starts counting when the falling edge of the RxD6 pin input is detected. When the set value of baud rate generator control register 6 (BRGC6) has been counted, the RxD6 pin input is sampled again (▽ in Figure 12-17). If the RxD6 pin is low level at this time, it is recognized as a start bit.

When the start bit is detected, reception is started, and serial data is sequentially stored in the receive shift register (RXS6) at the set baud rate. When the stop bit has been received, the reception completion interrupt (INTSR6) is generated and the data of RXS6 is written to receive buffer register 6 (RXB6). If an overrun error (OVE6) occurs, however, the receive data is not written to RXB6.

Even if a parity error (PE6) or a framing error (FE6) occurs while reception is in progress, reception continues to the reception position of the stop bit, and an error interrupt (INTSR6/INTSRE6) is generated on completion of reception.

**Figure 12-17. Reception Completion Interrupt Request Timing**



- Cautions**
1. Be sure to read receive buffer register 6 (RXB6) even if a reception error occurs. Otherwise, an overrun error will occur when the next data is received, and the reception error status will persist.
  2. Reception is always performed with the “number of stop bits = 1”. The second stop bit is ignored.
  3. Be sure to read asynchronous serial interface reception error status register 6 (ASIS6) before reading RXB6.

**(f) Reception error**

Three types of errors may occur during reception: a parity error, framing error, or overrun error. If the error flag of asynchronous serial interface reception error status register 6 (ASIS6) is set as a result of data reception, a reception error interrupt request (INTSR6/INTSRE6) is generated.

Which error has occurred during reception can be identified by reading the contents of ASIS6 in the reception error interrupt servicing (INTSR6/INTSRE6) (refer to **Table 12-3**).

The contents of ASIS6 are reset to 0 when ASIS6 is read.

**Table 12-3. Cause of Reception Error**

Reception Error	Cause	Value of ASIS6
Parity error	The parity specified for transmission does not match the parity of the receive data.	04H
Framing error	Stop bit is not detected.	02H
Overrun error	Reception of the next data is completed before data is read from receive buffer register 6 (RXB6).	01H

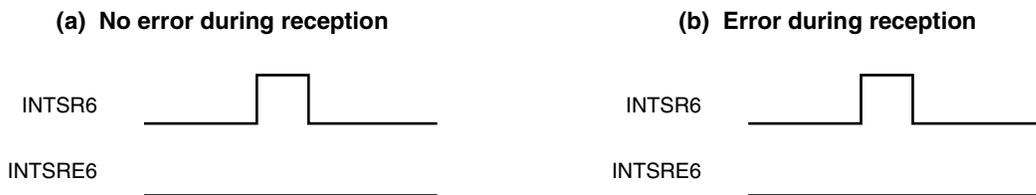
The error interrupt can be separated into INTSR6 and INTSRE6 by clearing bit 0 (ISRM6) of asynchronous serial interface mode register 6 (ASIM6) to 0.

**Figure 12-18. Reception Error Interrupt**

**1. If ISRM6 is cleared to 0 (INTSR6 and INTSRE6 are separated)**



**2. If ISRM6 is set to 1 (error interrupt is included in INTSR6)**



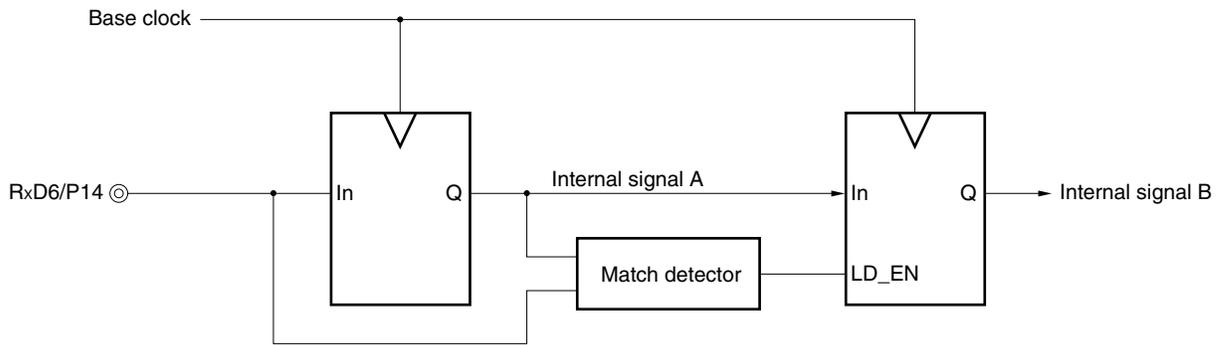
**(g) Noise filter of receive data**

The RXD6 signal is sampled with the base clock output by the prescaler block.

If two sampled values are the same, the output of the match detector changes, and the data is sampled as input data.

Because the circuit is configured as shown in Figure 12-19, the internal processing of the reception operation is delayed by two clocks from the external signal status.

**Figure 12-19. Noise Filter Circuit**



**(h) SBF transmission**

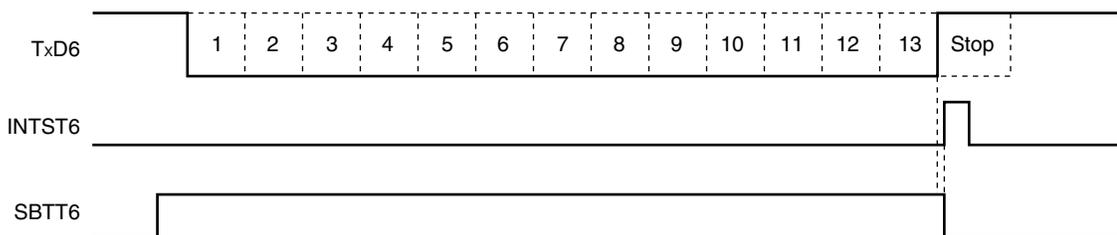
When the device is incorporated in LIN, the SBF (Synchronous Break Field) transmission control function is used for transmission. For the transmission operation of LIN, refer to **Figure 12-1 LIN Transmission Operation**.

The TxD6 pin outputs a high level when bit 7 (POWER6) of asynchronous serial interface mode register 6 (ASIM6) is set to 1. Transmission is enabled when bit 6 (TXE6) of ASIM6 is set to 1 next time, and SBF transmission operation is started when bit 5 (SBTT6) of asynchronous serial interface control register 6 (ASICL6) is set to 1.

After transmission has been started, the low levels of bits 13 to 20 (set by bits 4 to 2 (SBL62 to SBL60) of ASICL6) are output. When SBF transmission has been completed, a transmission completion interrupt request (INTST6) is generated, and SBTT6 is automatically cleared. After SBF transmission has been completed, the normal transmission mode is restored.

Transmission is stopped until the data to be transmitted next is written to transmit buffer register 6 (TXB6) or SBTT6 is set to 1.

**Figure 12-20. SBF Transmission**



- Remark** TxD6: TxD6 pin (output)  
 INTST6: Transmission completion interrupt request  
 SBTT6: Bit 5 of asynchronous serial interface control register 6 (ASICL6)

**(i) SBF reception**

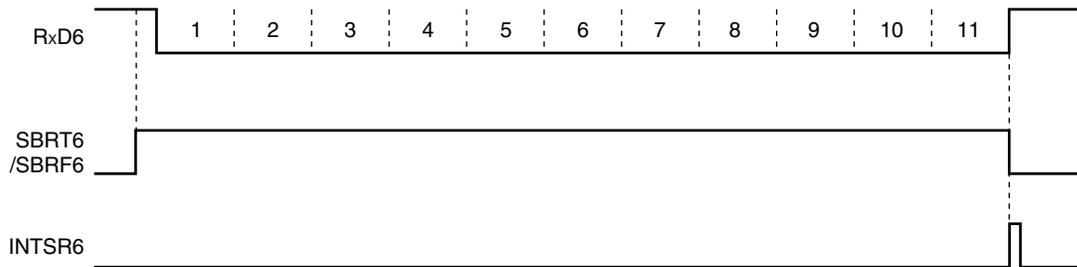
When the device is incorporated in LIN, the SBF (Synchronous Break Field) reception control function is used for reception. For the reception operation of LIN, refer to **Figure 12-2 LIN Reception Operation**.

Reception is enabled when bit 7 (POWER6) of asynchronous serial interface mode register 6 (ASIM6) is set to 1 and then bit 5 (RXE6) of ASIM6 is set to 1. SBF reception is enabled when bit 6 (SBRT6) of asynchronous serial interface control register 6 (ASICL6) is set to 1. In the SBF reception enabled status, the RxD6 pin is sampled and the start bit is detected in the same manner as the normal reception enable status.

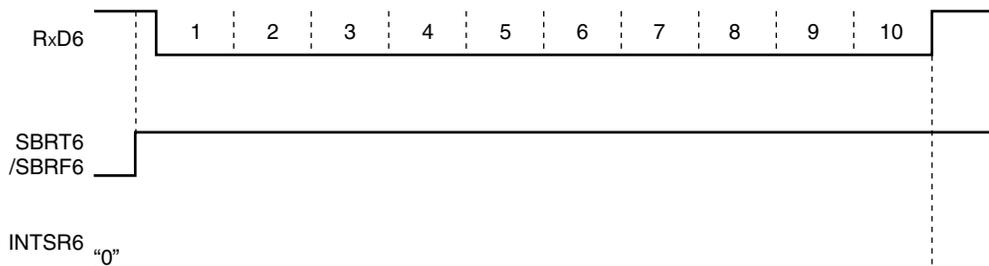
When the start bit has been detected, reception is started, and serial data is sequentially stored in the receive shift register 6 (RXS6) at the set baud rate. When the stop bit is received and if the width of SBF is 11 bits or more, a reception completion interrupt request (INTSR6) is generated as normal processing. At this time, the SBRF6 and SBRT6 bits are automatically cleared, and SBF reception ends. Detection of errors, such as OVE6, PE6, and FE6 (bits 0 to 2 of asynchronous serial interface reception error status register 6 (ASIS6)) is suppressed, and error detection processing of UART communication is not performed. In addition, data transfer between receive shift register 6 (RXS6) and receive buffer register 6 (RXB6) is not performed, and the reset value of FFH is retained. If the width of SBF is 10 bits or less, an interrupt does not occur as error processing after the stop bit has been received, and the SBF reception mode is restored. In this case, the SBRF6 and SBRT6 bits are not cleared.

**Figure 12-21. SBF Reception**

**1. Normal SBF reception (stop bit is detected with a width of more than 10.5 bits)**



**2. SBF reception error (stop bit is detected with a width of 10.5 bits or less)**



**Remark** RxD6: RxD6 pin (input)  
 SBRT6: Bit 6 of asynchronous serial interface control register 6 (ASICL6)  
 SBRF6: Bit 7 of ASICL6  
 INTSR6: Reception completion interrupt request

### 12.4.3 Dedicated baud rate generator

The dedicated baud rate generator consists of a source clock selector and an 8-bit programmable counter, and generates a serial clock for transmission/reception of UART6.

Separate 8-bit counters are provided for transmission and reception.

#### (1) Configuration of baud rate generator

- Base clock (Clock)

The clock selected by bits 3 to 0 (TPS63 to TPS60) of clock selection register 6 (CKSR6) is supplied to each module when bit 7 (POWER6) of the asynchronous serial interface mode register 6 (ASIM6) is 1. This clock is called the base clock (Clock) and its frequency is called  $f_{CLK}$ . Clock is fixed to the low level when POWER6 = 0.

- Transmission counter

This counter stops, cleared to 0, when bit 7 (POWER6) or bit 6 (TXE6) of asynchronous serial interface mode register 6 (ASIM6) is 0.

It starts counting when POWER6 = 1 and TXE6 = 1.

The counter is cleared to 0 when the first data transmitted is written to transmit buffer register 6 (TXB6).

If data are continuously transmitted, the counter is cleared to 0 again when one frame of data has been completely transmitted. If there is no data to be transmitted next, the counter is not cleared to 0 and continues counting until POWER6 or TXE6 is cleared to 0.

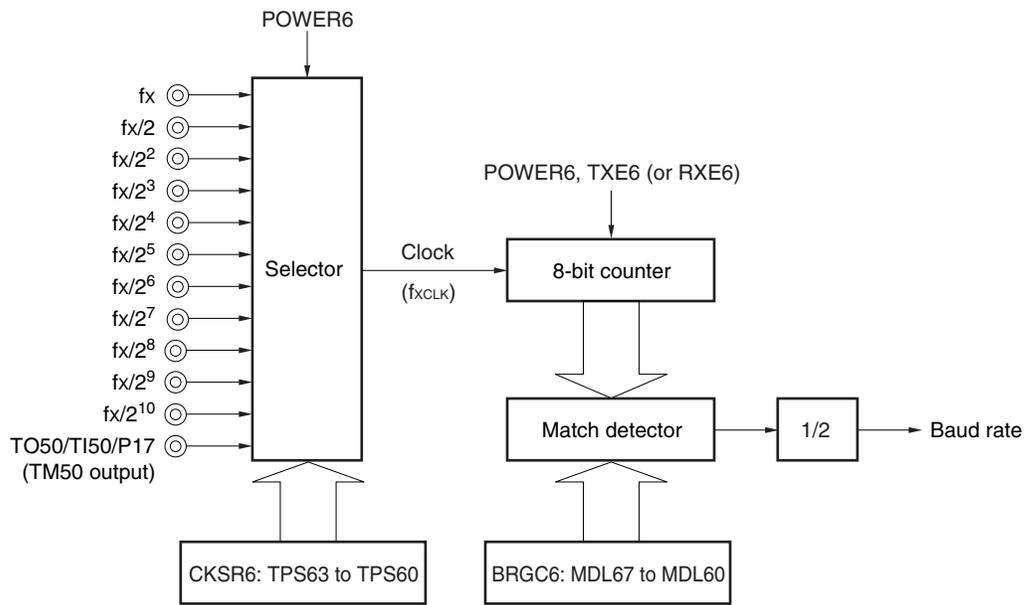
- Reception counter

This counter stops operation, cleared to 0, when bit 7 (POWER6) or bit 5 (RXE6) of asynchronous serial interface mode register 6 (ASIM6) is 0.

It starts counting when the start bit has been detected.

The counter stops operation after one frame has been received, until the next start bit is detected.

Figure 12-22. Configuration of Baud Rate Generator



**Remark**  $POWER6$ : Bit 7 of asynchronous serial interface mode register 6 (ASIM6)

$TXE6$ : Bit 6 of ASIM6

$RXE6$ : Bit 5 of ASIM6

$CKSR6$ : Clock selection register 6

$BRGC6$ : Baud rate generator control register 6

**(2) Generation of serial clock**

A serial clock can be generated by using clock selection register 6 (CKSR6) and baud rate generator control register 6 (BRGC6).

Select the clock to be input to the 8-bit counter by using bits 3 to 0 (TPS63 to TPS60) of CKSR6.

Bits 7 to 0 (MDL67 to MDL60) of BRGC6 can be used to select the division value of the 8-bit counter.

**(a) Clock selection register 6 (CKSR6)**

This register selects the base clock of serial interface UART6.

CKSR6 can be set by an 8-bit memory manipulation instruction.

RESET input clears this register to 00H.

**Remark** CKSR6 can be refreshed (the same value is written) by software during a communication operation (when bit 7 (POWER6) and bit 6 (TXE6) of ASIM6 = 1 or bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 = 1).

Address: FF56H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CKSR6	0	0	0	0	TPS63	TPS62	TPS61	TPS60

TPS63	TPS62	TPS61	TPS60	Base clock (f <sub>CLK</sub> )
0	0	0	0	f <sub>x</sub> (10 MHz)
0	0	0	1	f <sub>x</sub> /2 (5 MHz)
0	0	1	0	f <sub>x</sub> /2 <sup>2</sup> (2.5 MHz)
0	0	1	1	f <sub>x</sub> /2 <sup>3</sup> (1.25 MHz)
0	1	0	0	f <sub>x</sub> /2 <sup>4</sup> (625 kHz)
0	1	0	1	f <sub>x</sub> /2 <sup>5</sup> (312.5 kHz)
0	1	1	0	f <sub>x</sub> /2 <sup>6</sup> (156.25 kHz)
0	1	1	1	f <sub>x</sub> /2 <sup>7</sup> (78.13 kHz)
1	0	0	0	f <sub>x</sub> /2 <sup>8</sup> (39.06 kHz)
1	0	0	1	f <sub>x</sub> /2 <sup>9</sup> (19.53 kHz)
1	0	1	0	f <sub>x</sub> /2 <sup>10</sup> (9.77 kHz)
1	0	1	1	TM50 output
Other				Setting prohibited

**Caution** Make sure POWER6 = 0 when rewriting TPS63 to TPS60.

- Remarks**
1. Figures in parentheses are for operation with f<sub>x</sub> = 10 MHz
  2. f<sub>x</sub>: X1 input clock oscillation frequency

**(b) Baud rate generator control register 6 (BRGC6)**

This register selects the base clock of serial interface UART6.

BRGC6 can be set by an 8-bit memory manipulation instruction.

RESET input sets this register to FFH.

**Remark** BRGC6 can be refreshed (the same value is written) by software during a communication operation (when bit 7 (POWER6) and bit 6 (TXE6) of ASIM6 = 1 or bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 = 1).

Address: FF57H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
BRGC6	MDL67	MDL66	MDL65	MDL64	MDL63	MDL62	MDL61	MDL60

MDL67	MDL66	MDL65	MDL64	MDL63	MDL62	MDL61	MDL60	k	Output clock selection of 8-bit counter
0	0	0	0	0	×	×	×	×	Setting prohibited
0	0	0	0	1	0	0	0	8	$f_{XCLK}/8$
0	0	0	0	1	0	0	1	9	$f_{XCLK}/9$
0	0	0	0	1	0	1	0	10	$f_{XCLK}/10$
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
1	1	1	1	1	1	0	0	252	$f_{XCLK}/252$
1	1	1	1	1	1	0	1	253	$f_{XCLK}/253$
1	1	1	1	1	1	1	0	254	$f_{XCLK}/254$
1	1	1	1	1	1	1	1	255	$f_{XCLK}/255$

- Cautions**
1. Make sure that bit 6 (TXE6) and bit 5 (RXE6) of the ASIM6 register = 0 when rewriting the MDL67 to MDL60 bits.
  2. The baud rate is the output clock of the 8-bit counter divided by 2.

- Remarks**
1.  $f_{XCLK}$ : Frequency of base clock (Clock) selected by the TPS63 to TPS60 bits of CKSR6 register
  2. k: Value set by MDL67 to MDL60 bits (k = 8, 9, 10, ..., 255)
  3. ×: Don't care

**(c) Baud rate**

The baud rate can be calculated by the following expression.

- Baud rate =  $\frac{f_{\text{CLK}}}{2 \times k}$  [bps]

$f_{\text{CLK}}$ : Frequency of base clock (Clock) selected by TPS63 to TPS60 bits of CKSR6 register

k: Value set by MDL67 to MDL60 bits of BRGC6 register (k = 8, 9, 10, ..., 255)

**(d) Error of baud rate**

The baud rate error can be calculated by the following expression.

- Error (%) =  $\left( \frac{\text{Actual baud rate (baud rate with error)}}{\text{Desired baud rate (correct baud rate)}} - 1 \right) \times 100$  [%]

**Cautions 1. Keep the baud rate error during transmission to within the permissible error range at the reception destination.**

**2. Make sure that the baud rate error during reception satisfies the range shown in (4) Permissible baud rate range during reception.**

**Example:** Frequency of base clock (Clock) = 20 MHz = 20,000,000 Hz  
 Set value of MDL67 to MDL60 bits of BRGC6 register = 01000001B (k = 65)  
 Target baud rate = 153600 bps

$$\begin{aligned} \text{Baud rate} &= 20 \text{ M} / (2 \times 65) \\ &= 20000000 / (2 \times 65) = 153,846 \text{ [bps]} \end{aligned}$$

$$\begin{aligned} \text{Error} &= (153846 / 153600 - 1) \times 100 \\ &= 0.160 \text{ [%]} \end{aligned}$$

## (3) Example of setting baud rate

Table 12-4. Set Data of Baud Rate Generator

Baud Rate [bps]	fx = 10.0 MHz				fx = 8.38 MHz				fx = 4.19 MHz			
	TPS63 to TPS60	k	Calculated value	ERR[%]	TPS63 to TPS60	k	Calculated value	ERR[%]	TPS63 to TPS60	k	Calculated value	ERR[%]
600	6H	130	601	0.16	6H	109	601	0.11	5H	109	601	0.11
1200	5H	130	1202	0.16	5H	109	1201	0.11	4H	109	1201	0.11
2400	4H	130	2404	0.16	4H	109	2403	0.11	3H	109	2403	0.11
4800	3H	130	4808	0.16	3H	109	4805	0.11	2H	109	4805	0.11
9600	2H	130	9615	0.16	2H	109	9610	0.11	1H	109	9610	0.11
10400	2H	120	10417	0.16	2H	101	10371	0.28	1H	101	10475	-0.28
19200	1H	130	19231	0.16	1H	109	19200	0.11	0H	109	19220	0.11
31250	1H	80	31250	0.00	0H	134	31268	0.06	0H	67	31268	0.06
38400	0H	130	38462	0.16	0H	109	38440	0.11	0H	55	38090	-0.80
76800	0H	65	76923	0.16	0H	55	76182	-0.80	0H	27	77593	1.03
115200	0H	43	116279	0.94	0H	36	116388	1.03	0H	18	116389	1.03
153600	0H	33	151515	-1.36	0H	27	155185	1.03	0H	14	149643	-2.58
230400	0H	22	227272	-1.36	0H	18	232777	1.03	0H	9	232778	1.03

**Caution** The maximum permissible frequency ( $f_{xCLK}$ ) of the base clock is 25 MHz.

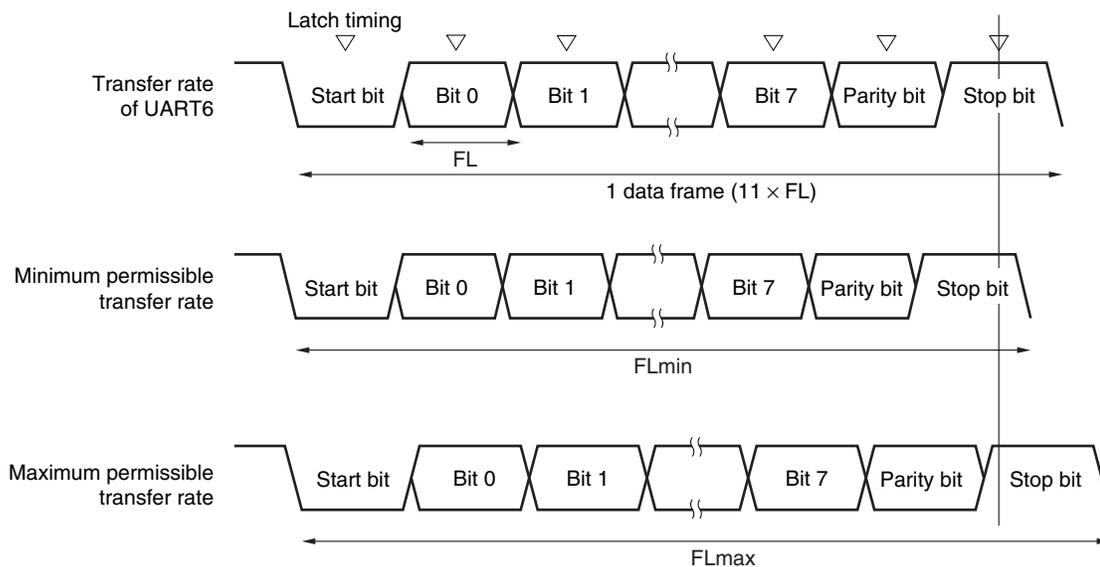
**Remark** TPS63 to TPS60: Bits 3 to 0 of clock selection register 6 (CKSR6) (setting of base clock ( $f_{xCLK}$ ))  
k: Value set by MDL67 to MDL60 bits of baud rate generator control register 6 (BRGC6) ( $k = 8, 9, 10, \dots, 255$ )  
fx: X1 input clock oscillation frequency  
ERR: Baud rate error

**(4) Permissible baud rate range during reception**

The permissible error from the baud rate at the transmission destination during reception is shown below.

**Caution** Make sure that the baud rate error during reception is within the permissible error range, by using the calculation expression shown below.

**Figure 12-23. Permissible Baud Rate Range During Reception**



As shown in Figure 12-23, the latch timing of the receive data is determined by the counter set by baud rate generator control register 6 (BRGC6) after the start bit has been detected. If the last data (stop bit) meets this latch timing, the data can be correctly received.

Assuming that 11-bit data is received, the theoretical values can be calculated as follows.

$$FL = (\text{Brate})^{-1}$$

- Brate: Baud rate of UART6
- k: Set value of BRGC6
- FL: 1-bit data length
- Margin of latch timing: 2 clocks

$$\text{Minimum permissible transfer rate: } FL_{\min} = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$$

Therefore, the maximum receivable baud rate at the transmission destination is as follows.

$$BR_{max} = (F_{min}/11)^{-1} = \frac{22k}{21k + 2} \text{ Brate}$$

Similarly, the maximum permissible transfer rate can be calculated as follows.

$$\frac{10}{11} \times FL_{max} = 11 \times FL - \frac{k + 2}{2 \times k} \times FL = \frac{21k - 2}{2 \times k} FL$$

$$FL_{max} = \frac{21k \times 2}{20k} FL \times 11$$

Therefore, the minimum receivable baud rate at the transmission destination is as follows.

$$BR_{min} = (F_{Lmax}/11)^{-1} = \frac{20k}{21k - 2} \text{ Brate}$$

The permissible baud rate error between UART6 and the transmission destination can be calculated from the above minimum and maximum baud rate expressions, as follows.

**Table 12-5. Maximum/Minimum Permissible Baud Rate Error**

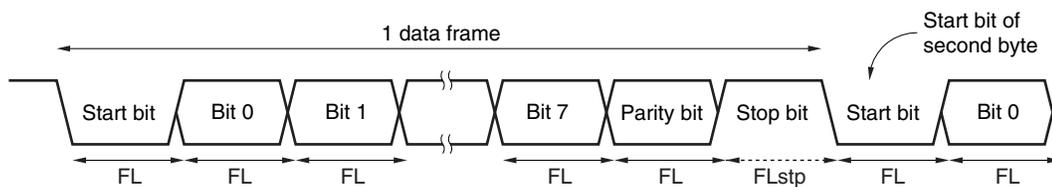
Division Ratio (k)	Maximum Permissible Baud Rate Error	Minimum Permissible Baud Rate Error
8	+3.53%	-3.61%
20	+4.26%	-4.31%
50	+4.56%	-4.58%
100	+4.66%	-4.67%
255	+4.72%	-4.73%

- Remarks**
1. The accuracy of reception depends on the number of bits in one frame, input clock frequency, and division ratio (k). The higher the input clock frequency and the higher the division ratio (k), the higher the accuracy.
  2. k: Set value of BRGC6

**(5) Transfer rate during continuous transmission**

When data is continuously transmitted, the transfer rate from a stop bit to the next start bit is extended by two clocks from the normal value. However, the result of transfer is not affected because the timing is initialized on the reception side when the start bit is detected.

**Figure 12-24. Transfer Rate During Continuous Transmission**



Where the 1-bit data length is  $FL$ , the stop bit length is  $FL_{stp}$ , and base clock frequency is  $f_{CLK}$ , the following expression is satisfied.

$$FL_{stp} = FL + 2/f_{CLK}$$

Therefore, the transfer rate during continuous transmission is:

$$\text{Transfer rate} = 11 \times FL + 2/f_{CLK}$$

## CHAPTER 13 SERIAL INTERFACE CSI10

### 13.1 Functions of Serial Interface CSI10

Serial interface CSI10 has the following two modes.

- Operation stop mode
- 3-wire serial I/O mode

#### (1) Operation stop mode

This mode is used when serial transfer is not performed and can enable a reduction in the power consumption.

#### (2) 3-wire serial I/O mode (MSB/LSB-first selectable)

This mode is used to transfer 8-bit data using three lines: a serial clock line ( $\overline{\text{SCK10}}$ ) and two serial data lines (SI10 and SO10).

The processing time of data transfer can be shortened in the 3-wire serial I/O mode because transmission and reception can be simultaneously executed.

In addition, whether 8-bit data is transferred with the MSB or LSB first can be specified, so this interface can be connected to any device.

The 3-wire serial I/O mode is useful for connecting peripheral I/Os and display controllers having a conventional clocked serial interface, such as the 75XL Series, 78K Series, and 17K Series.

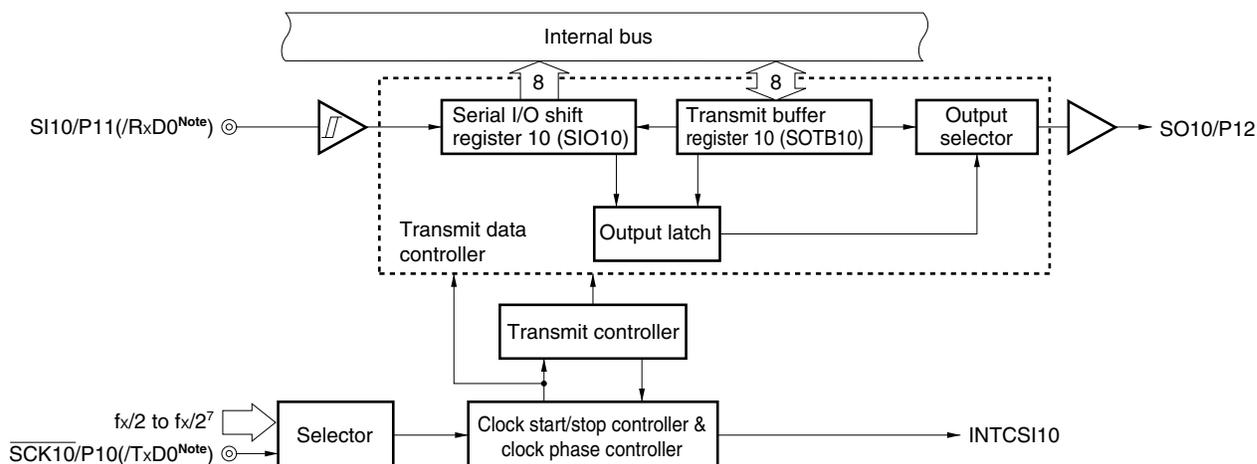
### 13.2 Configuration of Serial Interface CSI10

Serial interface CSI10 consists of the following hardware.

**Table 13-1. Configuration of Serial Interface CSI10**

Item	Configuration
Registers	Transmit buffer register 10 (SOTB10) Serial I/O shift register 10 (SIO10)
Control registers	Serial operation mode register 10 (CSIM10) Serial clock selection register 10 (CSIC10)

Figure 13-1. Block Diagram of Serial Interface CSI10



**Note**  $\mu$ PD780102, 780103, and 78F0103 only

#### (1) Transmit buffer register 10 (SOTB10)

This register sets the transmit data.

Transmission/reception is started by writing data to SOTB10 when bit 6 (TRMD10) of serial operation mode register 10 (CSIM10) is 1.

The data written to SOTB10 is converted from parallel data into serial data by serial I/O shift register 10, and output to the serial output pin (SO10).

SOTB10 can be written or read by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$  input makes this register undefined.

**Caution** Do not access SOTB10 when CSOT10 = 1 (during serial communication).

#### (2) Serial I/O shift register 10 (SIO10)

This is an 8-bit register that converts data from parallel data into serial data or vice versa.

This register can be read by an 8-bit memory manipulation instruction.

Reception is started by reading data from SIO10 if bit 6 (TRMD10) of serial operation mode register 10 (CSIM10) is 0.

During reception, the data is read from the serial input pin (SI10) to SIO10.

★  $\overline{\text{RESET}}$  input clears this register to 00H.

**Caution** Do not access SIO10 when CSOT10 = 1 (during serial communication).

### 13.3 Registers Controlling Serial Interface CSI10

The serial interface CSI10 is controlled by the following two registers.

- Serial operation mode register 10 (CSIM10)
- Serial clock selection register 10 (CSIC10)

#### (1) Serial operation mode register 10 (CSIM10)

This register is used to select the operation mode and enable or disable operation.

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

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

**Figure 13-2. Format of Serial Operation Mode Register 10 (CSIM10)**

Address: FF80H After reset: 00H R/W<sup>Note 1</sup>

Symbol	<span style="border: 1px solid black; padding: 0 2px;">7</span>	6	5	4	3	2	1	0
CSIM10	CSIE10	TRMD10	0	DIR10	0	0	0	CSOT10

CSIE10	Operation control in 3-wire serial I/O mode
0	Stops operation (SI10/P11(/RxD0 <sup>Note 2</sup> ), SO10/P12, and SCK10/P10(/TxD0 <sup>Note 2</sup> ) pins can be used as general-purpose port pins).
1	Enables operation (SI10/P11(/RxD0 <sup>Note 2</sup> ), SO10/P12, and $\overline{\text{SCK10}}$ /P10(/TxD0 <sup>Note 2</sup> ) pins are at active level).

TRMD10 <sup>Note 3</sup>	Transmit/receive mode control
0 <sup>Note 4</sup>	Receive mode (transmission disabled).
1	Transmit/receive mode

DIR10 <sup>Note 5</sup>	First bit specification
0	MSB
1	LSB

CSOT10 <sup>Note 6</sup>	Operation mode flag
0	Communication is stopped.
1	Communication is in progress.

- Notes**
1. Bit 0 is a read-only bit.
  2.  $\mu$ PD780102, 780103, and 78F0103 only
  3. Do not rewrite TRMD10 when CSOT10 = 1 (during serial communication).
  4. The SO10 pin is fixed to the low level when TRMD10 is 0. Reception is started when data is read from SIO10.
  5. Do not rewrite DIR10 when CSOT10 = 1 (during serial communication).
  6. CSOT10 is cleared if CSIE10 is set to 0 (operation stopped).

**Caution** Be sure to set bit 5 to 0.

**(2) Serial clock selection register 10 (CSIC10)**

This register is used to select the phase of the data clock and set the count clock.

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

RESET input sets this register to 00H.

**Figure 13-3. Format of Serial Clock Selection Register 10 (CSIC10)**

Address: FF81H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIC10	0	0	0	CKP10	DAP10	CKS102	CKS101	CKS100

CKP10	DAP10	Data clock phase selection	Type
0	0		1
0	1		2
1	0		3
1	1		4

CKS102	CKS101	CKS100	CSI10 count clock selection
0	0	0	$f_x/2$ (5 MHz)
0	0	1	$f_x/2^2$ (2.5 MHz)
0	1	0	$f_x/2^3$ (1.25 MHz)
0	1	1	$f_x/2^4$ (625 kHz)
1	0	0	$f_x/2^5$ (312.5 kHz)
1	0	1	$f_x/2^6$ (156.25 kHz)
1	1	0	$f_x/2^7$ (78.13 kHz)
1	1	1	External clock

- ★ **Cautions**
1. Do not write to CSIC10 during communication or when P10/SCK10 (Tx/D0<sup>Note</sup>), P11/SI10 (/Rx/D0<sup>Note</sup>), and P12/SO10 are used as general-purpose ports.
  2. The phase type of the data clock is type 1 after reset.

**Note**  $\mu$ PD780102, 780103, and 78F0103 only.

- Remarks**
1. Figures in parentheses are for operation with  $f_x = 10$  MHz
  2.  $f_x$ : X1 input clock oscillation frequency

## 13.4 Operation of Serial Interface CSI10

Serial interface CSI10 can be used in the following two modes.

- Operation stop mode
- 3-wire serial I/O mode

### 13.4.1 Operation stop mode

Serial transfer is not executed in this mode. Therefore, the power consumption can be reduced. In addition, the P10/ $\overline{\text{SCK10}}$ (/TxD0<sup>Note</sup>), P11/SI10(/RxD0<sup>Note</sup>), and P12/SO10 pins can be used as ordinary I/O port pins in this mode.

**Note**  $\mu$ PD780102, 780103, and 78F0103 only

#### (1) Register setting

The operation stop mode is set by serial operation mode register 10 (CSIM10).

##### (a) Serial operation mode register 10 (CSIM10)

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

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

Address: FF80H After reset: 00H R/W

Symbol	<span style="border: 1px solid black; padding: 0 2px;">7</span>	6	5	4	3	2	1	0
CSIM10	CSIE10	TRMD10	0	DIR10	0	0	0	CSOT10

CSIE10	Operation control in 3-wire serial I/O mode
0	Stops operation (SI10/P11(/RxD0 <sup>Note</sup> ), SO10/P12, and $\overline{\text{SCK10}}$ /P10(/TxD0 <sup>Note</sup> ) pins can be used as general-purpose port pins).
1	Enables operation (SI10/P11(/RxD0 <sup>Note</sup> ), SO10/P12, and $\overline{\text{SCK10}}$ /P10(/TxD0 <sup>Note</sup> ) pins are at active level).

**Note**  $\mu$ PD780102, 780103, and 78F0103 only

### 13.4.2 3-wire serial I/O mode

The 3-wire serial I/O mode is useful for connecting peripheral I/Os and display controllers having a conventional clocked serial interface, such as the 75XL Series, 78K Series, and 17K Series.

In this mode, communication is executed by using three lines: serial clock ( $\overline{\text{SCK10}}$ ), serial output (SO10), and serial input (SI10) lines.

#### (1) Register setting

The 3-wire serial I/O mode is set by serial operation mode register 10 (CSIM10) and serial clock selection register10 (CSIC10).

**(a) Serial operation mode register 10 (CSIM10)**

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

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

Address: FF80H After reset: 00H R/W<sup>Note 1</sup>

Symbol	<input checked="" type="checkbox"/>	6	5	4	3	2	1	0
CSIM10	CSIE10	TRMD10	0	DIR10	0	0	0	CSOT10

CSIE10	Operation control in 3-wire serial I/O mode
0	Stops operation (SI10/P11(/RxD0 <sup>Note 2</sup> ), SO10/P12, and $\overline{\text{SCK10}}$ /P10(/TxD0 <sup>Note 2</sup> ) pins can be used as general-purpose port pins).
1	Enables operation (SI10/P11(/RxD0 <sup>Note 2</sup> ), SO10/P12, and $\overline{\text{SCK10}}$ /P10(/TxD0 <sup>Note 2</sup> ) pins are at active level).

TRMD10 <sup>Note 3</sup>	Transmit/receive mode control
0 <sup>Note 4</sup>	Receive mode (transmission disabled).
1	Transmit/receive mode

DIR10 <sup>Note 5</sup>	First bit specification
0	MSB
1	LSB

CSOT10 <sup>Note 6</sup>	Operation mode flag
0	Communication is stopped.
1	Communication is in progress.

- Notes**
1. Bit 0 is a read-only bit.
  2.  $\mu\text{PD780102}$ ,  $780103$ , and  $78F0103$  only
  3. Do not rewrite TRMD10 when CSOT10 = 1 (during serial communication).
  4. The SO10 pin is fixed to the low level when TRMD10 is 0. Reception is started when data is read from SIO10.
  5. Do not rewrite DIR10 when CSOT10 = 1 (during serial communication).
  6. CSOT10 is cleared if CSIE10 is cleared to 0 (operation stopped).

**Caution** Be sure to set bit 5 to 0.

**(b) Serial clock selection register 10 (CSIC10)**

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

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

Address: FF81H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIC10	0	0	0	CKP10	DAP10	CKS102	CKS101	CKS100

CKP10	DAP10	Data clock phase selection	Type
0	0		1
0	1		2
1	0		3
1	1		4

CKS102	CKS101	CKS100	CSI10 count clock selection
0	0	0	$f_x/2$ (5 MHz)
0	0	1	$f_x/2^2$ (2.5 MHz)
0	1	0	$f_x/2^3$ (1.25 MHz)
0	1	1	$f_x/2^4$ (625 kHz)
1	0	0	$f_x/2^5$ (312.5 kHz)
1	0	1	$f_x/2^6$ (156.25 kHz)
1	1	0	$f_x/2^7$ (78.13 kHz)
1	1	1	External clock

- ★ **Cautions**
  1. Do not write to CSIC10 during communication or when P10/ $\overline{\text{SCK10}}$  (/TxD0<sup>Note5</sup>), P11/SI10 (/RxD0<sup>Note5</sup>), and P12/SO10 are used as general-purpose ports.
  2. The phase type of the data clock is type 1 after reset.

**Note**  $\mu$ PD780102, 780103, and 78F0103 only.

- Remarks**
  1. Figures in parentheses are for operation with  $f_x = 10$  MHz
  2.  $f_x$ : X1 input clock oscillation frequency

**(2) Setting of ports****<1> Transmit/receive mode****(a) To use externally input clock as system clock ( $\overline{\text{SCK10}}$ )**

- Bit 1 (PM11) of port mode register 1: Set to 1
- Bit 2 (PM12) of port mode register 1: Cleared to 0
- Bit 0 (PM10) of port mode register 1: Set to 1
- Bit 2 (P12) of port 1: Cleared to 0

**(b) To use internal clock as system clock ( $\overline{\text{SCK10}}$ )**

- Bit 1 (PM11) of port mode register 1: Set to 1
- Bit 2 (PM12) of port mode register 1: Cleared to 0
- Bit 0 (PM10) of port mode register 1: Cleared to 0
- Bit 2 (P12) of port 1: Cleared to 0
- Bit 0 (P10) of port 1: Set to 1

**<2> Receive mode (with transmission disabled)****(a) To use externally input clock as system clock ( $\overline{\text{SCK10}}$ )**

- Bit 1 (PM11) of port mode register 1: Set to 1
- Bit 0 (PM10) of port mode register 1: Set to 1

**(b) To use internal clock as system clock ( $\overline{\text{SCK10}}$ )**

- Bit 1 (PM11) of port mode register 1: Set to 1
- Bit 0 (PM10) of port mode register 1: Cleared to 0
- Bit 0 (P10) of port 1: Set to 1

**Remark** The transmit/receive mode or receive mode is selected by using bit 6 (TRMD10) of serial operation mode register 10 (CSIM10).

**(3) Communication operation**

In the 3-wire serial I/O mode, data is transmitted or received in 8-bit units. Each bit of the data is transmitted or received in synchronization with the serial clock.

Data can be transmitted or received if bit 6 (TRMD10) of serial operation mode register 10 (CSIM10) is 1. Transmission/reception is started when a value is written to transmit buffer register 10 (SOTB10). In addition, data can be received when bit 6 (TRMD10) of serial operation mode register 10 (CSIM10) is 0.

Reception is started when data is read from serial I/O shift register 10 (SIO10).

After communication has been started, bit 0 (CSOT10) of CSIM10 is set to 1. When communication of 8-bit data has been completed, a communication completion interrupt request flag (CSIIF10) is set, and CSOT10 is cleared to 0. Then the next communication is enabled.

**Caution** Do not access the control register and data register when CSOT10 = 1 (during serial communication).

**Figure 13-4. Timing in 3-Wire Serial I/O Mode (1/2)**

**(1) Transmission/reception timing (Type 1; TRMD10 = 1, DIR10 = 0, CKP10 = 0, DAP10 = 0)**

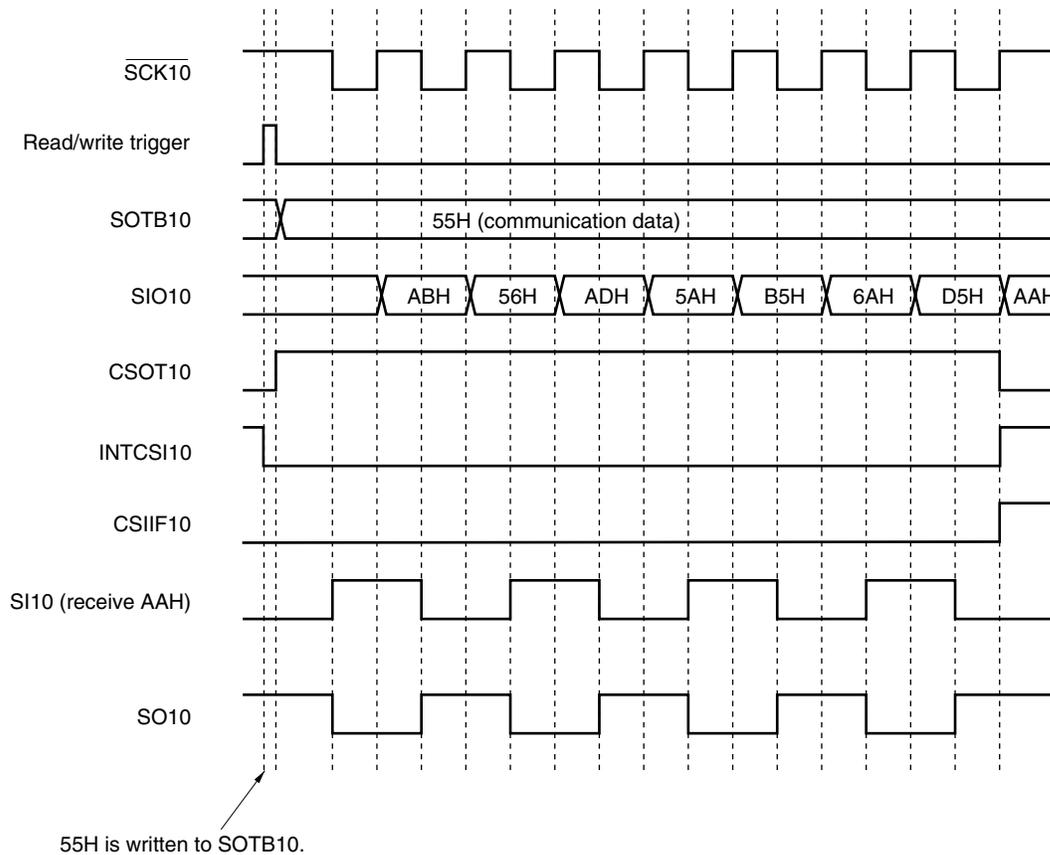


Figure 13-4. Timing in 3-Wire Serial I/O Mode (2/2)

(2) Transmission/reception timing (Type 2; TRMD10 = 1, DIR10 = 0, CKP10 = 0, DAP10 = 1)

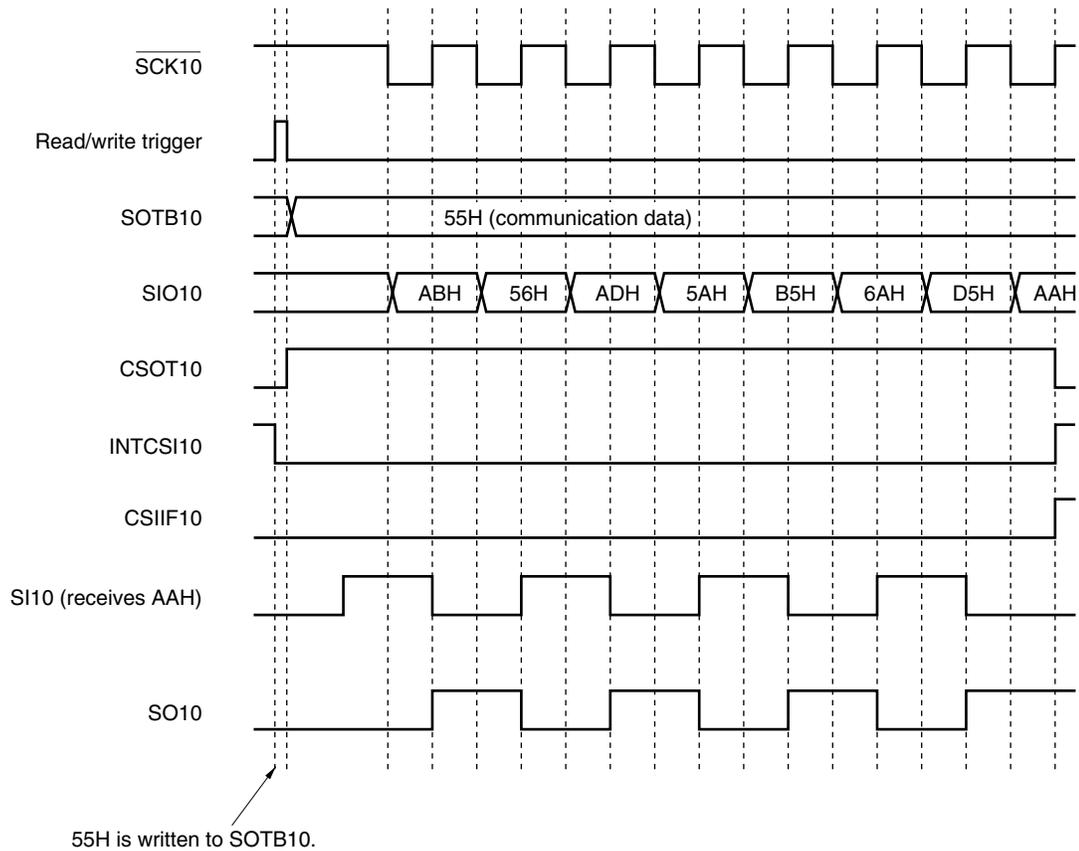
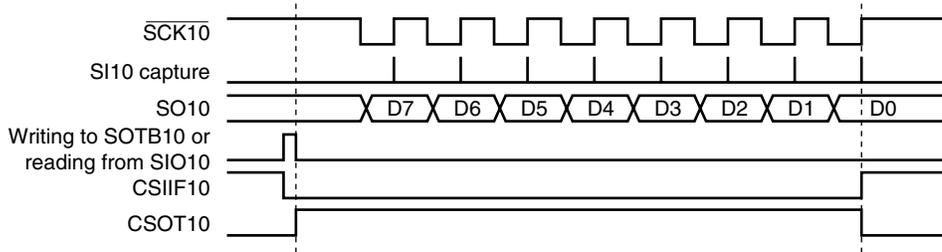
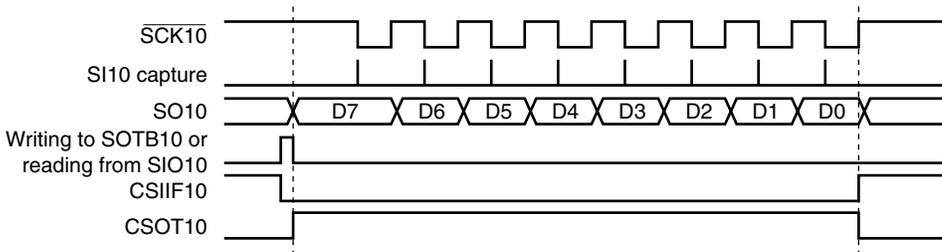


Figure 13-5. Timing of Clock/Data Phase

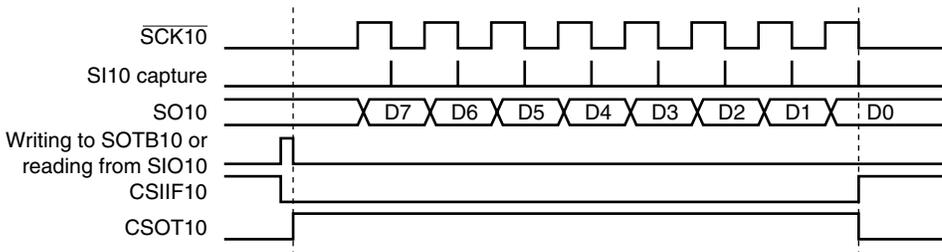
(a) Type 1; CKP10 = 0, DAP10 = 0



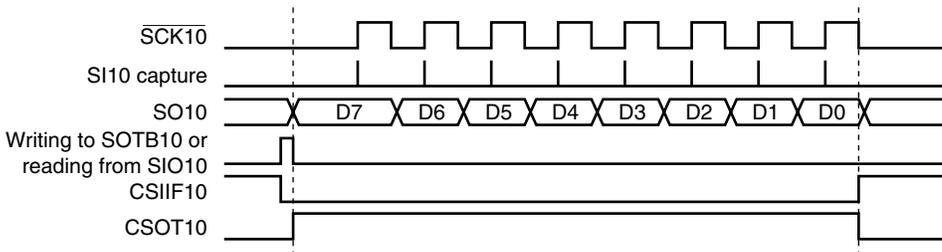
(b) Type 2; CKP10 = 0, DAP10 = 1



(c) Type 3; CKP10 = 1, DAP10 = 0

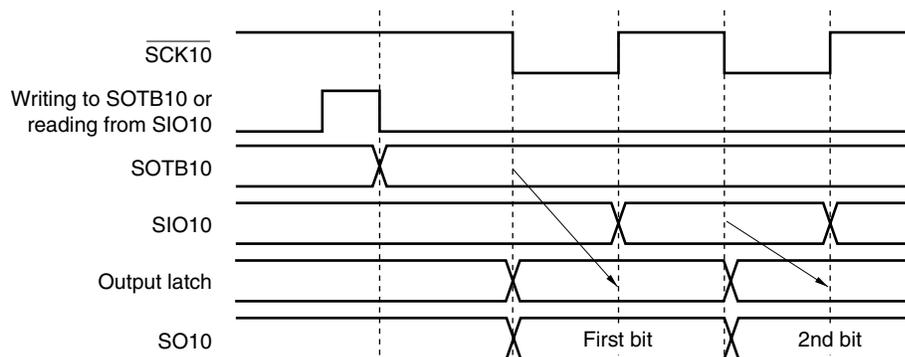


(d) Type 4; CKP10 = 1, DAP10 = 1



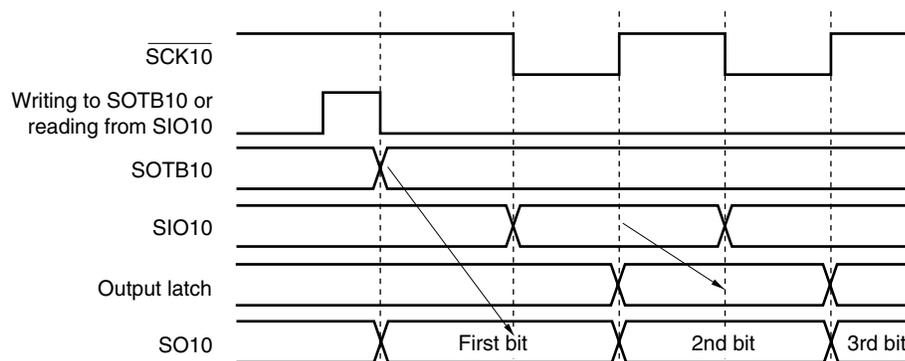
**(4) Timing of output to SO10 pin (first bit)**

When communication is started, the value of transmit buffer register 10 (SOTB10) is output from the SO10 pin. The output operation of the first bit at this time is described below.

**Figure 13-6. Output Operation of First Bit****(1) When CKP10 = 0, DAP10 = 0 (or CKP10 = 1, DAP10 = 0)**

The first bit is directly latched by the SOTB10 register to the output latch at the falling (or rising) edge of the  $\overline{\text{SCK10}}$ , and output from the SO10 pin via an output selector. Then, the value of the SOTB10 register is transferred to the SIO10 register at the next rising (or falling) edge of  $\overline{\text{SCK10}}$ , and shifted one bit. At the same time, the first bit of the receive data is stored in the SIO10 register via the SI10 pin.

The second and subsequent bits are latched by the SIO10 register to the output latch at the next falling (or rising) edge of  $\overline{\text{SCK10}}$ , and the data is output from the SO10 pin.

**(2) When CKP10 = 0, DAP10 = 1 (or CKP10 = 1, DAP10 = 1)**

The first bit is directly latched by the SOTB10 register at the falling edge of the write signal of the SOTB10 register or the read signal of the SIO10 register, and output from the SO10 pin via an output selector. Then, the value of the SOTB10 register is transferred to the SIO10 register at the next rising (or falling) edge of  $\overline{\text{SCK10}}$ , and shifted one bit. At the same time, the first bit of the receive data is stored in the SIO10 register via the SI10 pin.

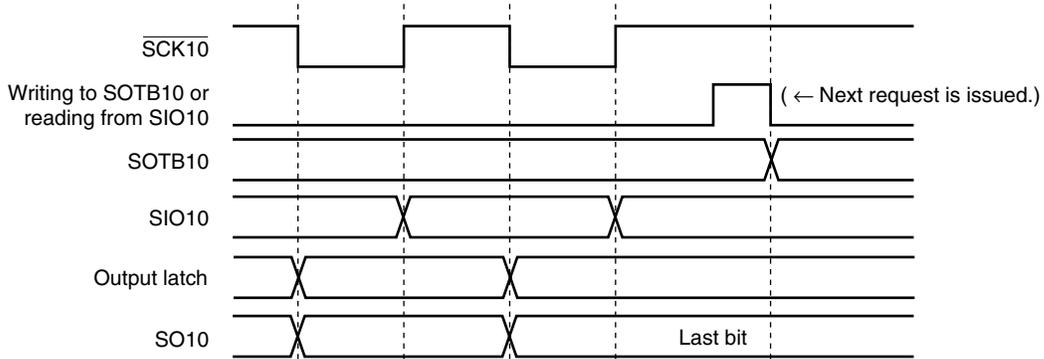
The second and subsequent bits are latched by the SIO10 register to the output latch at the next falling (or rising) edge of  $\overline{\text{SCK10}}$ , and the data is output from the SO10 pin.

**(5) Output value of SO10 pin (last bit)**

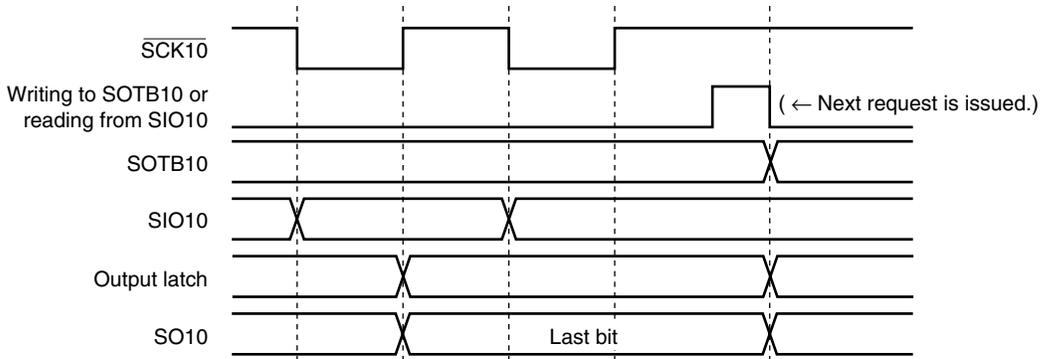
After communication has been completed, the SO10 pin holds the output value of the last bit.

**Figure 13-7. Output Value of SO10 Pin (Last Bit)**

**(1) Type 1; when CKP10 = 0 and DAP10 = 0 (or CKP10 = 1, DAP10 = 0)**



**(2) Type 2; when CKP10 = 0 and DAP10 = 1 (or CKP10 = 1, DAP10 = 1)**



**(6) SO10 pin**

The status of the SO10 pin is as follows if bit 7 (CSIE10) of serial operation mode register 10 (CSIM10) is cleared to 0.

**Table 13-2. SO10 Pin Status**

TRMD10	DAP10	DIR10	SO10 Pin
TRMD10 = 0 <sup>Note</sup>	–	–	Outputs low level <sup>Note</sup> .
TRMD10 = 1	DAP10 = 0	–	Value of SO10 latch (low-level output)
	DAP10 = 1	DIR10 = 0	Value of bit 7 of SOTB10
		DIR10 = 1	Value of bit 0 of SOTB10

**Note** Status after reset

**Caution** If a value is written to TRMD10, DAP10, and DIR10, the output value of the SO10 pin changes.

## CHAPTER 14 INTERRUPT FUNCTIONS

### 14.1 Interrupt Function Types

The following two types of interrupt functions are used.

#### (1) Maskable interrupts

These interrupts undergo mask control. Maskable interrupts can be divided into a high interrupt priority group and a low interrupt priority group by setting the priority specification flag registers (PR0L, PR0H, PR1L).

Multiple interrupt servicing of high-priority interrupts can be applied to low priority interrupts. If two or more interrupts with the same priority are simultaneously generated, each interrupt is serviced according to its predetermined priority (see **Table 14-1**).

A standby release signal is generated.

Six external interrupt requests and 12 internal interrupt requests are provided as maskable interrupts.

#### (2) Software interrupt

This is a vectored interrupt generated by executing the BRK instruction. It is acknowledged even when interrupts are disabled. The software interrupt does not undergo interrupt priority control.

### 14.2 Interrupt Sources and Configuration

A total of 19 interrupt sources exist for maskable and software interrupts. In addition, maximum total of 5 reset sources are also provided (see **Table 14-1**).

Table 14-1. Interrupt Source List

Interrupt Type	Default Priority <sup>Note 1</sup>	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type <sup>Note 2</sup>
		Name	Trigger			
Maskable	0	INTLVI	Low-voltage detection <sup>Note 3</sup>	Internal	0004H	(A)
	1	INTP0	Pin input edge detection	External	0006H	(B)
	2	INTP1			0008H	
	3	INTP2			000AH	
	4	INTP3			000CH	
	5	INTP4			000EH	
	6	INTP5			0010H	
	7	INTSRE6			UART6 reception error generation	
	8	INTSR6	End of UART6 reception	0014H		
	9	INTST6	End of UART6 transmission	0016H		
	10	INTCSI10/ INTST0 <sup>Note 4</sup>	End of CSI10 transfer/end of UART0 transmission	0018H		
	11	INTTMH1	Match between TMH1 and CRH1 (when compare register is specified)	001AH		
	12	INTTMH0	Match between TMH0 and CRH0 (when compare register is specified)	001CH		
	13	INTTM50	Match between TM50 and CR50 (when compare register is specified)	001EH		
	14	INTTM000	Match between TM00 and CR000 (when compare register is specified)	0020H		
	15	INTTM010	Match between TM00 and CR010 (when compare register is specified)	0022H		
	16	INTAD	End of A/D conversion	0024H		
17	INTSR0 <sup>Note 4</sup>	End of UART0 reception	0026H			
Software	–	BRK	BRK instruction execution	–	003EH	(C)
★ Reset	–	RESET	Reset input	–	0000H	–
		POC	Power-on-clear <sup>Note 5</sup>			
		LVI	Low-voltage detection <sup>Note 6</sup>			
		Clock monitor	X1 input clock stop detection			
		WDT	WDT overflow			

**Notes** 1. The default priority is the priority applicable when two or more maskable interrupt are generated simultaneously. 0 is the highest priority, and 17 is the lowest.

2. Basic configuration types (A) to (C) correspond to (A) to (C) in Figure 14-1.

★ 3. When bit 1 (LVIMD) = 0 is selected for the low-voltage detection register (LVIM).

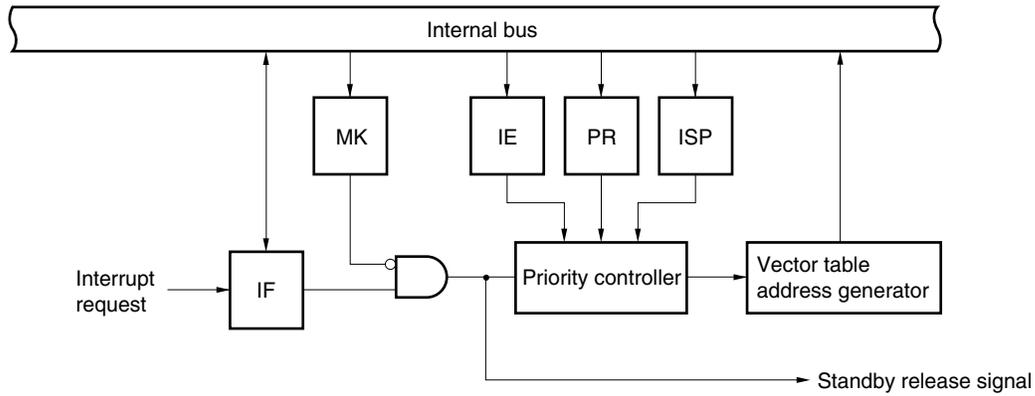
4. The interrupt sources INTST0 and INTSR0 are available only in the  $\mu$ PD780102, 780103, and 78F0103.

★ 5. When “POC used” is selected by mask option.

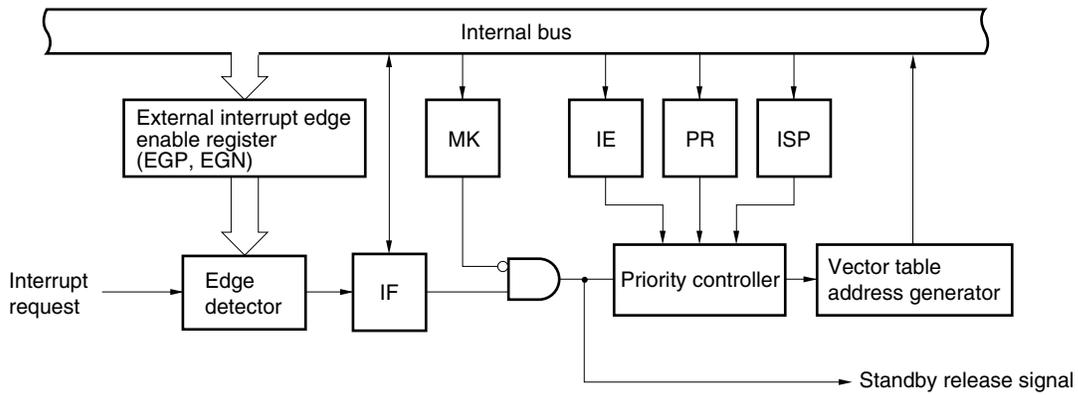
★ 6. When LVIMD = 1 is selected.

Figure 14-1. Basic Configuration of Interrupt Function

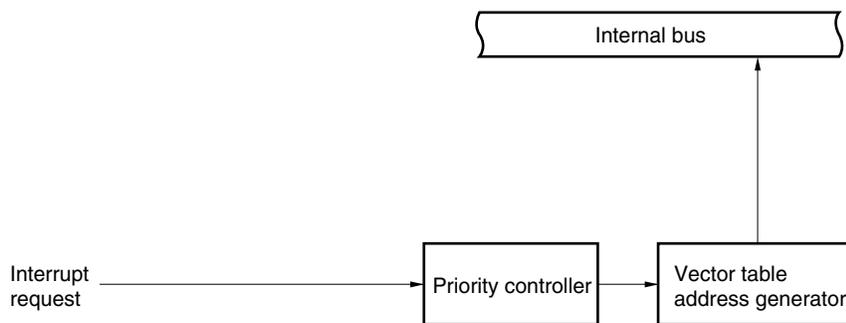
(A) Internal maskable interrupt



(B) External maskable interrupt (INTP0 to INTP5)



(C) Software interrupt



- IF: Interrupt request flag
- IE: Interrupt enable flag
- ISP: In-service priority flag
- MK: Interrupt mask flag
- PR: Priority specification flag

### 14.3 Registers Controlling Interrupt Function

The following 6 types of registers are used to control the interrupt functions.

- Interrupt request flag register (IF0L, IF0H, IF1L)
- Interrupt mask flag register (MK0L, MK0H, MK1L)
- Priority specification flag register (PR0L, PR0H, PR1L)
- External interrupt rising edge enable register (EGP)
- External interrupt falling edge enable register (EGN)
- Program status word (PSW)

Table 14-2 shows a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

**Table 14-2. Flags Corresponding to Interrupt Request Sources**

Interrupt Request	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
		Register		Register		Register
INTLVI	LVIIF	IF0L	LVIMK	MK0L	LVIPR	PR0L
INTP0	PIF0		PMK0		PPR0	
INTP1	PIF1		PMK1		PPR1	
INTP2	PIF2		PMK2		PPR2	
INTP3	PIF3		PMK3		PPR3	
INTP4	PIF4		PMK4		PPR4	
INTP5	PIF5		PMK5		PPR5	
INTSRE6	SREIF6		SREMK6		SREPR6	
INTSR6	SRIF6	IF0H	SRMK6	MK0H	SRPR6	PR0H
INTST6	STIF6		STMK6		STPR6	
INTCSI10	DUALIF0 <sup>Note 2</sup>		DUALMK0 <sup>Note 2</sup>		DUALPR0 <sup>Note 2</sup>	
INTST0 <sup>Note 1</sup>	CSIF10 <sup>Note 3</sup>		CSIMK10 <sup>Note 3</sup>		CSIPR10 <sup>Note 3</sup>	
INTTMH1	TMIFH1		TMMKH1		TMPRH1	
INTTMH0	TMIFH0		TMMKH0		TMPRH0	
INTTM50	TMIF50		TMMK50		TMPR50	
INTTM000	TMIF000		TMMK000		TMPR000	
INTTM010	TMIF010		TMMK010		TMPR010	
INTAD	ADIF		1F1L		ADMK	
INTSRO <sup>Note 1</sup>	SRIF0 <sup>Note 1</sup>	SRMK0 <sup>Note 1</sup>		SRPR0 <sup>Note 1</sup>		

- Notes**
1.  $\mu$ PD780102, 780103, and 78F0103 only.
  2. Flag name in the  $\mu$ PD780102, 780103, and 78F0103. If either of the two types of interrupt sources is generated, these flags are set (1).
  3. Flag name in the  $\mu$ PD780101

**(1) Interrupt request flag registers (IF0L, IF0H, IF1L)**

The interrupt request flags are set to 1 when the corresponding interrupt request is generated or an instruction is executed. They are cleared to 0 when an instruction is executed upon acknowledgment of an interrupt request or upon application of  $\overline{\text{RESET}}$  input.

IF0L, IF0H, and IF1L are set by a 1-bit or 8-bit memory manipulation instruction. When IF0L and IF0H are combined to form 16-bit register IF0, they are read with a 16-bit memory manipulation instruction.

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

**Figure 14-2. Format of Interrupt Request Flag Register (IF0L, IF0H, IF1L)**

Address: FFE0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IF0L	SREIF6	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	LVIIF

Address: FFE1H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IF0H	TMIF010	TMIF000	TMIF50	TMIFH0	TMIFH1	DUALIFO <sup>Note 1</sup>	STIF6	SRIF6

Address: FFE2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IF1L	0	0	0	0	0	0	SRIFO <sup>Note 2</sup>	ADIF

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

- Notes**
1. This is CSIF10 in the  $\mu\text{PD780101}$ .
  2. Incorporated only in the  $\mu\text{PD780102}$ , 780103, and 78F0103.

- Cautions**
1. Be sure to set bits 2 to 7 of IF1L to 0.
  2. When operating a timer, serial interface, or A/D converter after standby release, operate it once after clearing the interrupt request flag. An interrupt request flag may be set by noise.
  3. When an interrupt is acknowledged, the interrupt request flag is automatically cleared and then the interrupt routine is entered.

**(2) Interrupt mask flag registers (MK0L, MK0H, MK1L)**

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt servicing. MK0L, MK0H, and MK1L are set by a 1-bit or 8-bit memory manipulation instruction. When MK0L and MK0H are combined to form a 16-bit register MK0, they are set with a 16-bit memory manipulation instruction.  $\overline{\text{RESET}}$  input sets these registers to FFH.

**Figure 14-3. Format of Interrupt Mask Flag Register (MK0L, MK0H, MK1L)**

Address: FFE4H After reset: FFH R/W

Symbol	<span style="border: 1px solid black; padding: 0 2px;">7</span>	<span style="border: 1px solid black; padding: 0 2px;">6</span>	<span style="border: 1px solid black; padding: 0 2px;">5</span>	<span style="border: 1px solid black; padding: 0 2px;">4</span>	<span style="border: 1px solid black; padding: 0 2px;">3</span>	<span style="border: 1px solid black; padding: 0 2px;">2</span>	<span style="border: 1px solid black; padding: 0 2px;">1</span>	<span style="border: 1px solid black; padding: 0 2px;">0</span>
MK0L	SREMK6	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK

Address: FFE5H After reset: FFH R/W

Symbol	<span style="border: 1px solid black; padding: 0 2px;">7</span>	<span style="border: 1px solid black; padding: 0 2px;">6</span>	<span style="border: 1px solid black; padding: 0 2px;">5</span>	<span style="border: 1px solid black; padding: 0 2px;">4</span>	<span style="border: 1px solid black; padding: 0 2px;">3</span>	<span style="border: 1px solid black; padding: 0 2px;">2</span>	<span style="border: 1px solid black; padding: 0 2px;">1</span>	<span style="border: 1px solid black; padding: 0 2px;">0</span>
MK0H	TMMK010	TMMK000	TMMK50	TMMKH0	TMMKH1	DUALMK0 <sup>Note 1</sup>	STMK6	SRMK6

Address: FFE6H After reset: FFH R/W

Symbol	7	6	5	4	3	2	<span style="border: 1px solid black; padding: 0 2px;">1</span>	<span style="border: 1px solid black; padding: 0 2px;">0</span>
MK1L	1	1	1	1	1	1	SRMK0 <sup>Note 2</sup>	ADMK

XXMKX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

- Notes**
1. This is CSIMK10 in the  $\mu$ PD780101.
  2. Incorporated only in the  $\mu$ PD780102, 780103, and 78F0103.

- Cautions**
1. Because ports 3 and 12 have an alternate function as external interrupt request inputs, when the output level is changed by specifying the output mode of the port function, an interrupt request flag is set. Therefore, the interrupt mask flag should be set to 1 before using the output mode.
  2. Be sure to set bits 2 to 7 of MK1L to 1.

**(3) Priority specification flag registers (PR0L, PR0H, PR1L)**

The priority specification flag registers are used to set the corresponding maskable interrupt priority order. PR0L, PR0H, and PR1L are set by a 1-bit or 8-bit memory manipulation instruction. If PR0L and PR0H are combined to form 16-bit register PR0, they are set with a 16-bit memory manipulation instruction.  $\overline{\text{RESET}}$  input sets these registers to FFH.

**Figure 14-4. Format of Priority Specification Flag Register (PR0L, PR0H, PR1L)**

Address: FFE8H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PR0L	SREPR6	PPR5	PPR4	PPR3	PPR2	PPR1	PPR0	LVIPR

Address: FFE9H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PR0H	TMPR010	TMPR000	TMPR50	TMPRH0	TMPRH1	DUALPRO <sup>Note 1</sup>	STPR6	SRPR6

Address: FFEAH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PR1L	1	1	1	1	1	1	SRPR0 <sup>Note 2</sup>	ADPR

XXPRX	Priority level selection
0	High priority level
1	Low priority level

- Notes**
1. This is CSIPRI0 in the  $\mu$ PD780101.
  2. Incorporated only in the  $\mu$ PD780102, 780103, and 78F0103.

**Caution** Be sure to set bits 2 to 7 of PR1L to 1.

**(4) External interrupt rising edge enable register (EGP), external interrupt falling edge enable register (EGN)**

These registers specify the valid edge for INTP0 to INTP5.

EGP and EGN are set by a 1-bit or 8-bit memory manipulation instruction.

RESET input sets these registers to 00H.

**Figure 14-5. Format of External Interrupt Rising Edge Enable Register (EGP) and External Interrupt Falling Edge Enable Register (EGN)**

Address: FF48H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP	0	0	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

Address: FF49H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN	0	0	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

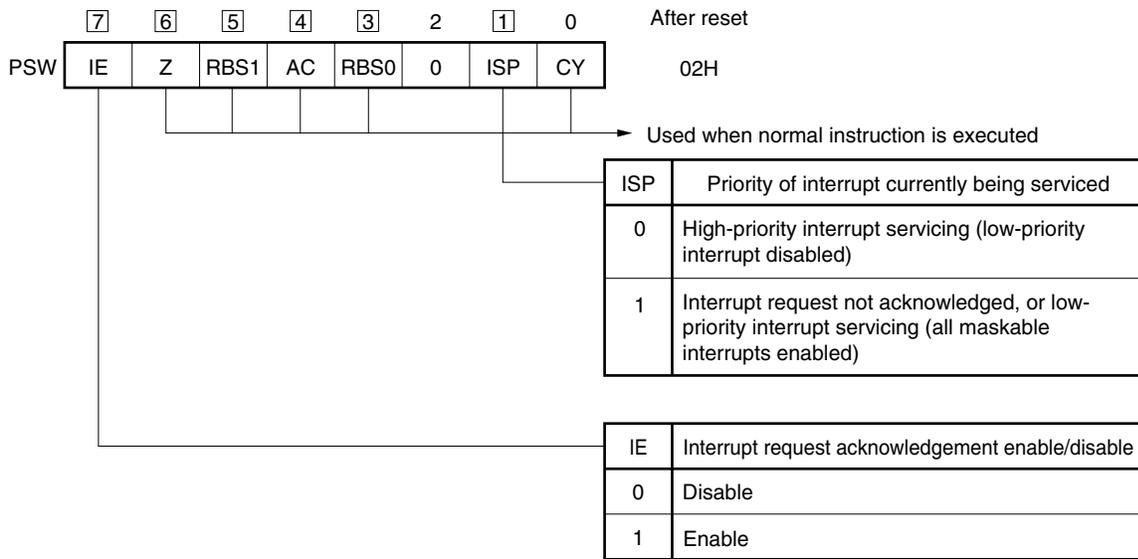
EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 5)
0	0	Interrupt disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

**(5) Program status word (PSW)**

The program status word is a register used to hold the instruction execution result and the current status for an interrupt request. The IE flag that sets maskable interrupt enable/disable and the ISP flag that controls multiple interrupt servicing are mapped to the PSW.

Besides 8-bit read/write, this register can carry out operations using bit manipulation instructions and dedicated instructions (EI and DI). When a vectored interrupt request is acknowledged, if the BRK instruction is executed, the contents of the PSW are automatically saved into a stack and the IE flag is reset to 0. If a maskable interrupt request is acknowledged, the contents of the priority specification flag of the acknowledged interrupt are transferred to the ISP flag. The PSW contents are also saved into the stack with the PUSH PSW instruction. They are restored from the stack with the RETI, RETB, and POP PSW instructions.  $\overline{\text{RESET}}$  input sets PSW to 02H.

**Figure 14-6. Format of Program Status Word**



## 14.4 Interrupt Servicing Operations

### 14.4.1 Maskable interrupt acknowledgement

A maskable interrupt becomes acknowledgeable when the interrupt request flag is set to 1 and the mask (MK) flag corresponding to that interrupt request is cleared to 0. A vectored interrupt request is acknowledged if interrupts are in the interrupt enabled state (when the IE flag is set to 1). However, a low-priority interrupt request is not acknowledged during servicing of a higher priority interrupt request (when the ISP flag is reset to 0). The times from generation of a maskable interrupt request until interrupt servicing is performed are listed in Table 14-3 below.

For the interrupt request acknowledgement timing, see **Figures 14-8** and **14-9**.

**Table 14-3. Time from Generation of Maskable Interrupt Until Servicing**

	Minimum Time	Maximum Time <sup>Note</sup>
When $\times\times PR = 0$	7 clocks	32 clocks
When $\times\times PR = 1$	8 clocks	33 clocks

**Note** If an interrupt request is generated just before a divide instruction, the wait time becomes longer.

**Remark** 1 clock:  $1/f_{CPU}$  ( $f_{CPU}$ : CPU clock)

If two or more interrupt requests are generated simultaneously, the request with a higher priority level specified in the priority specification flag is acknowledged first. If two or more interrupts requests have the same priority level, the request with the highest default priority is acknowledged first.

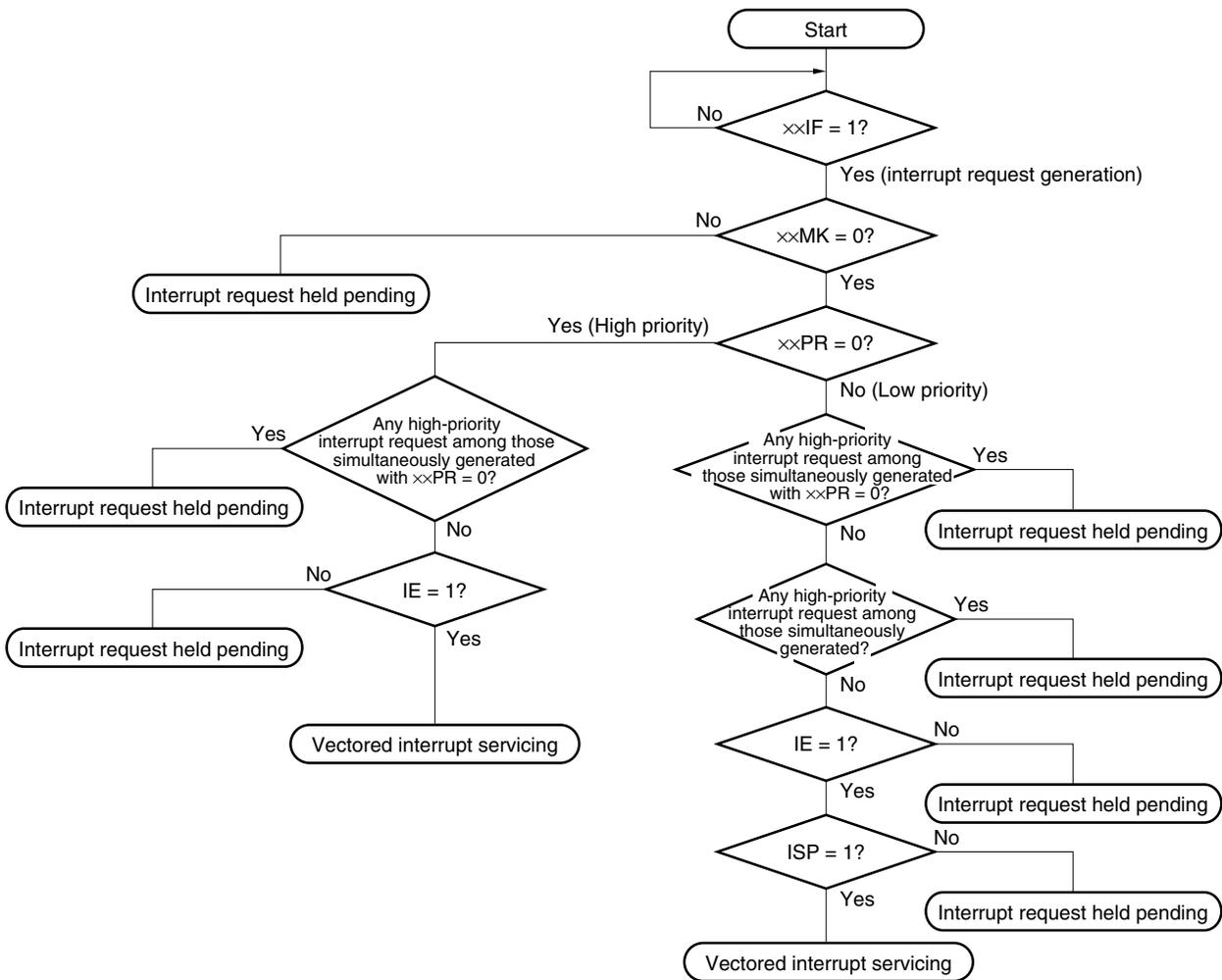
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 14-7 shows the interrupt request acknowledgement algorithm.

If a maskable interrupt request is acknowledged, the contents are saved into the stacks in the order of PSW, then PC, the IE flag is reset (0), and the contents of the priority specification flag corresponding to the acknowledged interrupt are transferred to the ISP flag. The vector table data determined for each interrupt request is the loaded into the PC and branched.

Restoring from an interrupt is possible by using the RETI instruction.

Figure 14-7. Interrupt Request Acknowledgement Processing Algorithm



××IF: Interrupt request flag

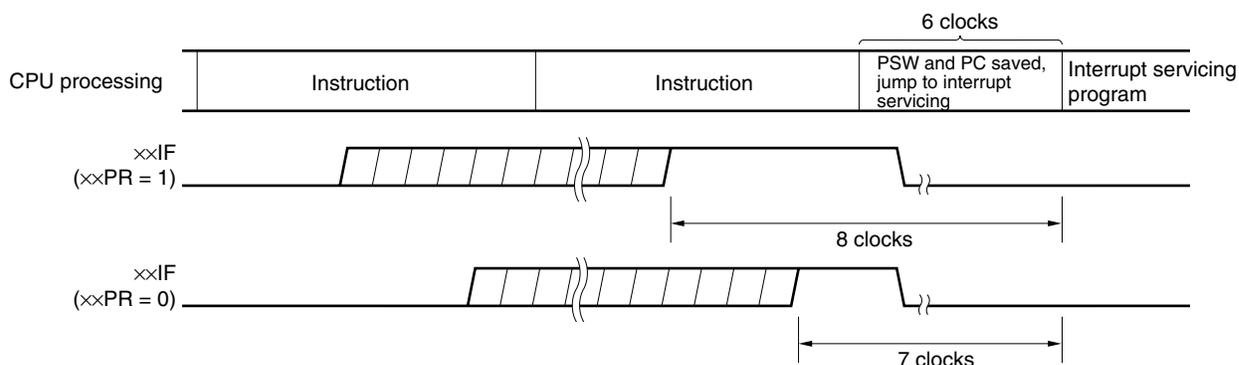
××MK: Interrupt mask flag

××PR: Priority specification flag

IE: Flag that controls acknowledgement of maskable interrupt request (1 = Enable, 0 = Disable)

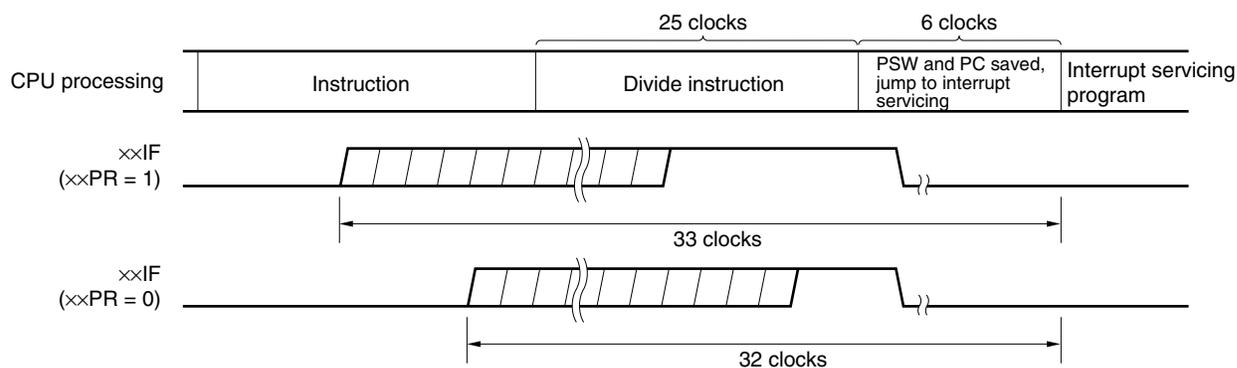
ISP: Flag that indicates the priority level of the interrupt currently being serviced (0 = high-priority interrupt servicing, 1 = No interrupt request acknowledged, or low-priority interrupt servicing)

**Figure 14-8. Interrupt Request Acknowledgement Timing (Minimum Time)**



**Remark** 1 clock:  $1/f_{CPU}$  ( $f_{CPU}$ : CPU clock)

**Figure 14-9. Interrupt Request Acknowledgement Timing (Maximum Time)**



**Remark** 1 clock:  $1/f_{CPU}$  ( $f_{CPU}$ : CPU clock)

#### 14.4.2 Software interrupt request acknowledgement

A software interrupt acknowledge is acknowledged by BRK instruction execution. Software interrupts cannot be disabled.

If a software interrupt request is acknowledged, the contents are saved into the stacks in the order of the program status word (PSW), then program counter (PC), the IE flag is reset (0), and the contents of the vector table (003EH, 003FH) are loaded into the PC and branched.

Restoring from a software interrupt is possible by using the RETB instruction.

**Caution** Do not use the RETI instruction for restoring from the software interrupt.

### 14.4.3 Multiple interrupt servicing

Multiple interrupt occurs when another interrupt request is acknowledged during execution of an interrupt.

Multiple interrupt does not occur unless the interrupt request acknowledgement enabled state is selected (IE = 1). Also, when an interrupt request is acknowledged, interrupt request acknowledgement becomes disabled (IE = 0). Therefore, to enable multiple interrupt servicing, it is necessary to set (1) the IE flag with the EI instruction during interrupt servicing to enable interrupt acknowledgement.

Moreover, even if interrupts are enabled, multiple interrupt servicing may not be enabled, this being subject to interrupt priority control. Two types of priority control are available: default priority control and programmable priority control. Programmable priority control is used for multiple interrupt servicing.

In the interrupt enabled state, if an interrupt request with a priority equal to or higher than that of the interrupt currently being serviced is generated, it is acknowledged for multiple interrupt. If an interrupt with a priority lower than that of the interrupt currently being serviced is generated during interrupt servicing, it is not acknowledged for multiple interrupt. Interrupt requests that are not enabled because interrupts are in the interrupt disabled state or they have a lower priority are held pending. When servicing of the current interrupt ends, the pending interrupt request is acknowledged following execution of one main processing instruction execution.

Table 14-4 shows interrupt requests enabled for multiple interrupt and Figure 14-10 shows multiple interrupt examples.

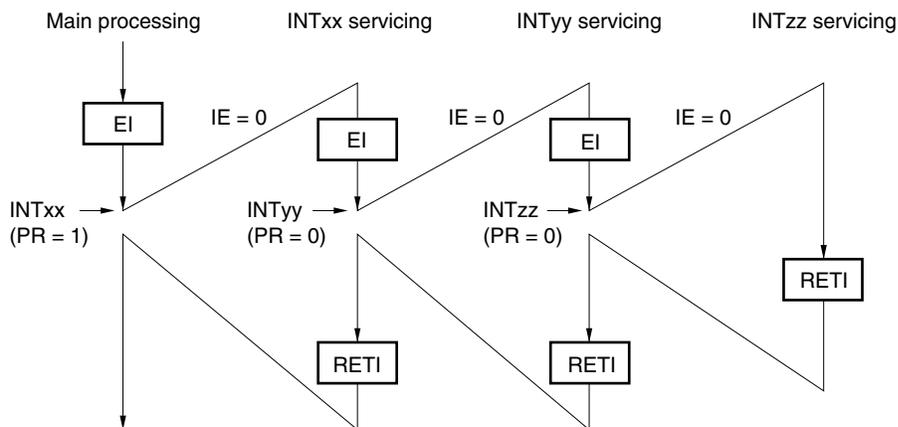
**Table 14-4. Interrupt Request Enabled for Multiple Interrupt During Interrupt Servicing**

Multiple Interrupt Request		Maskable Interrupt Request			
		PR = 0		PR = 1	
		IE = 1	IE = 0	IE = 1	IE = 0
Interrupt Being Serviced					
Maskable interrupt	ISP = 0	○	×	×	×
	ISP = 1	○	×	○	×
Software interrupt		○	×	○	×

- Remarks**
- : Multiple interrupt enabled
  - ×: Multiple interrupt disabled
  - ISP and IE are flags contained in the PSW.
    - ISP = 0: An interrupt with higher priority is being serviced.
    - ISP = 1: No interrupt request has been acknowledged, or an interrupt with a lower priority is being serviced.
    - IE = 0: Interrupt request acknowledgement is disabled.
    - IE = 1: Interrupt request acknowledgement is enabled.
  - PR is a flag contained in PR0L, PR0H, and PR1L.
    - PR = 0: Higher priority level
    - PR = 1: Lower priority level

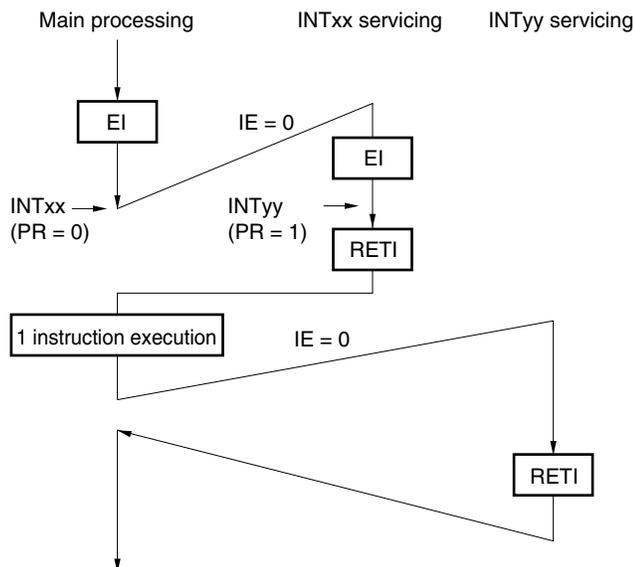
Figure 14-10. Examples of Multiple Interrupt Servicing (1/2)

**Example 1. Multiple interrupt servicing occurs twice**



During servicing of interrupt INTxx, two interrupt requests, INTyy and INTzz, are acknowledged, and multiple interrupt servicing takes place. Before each interrupt request is acknowledged, the EI instruction must always be issued to enable interrupt request acknowledgement.

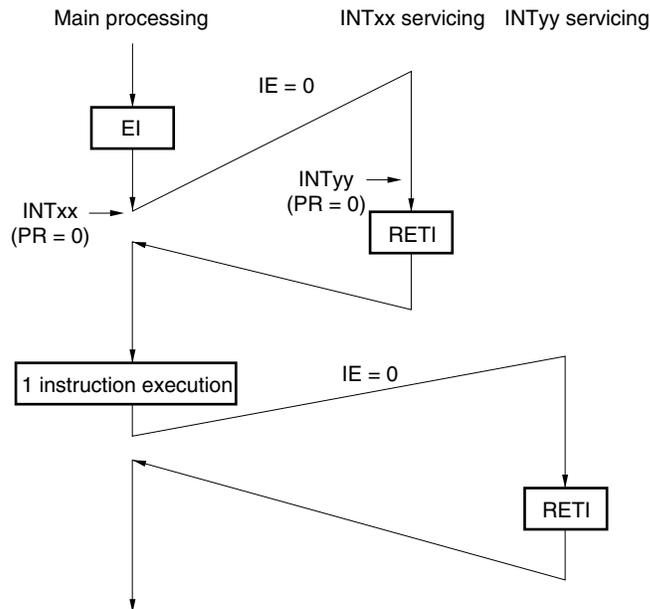
**Example 2. Multiple interrupt servicing does not occur due to priority control**



Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

- PR = 0: Higher priority level
- PR = 1: Lower priority level
- IE = 0: Interrupt request acknowledgement disabled

Figure 14-10. Examples of Multiple Interrupt Servicing (2/2)

**Example 3. Multiple interrupt servicing does not occur because interrupts are not enabled**

Interrupts are not enabled during servicing of interrupt INTxx (EI instruction is not issued), therefore, interrupt request INTyy is not acknowledged and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 0: Higher priority level

IE = 0: Interrupt request acknowledgement disabled

**14.4.4 Interrupt request hold**

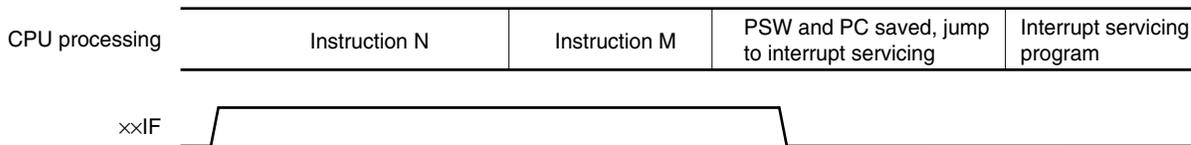
There are instructions where, even if an interrupt request is issued for them while another instruction is being executed, request acknowledgement is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- MOV PSW, #byte
- MOV A, PSW
- MOV PSW, A
- MOV1 PSW. bit, CY
- MOV1 CY, PSW. bit
- AND1 CY, PSW. bit
- OR1 CY, PSW. bit
- XOR1 CY, PSW. bit
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- PUSH PSW
- POP PSW
- BT PSW. bit, \$addr16
- BF PSW. bit, \$addr16
- BTCLR PSW. bit, \$addr16
- EI
- DI
- Manipulation instructions for the IF0L, IF0H, IF1L, MK0L, MK0H, MK1L, PR0L, PR0H, and PR1L registers.

**Caution** The BRK instruction is not one of the above-listed interrupt request hold instructions. However, the software interrupt activated by executing the BRK instruction causes the IE flag to be cleared. Therefore, even if a maskable interrupt request is generated during execution of the BRK instruction, the interrupt request is not acknowledged.

Figure 14-11 shows the timing at which interrupt requests are held pending.

**Figure 14-11. Interrupt Request Hold**



- Remarks**
1. Instruction N: Interrupt request hold instruction
  2. Instruction M: Instruction other than interrupt request hold instruction
  3. The xPR (priority level) values do not affect the operation of xIF (instruction request).

## CHAPTER 15 STANDBY FUNCTION

### 15.1 Standby Function and Configuration

#### 15.1.1 Standby function

Table 15-1. Relationship Between HALT and STOP Modes and Clock

	X1 Input Clock	Ring-OSC Clock	CPU Clock
HALT mode	Oscillation continues	Oscillation continues	Operation stopped
STOP mode	Oscillation stopped	Oscillation continues	Operation stopped

The standby function is designed to reduce the power consumption of the system. The following two modes are available.

##### (1) HALT mode

HALT instruction execution sets the HALT mode. The HALT mode is intended to stop the CPU operation clock. The X1 input clock and Ring-OSC clock oscillators continue oscillating. In this mode, current consumption is not decreased as much as in the STOP mode. However, the HALT mode is effective for restarting operation immediately upon interrupt request generation and carrying out intermittent operations.

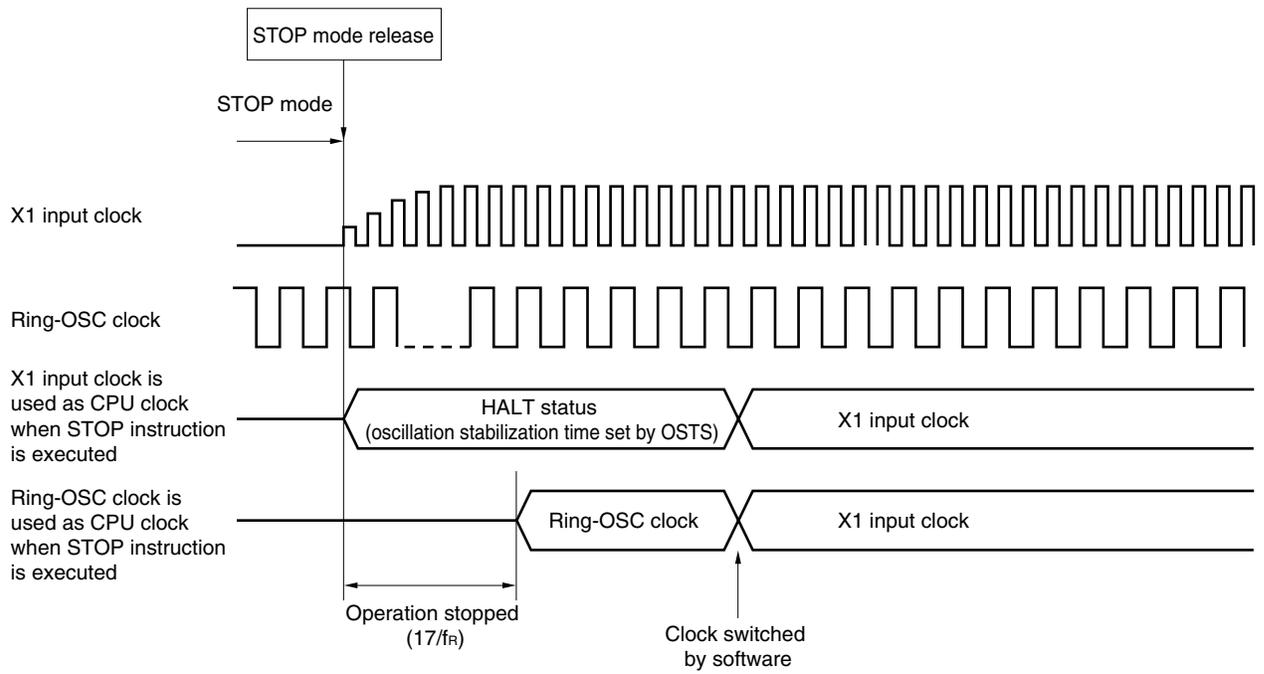
##### (2) STOP mode

STOP instruction execution sets the STOP mode. In the STOP mode, the X1 input clock oscillator stops, stopping the whole system, thereby considerably reducing the CPU power consumption. Because this mode can be cleared by an interrupt request, it enables intermittent operations to be carried out. However, because a wait time is required to secure the oscillation stabilization time after the STOP mode is released, select the HALT mode if it is necessary to start processing immediately upon interrupt request generation.

In either of these two modes, all the contents of registers, flags and data memory just before the standby mode is set are held. The I/O port output latches and output buffer statuses are also held.

- Cautions**
1. When shifting to the STOP mode, be sure to stop the peripheral hardware operation before executing STOP instruction.
  2. The following sequence is recommended for power consumption reduction of the A/D converter when the standby function is used: First clear bit 7 (ADCS) of the A/D converter mode register (ADM) to 0 to stop the A/D conversion operation, and then execute the HALT or STOP instruction.
  3. Ring-OSC clock oscillation cannot be stopped in the STOP mode. However, when the Ring-OSC clock is used as the CPU clock, operation is stopped for  $17/f_R$  (s) after STOP mode is released.

Figure 15-1. Operation Timing When STOP Mode Is Released



15.1.2 Registers controlling standby function

The standby function is controlled by the following two registers.

- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)

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

This is the status register of the X1 input clock oscillation stabilization time counter. If the Ring-OSC clock is used as the CPU clock, the X1 input clock oscillation stabilization time can be checked.

- ★ OSTC can be read by a 1-bit or 8-bit memory manipulation instruction.  
 $\overline{\text{RESET}}$  input, STOP instruction, and MSTOP =1 clear OSTC to 00H.

★ **Figure 15-2. Format of Oscillation Stabilization Time Counter Status Register (OSTC)**

Address: FFA3H After reset: 00H R

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

MOST11	MOST13	MOST14	MOST15	MOST16	Oscillation stabilization time status
1	0	0	0	0	$2^{11}/f_{XP}$ min. (204.8 $\mu$ s min.)
1	1	0	0	0	$2^{13}/f_{XP}$ min. (819.2 $\mu$ s min.)
1	1	1	0	0	$2^{14}/f_{XP}$ min. (1.64 ms min.)
1	1	1	1	0	$2^{15}/f_{XP}$ min. (3.27 ms min.)
1	1	1	1	1	$2^{16}/f_{XP}$ min. (6.55 ms min.)

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

- Remarks**
1. Values in parentheses are for operation with  $f_{XP} = 10$  MHz.
  2.  $f_{XP}$ : X1 input clock oscillation frequency

**(2) Oscillation stabilization time select register (OSTS)**

This register is used to select the oscillation stabilization time of the X1 input clock when STOP mode is released. OSTS can be set by an 8-bit memory manipulation instruction.  $\overline{\text{RESET}}$  input sets OSTS to 05H.

★

**Figure 15-3. Format of Oscillation Stabilization Time Select Register (OSTS)**

Address: FFA4H After reset: 05H R/W

Symbol	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

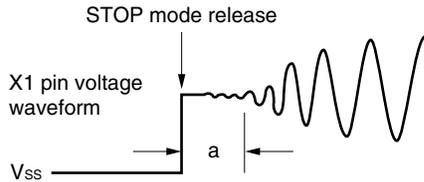
OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection
0	0	1	$2^{11}/f_{XP}$ (204.8 $\mu\text{s}$ )
0	1	0	$2^{13}/f_{XP}$ (819.2 $\mu\text{s}$ )
0	1	1	$2^{14}/f_{XP}$ (1.64 ms)
1	0	0	$2^{15}/f_{XP}$ (3.27 ms)
1	0	1	$2^{16}/f_{XP}$ (6.55 ms)
Other			Setting prohibited

**Cautions 1.** If the STOP mode is entered and then released while the Ring-OSC is being used as the CPU clock, set the oscillation stabilization time as follows.

- Desired OSTC oscillation stabilization time  $\leq$  Oscillation stabilization time set by OSTS

The setting value of OSTS can be initialized (to 05H) only by a reset. Therefore, note that the X1 oscillation stabilization time counter counts up to the set value of OSTS after STOP mode is released when the Ring-OSC is used as the CPU clock.

**2.** The wait time when STOP mode is released does not include the time after STOP mode release until clock oscillation starts (“a” below) regardless of whether STOP mode is released by  $\overline{\text{RESET}}$  input or interrupt generation.



- Remarks 1.** Values in parentheses are for operation with  $f_{XP} = 10$  MHz.  
**2.**  $f_{XP}$ : X1 input clock oscillation frequency

## 15.2 Standby Function Operation

### 15.2.1 HALT mode

#### (1) HALT mode

The HALT mode is set by executing the HALT instruction. HALT mode can be set regardless of whether CPU clock before the setting was the X1 input clock or Ring-OSC clock.

The operating statuses in the HALT mode are shown below.

**Table 15-2. Operating Statuses in HALT Mode**

HALT Mode Setting		When HALT Instruction Is Executed While CPU Is Operating Using X1 Input Clock		When HALT Instruction Is Executed While CPU Is Operating Using Ring-OSC Clock	
		Ring-OSC Oscillation Continues	Ring-OSC Oscillation Stopped <sup>Note 1</sup>	X1 Input Clock Oscillation Continues	X1 Input Clock Oscillation Stopped
Item					
System clock		Both X1 input clock and Ring-OSC clock can be oscillated. Clock supply to CPU stops			
CPU		Operation stopped			
Port (output latch)		Holds the status before HALT mode is set			
16-bit timer/event counter 00		Operable		Operation stopped	
8-bit timer/event counter 50		Operable		Operable only when TI50 is selected as count clock	
8-bit timer	H0	Operable		Operable only when TO50 is selected as count clock during 8-bit timer/event counter 50 operation	
	H1	Operable		Operable only when $f_{R/2^7}$ is selected as count clock	
Watchdog timer		Operable	Operable only when X1 input clock is selected as operation clock	Operable	Operable only when Ring-OSC clock input is selected as operation clock
★ A/D converter		Operable		Operation disabled	
Serial interface	UART0 <sup>Note 2</sup>	Operable		Operable only when TO50 is selected as serial clock during 8-bit timer/event counter 50 operation	
	UART6	Operable			
	CSI10	Operable		Operable only when external $\overline{\text{SCK10}}$ is selected as serial clock	
Clock monitor		Operable	Operation stopped	Operable	Operation stopped
Power-on-clear function <sup>Note 3</sup>		Operable			
Low-voltage detection function		Operable			
External interrupt		Operable			

- Notes**
1. When “Stopped by software” is selected for Ring-OSC by a mask option and Ring-OSC is stopped by software (for mask options, see **CHAPTER 20 MASK OPTIONS**).
  2.  $\mu$ PD780102, 780103, and 78F0103 only.
  3. When “POC used” is selected by mask option (for mask options, see **CHAPTER 20 MASK OPTIONS**).

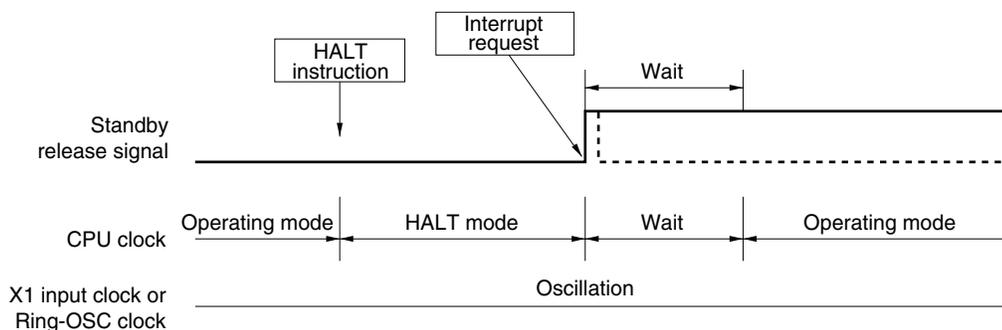
**(2) HALT mode release**

The HALT mode can be released by the following two sources.

**(a) Release by unmasked interrupt request**

When an unmasked interrupt request is generated, the HALT mode is released. If interrupt acknowledgement is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgement is disabled, the next address instruction is executed.

**Figure 15-4. HALT Mode Release by Interrupt Request Generation**



**Remarks 1.** The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

**2.** The wait time is as follows:

- When vectored interrupt servicing is carried out: 8 or 9 clocks
- When vectored interrupt servicing is not carried out: 2 or 3 clocks

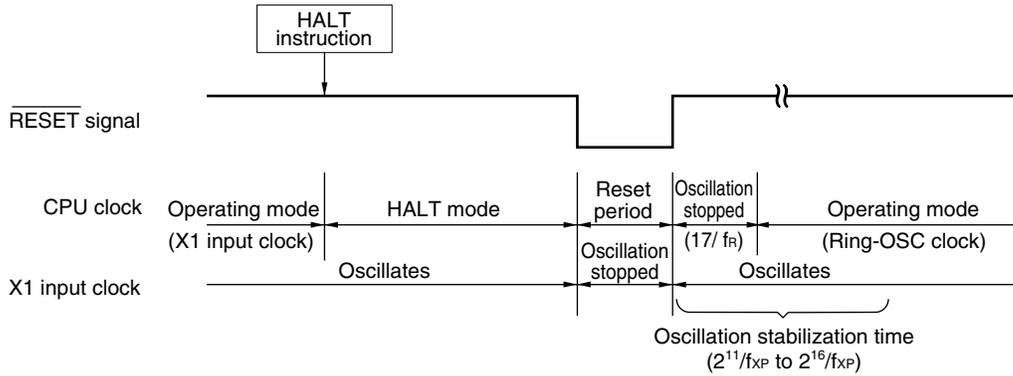
(b) Release by  $\overline{\text{RESET}}$  input

When the  $\overline{\text{RESET}}$  signal is input, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

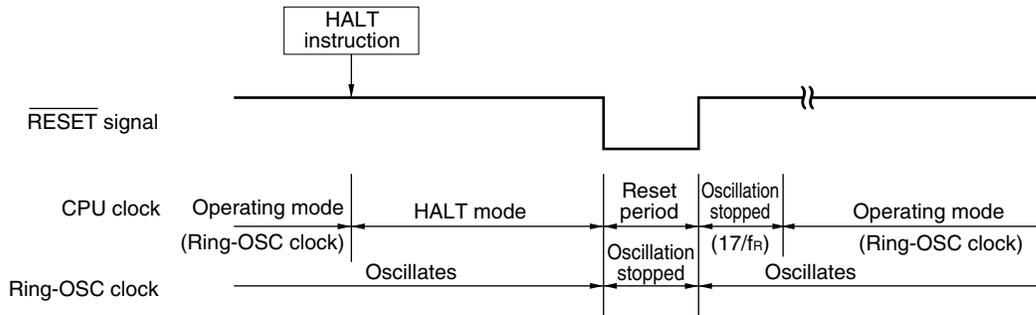
Figure 15-5. HALT Mode Release by  $\overline{\text{RESET}}$  Input

★

(1) When X1 input clock is used as CPU clock



(2) When Ring-OSC clock is used as CPU clock



- Remarks 1.  $f_{XP}$ : X1 input clock oscillation frequency
- 2.  $f_R$ : Ring-OSC clock oscillation frequency

Table 15-3. Operation After HALT Mode Release

Release Source	MK $\times\times$	PR $\times\times$	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Next address instruction execution
	0	0	1	×	Interrupt servicing execution
	0	1	0	1	Next address instruction execution
	0	1	×	0	
	0	1	1	1	Interrupt servicing execution
	1	×	×	×	HALT mode held
$\overline{\text{RESET}}$ input	–	–	×	×	Reset processing

×: Don't care

## 15.2.2 STOP mode

## (1) STOP mode setting and operating statuses

The STOP mode is set by executing the STOP instruction. It can be set regardless of whether the CPU clock before the setting was the X1 input clock or Ring-OSC clock.

**Caution** Because the interrupt request signal is used to clear the standby mode, if there is an interrupt source with the interrupt request flag set and the interrupt mask flag reset, the standby mode is immediately cleared if set. Thus, the STOP mode is reset to the HALT mode immediately after execution of the STOP instruction and the system returns to the operating mode as soon as the wait time set using the oscillation stabilization time select register (OSTS) has elapsed.

The operating statuses in the STOP mode are shown below.

Table 15-4. Operating Statuses in STOP Mode

HALT Mode Setting		When STOP Instruction Is Executed While CPU Is Operating Using X1 Input Clock		When STOP Instruction Is Executed While CPU Is Operating Using Ring-OSC Clock
		Ring-OSC Oscillation Continues	Ring-OSC Oscillation Stopped <sup>Note 1</sup>	
Item				
System clock		Only X1 input clock oscillation stops. Clock supply to CPU stops		
CPU		Operation stopped		
Port (output latch)		Holds the status before STOP mode is set		
16-bit timer/event counter 00		Operation stopped		
8-bit timer/event counter 50		Operable only when T150 is selected as count clock		
8-bit timer	H0	Operable when TO50 is selected as count clock during 8-bit timer/event counter 50 operation		
	H1	Operable <sup>Note 2</sup>	Operation stopped	Operable <sup>Note 2</sup>
Watchdog timer		Operable only when Ring-OSC clock input is selected as operation clock	Operation stopped	Operable only when Ring-OSC clock input is selected as operation clock
A/D converter		Operation stopped		
Serial interface	UART0 <sup>Note 3</sup>	Operable only when TO50 is selected as count clock during 8-bit timer/event counter 50 operation		
	UART6			
	CSI10	Operable only when external $\overline{\text{SCK10}}$ is selected as serial clock		
Clock monitor		Operation stopped		
Power-on-clear function <sup>Note 4</sup>		Operable		
Low-voltage detection function		Operable		
External interrupt		Operable		

- Notes**
1. When “Stopped by software” is selected for Ring-OSC by a mask option and Ring-OSC is stopped by software (for mask options, see **CHAPTER 20 MASK OPTIONS**).
  2. Operable only when  $f_R/2^7$  is selected as count clock.
  3.  $\mu$ PD780102, 780103, and 78F0103 only.
  4. When “POC used” is selected by a mask option (for mask options, see **CHAPTER 20 MASK OPTIONS**).

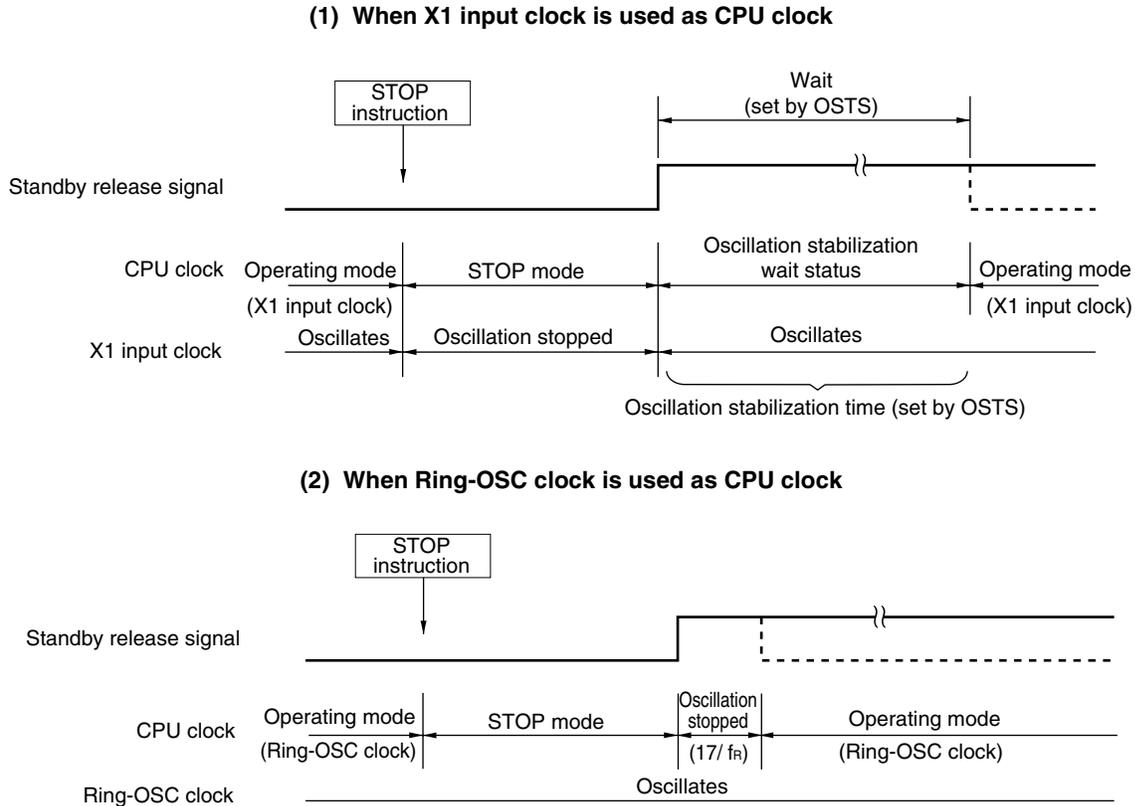
(2) STOP mode release

The STOP mode can be released by the following two sources.

(a) Release by unmasked interrupt request

When an unmasked interrupt request is generated, the STOP mode is released. After the oscillation stabilization time has elapsed, if interrupt acknowledgement is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgement is disabled, the next address instruction is executed.

Figure 15-6. STOP Mode Release by Interrupt Request Generation



**Remark** The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

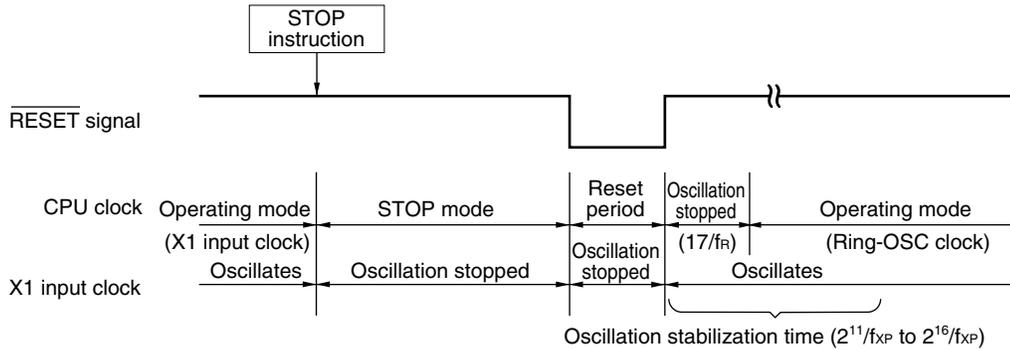
**(b) Release by  $\overline{\text{RESET}}$  input**

When the  $\overline{\text{RESET}}$  signal is input, STOP mode is released and a reset operation is performed after the oscillation stabilization time has elapsed.

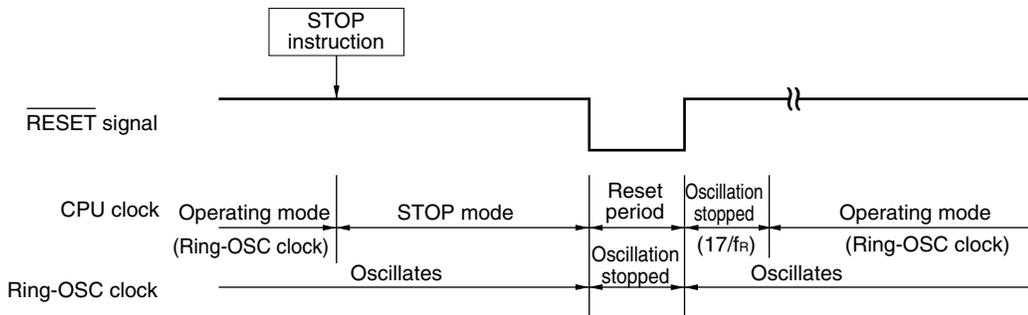
**Figure 15-7. STOP Mode Release by  $\overline{\text{RESET}}$  Input**

★

**(1) When X1 input clock is used as CPU clock**



**(2) When Ring-OSC clock is used as CPU clock**



**Table 15-5. Operation After STOP Mode Release**

Release Source	MK <sub>xx</sub>	PR <sub>xx</sub>	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Next address instruction execution
	0	0	1	×	Interrupt servicing execution
	0	1	0	1	Next address instruction execution
	0	1	×	0	
	0	1	1	1	Interrupt servicing execution
	1	×	×	×	×
$\overline{\text{RESET}}$ input	—	—	×	×	Reset processing

×: Don't care

## CHAPTER 16 RESET FUNCTION

The following five operations are available to generate a reset signal.

- (1) External reset input via  $\overline{\text{RESET}}$  pin
- (2) Internal reset by watchdog timer program loop detection
- (3) Internal reset by clock monitor X1 clock oscillation stop detection
- (4) Internal reset by comparison of supply voltage and detection voltage of power-on-clear (POC) circuit
- (5) Internal reset by comparison of supply voltage and detection voltage of low-power-supply detector (LVI)

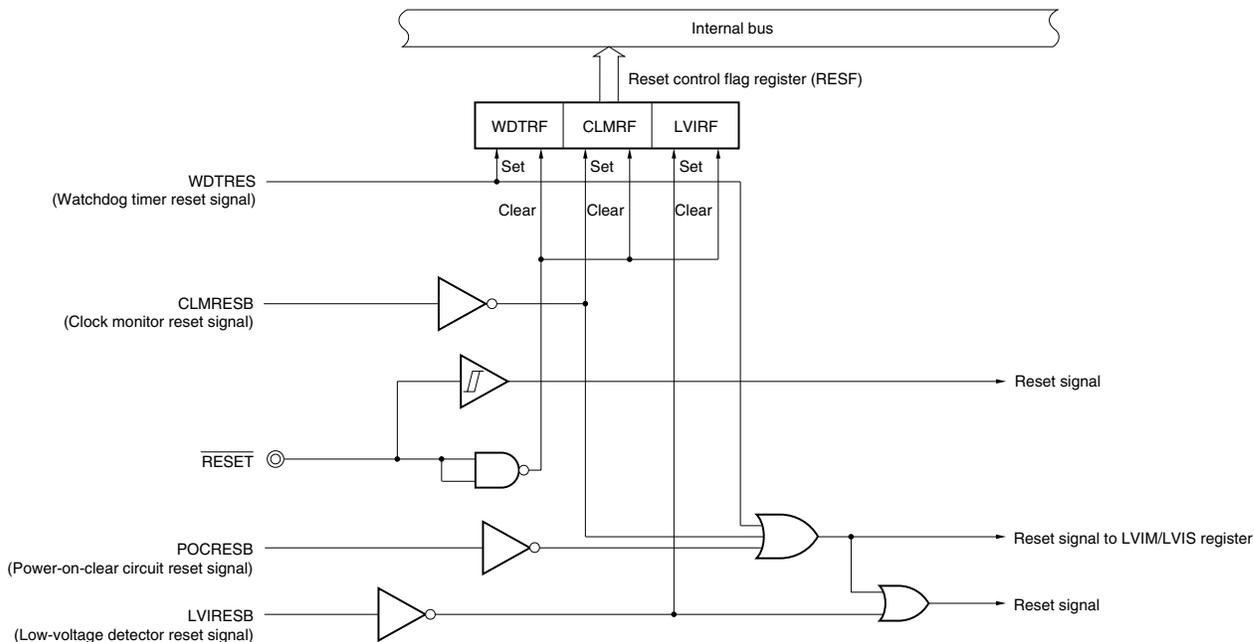
External and internal resets have no functional differences. In both cases, program execution starts at the address at 0000H and 0001H when the reset signal is input.

A reset is applied when a low level is input to the  $\overline{\text{RESET}}$  pin, the watchdog timer overflows, X1 clock oscillation stop is detected by the clock monitor, or by POC and LVI circuit voltage detection, and each hardware is set to the status shown in Table 16-1. Each pin is high impedance during reset input or during the oscillation stabilization time just after reset release, except for P130, which is low-level output.

When a high level is input to the  $\overline{\text{RESET}}$  pin, the reset is released and program execution starts using the Ring-OSC clock after the CPU clock operation has stopped for  $17/f_R$  (s). Reset by the watchdog timer or clock timer source is automatically released and program execution starts using the Ring-OSC clock after the CPU clock operation has stopped for  $17/f_R$  (s) (see **Figures 16-2 to 16-4**). Reset by POC and LVI circuit power supply detection is automatically released when  $V_{DD} > V_{POC}$  or  $V_{DD} > V_{LVI}$  after reset, and program execution starts using the Ring-OSC clock after the CPU clock operation has stopped for  $17/f_R$  (s) (see **CHAPTER 18 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 19 LOW-VOLTAGE DETECTOR**).

- Cautions**
1. For an external reset, input a low level for 10  $\mu\text{s}$  or more to the  $\overline{\text{RESET}}$  pin.
  2. During reset input, the X1 input clock and Ring-OSC clock stop oscillating.
  3. When the STOP mode is released by a reset, the STOP mode contents are held during reset input. However, the port pins become high-impedance, except for P130, which is set to low-level output.

Figure 16-1. Block Diagram of Reset Function



**Caution** An LVI circuit internal reset does not reset the LVI circuit.

- Remarks**
1. LVIM: Low-voltage detection register
  2. LVIS: Low-voltage detection level selection register

Figure 16-2. Timing of Reset by RESET Input

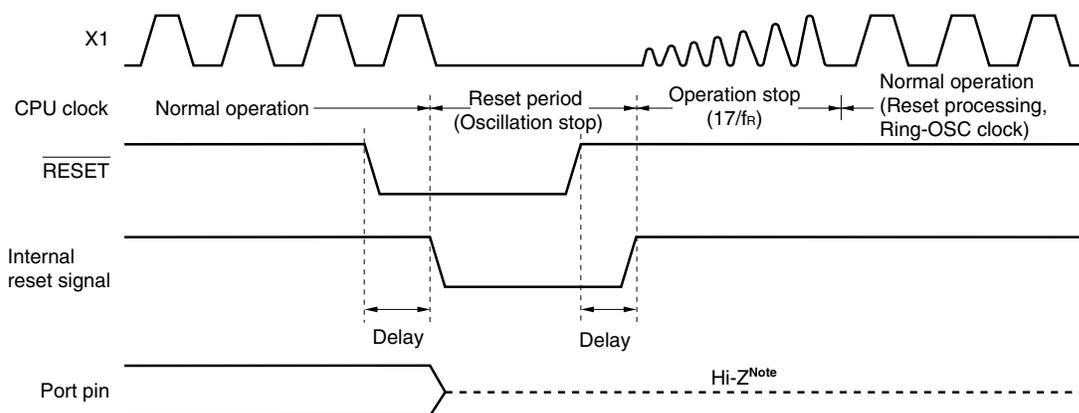
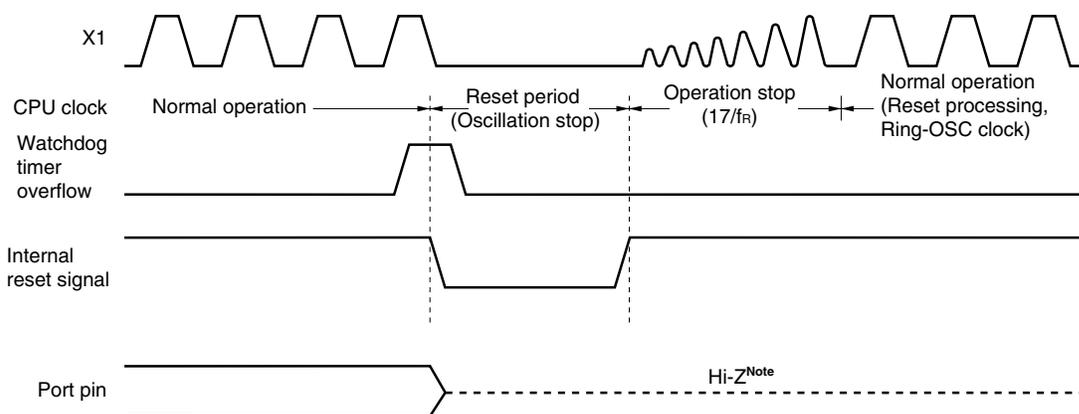
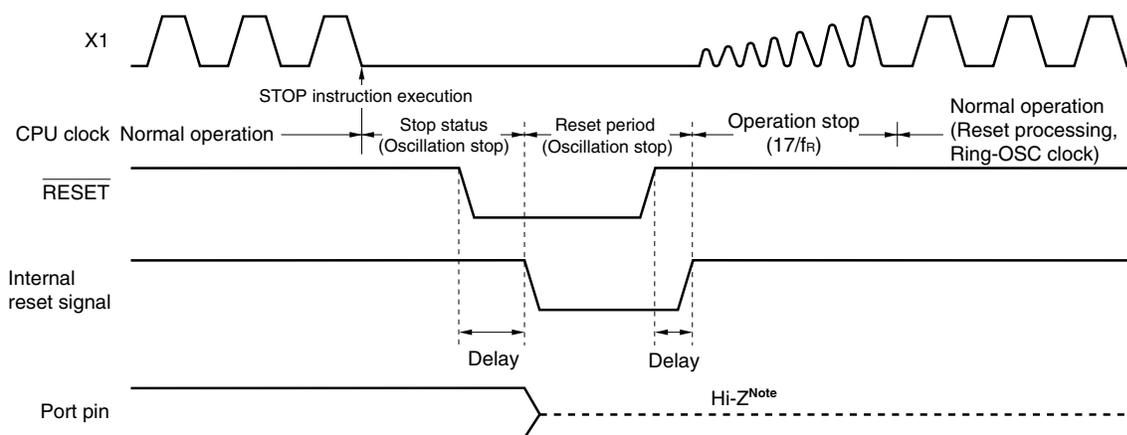


Figure 16-3. Timing of Reset Due to Watchdog Timer Overflow



**Caution** A watchdog timer internal reset resets the watchdog timer.

Figure 16-4. Timing of Reset in STOP Mode by RESET Input



★ **Note** Port pins become high-impedance, except for P130, which is set to low-level output.

**Remark** For the reset timing of the power-on-clear circuit and low-voltage detector, see **CHAPTER 18 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 19 LOW-VOLTAGE DETECTOR**.

Table 16-1. Hardware Statuses After Reset (1/2)

Hardware		Status After Reset
Program counter (PC) <sup>Note 1</sup>		The contents of the reset vector table (0000H, 0001H) are set.
Stack pointer (SP)		Undefined
Program status word (PSW)		02H
RAM	Data memory	Undefined <sup>Note 2</sup>
	General-purpose registers	Undefined <sup>Note 2</sup>
Ports (P0 to P3, P12, P13) (output latches)		00H
Port mode registers (PM0, PM1, PM3, PM12)		FFH
Pull-up resistor option registers (PU0, PU1, PU3, PU12)		00H
Input switch control register (ISC)		00H
Internal memory size switching register (IMS)		CFH
Internal expansion RAM size switching register (IXS)		0CH
Processor clock control register (PCC)		00H
Ring-OSC mode register (RCM)		00H
Main clock mode register (MCM)		00H
Main OSC control register (MOC)		00H
Oscillation stabilization time select register (OSTS)		05H
Oscillation stabilization time counter status register (OSTC)		00H
16-bit timer/event counter 00	Timer counter 00 (TM00)	0000H
	Capture/compare registers 000, 010 (CR000, CR010)	0000H
	Mode control register 00 (TMC00)	00H
	Prescaler mode register 00 (PRM00)	00H
	Capture/compare control register 00 (CRC00)	00H
	Timer output control register 00 (TOC00)	00H
8-bit timer/event counter 50	Timer counter 50 (TM50)	00H
	Compare register 50 (CR50)	00H
	Timer clock selection register 50 (TCL50)	00H
	Mode control register 50 (TMC50)	00H
8-bit timer/event counters H0, H1	Compare registers 00, 10, 01, 11 (CMP00, CMP10, CMP01, CMP11)	00H
	Mode registers (TMHMD0, TMHMD1)	00H
Watchdog timer	Mode register (WDTM)	67H
	Enable register (WDTE)	9AH
A/D converter	Conversion result register (ADCR)	0000H
	Mode register (ADM)	00H
	Analog input channel specification register (ADS)	00H
	Power-fail comparison mode register (PFM)	00H
	Power-fail comparison threshold register (PFT)	00H

**Notes** 1. During reset input or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

2. When a reset is executed in the standby mode, the pre-reset status is held even after reset.

**Table 16-1. Hardware Statuses After Reset (2/2)**

	Hardware	Status After Reset
Serial interface UART0 <sup>Note 1</sup>	Receive buffer register 0 (RXB0)	FFH
	Transmit shift register 0 (TXS0)	FFH
	Asynchronous serial interface operation mode register 0 (ASIM0)	01H
	Baud rate generator control register 0 (BRGC0)	1FH
Serial interface UART6	Receive buffer register 6 (RXB6)	FFH
	Transmit buffer register 6 (TXB6)	FFH
	Asynchronous serial interface operation mode register 6 (ASIM6)	01H
	Asynchronous serial interface reception error status register 6 (ASIS6)	00H
	Asynchronous serial interface transmission error status register 6 (ASIF6)	00H
	Clock selection register 6 (CKSR6)	00H
	Baud rate generator control register 6 (BRGC6)	FFH
	Asynchronous serial interface control register 6 (ASICL6)	16H
Serial interface CSI10	Transmit buffer register 10 (SOTB10)	Undefined
	Serial I/O shift register 10 (SIO10)	00H
	Serial operation mode register 10 (CSIM10)	00H
	Serial clock selection register 10 (CSIC10)	00H
Clock monitor	Mode register (CLM)	00H
Reset function	Reset control flag register (RESF)	00H <sup>Note 2</sup>
Low-voltage detector	Low-voltage detection register (LVIM)	00H <sup>Note 2</sup>
	Low-voltage detection level selection register (LVIS)	00H <sup>Note 2</sup>
Interrupt	Request flag registers 0L, 0H, 1L (IF0L, IF0H, IF1L)	00H
	Mask flag registers 0L, 0H, 1L (MK0L, MK0H, MK1L)	FFH
	Priority specification flag registers 0L, 0H, 1L (PR0L, PR0H, PR1L)	FFH
	External interrupt rising edge enable register (EGP)	00H
	External interrupt falling edge enable register (EGN)	00H

- Notes**
1.  $\mu$ PD780102, 780103, and 78F0103 only.
  2. These values vary depending on the reset source.

Reset Source \ Register	$\overline{\text{RESET}}$ Input	Reset by POC	Reset by WDT	Reset by CLM	Reset by LVI
RESF	See <b>Table 16-2</b> .				
LVIM	Cleared (00H)	Cleared (00H)	Cleared (00H)	Cleared (00H)	Held
LVIS					

### 16.1 Register for Confirming Reset Source

Many internal reset generation sources exist in the 78K0/KB1 Series. The reset control flag register (RESF) is used to store which source has generated the reset request.

RESF can be read by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$  input, reset input by power-on-clear (POC) circuit, and reading RESF clear RESF to 00H.

**Figure 16-5. Format of Reset Control Flag Register (RESF)**

Address: FFACH After reset: 00H<sup>Note</sup> R

Symbol	7	6	5	4	3	2	1	0
RESF	0	0	0	WDTRF	0	0	CLMRF	LVIRF

WDTRF	Internal reset request by watchdog timer (WDT)
0	Internal reset request is not generated, or RESF is cleared.
1	Internal reset request is generated.

CLMRF	Internal reset request by clock monitor (CLM)
0	Internal reset request is not generated, or RESF is cleared.
1	Internal reset request is generated.

LVIRF	Internal reset request by low-voltage detector (LVI)
0	Internal reset request is not generated, or RESF is cleared.
1	Internal reset request is generated.

**Note** The value after reset varies depending on the reset source.

**Caution** Do not read data by a 1-bit memory manipulation instruction.

The status of RESF when a reset request is generated is shown in Table 16-2.

**Table 16-2. RESF Status When Reset Request Is Generated**

Reset Source	$\overline{\text{RESET}}$ input	Reset by POC	Reset by WDT	Reset by CLM	Reset by LVI
WDTRF	Cleared (0)	Cleared (0)	Set (1)	Held	Held
CLMRF			Held	Set (1)	Held
LVIRF			Held	Held	Set (1)

## CHAPTER 17 CLOCK MONITOR

### 17.1 Functions of Clock Monitor

The clock monitor samples the X1 input clock using the on-chip Ring-OSC, and generates an internal reset signal when the X1 input clock is stopped.

When a reset signal is generated by the clock monitor, bit 1 (CLMRF) of the reset control flag register (RESF) is set to 1. For details of RESF, refer to **CHAPTER 16 RESET FUNCTION**.

The clock monitor automatically stops under the following conditions.

- In STOP mode and during the oscillation stabilization time
- When the X1 input clock is stopped (MSTOP = 1, when CPU operates on the Ring-OSC clock)
- Oscillation stabilization time after reset is released
- When the Ring-OSC clock is stopped

**Remark** MSTOP: Bit 7 of the main OSC control register (MOC)

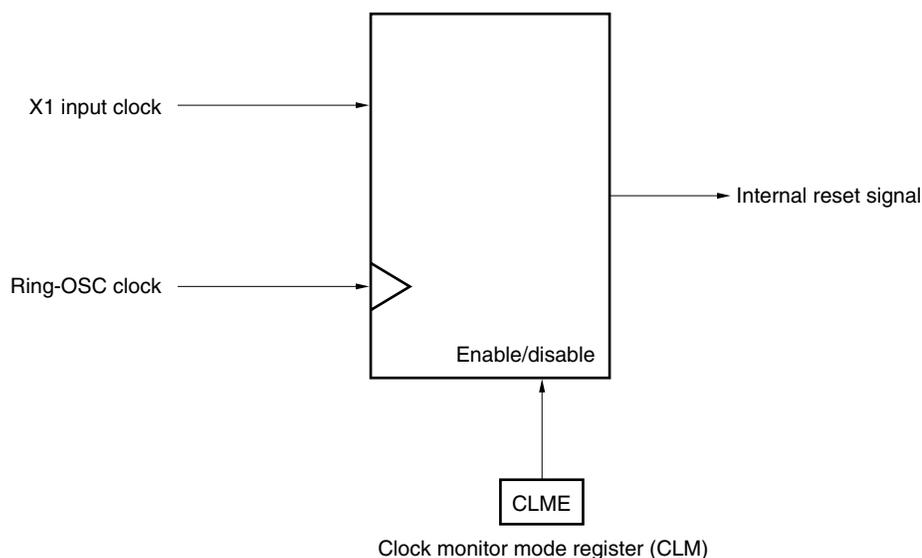
### 17.2 Configuration of Clock Monitor

Clock monitor consists of the following hardware.

**Table 17-1. Configuration of Clock Monitor**

Item	Configuration
Control register	Clock monitor mode register (CLM)

**Figure 17-1. Block Diagram of Clock Monitor**



### 17.3 Registers Controlling Clock Monitor

Clock monitor is controlled by the clock monitor mode register (CLM).

#### (1) Clock monitor mode register (CLM)

This register sets the operation mode of the clock monitor.

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

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

**Figure 17-2. Format of Clock Monitor Mode Register (CLM)**

Address: FFA9H    After reset: 00H    R/W

Symbol	7	6	5	4	3	2	1	<span style="border: 1px solid black; padding: 0 2px;">0</span>
CLM	0	0	0	0	0	0	0	CLME

CLME	Enables/disables clock monitor operation
0	Disables clock monitor operation
1	Enables clock monitor operation

- Cautions**
1. Once bit 0 (CLME) is set to 1, it cannot be cleared to 0 except by  $\overline{\text{RESET}}$  input and the internal reset signal.
  2. If the reset signal is generated by the clock monitor, CLME is cleared to 0 and bit 1 (CLMRF) of the reset control flag register (RESF) is set to 1. CLMRF is read by software and then automatically cleared to 0. CLMRF is cleared under the following conditions.
    - $\overline{\text{RESET}}$  input
    - Internal reset signal generation by POC
    - After read by software

### 17.4 Operation of Clock Monitor

This section explains the functions of the clock monitor. The start and stop conditions are as follows.

<Start condition>

Set bit 0 (CLME) of the clock monitor mode register (CLM) to operation enabled (1).

<Stop condition>

- In STOP mode and during the oscillation stabilization time
- When the X1 input clock is stopped (MSTOP = 1, when the CPU operates on the Ring-OSC clock)
- Oscillation stabilization time after reset is released
- When the Ring-OSC clock is stopped

**Remark** MSTOP: Bit 7 of the main OSC control register (MOC)

**Table 17-2. Operation Status of Clock Monitor (When CLME =1)**

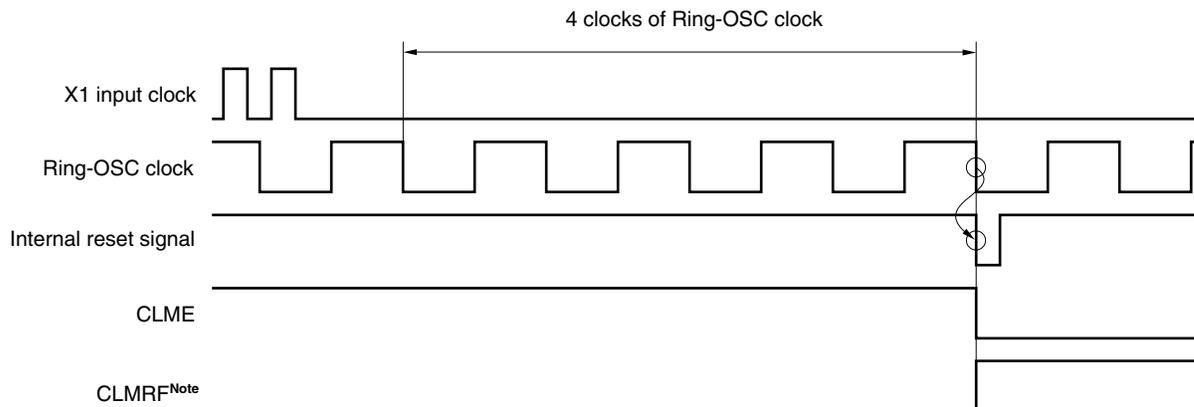
CPU Operation Clock	Operation Mode	X1 Input Clock Status	Ring-OSC Clock Status	Clock Monitor Status		
X1 input clock	STOP mode	Stopped	Oscillating	Stopped		
			Stopped <sup>Note</sup>			
	$\overline{\text{RESET}}$ input		Oscillating			
			Stopped <sup>Note</sup>			
	HALT mode	Oscillating	Oscillating	Operating		
			Stopped <sup>Note</sup>	Stopped		
Ring-OSC clock	STOP mode	Stopped	Oscillating	Stopped		
					$\overline{\text{RESET}}$ input	
	HALT mode					Oscillating
						Stopped

**Note** The Ring-OSC clock is stopped only when the "Ring-OSC can be stopped by software" is selected by a mask option. If "Ring-OSC cannot be stopped" is selected, the Ring-OSC clock cannot be stopped.

The clock monitor timing is as shown in Figure 17-3.

Figure 17-3. Timing of Clock Monitor (1/3)

(1) When internal reset is executed by oscillation stop of X1 input clock

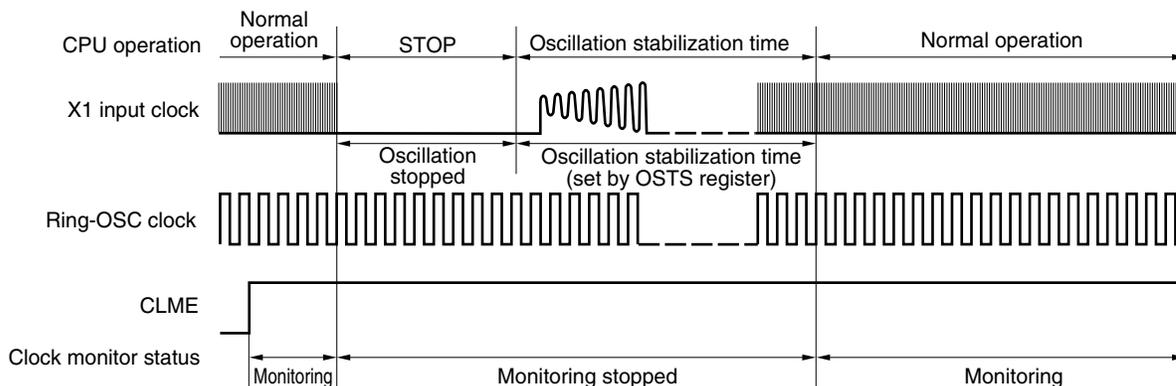


**Note** CLMRF is read by software and then automatically cleared to 0. CLMRF is cleared under the following conditions.

- $\overline{\text{RESET}}$  input
- Internal reset signal generation by POC
- After read by software

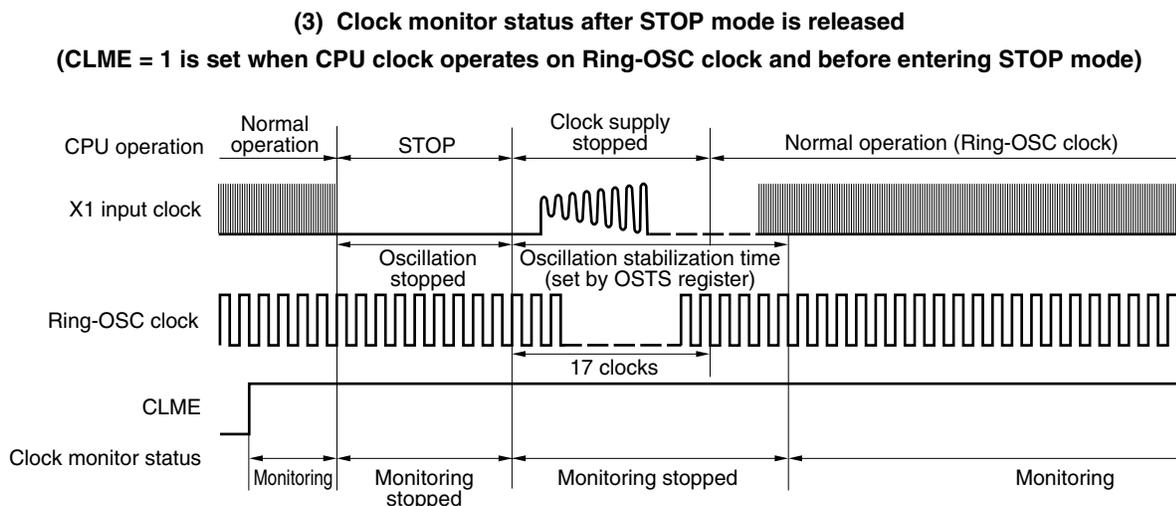
(2) Clock monitor status after STOP mode is released

(CLME = 1 is set when CPU clock operates on X1 input clock and before entering STOP mode)

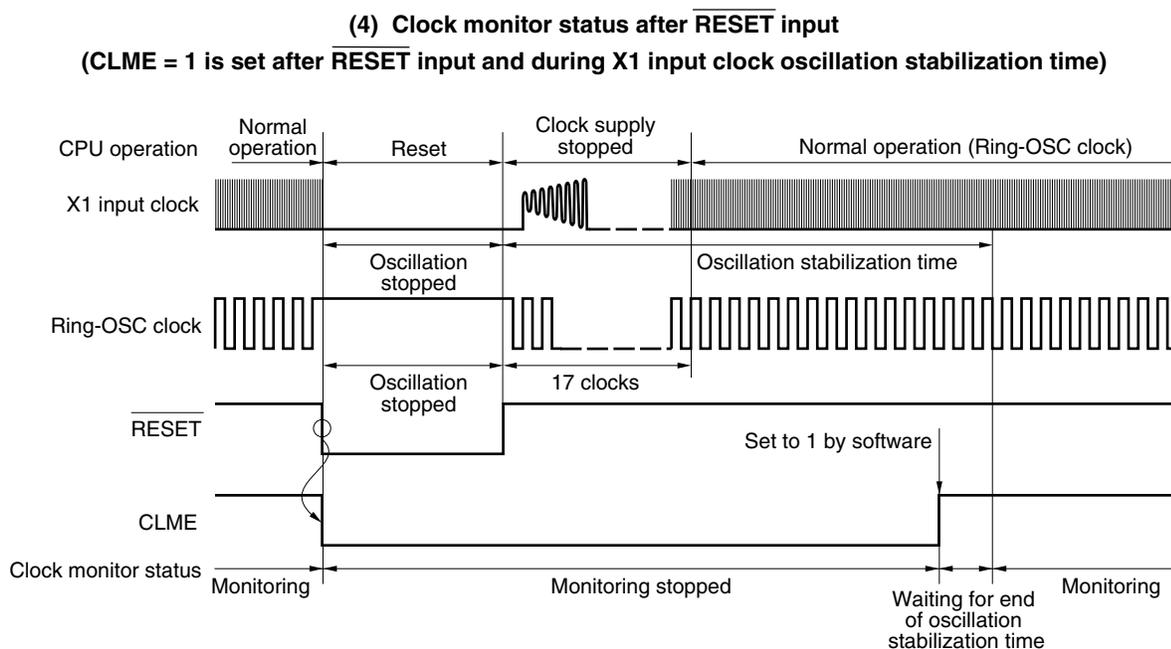


When bit 0 (CLME) of the clock monitor mode register (CLM) is set to 1 before entering STOP mode, monitoring automatically starts at the end of the X1 input clock oscillation stabilization time. Monitoring is stopped in STOP mode and during the oscillation stabilization time.

Figure 17-3. Timing of Clock Monitor (2/3)



When bit 0 (CLME) of the clock monitor mode register (CLM) is set to 1 before entering STOP mode, monitoring automatically starts at the end of the X1 input clock oscillation stabilization time. Monitoring is stopped in STOP mode and during the oscillation stabilization time.

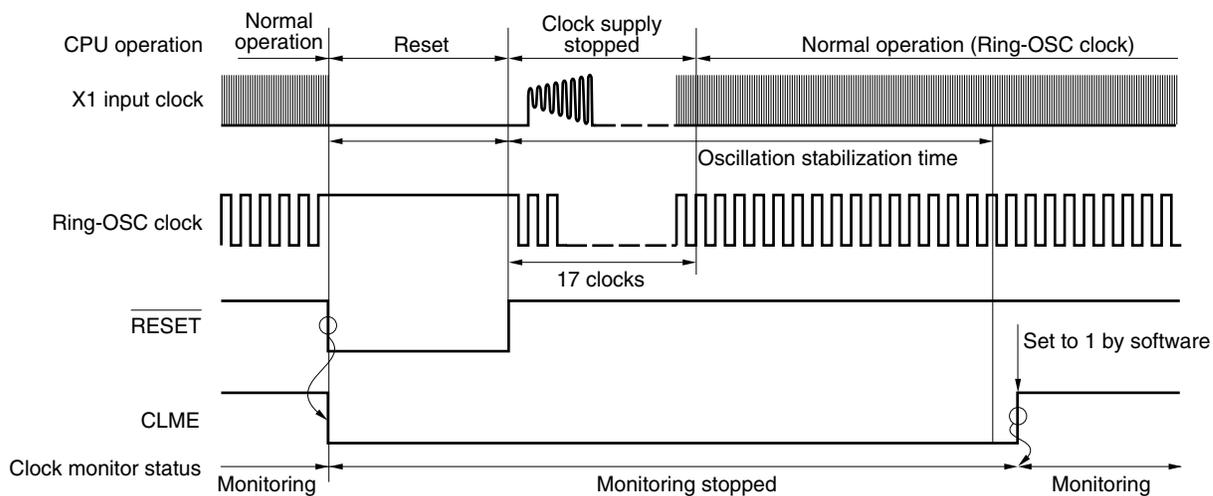


RESET input clears bit 0 (CLME) of the clock monitor mode register (CLM) to 0 and stops the clock monitor operation. Even if CLME is set to 1 by software during the oscillation stabilization time of the X1 input clock, monitoring is not performed until the oscillation stabilization time of the X1 input clock ends. Monitoring is automatically started at the end of the oscillation stabilization time.

Figure 17-3. Timing of Clock Monitor (3/3)

(5) Clock monitor status after  $\overline{\text{RESET}}$  input

(CLME = 1 is set after  $\overline{\text{RESET}}$  input and at the end of X1 input clock oscillation stabilization time)



$\overline{\text{RESET}}$  input clears bit 0 (CLME) of the clock monitor mode register (CLM) to 0 and stops the clock monitor operation. When CLME is set to 1 by software at the end of the oscillation stabilization time of the X1 input clock, monitoring is started.

## CHAPTER 18 POWER-ON-CLEAR CIRCUIT

### 18.1 Functions of Power-on-Clear Circuit

The power-on-clear circuit (POC) has the following functions.

- Generates internal reset signal at power on.
- Compares supply voltage ( $V_{DD}$ ) and detection voltage ( $V_{POC}$ ), and generates internal reset signal when  $V_{DD} < V_{POC}$ .
- The following can be selected by a mask option.
  - POC disabled
  - POC used (detection voltage:  $V_{POC} = 2.85\text{ V} \pm 0.15\text{ V}$ )
  - POC used (detection voltage:  $V_{POC} = 3.5\text{ V} \pm 0.2\text{ V}$ )

**Caution** If an internal reset signal is generated in the POC circuit, the reset control flag register (RESF) is cleared to 00H.

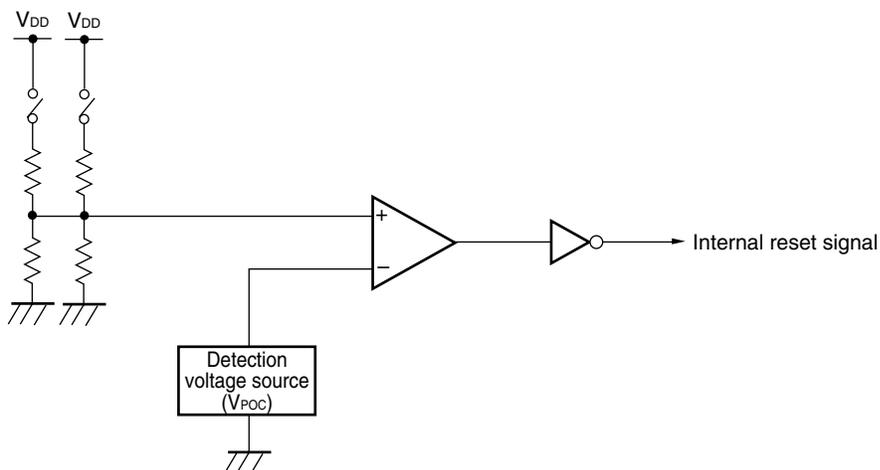
**Remark** This product incorporates multiple hardware functions that generate an internal reset signal. A flag that indicates the reset cause is located in the reset control flag register (RESF) for when an internal reset signal is generated by the watchdog timer (WDT), low-voltage-detection (LVI) circuit, or clock monitor. RESF is not cleared to 00H and the flag is set to 1 when an internal reset signal is generated by WDT, LVI, or the clock monitor.

For details of the RESF, refer to **CHAPTER 16 RESET FUNCTION**.

## 18.2 Configuration of Power-on-Clear Circuit

The block diagram of the power-on-clear circuit is shown in Figure 18-1.

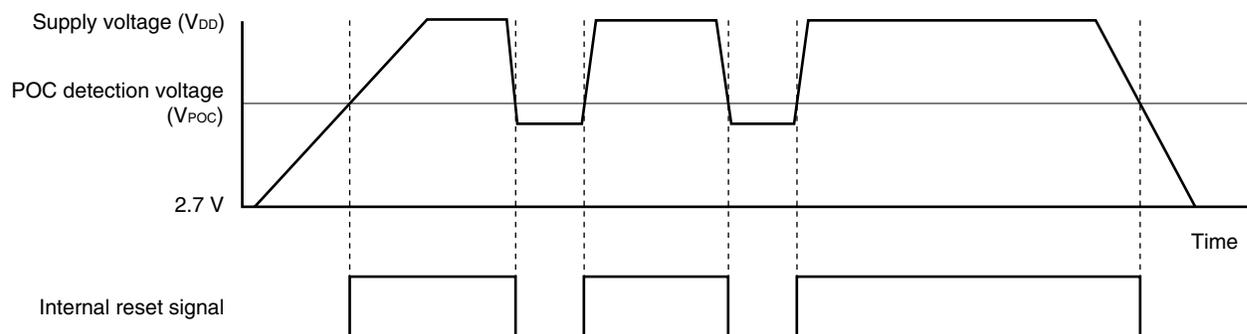
Figure 18-1. Block Diagram of Power-on-Clear Circuit



## 18.3 Operation of Power-on-Clear Circuit

In the power-on-clear circuit, the supply voltage ( $V_{DD}$ ) and detection voltage ( $V_{POC}$ ) are compared, and when  $V_{DD} < V_{POC}$ , an internal reset signal is generated.

Figure 18-2. Timing of Internal Reset Signal Generation in Power-on-Clear Circuit



★ 18.4 Cautions for Power-on-Clear Circuit

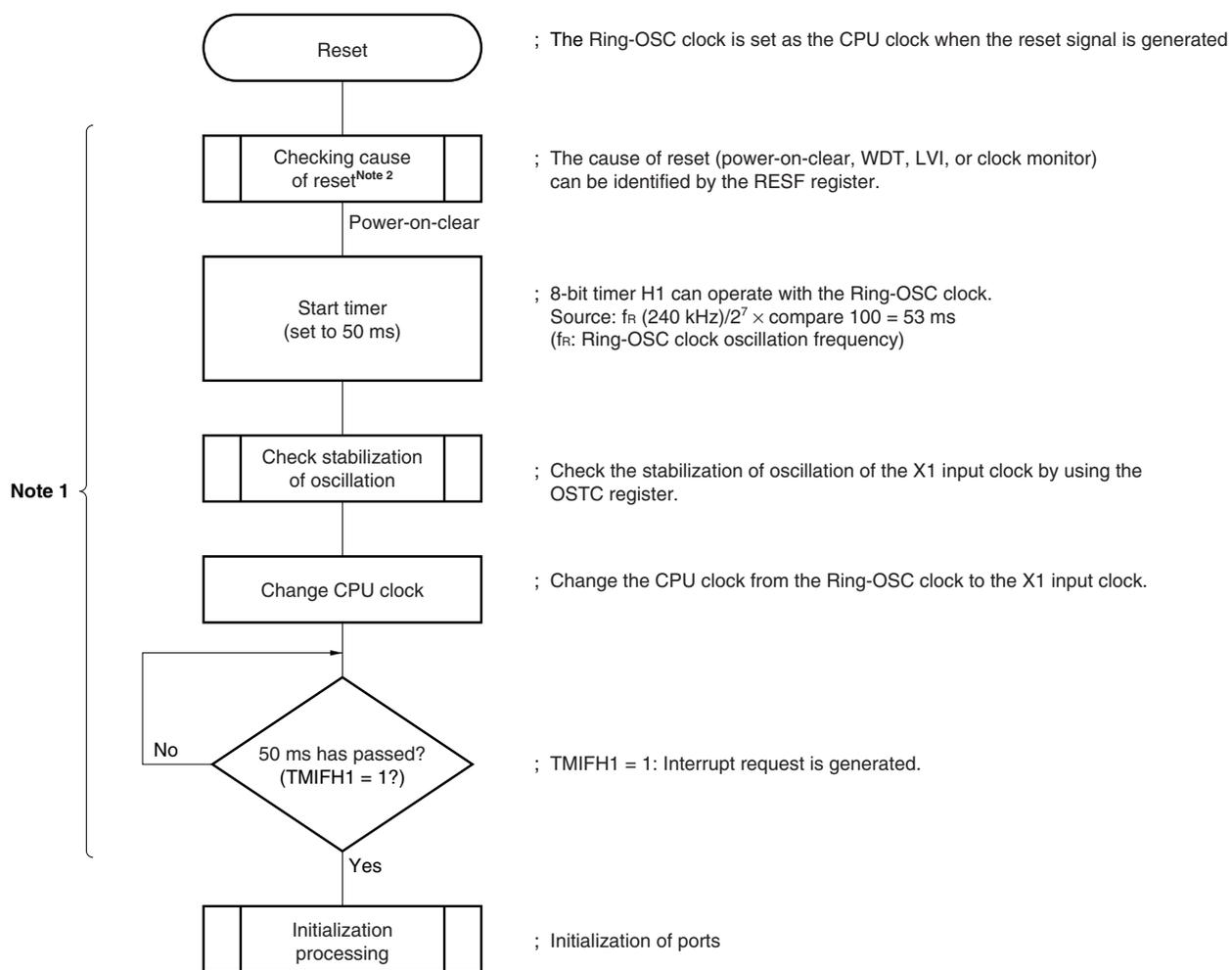
In a system where the supply voltage ( $V_{DD}$ ) fluctuates for a certain period in the vicinity of the POC detection voltage ( $V_{POC}$ ), the system may be repeatedly reset and released from the reset status. In this case, the time from release of reset to the start of the operation of the microcontroller can be arbitrarily set by taking the following action.

<Action>

After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports.

Figure 18-3. Example of Software Processing After Release of Reset (1/2)

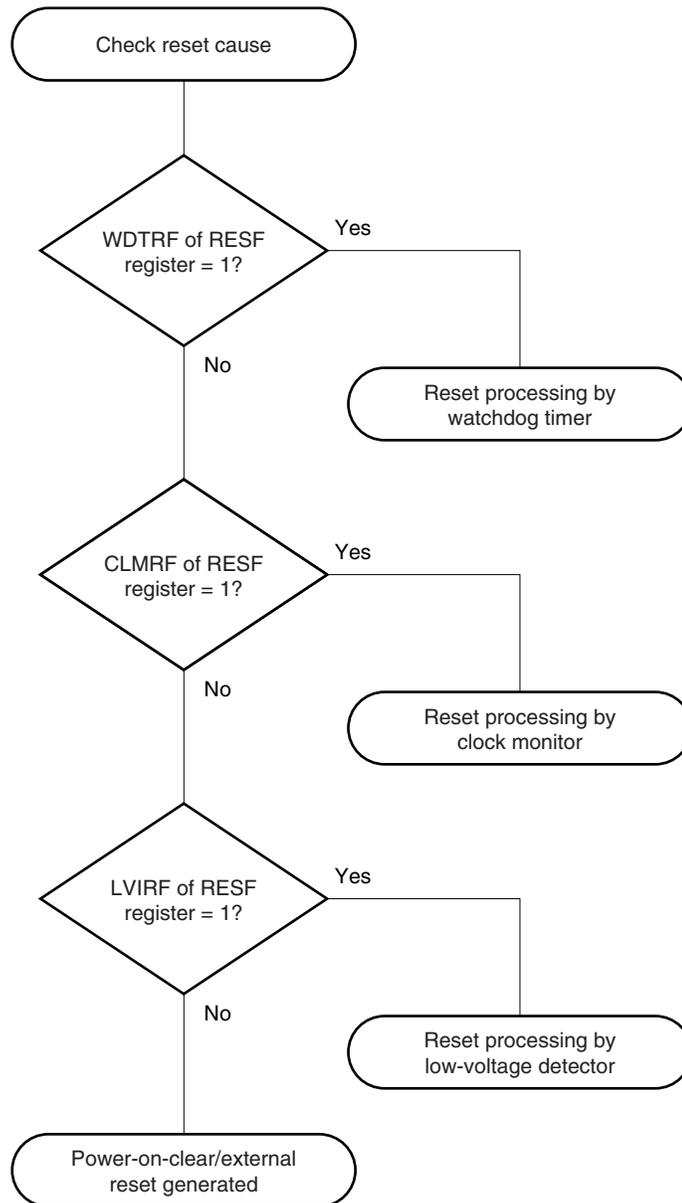
- If supply voltage fluctuation is 50 ms or less in vicinity of POC detection voltage



- Notes**
1. If reset is generated again during this period, initialization processing is not started.
  2. A flowchart is shown on the next page.

Figure 18-3. Example of Software Processing After Release of Reset (2/2)

- Checking reset cause



## CHAPTER 19 LOW-VOLTAGE DETECTOR

### 19.1 Functions of Low-Voltage Detector

The low-voltage detector (LVI) has following functions.

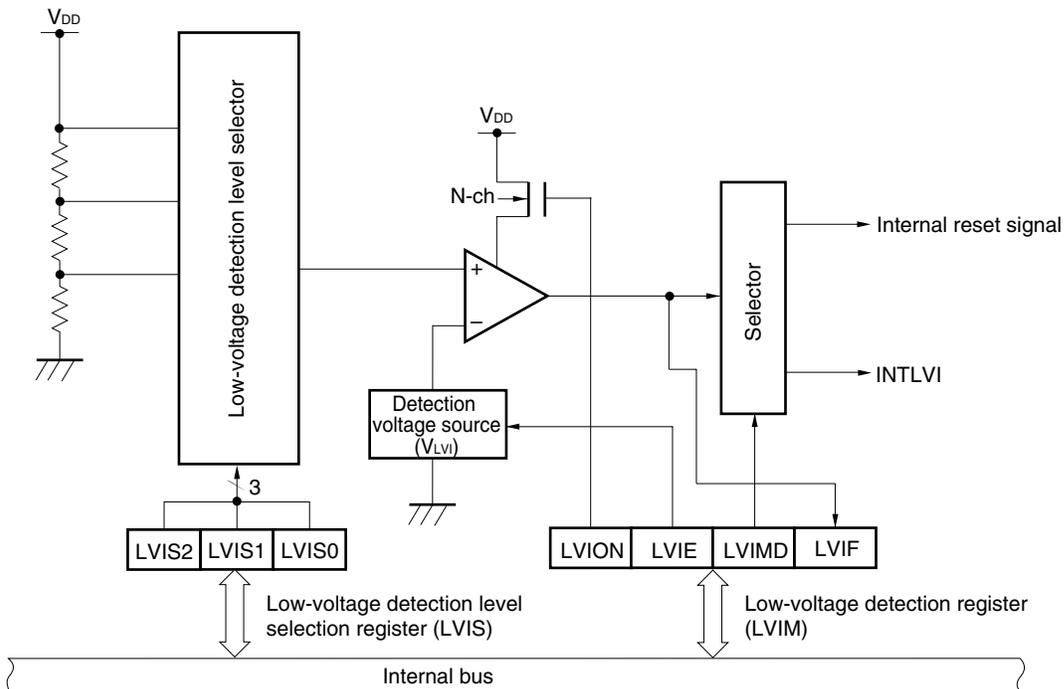
- Compares supply voltage ( $V_{DD}$ ) and detection voltage ( $V_{LVI}$ ), and generates an internal interrupt signal or internal reset signal when  $V_{DD} < V_{LVI}$ .
- Detection levels (five levels) of supply voltage can be changed by software.
- Interrupt or reset function can be selected by software.
- Operable in STOP mode.

When the low-voltage detector is used to reset, bit 0 (LVIRF) of the reset control flag register (RESF) is set to 1 if reset occurs. For details of RESF, refer to **CHAPTER 16 RESET FUNCTION**.

### 19.2 Configuration of Low-Voltage Detector

The block diagram of the low-voltage detector is shown below.

Figure 19-1. Block Diagram of Low-Voltage Detector



### 19.3 Registers Controlling Low-Voltage Detector

The low-voltage detector is controlled by the following registers.

- Low-voltage detection register (LVIM)
- Low-voltage detection level selection register (LVIS)

#### (1) Low-voltage detection register (LVIM)

This register sets low-voltage detection and the operation mode.

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

Figure 19-2. Format of Low-Voltage Detection Register (LVIM)

Address: FFBEH After reset: 00H R/W

Symbol	<span style="border: 1px solid black; padding: 0 2px;">7</span>	6	5	<span style="border: 1px solid black; padding: 0 2px;">4</span>	3	2	<span style="border: 1px solid black; padding: 0 2px;">1</span>	<span style="border: 1px solid black; padding: 0 2px;">0</span>
LVIM	LVION	0	0	LVIE	0	0	LVIMD	LVIF

LVION <sup>Notes 1, 2</sup>	Enables low-voltage detection operation
0	Disables operation
1	Operation starts

LVIE <sup>Notes 1, 3, 4</sup>	Specifies reference voltage generator
0	Disables operation
1	Enables operation

LVIMD <sup>Note 1</sup>	Low-voltage detection operation mode selection
0	Generates interrupt signal when supply voltage ( $V_{DD}$ ) < detection voltage ( $V_{LVI}$ )
1	Generates internal reset signal when supply voltage ( $V_{DD}$ ) < detection voltage ( $V_{LVI}$ )

LVIF <sup>Note 5</sup>	Low-voltage detection flag
0	Supply voltage ( $V_{DD}$ ) > detection voltage ( $V_{LVI}$ ), or when operation is disabled
1	Supply voltage ( $V_{DD}$ ) < detection voltage ( $V_{LVI}$ )

- Notes**
1. LVION, LVIE, and LVIMD are cleared to 0 at a reset other than an LVI reset. These are not cleared to 0 at an LVI reset.
  2. When LVION is set to 1, operation of the comparator in the LVI circuit is started. Use software to instigate a wait of at least 0.2 ms from when LVION is set to 1 until the voltage is confirmed at LVIF.
  3. When LVIE is set to 1, a reference voltage generator operation in the LVI circuit is started. Use software to instigate a wait of at least 2 ms from when LVIE is set to 1 until LVION is set to 1.
  4. If “use POC” is selected by a mask option, leave LVIE as 0. A wait time (2 ms) until LVION is set to 1 is not necessary.
  5. The value of LVIF is output as the interrupt request signal INTLVI when LVION = 1 and LVIMD = 0.

**Caution** To stop LVI, follow either of the procedures below.

- When using 8-bit manipulation instruction: Write 00H to LVIM.
- When using 1-bit memory manipulation instruction: Clear LVION to 0 first and then clear LVIE to 0.

**(2) Low-voltage detection level selection register (LVIS)**

This register selects the low-voltage detection level.

This register can be set by an 8-bit memory manipulation instruction.

★

**Figure 19-3. Format of Low-Voltage Detection Level Selection Register (LVIS)**

Address: FFBFH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LVIS	0	0	0	0	0	LVIS2	LVIS1	LVIS0

LVIS2	LVIS1	LVIS0	Detection level
0	0	0	$V_{LV10}$ (4.3 V $\pm$ 0.2 V)
0	0	1	$V_{LV11}$ (4.1 V $\pm$ 0.2 V)
0	1	0	$V_{LV12}$ (3.9 V $\pm$ 0.2 V)
0	1	1	$V_{LV13}$ (3.7 V $\pm$ 0.2 V)
1	0	0	$V_{LV14}$ (3.5 V $\pm$ 0.2 V) <sup>Note</sup>
1	0	1	$V_{LV15}$ (3.3 V $\pm$ 0.15 V) <sup>Note</sup>
1	1	0	$V_{LV16}$ (3.1 V $\pm$ 0.15 V) <sup>Note</sup>
1	1	1	Setting prohibited

**Note** When the detection voltage of the POC circuit is specified as  $V_{POC} = 3.5 \text{ V} \pm 0.2 \text{ V}$  by a mask option, do not select  $V_{LV14}$  to  $V_{LV16}$  as the LVI detection voltage. Even if  $V_{LV14}$  to  $V_{LV16}$  are selected, POC circuit has priority.

## 19.4 Operation of Low-Voltage Detector

The low-voltage detector can be used in the following two modes.

- Used as reset  
Compares the supply voltage ( $V_{DD}$ ) and detection voltage ( $V_{LVI}$ ), and generates an internal reset signal when  $V_{DD} < V_{LVI}$ .
- Used as interrupt  
Compares the supply voltage ( $V_{DD}$ ) and detection voltage ( $V_{LVI}$ ), and generates an interrupt signal (INTLVI) when  $V_{DD} < V_{LVI}$ .

The operation is set as follows.

(1) When used as reset

- When starting operation

- <1> Mask the LVI interrupt ( $LVIMK = 1$ ).
- <2> Set the detection voltage using bits 2 to 0 ( $LVIS2$  to  $LVIS0$ ) of the low-voltage detection level selection register ( $LVIS$ ).
- <3> Set bit 4 ( $LVIE$ ) of the low-voltage detection register ( $LVIM$ ) to 1 (enables reference voltage generator operation).
- <4> Use software to instigate a wait of at least 2 ms.
- <5> Set bit 7 ( $LVION$ ) of  $LVIM$  to 1 (enables LVI operation).
- <6> Use software to instigate a wait of at least 0.2 ms.
- <7> Confirm that "supply voltage ( $V_{DD}$ ) > detection voltage ( $V_{LVI}$ )" at bit 0 ( $LVIF$ ) of  $LVIM$ .
- <8> Set bit 1 ( $LVIMD$ ) of  $LVIM$  to 1 (generates internal reset signal when supply voltage ( $V_{DD}$ ) < detection voltage ( $V_{LVI}$ )).

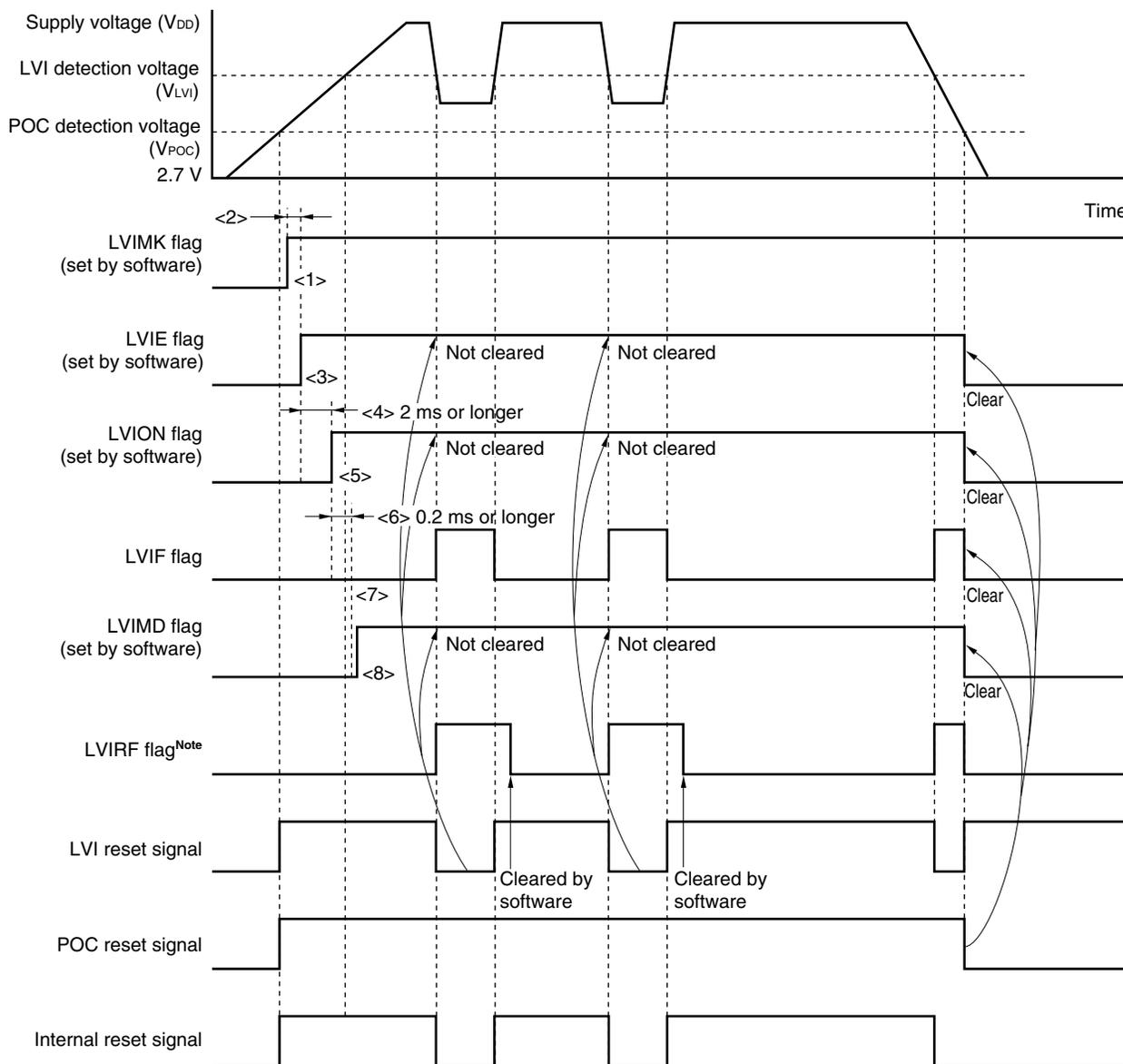
**Cautions 1. <1> must always be executed. When  $LVIMK = 0$ , an interrupt may occur immediately after the processing in <5>.**

**2. If "use POC" is selected by a mask option, procedures <3> and <4> are not required.**

**3. If supply voltage ( $V_{DD}$ ) > detection voltage ( $V_{LVI}$ ) when  $LVIM$  is set to 1, an internal reset signal is not generated.**

- When stopping operation  
Either of the following procedures must be executed.
  - When using 8-bit memory manipulation instruction:  
Write 00H to  $LVIM$ .
  - When using 1-bit memory manipulation instruction:  
Clear  $LVIMD$  to 0,  $LVION$  to 0, and  $LVIE$  to 0 in that order.

Figure 19-4. Timing of Low-Voltage Detector Internal Reset Signal Generation



**Note** LVIRF is bit 0 of the reset control flag register (RESF). For details of RESF, refer to **CHAPTER 16 RESET FUNCTION**.

**Remark**  $\langle 1 \rangle$  to  $\langle 8 \rangle$  in Figure 19-4 above correspond to  $\langle 1 \rangle$  to  $\langle 8 \rangle$  in the description of “when starting operation” in **19.4 (1) When used as reset**.

## (2) When used as interrupt

## • When starting operation

- <1> Mask the LVI interrupt (LVIMK = 1).
- <2> Set the detection voltage using bits 2 to 0 (LVIS2 to LVIS0) of the low-voltage detection level selection register (LVIS).
- <3> Set bit 4 (LVIE) of the low-voltage detection register (LVIM) to 1 (enables reference voltage generator operation).
- <4> Use software to instigate a wait of at least 2 ms.
- <5> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).
- <6> Use software to instigate a wait of at least 0.2 ms.
- <7> Confirm that “supply voltage ( $V_{DD}$ ) > detection voltage ( $V_{LVI}$ )” at bit 0 (LVIF) of LVIM.
- <8> Clear the interrupt request flag of LVI (LVIIIF) to 0.
- <9> Release the interrupt mask flag of LVI (LVIMK).
- <10> Execute the EI instruction (when vector interrupts are used).

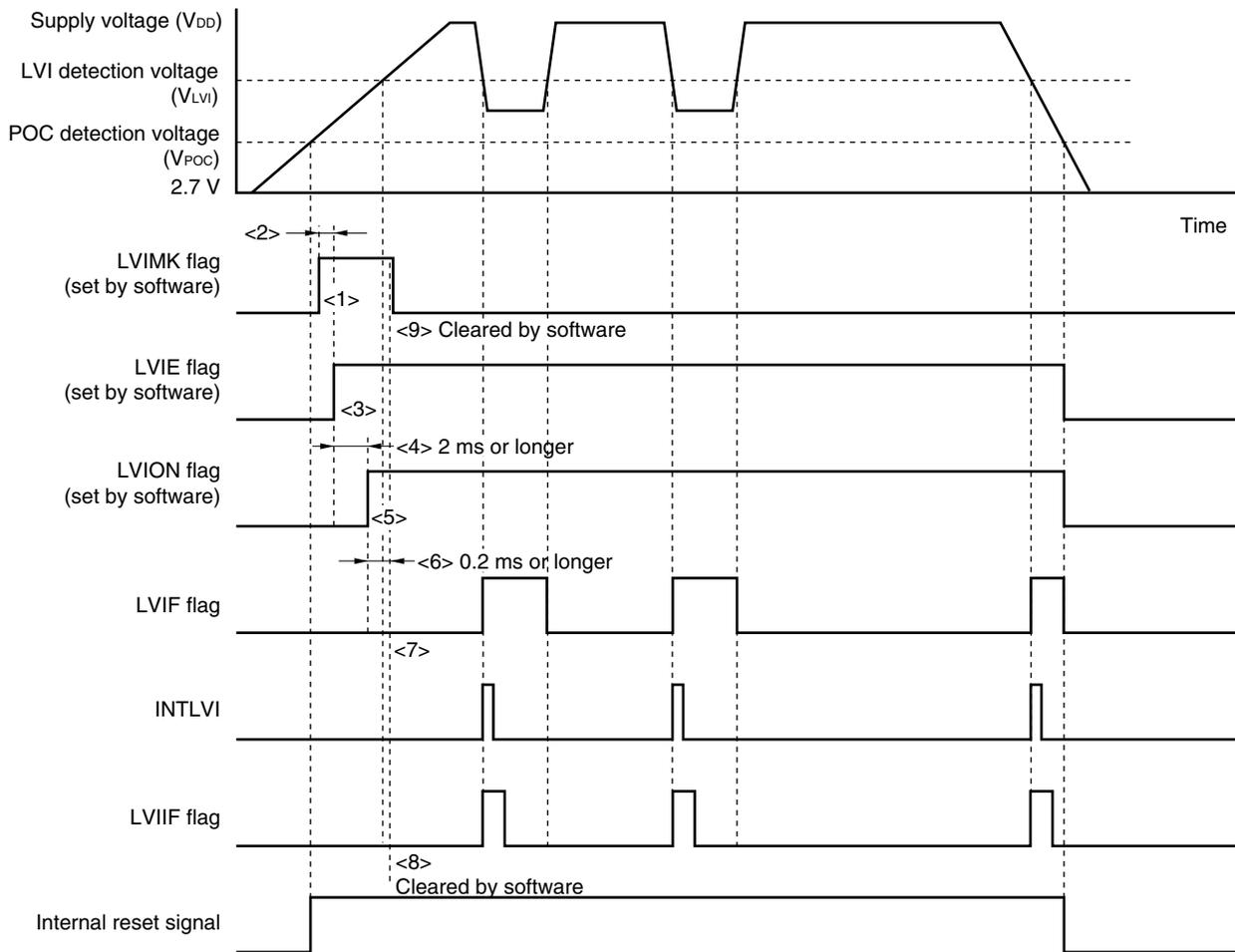
**Caution** If “use POC” is selected by a mask option, procedures <3> and <4> are not required.

## • When stopping operation

Either of the following procedures must be executed.

- When using 8-bit memory manipulation instruction:
  - Write 00H to LVIM.
- When using 1-bit memory manipulation instruction:
  - Clear LVION to 0 first, and then clear LVIE to 0.

Figure 19-5. Timing of Low-Voltage Detector Interrupt Signal Generation



**Remark**  $\langle 1 \rangle$  to  $\langle 9 \rangle$  in Figure 19-5 above correspond to  $\langle 1 \rangle$  to  $\langle 9 \rangle$  in the description of “when starting operation” in 19.4 (2) When used as interrupt.

**★ 19.5 Cautions for Low-Voltage Detector**

In a system where the supply voltage ( $V_{DD}$ ) fluctuates for a certain period in the vicinity of the LVI detection voltage ( $V_{LVI}$ ), the operation is as follows depending on how the low-voltage detector is used.

**(1) When used as reset**

The system may be repeatedly reset and released from the reset status.

In this case, the time from release of reset to the start of the operation of the microcontroller can be arbitrarily set by taking action (1) below.

**(2) When used as interrupt**

Interrupt requests may be frequently generated. Take action (2) below.

In this system, take the following actions.

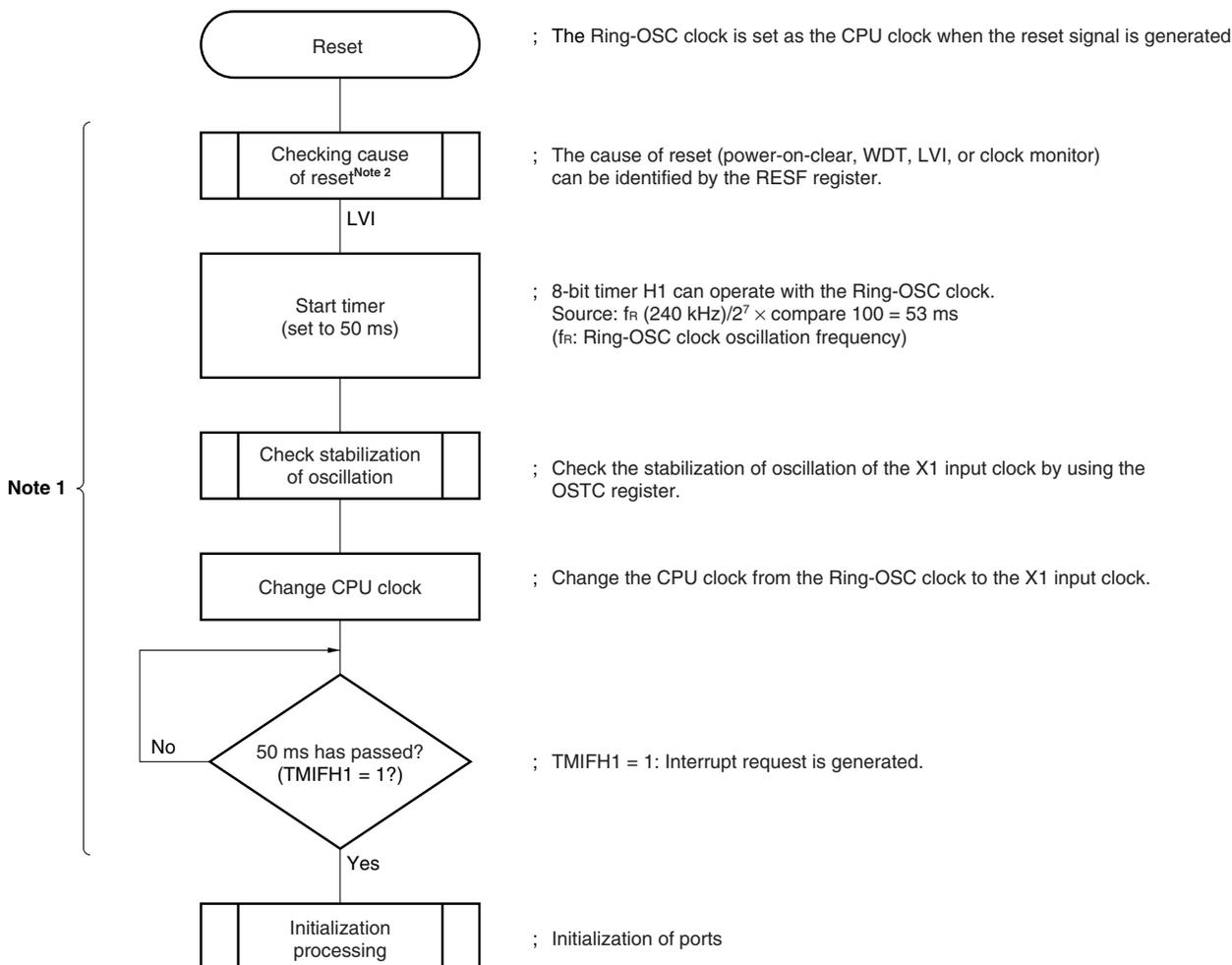
<Action>

**(1) When used as reset**

After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports.

Figure 19-6. Example of Software Processing After Release of Reset (1/2)

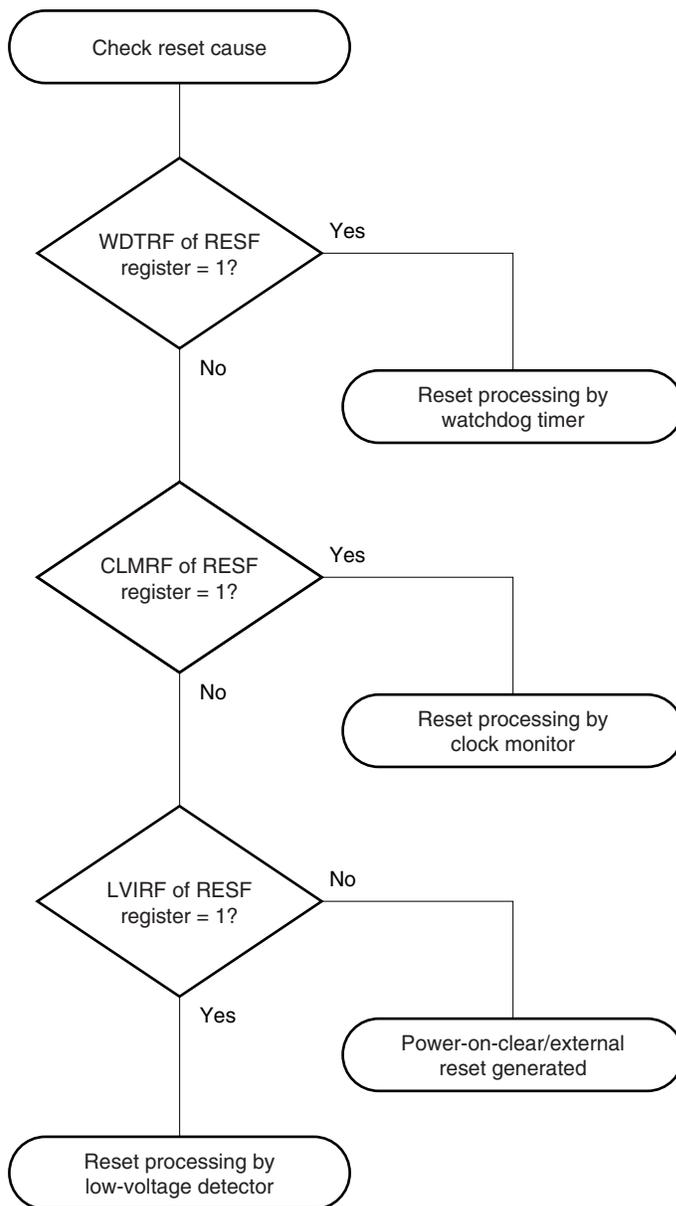
- If supply voltage fluctuation is 50 ms or less in vicinity of LVI detection voltage



- Notes**
1. If reset is generated again during this period, initialization processing is not started.
  2. A flowchart is shown on the next page.

Figure 19-6. Example of Software Processing After Release of Reset (2/2)

- Checking reset cause



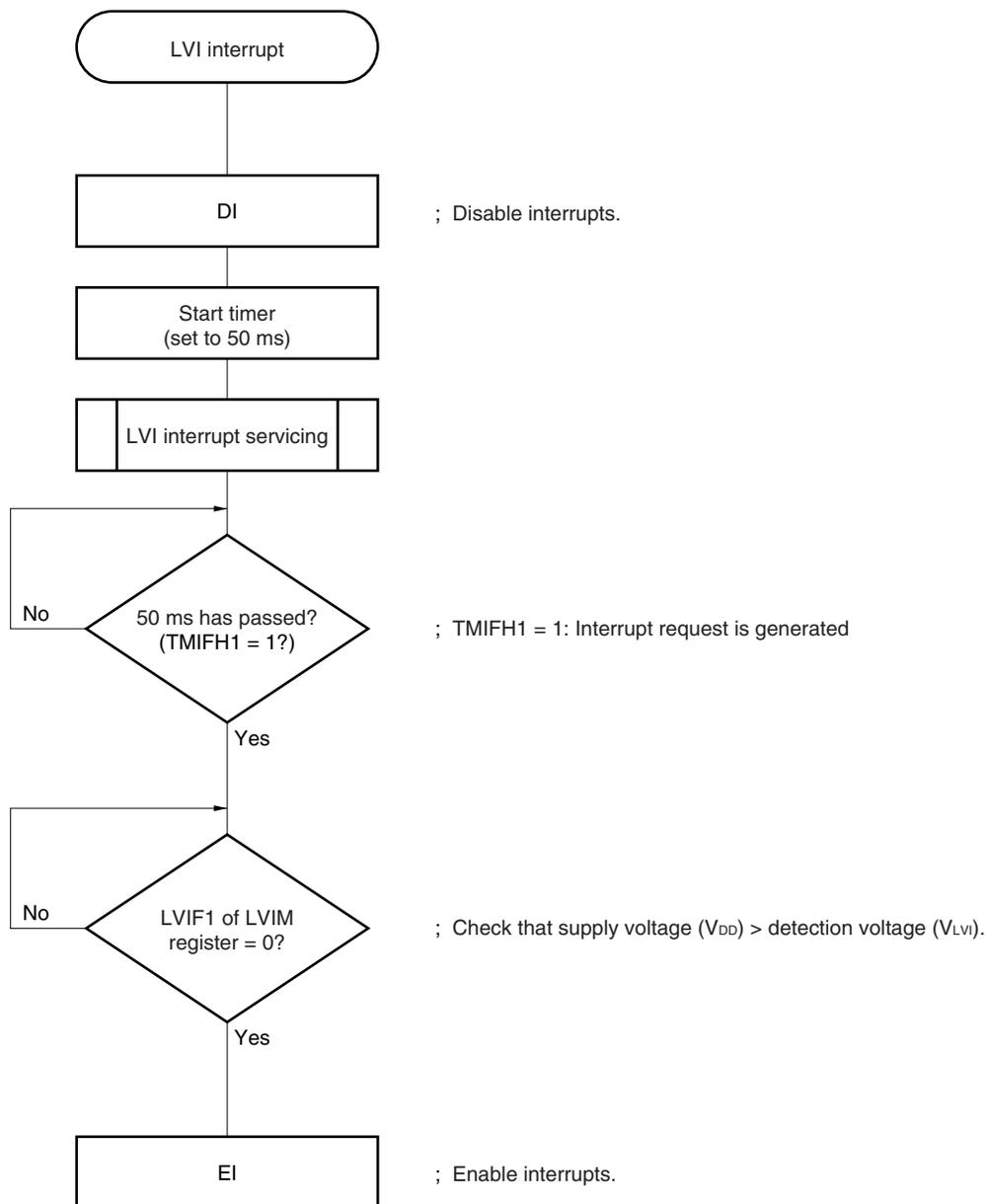
**(2) When used as interrupt**

Disable interrupts (DI) in the servicing routine of the LVI interrupt, and check to see if “supply voltage ( $V_{DD}$ ) > detection voltage ( $V_{LVI}$ )”, by using bit 0 (LVIF) of the low-voltage detection register (LVIM). Then enable interrupts (EI).

In a system where the supply voltage fluctuation period is long in the vicinity of the LVI detection voltage, disable interrupts (DI), wait for the supply voltage fluctuation period, check that “supply voltage ( $V_{DD}$ ) > detection voltage ( $V_{LVI}$ )” with the LVIF flag, and then enable interrupts (EI).

**Figure 19-7. Example of Software Processing of LVI Interrupt**

- If supply voltage fluctuation is 50 ms or less in vicinity of LVI detection voltage



## CHAPTER 20 MASK OPTIONS

Mask ROM versions are provided with the following mask options.

1. Power-on-clear (POC) circuit
  - POC cannot be used
  - POC used (detection voltage:  $V_{POC} = 2.85\text{ V} \pm 0.15\text{ V}$ )
  - POC used (detection voltage:  $V_{POC} = 3.5\text{ V} \pm 0.2\text{ V}$ )
2. Ring-OSC
  - Cannot be stopped
  - Can be stopped by software

Flash memory versions that support the mask options of the mask ROM versions are as follows.

**Table 20-1. Flash Memory Versions Supporting Mask Options of Mask ROM Versions**

Mask Option		Flash Memory Version
POC Circuit	Ring-OSC	
POC cannot be used	Cannot be stopped	$\mu$ PD78F0103M1
	Can be stopped by software	$\mu$ PD78F0103M2
POC used ( $V_{POC} = 2.85\text{ V} \pm 0.15\text{ V}$ )	Cannot be stopped	$\mu$ PD78F0103M3
	Can be stopped by software	$\mu$ PD78F0103M4
POC used ( $V_{POC} = 3.5\text{ V} \pm 0.2\text{ V}$ )	Cannot be stopped	$\mu$ PD78F0103M5
	Can be stopped by software	$\mu$ PD78F0103M6

## CHAPTER 21 $\mu$ PD78F0103

The  $\mu$ PD78F0103 is provided as the flash memory version of the 78K0/KB1 Series.

The  $\mu$ PD78F0103 replaces the internal mask ROM of the  $\mu$ PD780103 with flash memory to which a program can be written, erased, and overwritten while mounted on the board. Table 21-1 lists the differences between the  $\mu$ PD78F0103 and the mask ROM versions.

**Table 21-1. Differences Between  $\mu$ PD78F0103 and Mask ROM Versions**

Item	$\mu$ PD78F0103	Mask ROM Versions
Internal ROM configuration	Flash memory	Mask ROM
Internal ROM capacity	24 KB <sup>Note</sup>	$\mu$ PD780101: 8 KB $\mu$ PD780102: 16 KB $\mu$ PD780103: 24 KB
Internal high-speed RAM capacity	768 bytes <sup>Note</sup>	$\mu$ PD780101: 512 bytes $\mu$ PD780102: 768 bytes $\mu$ PD780103: 768 bytes
IC pin	None	Available
V <sub>PP</sub> pin	Available	None
Electrical specifications	Refer to <b>CHAPTER 23 ELECTRICAL SPECIFICATIONS (TARGET VALUES)</b> .	

**Note** The same capacity as the mask ROM versions can be specified by means of the internal memory size switching register (IMS).

**Caution** There are differences in noise immunity and noise radiation between the flash memory and mask ROM versions. When pre-producing an application set with the flash memory version and then mass-producing it with the mask ROM version, be sure to conduct sufficient evaluations for the commercial samples (not engineering samples) of the mask ROM versions.

## 21.1 Internal Memory Size Switching Register

The  $\mu$ PD78F0103 allows users to select the internal memory capacity using the internal memory size switching register (IMS) so that the same memory map as that of the mask ROM versions with a different internal memory capacity can be achieved.

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

$\overline{\text{RESET}}$  input sets IMS to CFH.

**Caution** The initial value of IMS is “setting prohibited (CFH)”. Be sure to set the value of the relevant mask ROM version at initialization.

**Figure 21-1. Format of Internal Memory Size Switching Register (IMS)**

Address: FFF0H After reset: CFH R/W

Symbol	7	6	5	4	3	2	1	0
IMS	RAM2	RAM1	RAM0	0	ROM3	ROM2	ROM1	ROM0
	RAM2	RAM1	RAM0	Internal high-speed RAM capacity selection				
	0	0	0	768 bytes				
	0	1	0	512 bytes				
	Other			Setting prohibited				
	ROM3	ROM2	ROM1	ROM0	Internal ROM capacity selection			
	0	0	1	0	8 KB			
	0	1	0	0	16 KB			
	0	1	1	0	24 KB			
	Other				Setting prohibited			

The IMS settings required to obtain the same memory map as mask ROM versions are shown in Table 21-2.

**Table 21-2. Internal Memory Size Switching Register Settings**

Target Mask ROM Versions	IMS Setting
$\mu$ PD780101	42H
$\mu$ PD780102	04H
$\mu$ PD780103	06H

**Caution** When using a mask ROM version, be sure to set the value indicated in Table 21-2 to IMS.

## 21.2 Flash Memory Programming

On-board writing of flash memory (with device mounted on target system) is supported.

On-board writing is performed after connecting a dedicated flash programmer (Flashpro III (FL-PR3, PG-FP3)) to the host machine and target system.

Moreover, writing to flash memory can also be performed using a flash memory writing adapter connected to Flashpro III.

**Remark** FL-PR3 is a product of Naito Densai Machida Mfg. Co., Ltd.

### 21.2.1 Selection of communication mode

Writing to flash memory is performed using Flashpro III and serial communication. Select the communication mode for writing from Table 21-3. For the selection of the communication mode, a format like the one shown in Figure 21-2 is used. The communication mode is selected according to the number of  $V_{PP}$  pulses shown in Table 21-3.

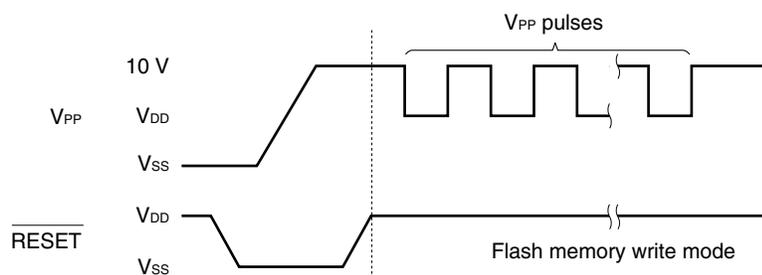
**Table 21-3. Communication Mode List**

Communication Mode	Number of Channels	Pin Used <sup>Note 1</sup>	Number of $V_{PP}$ Pulses
3-wire serial I/O	1	$\overline{SCK10}/\overline{TxD0}/P10$ $SI10/\overline{RxD0}/P11$ $SO10/P12$	0
		$\overline{SCK10}/\overline{TxD0}/P10$ $SI10/\overline{RxD0}/P11$ $SO10/P12$ $HS/P15/TOH0$	3
UART (UART0) <sup>Note 2</sup>	1	$\overline{TxD0}/\overline{SCK10}/P10$ $\overline{RxD0}/SI10/P11$	8
		$\overline{TxD0}/\overline{SCK10}/P10$ $\overline{RxD0}/SI10/P11$ $HS/P15/TOH0$	11
UART (UART6)	1	$\overline{TxD6}/P13$ $\overline{RxD6}/P14$	9

**Notes** 1. After shifting to flash memory programming mode, all pins not used for flash memory programming are set to the same state as after reset. Therefore, since all ports become output high-impedance, pin processing, such as connecting to  $V_{DD}$  or  $V_{SS}$  via a resistor is required if the output high-impedance state is not acknowledged by external devices.

2.  $\mu$ PD780102, 780103, and 78F0103 only.

**Caution** Be sure to select the number of  $V_{PP}$  pulses shown in Table 21-3 for the communication mode.

**Figure 21-2. Communication Mode Selection Format**

### 21.2.2 Flash memory programming function

Flash memory writing is performed via command and data transmit/receive operations using the selected communication mode. The main functions are listed in Table 21-4.

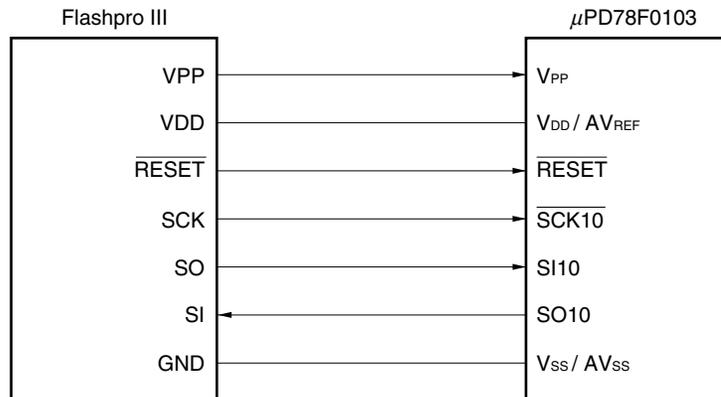
**Table 21-4. Main Functions of Flash Memory Programming**

Function	Description
Reset	Used to detect write stop and transmission synchronization.
Batch verify	Compares entire memory contents and input data.
Batch erase	Erases the entire memory contents.
Batch blank check	Checks the erase status of the entire memory.
High-speed write	Performs writing to flash memory according to write start address and number of write data (bytes).
Continuous write	Performs successive write operations using the data input with high-speed write operation.
Status	Checks the current operation mode and operation end.
Oscillation frequency setting	Inputs the resonator oscillation frequency information.
Delete time setting	Inputs the memory delete time.
Baud rate setting	Sets the communication rate when the UART method is used.
Silicon signature read	Outputs the device name, memory capacity, and device block information.

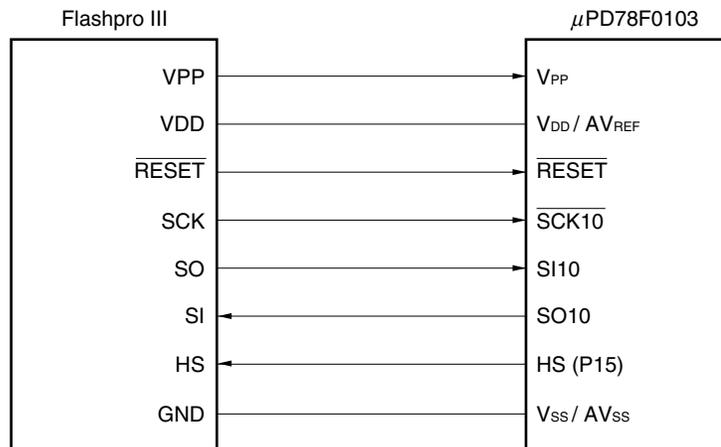
### 21.2.3 Connecting Flashpro III

The connection between Flashpro III and the  $\mu$ PD78F0103 differs depending on the communication mode (3-wire serial I/O or UART). **Figures 21-3 to 21-7** show the connection diagrams of each case.

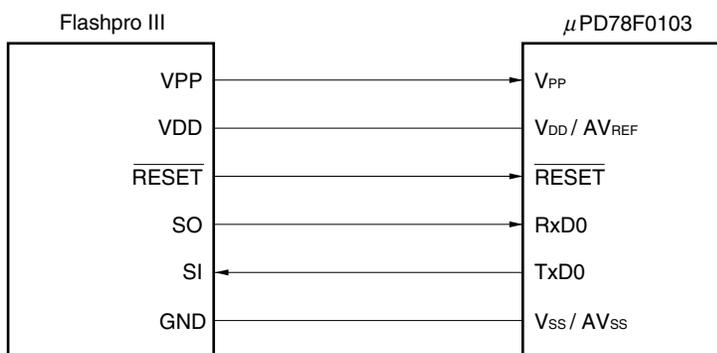
**Figure 21-3. Connection of Flashpro III in 3-Wire Serial I/O Mode**



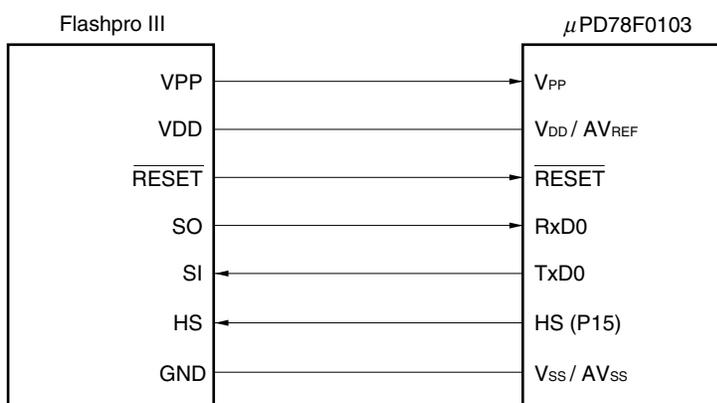
**Figure 21-4. Connection of Flashpro III in 3-Wire Serial I/O Mode (Using Handshake)**



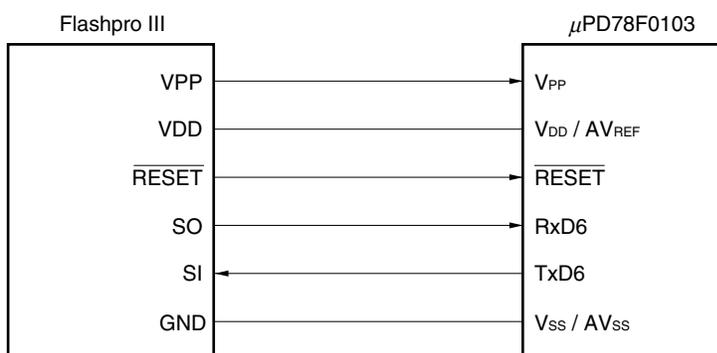
**Figure 21-5. Connection of Flashpro III in UART (UART0) Mode**



**Figure 21-6. Connection of Flashpro III in UART (UART0) Mode (Using Handshake)**



**Figure 21-7. Connection of Flashpro III in UART (UART6) Mode**



21.2.4 Connection on adapter for flash memory writing

Examples of the recommended connection when using the adapter for flash memory writing are shown below.

Figure 21-8. Example of Wiring Adapter for Flash Memory Writing in 3-Wire Serial I/O Mode

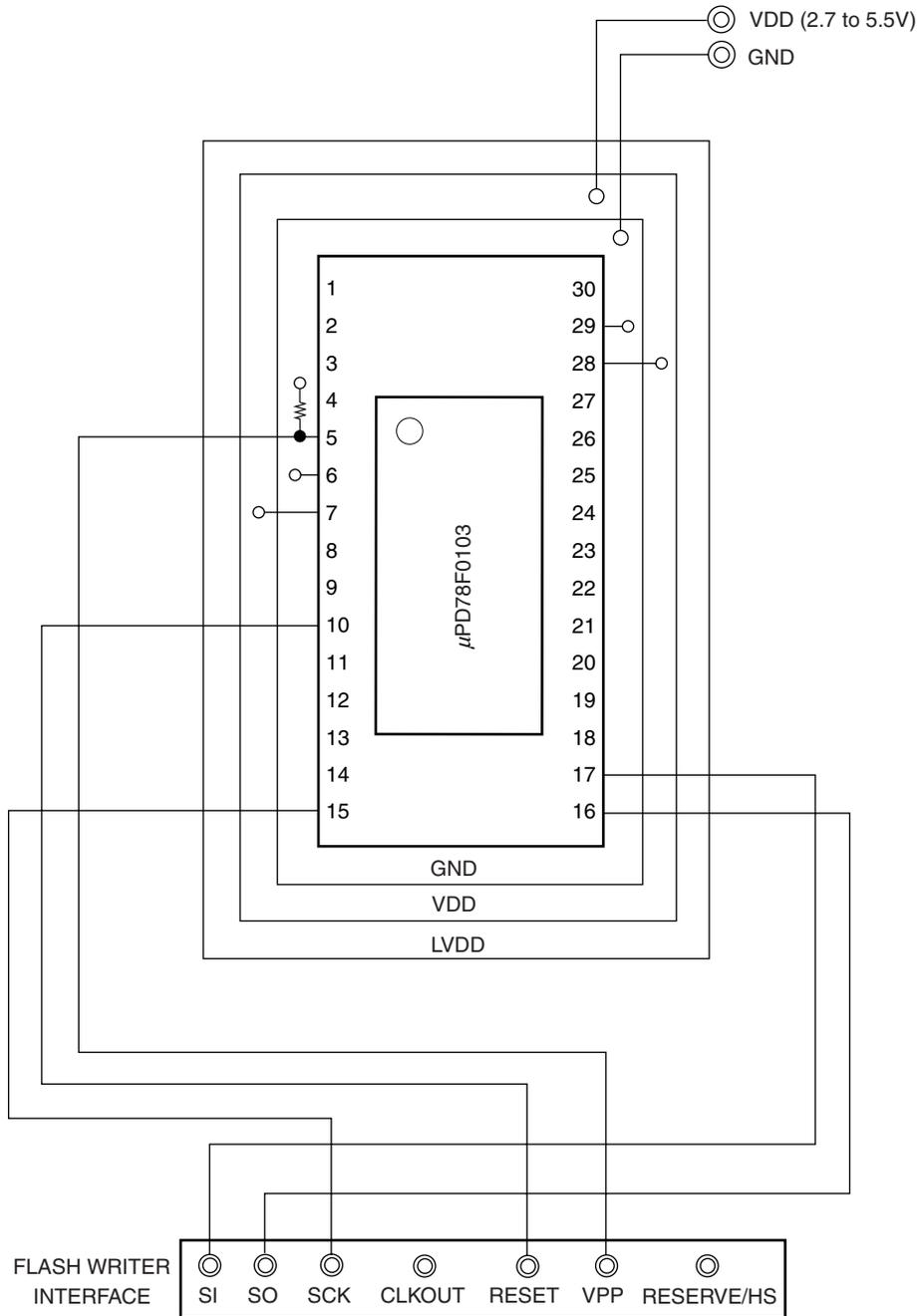


Figure 21-9. Example of Wiring Adapter for Flash Memory Writing in 3-Wire Serial I/O Mode (Using Handshake)

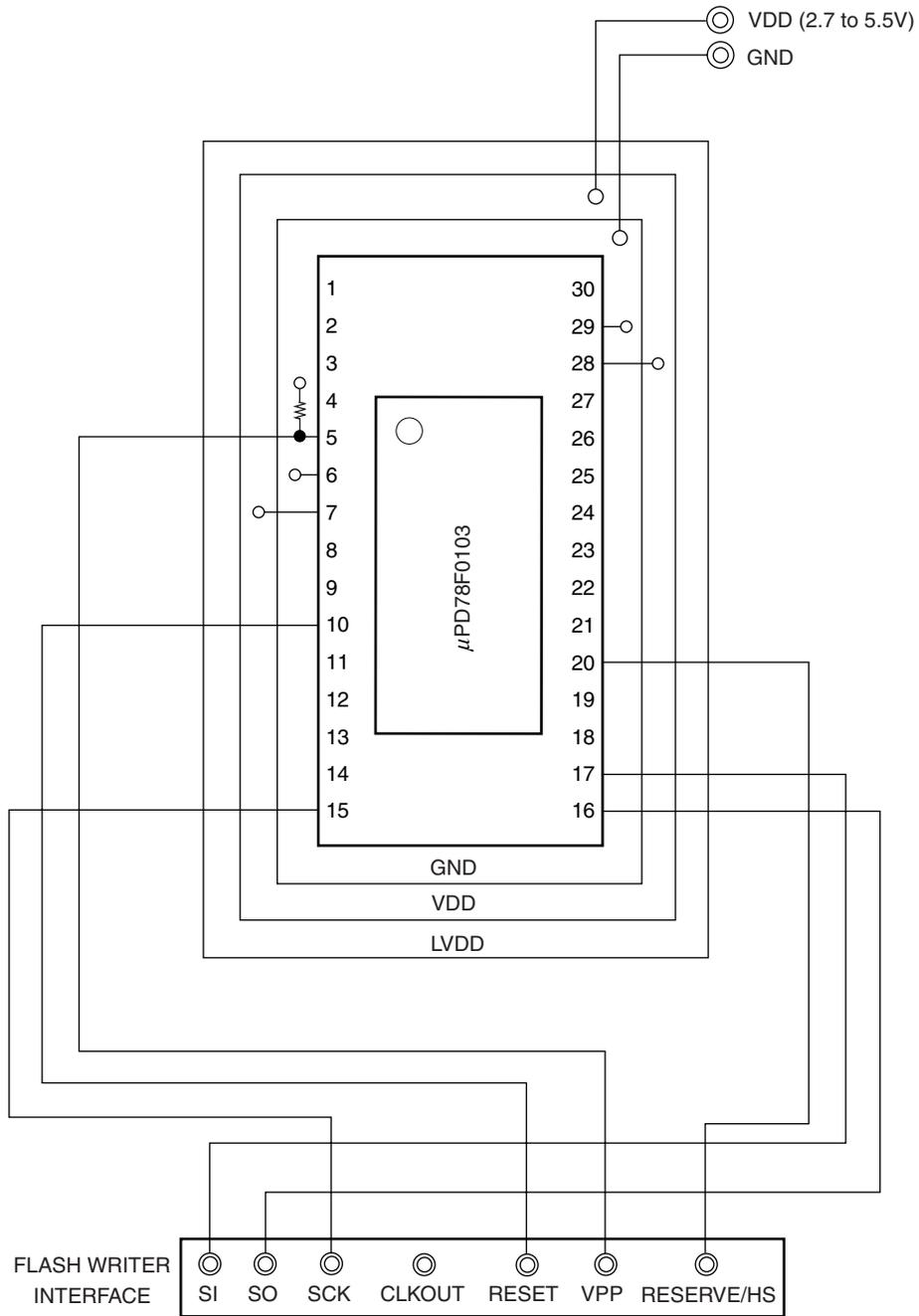


Figure 21-10. Example of Wiring Adapter for Flash Memory Writing in UART Mode (UART0)

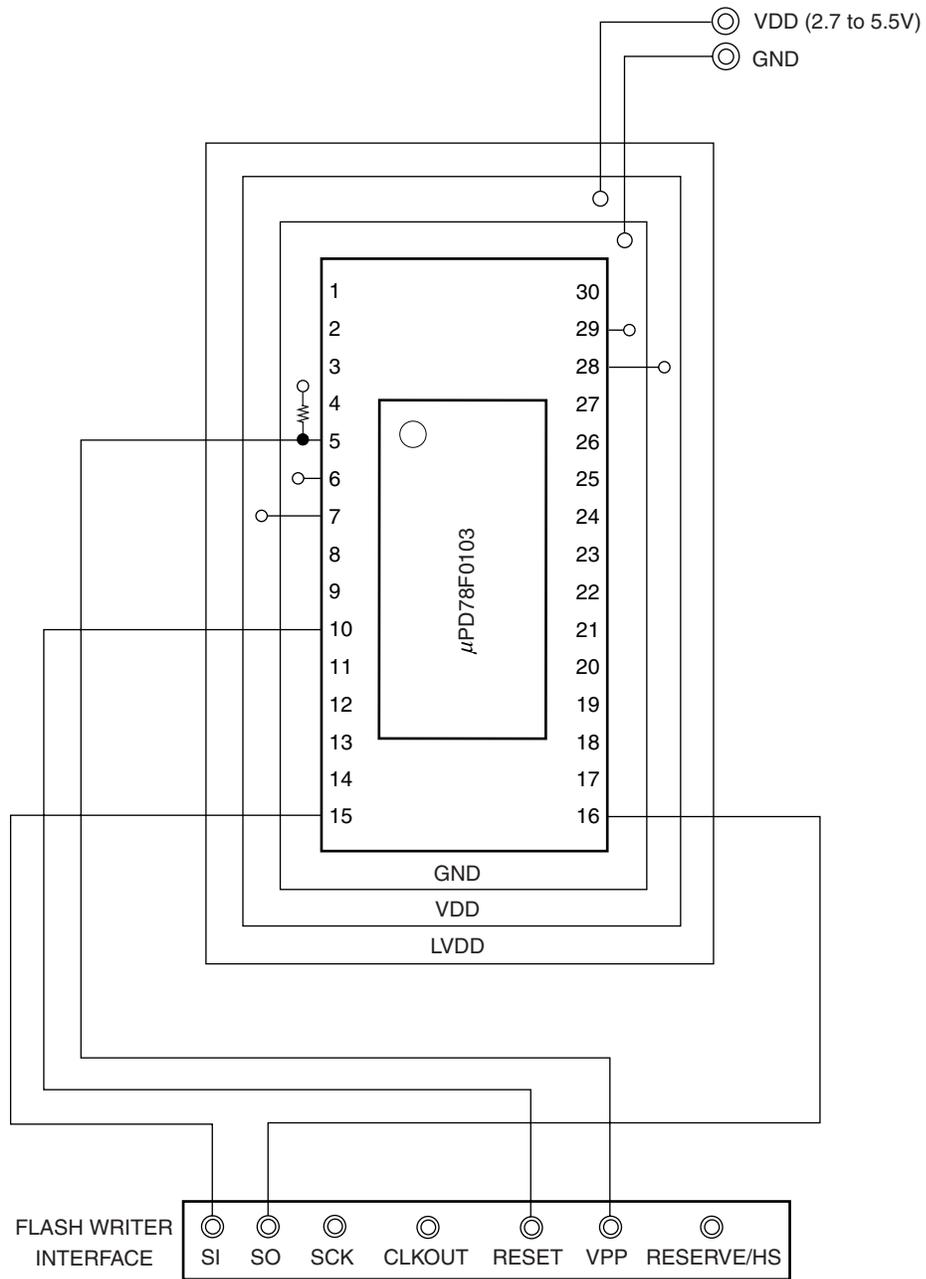


Figure 21-11. Example of Wiring Adapter for Flash Memory Writing in UART Mode (UART0)  
(Using Handshake)

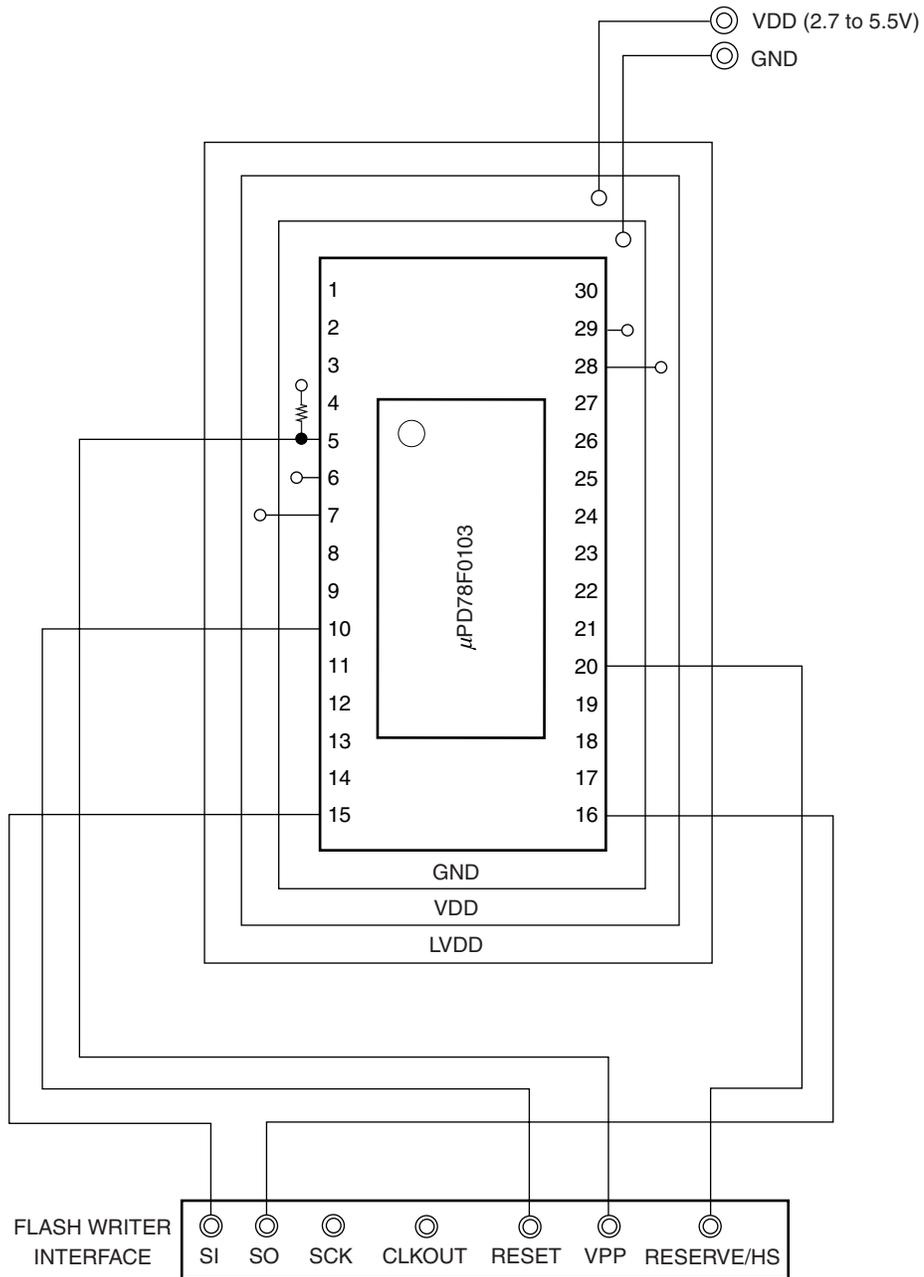
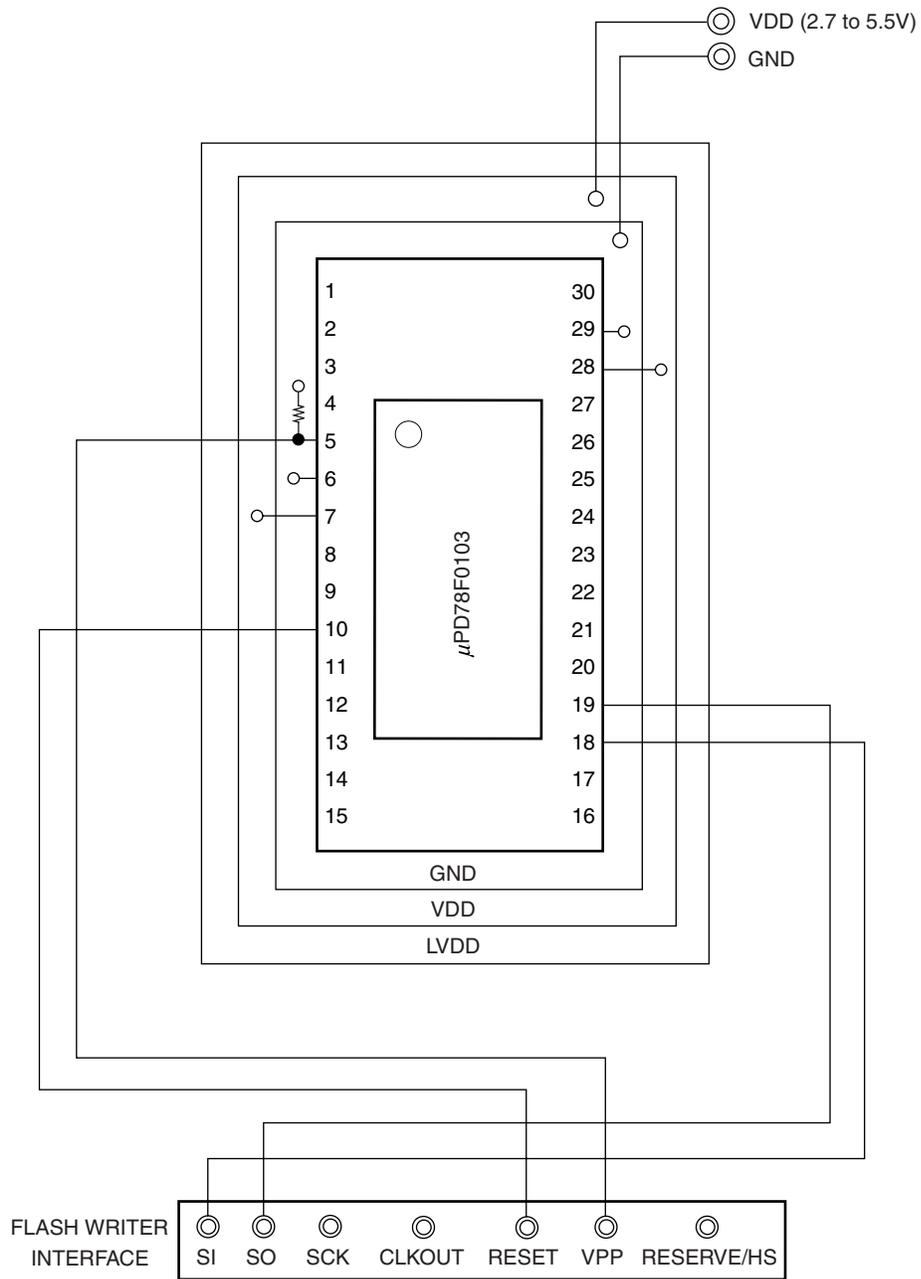


Figure 21-12. Example of Wiring Adapter for Flash Memory Writing in UART Mode (UART6)



## CHAPTER 22 INSTRUCTION SET

This chapter lists each instruction set of the 78K0/KB1 Series in table form. For details of each operation and operation code, refer to the separate document **78K/0 Series Instruction User's Manual (U12326E)**.

### 22.1 Conventions Used in Operation List

#### 22.1.1 Operand identifiers and specification methods

Operands are written in the "Operand" column of each instruction in accordance with the specification method of the instruction operand identifier (refer to the assembler specifications for details). When there are two or more methods, select one of them. Upper case letters and the symbols #, !, \$ and [ ] are keywords and must be written as they are. Each symbol has the following meaning.

- #: Immediate data specification
- !: Absolute address specification
- \$: Relative address specification
- [ ]: Indirect address specification

In the case of immediate data, describe an appropriate numeric value or a label. When using a label, be sure to write the #, !, \$, and [ ] symbols.

For operand register identifiers r and rp, either function names (X, A, C, etc.) or absolute names (names in parentheses in the table below, R0, R1, R2, etc.) can be used for specification.

**Table 22-1. Operand Identifiers and Specification Methods**

Identifier	Specification Method
r	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7),
rp	AX (RP0), BC (RP1), DE (RP2), HL (RP3)
sfr	Special function register symbol <sup>Note</sup>
sfrp	Special function register symbol (16-bit manipulatable register even addresses only) <sup>Note</sup>
saddr	FE20H to FF1FH Immediate data or labels
saddrp	FE20H to FF1FH Immediate data or labels (even address only)
addr16	0000H to FFFFH Immediate data or labels (Only even addresses for 16-bit data transfer instructions)
addr11	0800H to 0FFFH Immediate data or labels
addr5	0040H to 007FH Immediate data or labels (even address only)
word	16-bit immediate data or label
byte	8-bit immediate data or label
bit	3-bit immediate data or label
RBn	RB0 to RB3

**Note** Addresses from FFD0H to FFDFH cannot be accessed with these operands.

**Remark** For special function register symbols, refer to **Table 3-5 Special Function Register List**.

**22.1.2 Description of operation column**

A:	A register; 8-bit accumulator
X:	X register
B:	B register
C:	C register
D:	D register
E:	E register
H:	H register
L:	L register
AX:	AX register pair; 16-bit accumulator
BC:	BC register pair
DE:	DE register pair
HL:	HL register pair
PC:	Program counter
SP:	Stack pointer
PSW:	Program status word
CY:	Carry flag
AC:	Auxiliary carry flag
Z:	Zero flag
RBS:	Register bank select flag
IE:	Interrupt request enable flag
NMIS:	Non-maskable interrupt servicing flag
( ):	Memory contents indicated by address or register contents in parentheses
X <sub>H</sub> , X <sub>L</sub> :	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
jdsp8:	Signed 8-bit data (displacement value)

**22.1.3 Description of flag operation column**

(Blank):	Not affected
0:	Cleared to 0
1:	Set to 1
×	Set/cleared according to the result
R:	Previously saved value is restored

22.2 Operation List

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag			
				Note 1	Note 2		Z	AC	CY	
8-bit data transfer	<b>MOV</b>	r, #byte	2	4	–	r ← byte				
		saddr, #byte	3	6	7	(saddr) ← byte				
		sfr, #byte	3	–	7	sfr ← byte				
		A, r	Note 3	1	2	–	A ← r			
		r, A	Note 3	1	2	–	r ← A			
		A, saddr		2	4	5	A ← (saddr)			
		saddr, A		2	4	5	(saddr) ← A			
		A, sfr		2	–	5	A ← sfr			
		sfr, A		2	–	5	sfr ← A			
		A, !addr16		3	8	9 + n	A ← (addr16)			
		!addr16, A		3	8	9 + m	(addr16) ← A			
		PSW, #byte		3	–	7	PSW ← byte	×	×	×
		A, PSW		2	–	5	A ← PSW			
		PSW, A		2	–	5	PSW ← A	×	×	×
		A, [DE]		1	4	5 + n	A ← (DE)			
		[DE], A		1	4	5 + m	(DE) ← A			
		A, [HL]		1	4	5 + n	A ← (HL)			
		[HL], A		1	4	5 + m	(HL) ← A			
		A, [HL + byte]		2	8	9 + n	A ← (HL + byte)			
		[HL + byte], A		2	8	9 + m	(HL + byte) ← A			
	A, [HL + B]		1	6	7 + n	A ← (HL + B)				
	[HL + B], A		1	6	7 + m	(HL + B) ← A				
	A, [HL + C]		1	6	7 + n	A ← (HL + C)				
	[HL + C], A		1	6	7 + m	(HL + C) ← A				
	<b>XCH</b>	A, r	Note 3	1	2	–	A ↔ r			
		A, saddr		2	4	6	A ↔ (saddr)			
		A, sfr		2	–	6	A ↔ (sfr)			
		A, !addr16		3	8	10 + n + m	A ↔ (addr16)			
		A, [DE]		1	4	6 + n + m	A ↔ (DE)			
		A, [HL]		1	4	6 + n + m	A ↔ (HL)			
		A, [HL + byte]		2	8	10 + n + m	A ↔ (HL + byte)			
		A, [HL + B]		2	8	10 + n + m	A ↔ (HL + B)			
A, [HL + C]		2	8	10 + n + m	A ↔ (HL + C)					

- Notes**
1. When the internal high-speed RAM area is accessed or for an instruction with no data access
  2. When an area except the internal high-speed RAM area is accessed
  3. Except “r = A”

- Remarks**
1. One instruction clock cycle is one cycle of the CPU clock (f<sub>CPU</sub>) selected by the processor clock control register (PCC).
  2. This clock cycle applies to the internal ROM program.
  3. n is the number of waits when the external memory expansion area is read.
  4. m is the number of waits when the external memory expansion area is written.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit data transfer	<b>MOVW</b>	rp, #word	3	6	–	rp ← word			
		saddrp, #word	4	8	10	(saddrp) ← word			
		sfrp, #word	4	–	10	sfrp ← word			
		AX, saddrp	2	6	8	AX ← (saddrp)			
		saddrp, AX	2	6	8	(saddrp) ← AX			
		AX, sfrp	2	–	8	AX ← sfrp			
		sfrp, AX	2	–	8	sfrp ← AX			
		AX, rp <small>Note 3</small>	1	4	–	AX ← rp			
		rp, AX <small>Note 3</small>	1	4	–	rp ← AX			
		AX, !addr16	3	10	12 + 2n	AX ← (addr16)			
	!addr16, AX	3	10	12 + 2m	(addr16) ← AX				
	<b>XCHW</b>	AX, rp <small>Note 3</small>	1	4	–	AX ↔ rp			
8-bit operation	<b>ADD</b>	A, #byte	2	4	–	A, CY ← A + byte	x	x	x
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) + byte	x	x	x
		A, r <small>Note 4</small>	2	4	–	A, CY ← A + r	x	x	x
		r, A	2	4	–	r, CY ← r + A	x	x	x
		A, saddr	2	4	5	A, CY ← A + (saddr)	x	x	x
		A, !addr16	3	8	9 + n	A, CY ← A + (addr16)	x	x	x
		A, [HL]	1	4	5 + n	A, CY ← A + (HL)	x	x	x
		A, [HL + byte]	2	8	9 + n	A, CY ← A + (HL + byte)	x	x	x
		A, [HL + B]	2	8	9 + n	A, CY ← A + (HL + B)	x	x	x
		A, [HL + C]	2	8	9 + n	A, CY ← A + (HL + C)	x	x	x
	<b>ADDC</b>	A, #byte	2	4	–	A, CY ← A + byte + CY	x	x	x
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) + byte + CY	x	x	x
		A, r <small>Note 4</small>	2	4	–	A, CY ← A + r + CY	x	x	x
		r, A	2	4	–	r, CY ← r + A + CY	x	x	x
		A, saddr	2	4	5	A, CY ← A + (saddr) + CY	x	x	x
		A, !addr16	3	8	9 + n	A, CY ← A + (addr16) + C	x	x	x
		A, [HL]	1	4	5 + n	A, CY ← A + (HL) + CY	x	x	x
		A, [HL + byte]	2	8	9 + n	A, CY ← A + (HL + byte) + CY	x	x	x
		A, [HL + B]	2	8	9 + n	A, CY ← A + (HL + B) + CY	x	x	x
A, [HL + C]	2	8	9 + n	A, CY ← A + (HL + C) + CY	x	x	x		

- Notes**
1. When the internal high-speed RAM area is accessed or for an instruction with no data access
  2. When an area except the internal high-speed RAM area is accessed
  3. Only when rp = BC, DE or HL
  4. Except “r = A”

- Remarks**
1. One instruction clock cycle is one cycle of the CPU clock (f<sub>CPU</sub>) selected by the processor clock control register (PCC).
  2. This clock cycle applies to the internal ROM program.
  3. n is the number of waits when the external memory expansion area is read.
  4. m is the number of waits when the external memory expansion area is written.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	<b>SUB</b>	A, #byte	2	4	–	A, CY ← A – byte	x	x	x
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) – byte	x	x	x
		A, r <small>Note 3</small>	2	4	–	A, CY ← A – r	x	x	x
		r, A	2	4	–	r, CY ← r – A	x	x	x
		A, saddr	2	4	5	A, CY ← A – (saddr)	x	x	x
		A, !addr16	3	8	9 + n	A, CY ← A – (addr16)	x	x	x
		A, [HL]	1	4	5 + n	A, CY ← A – (HL)	x	x	x
		A, [HL + byte]	2	8	9 + n	A, CY ← A – (HL + byte)	x	x	x
		A, [HL + B]	2	8	9 + n	A, CY ← A – (HL + B)	x	x	x
	A, [HL + C]	2	8	9 + n	A, CY ← A – (HL + C)	x	x	x	
	<b>SUBC</b>	A, #byte	2	4	–	A, CY ← A – byte – CY	x	x	x
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) – byte – CY	x	x	x
		A, r <small>Note 3</small>	2	4	–	A, CY ← A – r – CY	x	x	x
		r, A	2	4	–	r, CY ← r – A – CY	x	x	x
		A, saddr	2	4	5	A, CY ← A – (saddr) – CY	x	x	x
		A, !addr16	3	8	9 + n	A, CY ← A – (addr16) – CY	x	x	x
		A, [HL]	1	4	5 + n	A, CY ← A – (HL) – CY	x	x	x
		A, [HL + byte]	2	8	9 + n	A, CY ← A – (HL + byte) – CY	x	x	x
		A, [HL + B]	2	8	9 + n	A, CY ← A – (HL + B) – CY	x	x	x
	A, [HL + C]	2	8	9 + n	A, CY ← A – (HL + C) – CY	x	x	x	
	<b>AND</b>	A, #byte	2	4	–	A ← A ∧ byte	x		
		saddr, #byte	3	6	8	(saddr) ← (saddr) ∧ byte	x		
		A, r <small>Note 3</small>	2	4	–	A ← A ∧ r	x		
		r, A	2	4	–	r ← r ∧ A	x		
		A, saddr	2	4	5	A ← A ∧ (saddr)	x		
		A, !addr16	3	8	9 + n	A ← A ∧ (addr16)	x		
		A, [HL]	1	4	5 + n	A ← A ∧ [HL]	x		
		A, [HL + byte]	2	8	9 + n	A ← A ∧ [HL + byte]	x		
		A, [HL + B]	2	8	9 + n	A ← A ∧ [HL + B]	x		
	A, [HL + C]	2	8	9 + n	A ← A ∧ [HL + C]	x			

- Notes**
1. When the internal high-speed RAM area is accessed or for an instruction with no data access
  2. When an area except the internal high-speed RAM area is accessed
  3. Except “r = A”

- Remarks**
1. One instruction clock cycle is one cycle of the CPU clock (f<sub>CPU</sub>) selected by the processor clock control register (PCC).
  2. This clock cycle applies to the internal ROM program.
  3. n is the number of waits when the external memory expansion area is read.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	<b>OR</b>	A, #byte	2	4	–	$A \leftarrow A \vee \text{byte}$	x		
		saddr, #byte	3	6	8	$(\text{saddr}) \leftarrow (\text{saddr}) \vee \text{byte}$	x		
		A, r <small>Note 3</small>	2	4	–	$A \leftarrow A \vee r$	x		
		r, A	2	4	–	$r \leftarrow r \vee A$	x		
		A, saddr	2	4	5	$A \leftarrow A \vee (\text{saddr})$	x		
		A, laddr16	3	8	9 + n	$A \leftarrow A \vee (\text{addr16})$	x		
		A, [HL]	1	4	5 + n	$A \leftarrow A \vee (\text{HL})$	x		
		A, [HL + byte]	2	8	9 + n	$A \leftarrow A \vee (\text{HL} + \text{byte})$	x		
		A, [HL + B]	2	8	9 + n	$A \leftarrow A \vee (\text{HL} + B)$	x		
	A, [HL + C]	2	8	9 + n	$A \leftarrow A \vee (\text{HL} + C)$	x			
	<b>XOR</b>	A, #byte	2	4	–	$A \leftarrow A \nabla \text{byte}$	x		
		saddr, #byte	3	6	8	$(\text{saddr}) \leftarrow (\text{saddr}) \nabla \text{byte}$	x		
		A, r <small>Note 3</small>	2	4	–	$A \leftarrow A \nabla r$	x		
		r, A	2	4	–	$r \leftarrow r \nabla A$	x		
		A, saddr	2	4	5	$A \leftarrow A \nabla (\text{saddr})$	x		
		A, laddr16	3	8	9 + n	$A \leftarrow A \nabla (\text{addr16})$	x		
		A, [HL]	1	4	5 + n	$A \leftarrow A \nabla (\text{HL})$	x		
		A, [HL + byte]	2	8	9 + n	$A \leftarrow A \nabla (\text{HL} + \text{byte})$	x		
		A, [HL + B]	2	8	9 + n	$A \leftarrow A \nabla (\text{HL} + B)$	x		
	A, [HL + C]	2	8	9 + n	$A \leftarrow A \nabla (\text{HL} + C)$	x			
	<b>CMP</b>	A, #byte	2	4	–	$A - \text{byte}$	x	x	x
		saddr, #byte	3	6	8	$(\text{saddr}) - \text{byte}$	x	x	x
		A, r <small>Note 3</small>	2	4	–	$A - r$	x	x	x
		r, A	2	4	–	$r - A$	x	x	x
		A, saddr	2	4	5	$A - (\text{saddr})$	x	x	x
		A, laddr16	3	8	9 + n	$A - (\text{addr16})$	x	x	x
		A, [HL]	1	4	5 + n	$A - (\text{HL})$	x	x	x
A, [HL + byte]		2	8	9 + n	$A - (\text{HL} + \text{byte})$	x	x	x	
A, [HL + B]		2	8	9 + n	$A - (\text{HL} + B)$	x	x	x	
A, [HL + C]	2	8	9 + n	$A - (\text{HL} + C)$	x	x	x		

- Notes**
1. When the internal high-speed RAM area is accessed or for an instruction with no data access
  2. When an area except the internal high-speed RAM area is accessed
  3. Except “r = A”

- Remarks**
1. One instruction clock cycle is one cycle of the CPU clock (f<sub>CPU</sub>) selected by the processor clock control register (PCC).
  2. This clock cycle applies to the internal ROM program.
  3. n is the number of waits when the external memory expansion area is read.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit operation	<b>ADDW</b>	AX, #word	3	6	–	AX, CY ← AX + word	×	×	×
	<b>SUBW</b>	AX, #word	3	6	–	AX, CY ← AX – word	×	×	×
	<b>CMPW</b>	AX, #word	3	6	–	AX – word	×	×	×
Multiply/divide	<b>MULU</b>	X	2	16	–	AX ← A × X			
	<b>DIVUW</b>	C	2	25	–	AX (Quotient), C (Remainder) ← AX ÷ C			
Increment/decrement	<b>INC</b>	r	1	2	–	r ← r + 1	×	×	
		saddr	2	4	6	(saddr) ← (saddr) + 1	×	×	
	<b>DEC</b>	r	1	2	–	r ← r – 1	×	×	
		saddr	2	4	6	(saddr) ← (saddr) – 1	×	×	
	<b>INCW</b>	rp	1	4	–	rp ← rp + 1			
	<b>DECW</b>	rp	1	4	–	rp ← rp – 1			
Rotate	<b>ROR</b>	A, 1	1	2	–	(CY, A7 ← A0, A <sub>m-1</sub> ← A <sub>m</sub> ) × 1 time			×
	<b>ROL</b>	A, 1	1	2	–	(CY, A0 ← A7, A <sub>m+1</sub> ← A <sub>m</sub> ) × 1 time			×
	<b>RORC</b>	A, 1	1	2	–	(CY ← A0, A7 ← CY, A <sub>m-1</sub> ← A <sub>m</sub> ) × 1 time			×
	<b>ROLC</b>	A, 1	1	2	–	(CY ← A7, A0 ← CY, A <sub>m+1</sub> ← A <sub>m</sub> ) × 1 time			×
	<b>ROR4</b>	[HL]	2	10	12 + n + m	A <sub>3-0</sub> ← (HL) <sub>3-0</sub> , (HL) <sub>7-4</sub> ← A <sub>3-0</sub> , (HL) <sub>3-0</sub> ← (HL) <sub>7-4</sub>			
	<b>ROL4</b>	[HL]	2	10	12 + n + m	A <sub>3-0</sub> ← (HL) <sub>7-4</sub> , (HL) <sub>3-0</sub> ← A <sub>3-0</sub> , (HL) <sub>7-4</sub> ← (HL) <sub>3-0</sub>			
BCD adjustment	<b>ADJBA</b>		2	4	–	Decimal Adjust Accumulator after Addition	×	×	×
	<b>ADJBS</b>		2	4	–	Decimal Adjust Accumulator after Subtract	×	×	×
Bit manipulate	<b>MOV1</b>	CY, saddr.bit	3	6	7	CY ← (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← sfr.bit			×
		CY, A.bit	2	4	–	CY ← A.bit			×
		CY, PSW.bit	3	–	7	CY ← PSW.bit			×
		CY, [HL].bit	2	6	7 + n	CY ← (HL).bit			×
		saddr.bit, CY	3	6	8	(saddr.bit) ← CY			
		sfr.bit, CY	3	–	8	sfr.bit ← CY			
		A.bit, CY	2	4	–	A.bit ← CY			
		PSW.bit, CY	3	–	8	PSW.bit ← CY			×
[HL].bit, CY	2	6	8 + n + m	(HL).bit ← CY					

- Notes**
1. When the internal high-speed RAM area is accessed or for an instruction with no data access
  2. When an area except the internal high-speed RAM area is accessed

- Remarks**
1. One instruction clock cycle is one cycle of the CPU clock (f<sub>CPU</sub>) selected by the processor clock control register (PCC).
  2. This clock cycle applies to the internal ROM program.
  3. n is the number of waits when the external memory expansion area is read.
  4. m is the number of waits when the external memory expansion area is written.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag			
				Note 1	Note 2		Z	AC	CY	
Bit manipulate	<b>AND1</b>	CY, saddr.bit	3	6	7	$CY \leftarrow CY \wedge \text{saddr.bit}$			×	
		CY, sfr.bit	3	–	7	$CY \leftarrow CY \wedge \text{sfr.bit}$			×	
		CY, A.bit	2	4	–	$CY \leftarrow CY \wedge \text{A.bit}$			×	
		CY, PSW.bit	3	–	7	$CY \leftarrow CY \wedge \text{PSW.bit}$			×	
		CY, [HL].bit	2	6	7 + n	$CY \leftarrow CY \wedge \text{(HL).bit}$			×	
	<b>OR1</b>	CY, saddr.bit	3	6	7	$CY \leftarrow CY \vee \text{(saddr.bit)}$			×	
		CY, sfr.bit	3	–	7	$CY \leftarrow CY \vee \text{sfr.bit}$			×	
		CY, A.bit	2	4	–	$CY \leftarrow CY \vee \text{A.bit}$			×	
		CY, PSW.bit	3	–	7	$CY \leftarrow CY \vee \text{PSW.bit}$			×	
		CY, [HL].bit	2	6	7 + n	$CY \leftarrow CY \vee \text{(HL).bit}$			×	
	<b>XOR1</b>	CY, saddr.bit	3	6	7	$CY \leftarrow CY \nabla \text{(saddr.bit)}$			×	
		CY, sfr.bit	3	–	7	$CY \leftarrow CY \nabla \text{sfr.bit}$			×	
		CY, A.bit	2	4	–	$CY \leftarrow CY \nabla \text{A.bit}$			×	
		CY, PSW. bit	3	–	7	$CY \leftarrow CY \nabla \text{PSW.bit}$			×	
		CY, [HL].bit	2	6	7 + n	$CY \leftarrow CY \nabla \text{(HL).bit}$			×	
	<b>SET1</b>	saddr.bit	2	4	6	$\text{(saddr.bit)} \leftarrow 1$				
		sfr.bit	3	–	8	$\text{sfr.bit} \leftarrow 1$				
		A.bit	2	4	–	$\text{A.bit} \leftarrow 1$				
		PSW.bit	2	–	6	$\text{PSW.bit} \leftarrow 1$		×	×	×
		[HL].bit	2	6	8 + n + m	$\text{(HL).bit} \leftarrow 1$				
	<b>CLR1</b>	saddr.bit	2	4	6	$\text{(saddr.bit)} \leftarrow 0$				
		sfr.bit	3	–	8	$\text{sfr.bit} \leftarrow 0$				
		A.bit	2	4	–	$\text{A.bit} \leftarrow 0$				
		PSW.bit	2	–	6	$\text{PSW.bit} \leftarrow 0$		×	×	×
		[HL].bit	2	6	8 + n + m	$\text{(HL).bit} \leftarrow 0$				
	<b>SET1</b>	CY	1	2	–	$CY \leftarrow 1$			1	
	<b>CLR1</b>	CY	1	2	–	$CY \leftarrow 0$			0	
	<b>NOT1</b>	CY	1	2	–	$CY \leftarrow \overline{CY}$			×	

- Notes**
1. When the internal high-speed RAM area is accessed or for an instruction with no data access
  2. When an area except the internal high-speed RAM area is accessed

- Remarks**
1. One instruction clock cycle is one cycle of the CPU clock ( $f_{CPU}$ ) selected by the processor clock control register (PCC).
  2. This clock cycle applies to the internal ROM program.
  3. n is the number of waits when the external memory expansion area is read.
  4. m is the number of waits when the external memory expansion area is written.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Call/return	<b>CALL</b>	!addr16	3	7	–	$(SP - 1) \leftarrow (PC + 3)_H, (SP - 2) \leftarrow (PC + 3)_L,$ $PC \leftarrow \text{addr16}, SP \leftarrow SP - 2$			
	<b>CALLF</b>	!addr11	2	5	–	$(SP - 1) \leftarrow (PC + 2)_H, (SP - 2) \leftarrow (PC + 2)_L,$ $PC_{15-11} \leftarrow 00001, PC_{10-0} \leftarrow \text{addr11},$ $SP \leftarrow SP - 2$			
	<b>CALLT</b>	[addr5]	1	6	–	$(SP - 1) \leftarrow (PC + 1)_H, (SP - 2) \leftarrow (PC + 1)_L,$ $PC_H \leftarrow (00000000, \text{addr5} + 1),$ $PC_L \leftarrow (00000000, \text{addr5}),$ $SP \leftarrow SP - 2$			
	<b>BRK</b>		1	6	–	$(SP - 1) \leftarrow PSW, (SP - 2) \leftarrow (PC + 1)_H,$ $(SP - 3) \leftarrow (PC + 1)_L, PC_H \leftarrow (003FH),$ $PC_L \leftarrow (003EH), SP \leftarrow SP - 3, IE \leftarrow 0$			
	<b>RET</b>		1	6	–	$PC_H \leftarrow (SP + 1), PC_L \leftarrow (SP),$ $SP \leftarrow SP + 2$			
	<b>RETI</b>		1	6	–	$PC_H \leftarrow (SP + 1), PC_L \leftarrow (SP),$ $PSW \leftarrow (SP + 2), SP \leftarrow SP + 3,$ $NMIS \leftarrow 0$	R	R	R
	<b>RETB</b>		1	6	–	$PC_H \leftarrow (SP + 1), PC_L \leftarrow (SP),$ $PSW \leftarrow (SP + 2), SP \leftarrow SP + 3$	R	R	R
Stack manipulate	<b>PUSH</b>	PSW	1	2	–	$(SP - 1) \leftarrow PSW, SP \leftarrow SP - 1$			
		rp	1	4	–	$(SP - 1) \leftarrow rp_H, (SP - 2) \leftarrow rp_L,$ $SP \leftarrow SP - 2$			
	<b>POP</b>	PSW	1	2	–	$PSW \leftarrow (SP), SP \leftarrow SP + 1$	R	R	R
		rp	1	4	–	$rp_H \leftarrow (SP + 1), rp_L \leftarrow (SP),$ $SP \leftarrow SP + 2$			
	<b>MOVW</b>	SP, #word	4	–	10	$SP \leftarrow \text{word}$			
		SP, AX	2	–	8	$SP \leftarrow AX$			
AX, SP		2	–	8	$AX \leftarrow SP$				
Unconditional branch	<b>BR</b>	!addr16	3	6	–	$PC \leftarrow \text{addr16}$			
		\$addr16	2	6	–	$PC \leftarrow PC + 2 + \text{jdisp8}$			
		AX	2	8	–	$PCH \leftarrow A, PCL \leftarrow X$			
Conditional branch	<b>BC</b>	\$addr16	2	6	–	$PC \leftarrow PC + 2 + \text{jdisp8}$ if $CY = 1$			
	<b>BNC</b>	\$addr16	2	6	–	$PC \leftarrow PC + 2 + \text{jdisp8}$ if $CY = 0$			
	<b>BZ</b>	\$addr16	2	6	–	$PC \leftarrow PC + 2 + \text{jdisp8}$ if $Z = 1$			
	<b>BNZ</b>	\$addr16	2	6	–	$PC \leftarrow PC + 2 + \text{jdisp8}$ if $Z = 0$			

- Notes**
1. When the internal high-speed RAM area is accessed or for an instruction with no data access
  2. When an area except the internal high-speed RAM area is accessed

- Remarks**
1. One instruction clock cycle is one cycle of the CPU clock ( $f_{CPU}$ ) selected by the processor clock control register (PCC).
  2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Conditional branch	<b>BT</b>	saddr.bit, \$addr16	3	8	9	PC ← PC + 3 + jdisp8 if(saddr.bit) = 1			
		sfr.bit, \$addr16	4	–	11	PC ← PC + 4 + jdisp8 if sfr.bit = 1			
		A.bit, \$addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 1			
		PSW.bit, \$addr16	3	–	9	PC ← PC + 3 + jdisp8 if PSW.bit = 1			
		[HL].bit, \$addr16	3	10	11 + n	PC ← PC + 3 + jdisp8 if (HL).bit = 1			
	<b>BF</b>	saddr.bit, \$addr16	4	10	11	PC ← PC + 4 + jdisp8 if(saddr.bit) = 0			
		sfr.bit, \$addr16	4	–	11	PC ← PC + 4 + jdisp8 if sfr.bit = 0			
		A.bit, \$addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 0			
		PSW.bit, \$addr16	4	–	11	PC ← PC + 4 + jdisp8 if PSW.bit = 0			
		[HL].bit, \$addr16	3	10	11 + n	PC ← PC + 3 + jdisp8 if (HL).bit = 0			
	<b>BTCLR</b>	saddr.bit, \$addr16	4	10	12	PC ← PC + 4 + jdisp8 if(saddr.bit) = 1 then reset(saddr.bit)			
		sfr.bit, \$addr16	4	–	12	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit			
		A.bit, \$addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit			
		PSW.bit, \$addr16	4	–	12	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	×	×	×
		[HL].bit, \$addr16	3	10	12 + n + m	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit			
	<b>DBNZ</b>	B, \$addr16	2	6	–	B ← B – 1, then PC ← PC + 2 + jdisp8 if B ≠ 0			
		C, \$addr16	2	6	–	C ← C – 1, then PC ← PC + 2 + jdisp8 if C ≠ 0			
		saddr, \$addr16	3	8	10	(saddr) ← (saddr) – 1, then PC ← PC + 3 + jdisp8 if(saddr) ≠ 0			
CPU control	<b>SEL</b>	R <sub>Bn</sub>	2	4	–	RBS1, 0 ← n			
	<b>NOP</b>		1	2	–	No Operation			
	<b>EI</b>		2	–	6	IE ← 1(Enable Interrupt)			
	<b>DI</b>		2	–	6	IE ← 0(Disable Interrupt)			
	<b>HALT</b>		2	6	–	Set HALT Mode			
	<b>STOP</b>		2	6	–	Set STOP Mode			

- Notes**
1. When the internal high-speed RAM area is accessed or for an instruction with no data access
  2. When an area except the internal high-speed RAM area is accessed

- Remarks**
1. One instruction clock cycle is one cycle of the CPU clock (f<sub>CPU</sub>) selected by the processor clock control register (PCC).
  2. This clock cycle applies to the internal ROM program.
  3. n is the number of waits when the external memory expansion area is read.
  4. m is the number of waits when the external memory expansion area is written.

22.3 Instructions Listed by Addressing Type

(1) 8-bit instructions

MOV, XCH, ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP, MULU, DIVUW, INC, DEC, ROR, ROL, RORC, ROLC, ROR4, ROL4, PUSH, POP, DBNZ

Second Operand First Operand	#byte	A	r <sup>Note</sup>	sfr	saddr	laddr16	PSW	[DE]	[HL]	[HL + byte] [HL + B] [HL + C]	\$saddr16	1	None
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP		ROR ROL RORC ROLC	
r	MOV	MOV ADD ADDC SUB SUBC AND OR XOR CMP											INC DEC
B, C											DBNZ		
sfr	MOV	MOV											
saddr	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV									DBNZ		INC DEC
laddr16		MOV											
PSW	MOV	MOV											PUSH POP
[DE]		MOV											
[HL]		MOV											ROR4 ROL4
[HL + byte] [HL + B] [HL + C]		MOV											
X													MULU
C													DIVUW

Note Except r = A

**(2) 16-bit instructions**

MOVW, XCHW, ADDW, SUBW, CMPW, PUSH, POP, INCW, DECW

Second Operand First Operand	#word	AX	rp <sup>Note</sup>	sfrp	saddrp	laddr16	SP	None
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	MOVW	MOVW	
rp	MOVW	MOVW <sup>Note</sup>						INCW DECW PUSH POP
sfrp	MOVW	MOVW						
saddrp	MOVW	MOVW						
laddr16		MOVW						
SP	MOVW	MOVW						

**Note** Only when rp = BC, DE, HL

**(3) Bit manipulation instructions**

MOV1, AND1, OR1, XOR1, SET1, CLR1, NOT1, BT, BF, BTCLR

Second Operand First Operand	A.bit	sfr.bit	saddr.bit	PSW.bit	[HL].bit	CY	\$addr16	None
A.bit						MOV1	BT BF BTCLR	SET1 CLR1
sfr.bit						MOV1	BT BF BTCLR	SET1 CLR1
saddr.bit						MOV1	BT BF BTCLR	SET1 CLR1
PSW.bit						MOV1	BT BF BTCLR	SET1 CLR1
[HL].bit						MOV1	BT BF BTCLR	SET1 CLR1
CY	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1			SET1 CLR1 NOT1

**(4) Call instructions/branch instructions**

CALL, CALLF, CALLT, BR, BC, BNC, BZ, BNZ, BT, BF, BTCLR, DBNZ

Second Operand First Operand	AX	!addr16	!addr11	[addr5]	\$addr16
Basic instruction	BR	CALL BR	CALLF	CALLT	BR BC BNC BZ BNZ
Compound instruction					BT BF BTCLR DBNZ

**(5) Other instructions**

ADJBA, ADJBS, BRK, RET, RETI, RETB, SEL, NOP, EI, DI, HALT, STOP

## CHAPTER 23 ELECTRICAL SPECIFICATIONS (TARGET VALUES)

These specifications are only target values, and may not be satisfied by mass-produced products.

The electrical specifications (target values) of (A1) products are under evaluation.

### Absolute Maximum Ratings (T<sub>A</sub> = 25°C)

Parameter	Symbol	Conditions	Ratings	Unit	
Supply voltage	V <sub>DD</sub>		-0.3 to +6.5	V	
	V <sub>SS</sub>		-0.3 to +0.3	V	
	AV <sub>REF</sub>		-0.3 to V <sub>DD</sub> + 0.3 <sup>Note 1</sup>	V	
	AV <sub>SS</sub>		-0.3 to +0.3	V	
	V <sub>PP</sub>	μPD78F0103 only <b>Note 2</b>	-0.3 to +10.5	V	
Supply current	I <sub>VDD</sub>	Total value of operating current and high-level output current	-60	mA	
	I <sub>VSS</sub>	Total value of operating current and low-level output current	70	mA	
Input voltage	V <sub>I1</sub>	P00 to P03, P10 to P17, P20 to P23, P30 to P33, P120, P130, X1, X2, $\overline{\text{RESET}}$	-0.3 to V <sub>DD</sub> + 0.3 <sup>Note 1</sup>	V	
	V <sub>I2</sub>	V <sub>PP</sub> in flash programming mode (μPD78F0103 only)	-0.3 to +10.5	V	
Output voltage	V <sub>O</sub>		-0.3 to V <sub>DD</sub> + 0.3 <sup>Note 1</sup>	V	
Analog input voltage	V <sub>AN</sub>		AV <sub>SS</sub> - 0.3 to AV <sub>REF</sub> + 0.3 <sup>Note 1</sup> and -0.3 to V <sub>DD</sub> + 0.3 <sup>Note 1</sup>	V	
Output current, high	I <sub>OH</sub>	Per pin	-10	mA	
		Total of all pins -60 mA	P30 to P33, P120	-30	mA
			P00 to P03, P10 to P17, P130	-30	mA
Output current, low	I <sub>OL</sub>	Per pin	20	mA	
		Total of all pins 70 mA	P30 to P33, P120	35	mA
			P00 to P03, P10 to P17, P130	35	mA
Operating ambient temperature	T <sub>A</sub>	In normal operation mode	-40 to +85	°C	
Storage temperature	T <sub>stg</sub>	μPD780101, 780102, 780103	-65 to +150	°C	
		μPD78F0103	-40 to +125		

**Note 1.** Must be 6.5 V or lower.

(Refer to **Note 2** on the next page.)

**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.

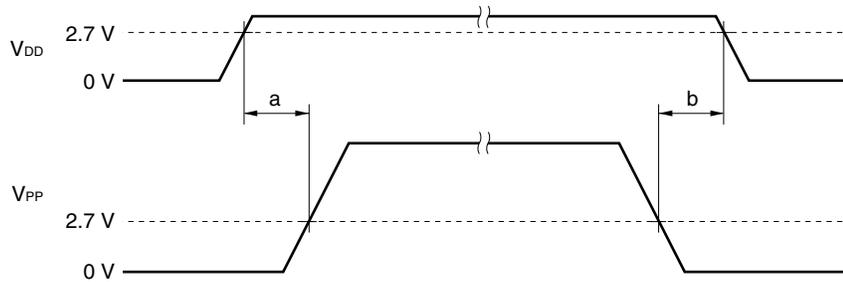
**Note 2.** Make sure that the following conditions of the  $V_{PP}$  voltage application timing are satisfied when the flash memory is written.

- **When supply voltage rises**

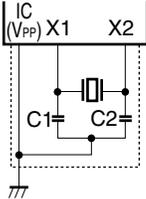
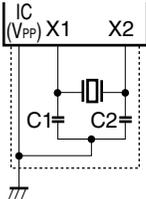
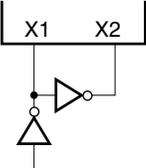
$V_{PP}$  must exceed  $V_{DD}$   $10\ \mu\text{s}$  or more after  $V_{DD}$  has reached the lower-limit value (2.7 V) of the operating voltage range (see a in the figure below).

- **When supply voltage drops**

$V_{DD}$  must be lowered  $10\ \mu\text{s}$  or more after  $V_{PP}$  falls below the lower-limit value (2.7 V) of the operating voltage range of  $V_{DD}$  (see b in the figure below).



**X1 Oscillator Characteristics ( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,  $2.7\text{ V} \leq AV_{REF} \leq V_{DD}$ ,  $V_{SS} = AV_{SS} = 0\text{ V}$ )**

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic resonator		Oscillation frequency ( $f_{XP}$ ) <sup>Note</sup>	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	2.0		10	MHz
			$3.3\text{ V} \leq V_{DD} < 4.0\text{ V}$	2.0		8.38	
			$2.7\text{ V} \leq V_{DD} < 3.3\text{ V}$	2.0		5.0	
Crystal resonator		Oscillation frequency ( $f_{XP}$ ) <sup>Note</sup>	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	2.0		10	MHz
			$3.3\text{ V} \leq V_{DD} < 4.0\text{ V}$	2.0		8.38	
			$2.7\text{ V} \leq V_{DD} < 3.3\text{ V}$	2.0		5.0	
External clock		X1 input frequency ( $f_{XP}$ ) <sup>Note</sup>	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	2.0		10	MHz
			$3.3\text{ V} \leq V_{DD} < 4.0\text{ V}$	2.0		8.38	
			$2.7\text{ V} \leq V_{DD} < 3.3\text{ V}$	2.0		5.0	
		X1 input high-/low-level width ( $t_{XH}$ , $t_{XL}$ )	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	46		500	ns
			$3.3\text{ V} \leq V_{DD} < 4.0\text{ V}$	56		500	
			$2.7\text{ V} \leq V_{DD} < 3.3\text{ V}$	96		500	

**Note** Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

**Caution** When using the X1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

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

**Remark** For the resonator selection and oscillator constant, users are required to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

**Ring-OSC Oscillator Characteristics ( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,  $2.7\text{ V} \leq AV_{REF} \leq V_{DD}$ ,  $V_{SS} = AV_{SS} = 0\text{ V}$ )**

Resonator	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
On-chip Ring-OSC oscillator	Oscillation frequency ( $f_R$ )		120	240	480	kHz

**DC Characteristics ( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,  $2.7\text{ V} \leq AV_{REF} \leq V_{DD}$ ,  $V_{SS} = AV_{SS} = 0\text{ V}$ ) (1/3)**

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high	$I_{OH}$	Per pin	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			-5	mA
		Total of P30 to P33, P120	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			-25	mA
		Total of P00 to P03, P10 to P17, P130	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			-25	mA
		Total of all pins	$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$			-10	mA
Output current, low	$I_{OL}$	Per pin for P00 to P03, P10 to P17, P30 to P33, P120, P130	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			10	mA
		Total of P30 to P33, P120	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			30	mA
		Total of P00 to P03, P10 to P17, P130	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			30	mA
		Total of all pins	$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$			10	mA
Input voltage, high	$V_{IH1}$	P12, P13, P15		$0.7V_{DD}$		$V_{DD}$	V
	$V_{IH2}$	P00 to P03, P10, P11, P14, P16, P17, P30 to P33, P120, $\overline{\text{RESET}}$		$0.8V_{DD}$		$V_{DD}$	V
	$V_{IH3}$	P20 to P23 <sup>Note</sup>		$0.7AV_{REF}$		$AV_{REF}$	V
	$V_{IH4}$	X1, X2		$V_{DD} - 0.5$		$V_{DD}$	V
Input voltage, low	$V_{IL1}$	P12, P13, P15		0		$0.3V_{DD}$	V
	$V_{IL2}$	P00 to P03, P10, P11, P14, P16, P17, P30 to P33, P120, $\overline{\text{RESET}}$		0		$0.2V_{DD}$	V
	$V_{IL3}$	P20 to P23 <sup>Note</sup>		0		$0.3AV_{REF}$	V
	$V_{IL4}$	X1, X2		0		0.4	V
Output voltage, high	$V_{OH}$	Total of P30 to P33, P120 pins $I_{OH} = -25\text{ mA}$	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OH} = -5\text{ mA}$	$V_{DD} - 1.0$			V
		Total of P00 to P03, P10 to P17, P130 pins $I_{OH} = -25\text{ mA}$	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OH} = -5\text{ mA}$	$V_{DD} - 1.0$			V
		$I_{OH} = -100\text{ }\mu\text{A}$	$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$	$V_{DD} - 0.5$			V
Output voltage, low	$V_{OL}$	Total of P30 to P33, P120 pins $I_{OL} = 30\text{ mA}$	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OL} = 10\text{ mA}$			1.3	V
		Total of P00 to P03, P10 to P17, P130 pins $I_{OL} = 30\text{ mA}$	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OL} = 10\text{ mA}$			1.3	V
		$I_{OL} = 400\text{ }\mu\text{A}$	$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$			0.4	V
Input leakage current, high	$I_{LIH1}$	$V_i = V_{DD}$	P00 to P03, P10 to P17, P30 to P33, P120, P130, $\overline{\text{RESET}}$			3	$\mu\text{A}$
		$V_i = AV_{REF}$	P20 to P23			3	$\mu\text{A}$
	$I_{LIH2}$	$V_i = V_{DD}$	X1, X2			20	$\mu\text{A}$
Input leakage current, low	$I_{LIL1}$	$V_i = 0\text{ V}$	P00 to P03, P10 to P17, P20 to P23, P30 to P33, P120, P130, $\overline{\text{RESET}}$			-3	$\mu\text{A}$
	$I_{LIL2}$		X1, X2			-20	$\mu\text{A}$
Output leakage current, high	$I_{LOH}$	$V_o = V_{DD}$				3	$\mu\text{A}$
Output leakage current, low	$I_{LOL}$	$V_o = 0\text{ V}$				-3	$\mu\text{A}$
Pull-up resistance value	R	$V_i = 0\text{ V}$		10	30	100	k $\Omega$
$V_{PP}$ supply voltage ( $\mu\text{PD78F0103}$ only)	$V_{PP1}$	In normal operation mode		0		$0.2V_{DD}$	V

**Note** When used as A/D converter analog input pins, set  $AV_{REF} = V_{DD}$ .

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (2/3):  $\mu$ PD78F0103

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,  $2.7\text{ V} \leq AV_{REF} \leq V_{DD}$ ,  $V_{SS} = AV_{SS} = 0\text{ V}$ )

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit		
Supply current <sup>Note 1</sup>	I <sub>DD1</sub>	X1 crystal oscillation operating mode <sup>Note 2</sup>	f <sub>XP</sub> = 10 MHz, V <sub>DD</sub> = 5.0 V ±10% <sup>Note 3</sup>	When A/D converter is stopped		12	22	mA	
				When A/D converter is operating <sup>Note 4</sup>		13	24	mA	
			f <sub>XP</sub> = 5 MHz, V <sub>DD</sub> = 3.0 V ±10% <sup>Note 3</sup>	When A/D converter is stopped		4	7.3	mA	
				When A/D converter is operating <sup>Note 4</sup>		4.6	8.5	mA	
	I <sub>DD2</sub>	X1 crystal oscillation HALT mode	f <sub>XP</sub> = 10 MHz, V <sub>DD</sub> = 5.0 V ±10% <sup>Note 3</sup>	When peripheral functions are stopped		1.7	3.4	mA	
				When peripheral functions are operating			5.2	mA	
			f <sub>XP</sub> = 5 MHz, V <sub>DD</sub> = 3.0 V ±10% <sup>Note 3</sup>	When peripheral functions are stopped		0.4	0.8	mA	
				When peripheral functions are operating			1.7	mA	
	I <sub>DD3</sub>	Ring-OSC operating mode <sup>Note 5</sup>	V <sub>DD</sub> = 5.0 V ±10%			0.4	1.6	mA	
			V <sub>DD</sub> = 3.0 V ±10%			0.3	1.2	mA	
	I <sub>DD4</sub>	STOP mode	V <sub>DD</sub> = 5.0 V ±10%	POC: OFF, RING: OFF			0.1	30	μA
				POC: OFF, RING: ON			14	58	μA
POC: ON <sup>Note 6</sup> , RING: OFF					3.5	35.5	μA		
POC: ON <sup>Note 6</sup> , RING: ON					17.5	63.5	μA		
V <sub>DD</sub> = 3.0 V ±10%			POC: OFF, RING: OFF			0.05	10	μA	
			POC: OFF, RING: ON			7.5	25	μA	
			POC: ON <sup>Note 6</sup> , RING: OFF			3.5	15.5	μA	
			POC: ON <sup>Note 6</sup> , RING: ON			11	30.5	μA	

- Notes**
1. Total current flowing through the internal power supply (V<sub>DD</sub>). Peripheral operation current is included (however, the current that flows through the pull-up resistors of ports is not included).
  2. I<sub>DD1</sub> includes peripheral operation current.
  3. When PCC = 00H.
  4. Including the current that flows through the AV<sub>REF</sub> pin.
  5. When main system clock is stopped.
  6. Including when LVIE (bit 4 of LVIM) = 1 with POC-OFF selected by a mask option.

**DC Characteristics (3/3):  $\mu$ PD780101, 780102, and 780103**
**( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,  $2.7\text{ V} \leq AV_{REF} \leq V_{DD}$ ,  $V_{SS} = AV_{SS} = 0\text{ V}$ )**

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit	
Supply current <sup>Note 1</sup>	I <sub>DD1</sub>	X1 crystal oscillation operating mode <sup>Note 2</sup>	f <sub>XP</sub> = 10 MHz, V <sub>DD</sub> = 5.0 V ±10% <sup>Note 3</sup>	When A/D converter is stopped		5.5	11	mA
				When A/D converter is operating <sup>Note 4</sup>		6.5	13	mA
		f <sub>XP</sub> = 5 MHz, V <sub>DD</sub> = 3.0 V ±10% <sup>Note 3</sup>	When A/D converter is stopped		1.5	3	mA	
			When A/D converter is operating <sup>Note 4</sup>		2.5	5	mA	
	I <sub>DD2</sub>	X1 crystal oscillation HALT mode	f <sub>XP</sub> = 10 MHz, V <sub>DD</sub> = 5.0 V ±10% <sup>Note 3</sup>	When peripheral functions are stopped		1.1	2.2	mA
				When peripheral functions are operating			4.4	mA
		f <sub>XP</sub> = 5 MHz, V <sub>DD</sub> = 3.0 V ±10% <sup>Note 3</sup>	When peripheral functions are stopped		0.6	1.2	mA	
			When peripheral functions are operating			2.4	mA	
	I <sub>DD3</sub>	Ring-OSC operating mode <sup>Note 5</sup>	V <sub>DD</sub> = 5.0 V ±10%			0.3	0.9	mA
			V <sub>DD</sub> = 3.0 V ±10%			0.19	0.57	mA
	I <sub>DD4</sub>	STOP mode	V <sub>DD</sub> = 5.0 V ±10%	POC: OFF, RING: OFF		0.1	30	μA
				POC: OFF, RING: ON		14	58	μA
POC: ON <sup>Note 6</sup> , RING: OFF					3.5	35.5	μA	
POC: ON <sup>Note 6</sup> , RING: ON					17.5	63.5	μA	
V <sub>DD</sub> = 3.0 V ±10%			POC: OFF, RING: OFF		0.05	10	μA	
			POC: OFF, RING: ON		7.5	25	μA	
			POC: ON <sup>Note 6</sup> , RING: OFF		3.5	15.5	μA	
			POC: ON <sup>Note 6</sup> , RING: ON		11	30.5	μA	

**Notes** 1. Total current flowing through the internal power supply ( $V_{DD}$ ). Peripheral operation current is included (however, the current that flows through the pull-up resistors of ports is not included).

2. I<sub>DD1</sub> includes peripheral operation current.
3. When PCC = 00H.
4. Including the current that flows through the AV<sub>REF</sub> pin.
5. When main system clock is stopped.
6. Including when LVIE (bit 4 of LVIM) = 1 with POC-OFF selected by a mask option.

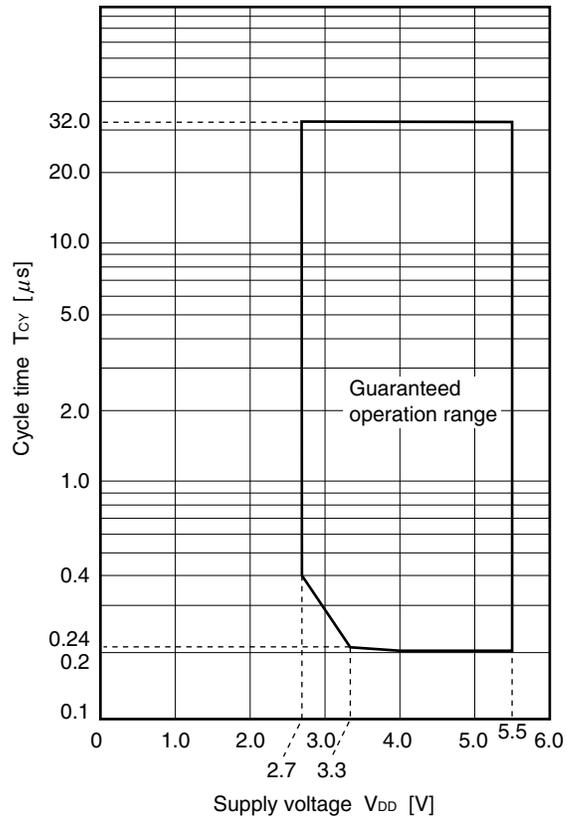
## AC Characteristics

(1) Basic operation ( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,  $2.7\text{ V} \leq AV_{REF} \leq V_{DD}$ ,  $V_{SS} = AV_{SS} = 0\text{ V}$ )

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
Instruction cycle (minimum instruction execution time)	T <sub>CY</sub>	X1 input clock	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	0.2		32	$\mu\text{s}$
			$3.3\text{ V} \leq V_{DD} < 4.0\text{ V}$	0.24		32	$\mu\text{s}$
			$2.7\text{ V} \leq V_{DD} < 3.3\text{ V}$	0.4		32	$\mu\text{s}$
		Ring-OSC clock	4.17	8.33	16.67	$\mu\text{s}$	
TI00 input high-level width, low-level width	t <sub>TIH0</sub> , t <sub>TIL0</sub>	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	$2/f_{\text{sam}} + 0.1$ <sup>Note</sup>			$\mu\text{s}$	
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$	$2/f_{\text{sam}} + 0.2$ <sup>Note</sup>			$\mu\text{s}$	
TI50 input frequency	f <sub>TI5</sub>	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			10	MHz	
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$			5		
TI50 input high-level width, low-level width	t <sub>TIH5</sub> , t <sub>TIL5</sub>	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	50			ns	
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$	100			ns	
Interrupt input high-level width, low-level width	t <sub>INTH</sub> , t <sub>INTL</sub>		1			$\mu\text{s}$	
RESET low-level width	t <sub>RST</sub>		10			$\mu\text{s}$	

**Note** Selection of  $f_{\text{sam}} = f_{\text{XP}}$ ,  $f_{\text{XP}}/4$ ,  $f_{\text{XP}}/256$  is possible using bits 0 and 1 (PRM000, PRM001) of prescaler mode register 00 (PRM00). Note that when selecting the TI000 or TI010 valid edge as the count clock,  $f_{\text{sam}} = f_{\text{XP}}$ .

$T_{CY}$  vs.  $V_{DD}$  (X1 Input Clock Operation)



**(2) Serial interface ( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,  $2.7\text{ V} \leq AV_{REF} \leq V_{DD}$ ,  $V_{SS} = AV_{SS} = 0\text{ V}$ )**
**(a) UART mode (UART6, dedicated baud rate generator output)**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					312.5	kbs

**(b) UART mode (UART0, dedicated baud rate generator output):  $\mu\text{PD780102}$ ,  $780103$ , and  $78F0103$  only**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					312.5	kbs

**(c) 3-wire serial I/O mode (master mode,  $\overline{\text{SCK10}}$ ... internal clock output)**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{SCK10}}$ cycle time	$t_{\text{CY1}}$	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	200			ns
		$3.3\text{ V} \leq V_{DD} < 4.0\text{ V}$	240			ns
		$2.7\text{ V} \leq V_{DD} < 3.3\text{ V}$	400			ns
$\overline{\text{SCK10}}$ high-/low-level width	$t_{\text{KH1}}$ , $t_{\text{KL1}}$		$t_{\text{CY1}}/2-10$			ns
SI10 setup time (to $\overline{\text{SCK10}}\uparrow$ )	$t_{\text{SIK1}}$		30			ns
SI10 hold time (from $\overline{\text{SCK10}}\uparrow$ )	$t_{\text{KSI1}}$		30			ns
Delay time from $\overline{\text{SCK10}}\downarrow$ to SO10 output	$t_{\text{KSO1}}$	$C = 100\text{ pF}^{\text{Note}}$			30	ns

**Note** C is the load capacitance of the  $\overline{\text{SCK10}}$  and SO10 output lines.

**(d) 3-wire serial I/O mode (master mode,  $\overline{\text{SCK10}}$ ... external clock input)**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{SCK10}}$ cycle time	$t_{\text{CY2}}$		200			ns
$\overline{\text{SCK10}}$ high-/low-level width	$t_{\text{KH2}}$ , $t_{\text{KL2}}$		$t_{\text{CY2}}/2-10$			ns
SI10 setup time (to $\overline{\text{SCK10}}\uparrow$ )	$t_{\text{SIK2}}$		30			ns
SI10 hold time (from $\overline{\text{SCK10}}\uparrow$ )	$t_{\text{KSI2}}$		30			ns
Delay time from $\overline{\text{SCK10}}\downarrow$ to SO10 output	$t_{\text{KSO2}}$	$C = 100\text{ pF}^{\text{Note}}$			30	ns

**Note** C is the load capacitance of the SO10 output line.

**(e) 3-wire serial I/O mode (slave mode,  $\overline{\text{SCK10}}$ ... internal clock output)**

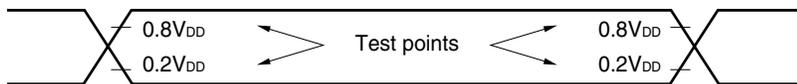
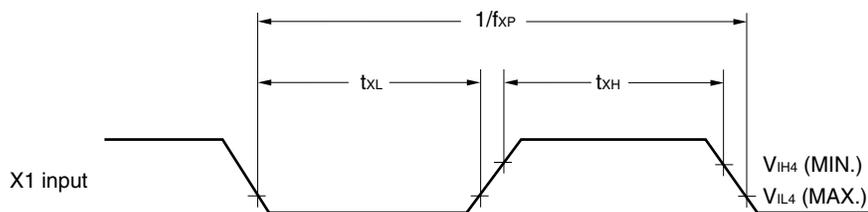
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{SCK10}}$ cycle time	$t_{\text{KCY3}}$	$4.0 \text{ V} \leq V_{\text{DD}} \leq 5.5 \text{ V}$	200			ns
		$3.3 \text{ V} \leq V_{\text{DD}} < 4.0 \text{ V}$	240			ns
		$2.7 \text{ V} \leq V_{\text{DD}} < 3.3 \text{ V}$	400			ns
$\overline{\text{SCK10}}$ high-/low-level width	$t_{\text{KH3}}$ , $t_{\text{KL3}}$		$t_{\text{KCY3}}/2-10$			ns
SI10 setup time (to $\overline{\text{SCK10}}\uparrow$ )	$t_{\text{SIK3}}$		50			ns
SI10 hold time (from $\overline{\text{SCK10}}\uparrow$ )	$t_{\text{KSI3}}$		50			ns
Delay time from $\overline{\text{SCK10}}\downarrow$ to SO10 output	$t_{\text{KSO3}}$	$C = 100 \text{ pF}^{\text{Note}}$			50	ns

**Note** C is the load capacitance of the  $\overline{\text{SCK10}}$  and SO10 output lines.

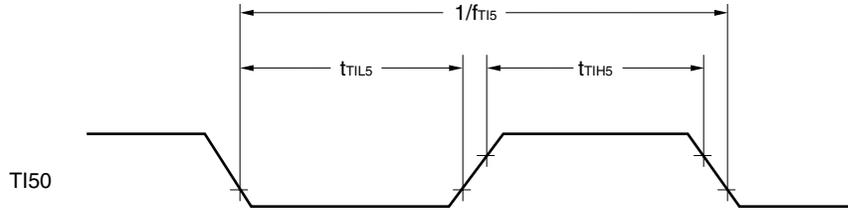
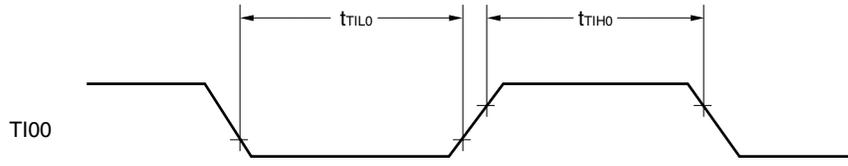
**(f) 3-wire serial I/O mode (slave mode,  $\overline{\text{SCK10}}$ ... external clock input)**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{SCK10}}$ cycle time	$t_{\text{KCY4}}$		200			ns
$\overline{\text{SCK10}}$ high-/low-level width	$t_{\text{KH4}}$ , $t_{\text{KL4}}$		$t_{\text{KCY4}}/2-10$			ns
SI10 setup time (to $\overline{\text{SCK10}}\uparrow$ )	$t_{\text{SIK4}}$		50			ns
SI10 hold time (from $\overline{\text{SCK10}}\uparrow$ )	$t_{\text{KSI4}}$		50			ns
Delay time from $\overline{\text{SCK10}}\downarrow$ to SO10 output	$t_{\text{KSO4}}$	$C = 100 \text{ pF}^{\text{Note}}$			50	ns

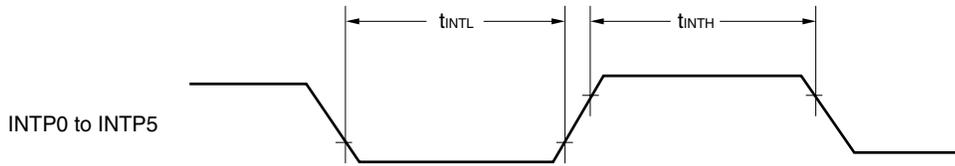
**Note** C is the load capacitance of the SO10 output line.

**AC Timing Test Points (Excluding X1 Input)**

**Clock Timing**


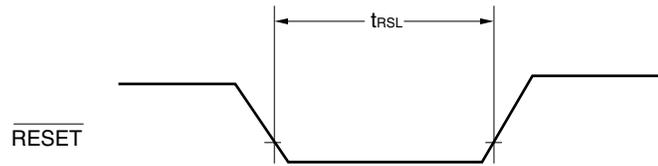
**TI Timing**



**Interrupt Request Input Timing**

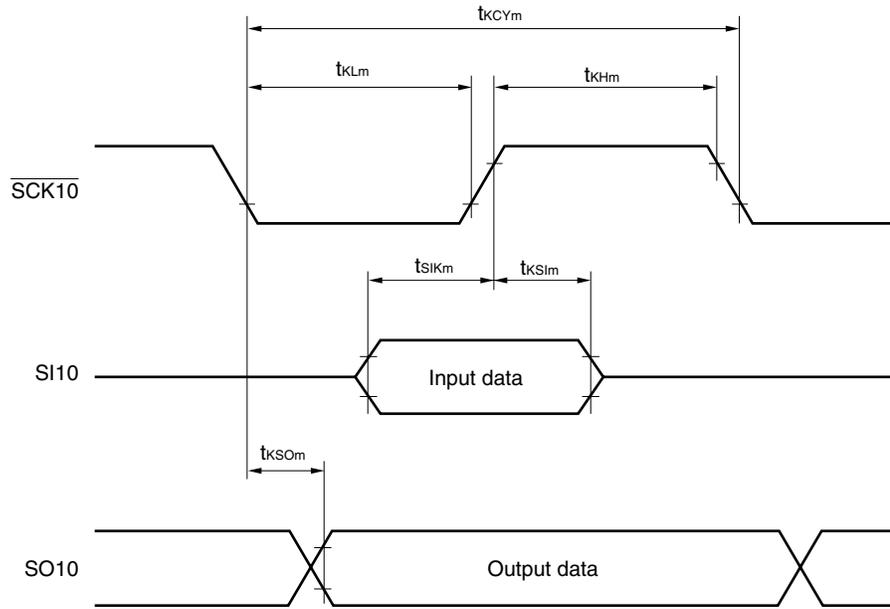


**RESET Input Timing**



Serial Transfer Timing

3-wire serial I/O mode:



**Remark**  $m = 1$  to 4

**A/D Converter Characteristics (T<sub>A</sub> = -40 to +85°C, 2.7 V ≤ V<sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ AV<sub>REF</sub> ≤ V<sub>DD</sub>, V<sub>SS</sub> = AV<sub>SS</sub> = 0 V)**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution			10	10	10	bit
Overall error <sup>Notes 1, 2</sup>		4.0 V ≤ AV <sub>REF</sub> ≤ 5.5 V		±0.2	±0.4	%FSR
		2.7 V ≤ AV <sub>REF</sub> < 4.0 V		±0.3	±0.6	%FSR
Conversion time	t <sub>CONV</sub>	4.0 V ≤ AV <sub>REF</sub> ≤ 5.5 V	14		100	μs
		2.7 V ≤ AV <sub>REF</sub> < 4.0 V	17		100	μs
Zero-scale error <sup>Notes 1, 2</sup>		4.0 V ≤ AV <sub>REF</sub> ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV <sub>REF</sub> < 4.0 V			±0.6	%FSR
Full-scale error <sup>Notes 1, 2</sup>		4.0 V ≤ AV <sub>REF</sub> ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV <sub>REF</sub> < 4.0 V			±0.6	%FSR
Integral non-linearity error <sup>Note 1</sup>		4.0 V ≤ AV <sub>REF</sub> ≤ 5.5 V			±2.5	LSB
		2.7 V ≤ AV <sub>REF</sub> < 4.0 V			±4.5	LSB
Differential non-linearity error <sup>Note 1</sup>		4.0 V ≤ AV <sub>REF</sub> ≤ 5.5 V			±1.5	LSB
		2.7 V ≤ AV <sub>REF</sub> < 4.0 V			±2.0	LSB
Analog input voltage	V <sub>IAN</sub>		AV <sub>SS</sub>		AV <sub>REF</sub>	V

**Notes** 1. Excludes quantization error (±1/2 LSB).

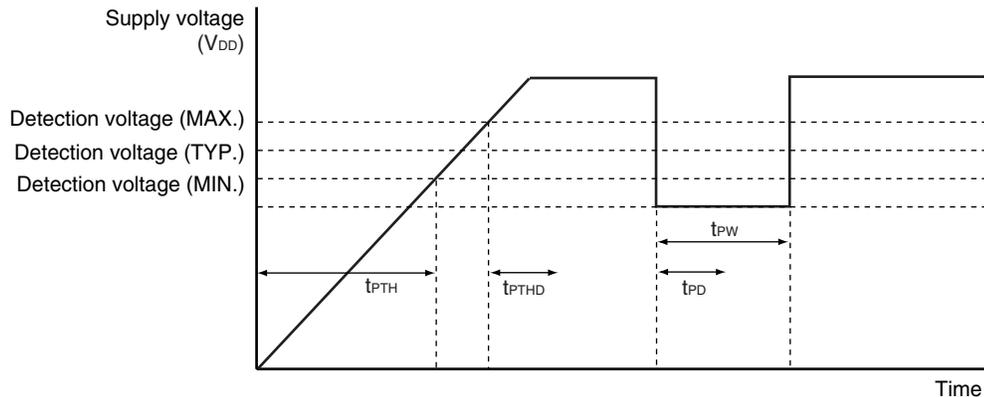
2. This value is indicated as a ratio (%FSR) to the full-scale value.

**POC Circuit Characteristics (T<sub>A</sub> = -40 to +85°C)**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V <sub>POC0</sub>	Mask option = 3.5 V	3.3	3.5	3.7	V
	V <sub>POC1</sub>	Mask option = 2.85 V	2.7	2.85	3.0	V
Power supply rise time	t <sub>PTH</sub>	V <sub>DD</sub> : 0 V → 2.7 V	0.0015		1500	ms
		V <sub>DD</sub> : 0 V → 3.3 V	0.002		1800	ms
Response delay time 1 <sup>Note</sup>	t <sub>PTHD</sub>	When power supply rises, after reaching detection voltage (MAX.)			3.0	ms
Response delay time 2 <sup>Note</sup>	t <sub>PD</sub>	When power supply falls, V <sub>DD</sub> = 1.7 V			1.0	ms
Minimum pulse width	t <sub>PW</sub>		0.2			ms

**Note** Time required from voltage detection to reset release.

**POC Circuit Timing**



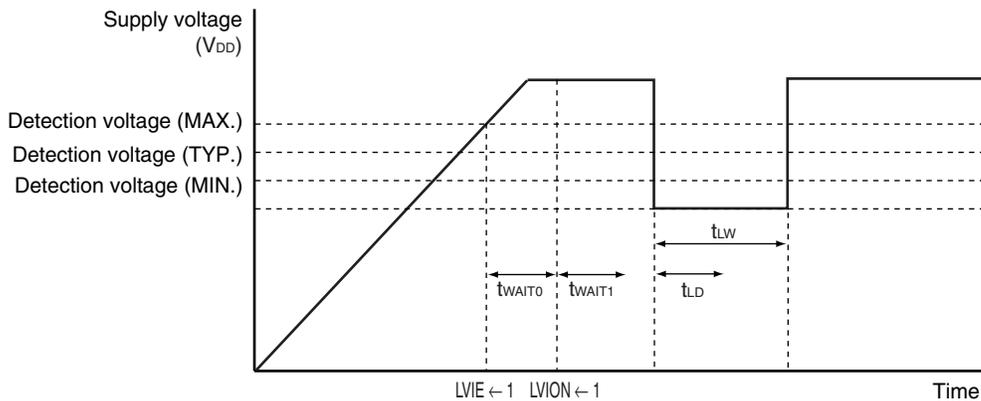
**LVI Circuit Characteristics (T<sub>A</sub> = -40 to +85°C)**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V <sub>LVI0</sub>		4.1	4.3	4.5	V
	V <sub>LVI1</sub>		3.9	4.1	4.3	V
	V <sub>LVI2</sub>		3.7	3.9	4.1	V
	V <sub>LVI3</sub>		3.5	3.7	3.9	V
	V <sub>LVI4</sub>		3.3	3.5	3.7	V
	V <sub>LVI5</sub>		3.15	3.3	3.45	V
	V <sub>LVI6</sub>		2.95	3.1	3.25	V
Response time <sup>Note 1</sup>	t <sub>LD</sub>			0.2	2.0	ms
Minimum pulse width	t <sub>LW</sub>		0.2			ms
Reference voltage stabilization wait time <sup>Note 2</sup>	t <sub>LWAIT0</sub>			0.5	2.0	ms
Operation stabilization wait time <sup>Note 3</sup>	t <sub>LWAIT1</sub>			0.1	0.2	ms

- Notes**
1. Time required from voltage detection to interrupt output or  $\overline{\text{RESET}}$  output.
  2. Time required from setting LVIE to 1 to reference voltage stabilization when POC = OFF is selected by the POC mask option.
  3. Time required from setting LVION to 1 to operation stabilization.

- Remarks**
1. V<sub>LVI0</sub> > V<sub>LVI1</sub> > V<sub>LVI2</sub> > V<sub>LVI3</sub> > V<sub>LVI4</sub> > V<sub>LVI5</sub> > V<sub>LVI6</sub>
  2. V<sub>POCn</sub> < V<sub>LVI m</sub> (n = 0 and 1, m = 0 to 6)

**LVI Circuit Timing**



**Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics (T<sub>A</sub> = -40 to +85°C)**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage	V <sub>DDDR</sub>		1.6		5.5	V
Release signal set time	t <sub>SREL</sub>		0			μs

**Flash Memory Programming Characteristics:  $\mu$ PD78F0103****( $T_A = +10$  to  $+60^\circ\text{C}$ ,  $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,  $2.7\text{ V} \leq AV_{REF} \leq V_{DD}$ ,  $V_{SS} = AV_{SS} = 0\text{ V}$ )****(1) Write erase characteristics**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$V_{PP}$ supply voltage	$V_{PP2}$	During flash memory programming	9.7	10.0	10.3	V
$V_{DD}$ supply current	$I_{DD}$	When $V_{PP} = V_{PP2}$ , $f_{XP} = 10\text{ MHz}$ , $V_{DD} = 5.5\text{ V}$			37	mA
$V_{PP}$ supply current	$I_{PP}$	$V_{PP} = V_{PP2}$			100	mA
Step erase time <sup>Note 1</sup>	$T_{er}$		0.199	0.2	0.201	s
Overall erase time <sup>Note 2</sup>	$T_{era}$	When step erase time = 0.2 s			20	s/chip
Writeback time <sup>Note 3</sup>	$T_{wb}$		49.4	50	50.6	ms
Number of writebacks per 1 writeback command <sup>Note 4</sup>	$C_{wb}$	When writeback time = 50 ms			60	Times
Number of erases/writebacks	$C_{erwb}$				16	Times
Step write time <sup>Note 5</sup>	$T_{wr}$		48	50	52	$\mu\text{s}$
Overall write time per word <sup>Note 6</sup>	$T_{wrw}$	When step write time = 50 $\mu\text{s}$ (1 word = 1 byte)	48		520	$\mu\text{s}$
Number of rewrites per chip <sup>Note 7</sup>	$C_{erwr}$	1 erase + 1 write after erase = 1 rewrite			20	Times

- Notes**
- The recommended setting value of the step erase time is 0.2 s.
  - The prewrite time before erasure and the erase verify time (writeback time) are not included.
  - The recommended setting value of the writeback time is 50 ms.
  - Writeback is executed once by the issuance of the writeback command. Therefore, the number of retries must be the maximum value minus the number of commands issued.
  - The recommended setting value of the step write time is 50  $\mu\text{s}$ .
  - The actual write time per word is 100  $\mu\text{s}$  longer. The internal verify time during or after a write is not included.
  - When a product is first written after shipment, “erase  $\rightarrow$  write” and “write only” are both taken as one rewrite.

Example: P: Write, E: Erase

Shipped product  $\rightarrow$  P  $\rightarrow$  E  $\rightarrow$  P  $\rightarrow$  E  $\rightarrow$  P: 3 rewrites

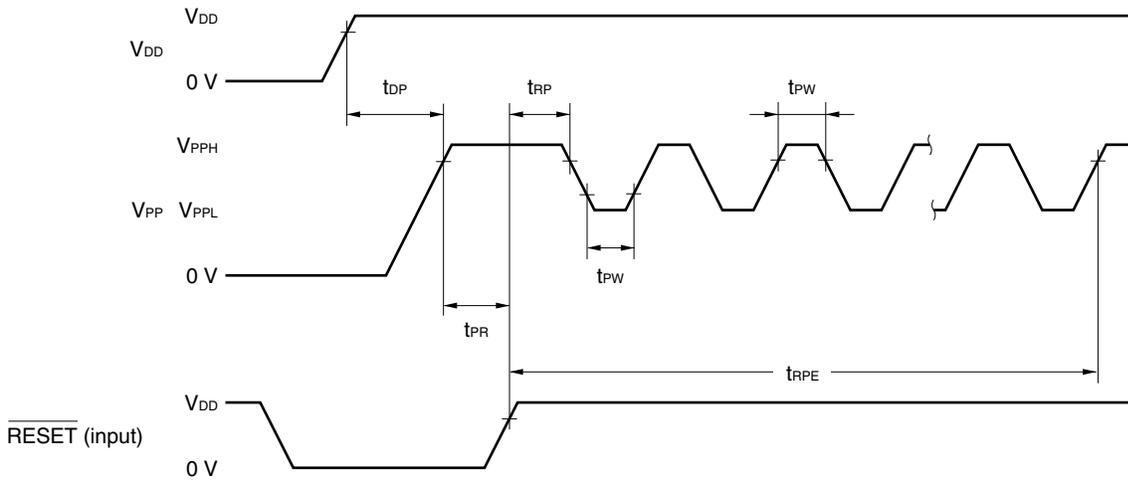
Shipped product  $\rightarrow$  E  $\rightarrow$  P  $\rightarrow$  E  $\rightarrow$  P  $\rightarrow$  E  $\rightarrow$  P: 3 rewrites

**Remark** The range of the operating clock during flash memory programming is the same as the range during normal operation.

(2) Serial write operation characteristics

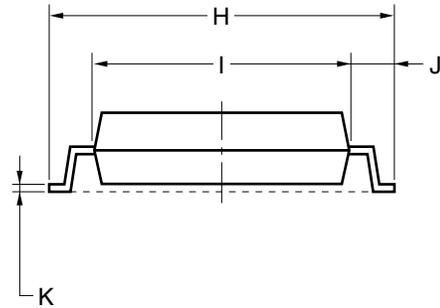
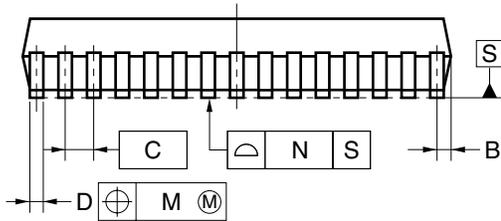
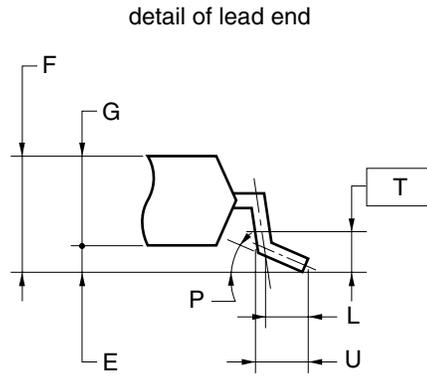
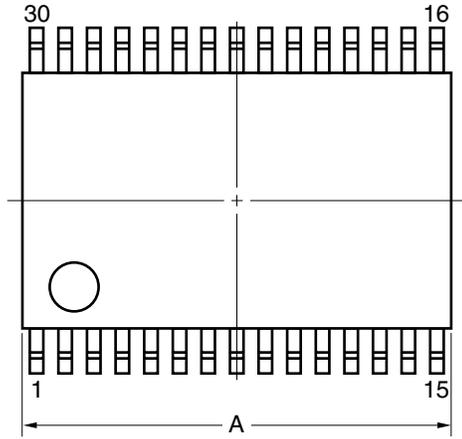
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Set time from $V_{DD}\uparrow$ to $V_{PP}\uparrow$	$t_{DP}$		10			$\mu s$
Release time from $V_{PP}\uparrow$ to $\overline{RESET}\uparrow$	$t_{PR}$		10			$\mu s$
$V_{PP}$ pulse input start time from $\overline{RESET}\uparrow$	$t_{RP}$		2			ms
$V_{PP}$ pulse high-/low-level width	$t_{PW}$		8			$\mu s$
$V_{PP}$ pulse input end time from $\overline{RESET}\uparrow$	$t_{RPE}$				20	ms
$V_{PP}$ pulse low-level input voltage	$V_{PPL}$		$0.8V_{DD}$		$1.2V_{DD}$	V
$V_{PP}$ pulse high-level input voltage	$V_{PPH}$		9.7	10.0	10.3	V

Flash Write Mode Setting Timing



## CHAPTER 24 PACKAGE DRAWING

### 30-PIN PLASTIC SSOP (7.62 mm (300))



**NOTE**

Each lead centerline is located within 0.13 mm of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS
A	9.85±0.15
B	0.45 MAX.
C	0.65 (T.P.)
D	0.24 <sup>+0.08</sup> <sub>-0.07</sub>
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° <sup>+5°</sup> <sub>-3°</sub>
T	0.25
U	0.6±0.15

**S30MC-65-5A4-2**

### 25.1 Cautions for Wait

This product has two internal system buses.

One is a CPU bus and the other is a peripheral bus that interfaces with the low-speed peripheral hardware.

Because the clock of the CPU bus and the clock of the peripheral bus are asynchronous, unexpected illegal data may be passed if an access to the CPU conflicts with an access to the peripheral hardware.

When accessing the peripheral hardware that may cause a conflict, therefore, the CPU repeatedly executes processing, until the correct data is passed.

As a result, the CPU does not start the next instruction processing but waits. If this happens, the number of execution clocks of an instruction increases by the number of wait clocks (for the number of wait clocks, refer to **Table 25-1**). This must be noted when real-time processing is performed.

## 25.2 Peripheral Hardware That Generates Wait

Table 25-1 lists the registers that issue a wait request when accessed by the CPU, and the number of CPU wait clocks.

**Table 25-1. Registers That Generate Wait and Number of CPU Wait Clocks**

Peripheral Hardware	Register	Access	Number of Wait Clocks
Watchdog timer	WDTM	Write	3 clocks (fixed)
Serial interface UART0	ASIS0	Read	1 clock (fixed)
Serial interface UART6	ASIS6	Read	1 clock (fixed)
A/D converter	ADM	Write	2 to 5 clocks <sup>Note</sup>
	ADS	Write	(when ADM.5 flag = "1")
	PFM	Write	2 to 9 clocks <sup>Note</sup>
	PFT	Write	(when ADM.5 flag = "0")
	ADCR	Read	1 to 5 clocks (when ADM.5 flag = "1") 1 to 9 clocks (when ADM.5 flag = "0")
<p>&lt;Calculating maximum number of wait clocks&gt;</p> $\{(1/f_{\text{MACRO}}) \times 2 / (1/f_{\text{CPU}})\} + 1$ <p>*The result after the decimal point is truncated if it is less than <math>t_{\text{CPUL}}</math> after it has been multiplied by <math>(1/f_{\text{CPU}})</math>, and is rounded up if it exceeds <math>t_{\text{CPUL}}</math>.</p> <p><math>f_{\text{MACRO}}</math>: Macro operating frequency (When bit 5 (FR2) of ADM = "1": <math>f_x/2</math>, when bit 5 (FR2) of ADM = "0": <math>f_x/2^2</math>)</p> <p><math>f_{\text{CPU}}</math>: CPU clock frequency</p> <p><math>t_{\text{CPUL}}</math>: Low-level width of CPU clock</p>			

**Note** No wait cycle is generated for the CPU if the number of wait clocks calculated by the above expression is 1.

**Remark** The clock is the CPU clock ( $f_{\text{CPU}}$ ).

### 25.3 Example of Wait Occurrence

<1> Watchdog timer

<On execution of MOV WDTM, A>

Number of execution clocks: 8

(5 clocks when data is written to a register that does not issue a wait (MOV sfr, A).)

<On execution of MOV WDTM, #byte>

Number of execution clocks: 10

(7 clocks when data is written to a register that does not issue a wait (MOV sfr, #byte).)

<2> Serial interface UART6

<On execution of MOV A, ASIS6>

Number of execution clocks: 6

(5 clocks when data is read from a register that does not issue a wait (MOV A, sfr).)

<3> A/D converter

**Table 25-2. Number of Wait Clocks and Number of Execution Clocks on Occurrence of Wait (A/D Converter)**

<On execution of MOV ADM, A; MOV ADS, A; or MOV A, ADCR>

- When  $f_x = 10 \text{ MHz}$ ,  $t_{\text{CPU}} = 50 \text{ ns}$

Value of Bit 5 (FR2) of ADM Register	$f_{\text{CPU}}$	Number of Wait Clocks	Number of Execution Clocks
0	$f_x$	9 clocks	14 clocks
	$f_x/2$	5 clocks	10 clocks
	$f_x/2^2$	3 clocks	8 clocks
	$f_x/2^3$	2 clocks	7 clocks
	$f_x/2^4$	0 clocks (1 clock <sup>Note</sup> )	5 clocks (6 clocks <sup>Note</sup> )
1	$f_x$	5 clocks	10 clocks
	$f_x/2$	3 clocks	8 clocks
	$f_x/2^2$	2 clocks	7 clocks
	$f_x/2^3$	0 clocks (1 clock <sup>Note</sup> )	5 clocks (6 clocks <sup>Note</sup> )
	$f_x/2^4$	0 clocks (1 clock <sup>Note</sup> )	5 clocks (6 clocks <sup>Note</sup> )

**Note** On execution of MOV A, ADCR

**Remark** The clock is the CPU clock ( $f_{\text{CPU}}$ ).

$f_x$ : X1 input clock frequency

$t_{\text{CPU}}$ : Low-level width of CPU clock

## APPENDIX A DEVELOPMENT TOOLS

The following development tools are available for the development of systems that employ the 78K0/KB1 Series. Figure A-1 shows the development tool configuration.

- **Support for PC98-NX series**

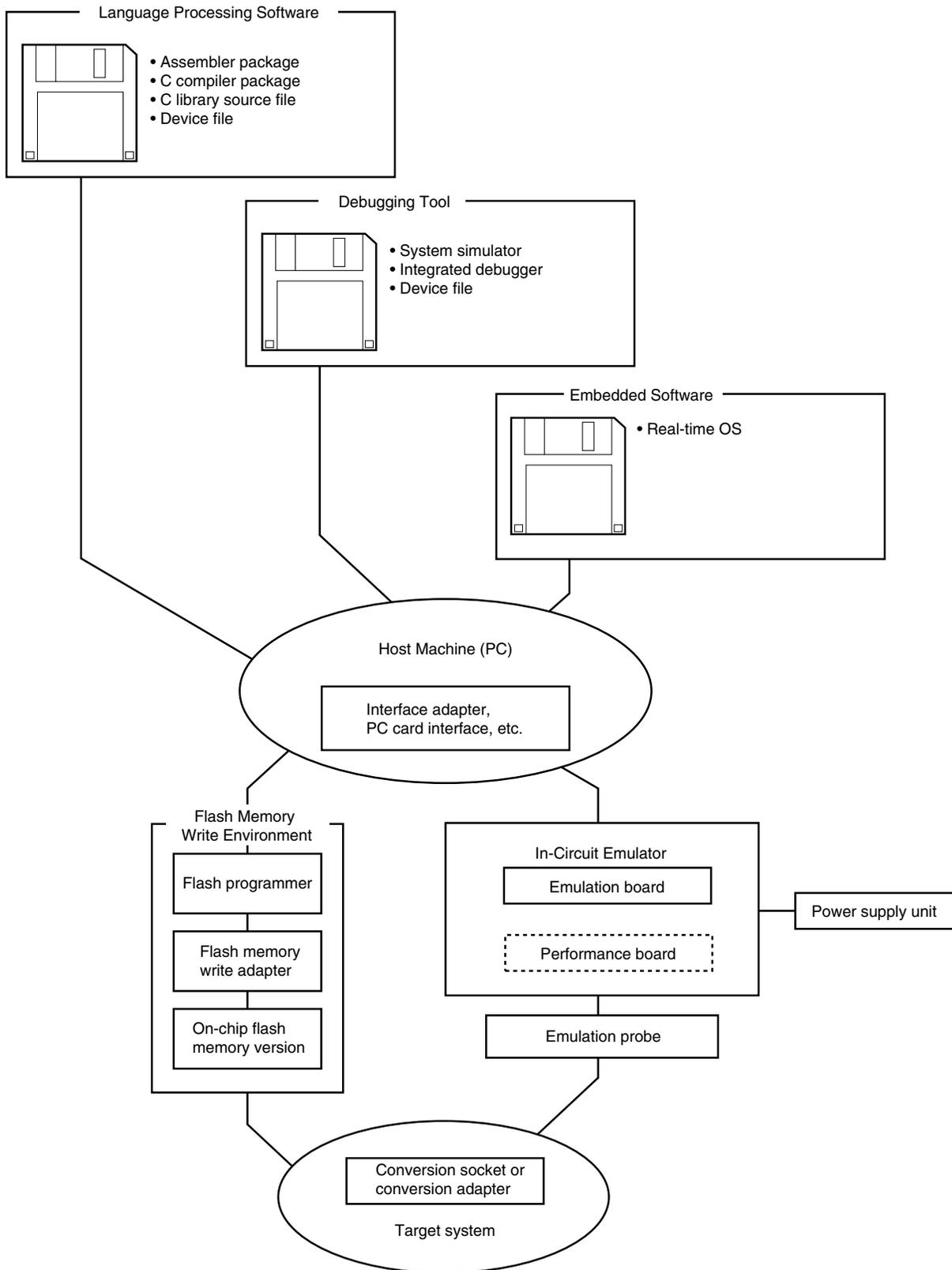
Unless otherwise specified, products supported by IBM PC/AT™ compatibles are compatible with PC98-NX series computers. When using PC98-NX series computers, refer to the explanation for IBM PC/AT compatibles.

- **Windows**

Unless otherwise specified, "Windows" means the following OSs.

- Windows 3.1
- Windows 95, 98, 2000
- Windows NT™ Ver 4.0

Figure A-1. Development Tool Configuration



**Remark** The item in the broken-line box differs according to the development environment. See **A.4.1 Hardware**.

**A.1 Software Package**

SP78K0 78K/0 Series software package	Development tools (software) common to the 78K/0 Series are combined in this package. Part number: $\mu$ SxxxxSP78K0
---	---

**Remark** xxxx in the part number differs depending on the host machine and OS used.

$\mu$ SxxxxSP78K0

xxxx	Host Machine	OS	Supply Medium
AB17	PC-9800 series,	Windows (Japanese version)	CD-ROM
BB17	IBM PC/AT compatibles	Windows (English version)	

**A.2 Language Processing Software**

<p>RA78K0 Assembler package</p>	<p>This assembler converts programs written in mnemonics into object codes executable with a microcontroller.</p> <p>This assembler is also provided with functions capable of automatically creating symbol tables and branch instruction optimization.</p> <p>This assembler should be used in combination with a device file (DF780103) (sold separately).</p> <p><b>&lt;Precaution when using RA78K0 in PC environment&gt;</b></p> <p>This assembler package is a DOS-based application. It can also be used in Windows, however, by using the Project Manager (included in assembler package) on Windows.</p> <p>Part number: <math>\mu</math>SxxxxRA78K0</p>
<p>CC78K0 C compiler package</p>	<p>This compiler converts programs written in C language into object codes executable with a microcontroller.</p> <p>This compiler should be used in combination with an assembler package and device file (both sold separately).</p> <p><b>&lt;Precaution when using CC78K0 in PC environment&gt;</b></p> <p>This C compiler package is a DOS-based application. It can also be used in Windows, however, by using the Project Manager (included in assembler package) on Windows.</p> <p>Part number: <math>\mu</math>SxxxxCC78K0</p>
<p>DF780103<sup>Notes 1, 2</sup> Device file</p>	<p>This file contains information peculiar to the device.</p> <p>This device file should be used in combination with a tool (RA78K0, CC78K0, SM78K0, ID78K0-NS, and ID78K0) (all sold separately).</p> <p>The corresponding OS and host machine differ depending on the tool to be used (all sold separately).</p> <p>Part number: <math>\mu</math>SxxxxDF780103</p>
<p>CC78K0-L<sup>Note 3</sup> C library source file</p>	<p>This is a source file of the functions that configure the object library included in the C compiler package (CC78K0).</p> <p>This file is required to match the object library included in the C compiler package to the user's specifications.</p> <p>Part number: <math>\mu</math>SxxxxCC78K0-L</p>

- Notes**
1. The DF780103 can be used in common with the RA78K0, CC78K0, SM78K0, ID78K0-NS, and ID78K0.
  2. Under development
  3. The CC78K0-L is not included in the software package (SP78K0).

**Remark** xxxx in the part number differs depending on the host machine and OS used.

μSxxxxRA78K0

μSxxxxCC78K0

xxxx	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	3.5-inch 2HD FD
BB13		Windows (English version)	
AB17		Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	
3P17	HP9000 series 700™	HP-UX™ (Rel. 10.10)	
3K17	SPARCstation™	SunOS™ (Rel. 4.1.4) Solaris™ (Rel. 2.5.1)	

μSxxxxDF780103

μSxxxxCC78K0-L

xxxx	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	3.5-inch 2HD FD
BB13		Windows (English version)	
3P16	HP9000 series 700	HP-UX (Rel. 10.10)	DAT
3K13	SPARCstation	SunOS (Rel. 4.1.4)	3.5-inch 2HD FD
3K15		Solaris (Rel. 2.5.1)	1/4-inch CGMT

### A.3 Flash Memory Writing Tools

Flashpro III (part number: FL-PR3, PG-FP3) Flash programmer	Flash programmer dedicated to microcontrollers with on-chip flash memory.
FA-30MC Flash memory writing adapter	Flash memory writing adapter used connected to the Flashpro III. 30-pin plastic SSOP (MC-5A4 type)
Flashpro III controller	Program to control Flashpro III from a PC. Provided with Flashpro III.

**Remark** FL-PR3 and FA-30MC are products of Naito Densai Machida Mfg. Co., Ltd.  
TEL: +81-45-475-4191 Naito Densai Machida Mfg. Co., Ltd.

## A.4 Debugging Tools

### A.4.1 Hardware

IE-78K0-NS In-circuit emulator	The in-circuit emulator serves to debug hardware and software when developing application systems using a 78K/0 Series product. It corresponds to the integrated debugger (ID78K0-NS). This emulator should be used in combination with a power supply unit, emulation probe, and the interface adapter required to connect this emulator to the host machine.
IE-78K0-NS-PA Performance board	This board is connected to the IE-78K0-NS to expand its functions. Adding this board adds a coverage function and enhances debugging functions such as tracer and timer functions.
IE-78K0-NS-A In-circuit emulator	Product that combines the IE-78K0-NS and IE-78K0-NS-PA
IE-70000-MC-PS-B Power supply unit	This adapter is used for supplying power from a 100 V to 240 V AC outlet.
IE-70000-98-IF-C Interface adapter	This adapter is required when using a PC-9800 series computer (except notebook type) as the IE-78K0-NS(-A) host machine (C bus compatible).
IE-70000-CD-IF-A PC card interface	This is PC card and interface cable required when using a notebook-type computer as the IE-78K0-NS(-A) host machine (PCMCIA socket compatible).
IE-70000-PC-IF-C Interface adapter	This adapter is required when using an IBM PC compatible computer as the IE-78K0-NS(-A) host machine (ISA bus compatible).
IE-70000-PCI-IF-A Interface adapter	This adapter is required when using a computer with a PCI bus as the IE-78K0-NS(-A) host machine.
IE-780148-NS-EM1 <sup>Note</sup> Emulation board	This board emulates the operations of the peripheral hardware peculiar to a device. It should be used in combination with an in-circuit emulator.
★ NP-36GS Emulation probe	This probe is used to connect the in-circuit emulator to a target system and is designed for use with 30-pin plastic SSOP (MC-5A4 type).
★ NGS-30 Conversion socket	This conversion socket connects the NP-36GS to a target system board designed for a 30-pin plastic SSOP (MC-5A4 type).
★ NP-30MC Emulation probe	This probe is used to connect the in-circuit emulator to the target system and is designed for use with a 30-pin plastic SSOP (MC-5A4 type).
★ NSPACK30BK YSPACK30BK HSPACK30BK YQGUIDE Conversion socket	This conversion socket connects the NP-30MC to a target system board designed to mount a 30-pin plastic SSOP (MC-5A4 type).

**Note** Under development

**Remarks 1.** NP-36GS, NP-30MC, and NGS-30 are products of Naito Densai Machida Mfg. Co., Ltd.

TEL: +81-45-475-4191 Naito Densai Machida Mfg. Co., Ltd.

**2.** NSPACK30BK, YSPACK30BK, HSPACK30BK, and YQGUIDE are products of TOKYO ELETECH CORPORATION.

For further information, contact Daimaru Kogyo Co., Ltd.

Tokyo Electronics Department (TEL: +81-3-3820-7112)

Osaka Electronics Department (TEL: +81-6-6244-6672)

A.4.2 Software

<p>SM78K0 System simulator</p>	<p>This system simulator is used to perform debugging at C source level or assembler level while simulating the operation of the target system on a host machine. This simulator runs on Windows. Use of the SM78K0 allows the execution of application logical testing and performance testing on an independent basis from hardware development without having to use an in-circuit emulator, thereby providing higher development efficiency and software quality. The SM78K0 should be used in combination with a device file (DF780103) (sold separately).</p> <p>Part number: <math>\mu</math>SxxxxSM78K0</p>
<p>ID78K0-NS Integrated debugger (supporting in-circuit emulator IE-78K0-NS(-A))</p>	<p>This debugger is a control program used to debug 78K/0 Series microcontrollers. It adopts a graphical user interface, which is equivalent visually and operationally to Windows or OSF/Motif™. It also has an enhanced debugging function for C language programs, and thus trace results can be displayed on screen at C-language level by using the windows integration function which links a trace result with its source program, disassembled display, and memory display. In addition, by incorporating function modules such as a task debugger and system performance analyzer, the efficiency of debugging programs that run on real-time OSs can be improved. It should be used in combination with a device file (sold separately).</p> <p>Part number: <math>\mu</math>SxxxxID78K0-NS</p>

**Remark** xxxx in the part number differs depending on the host machine and OS used.

$\mu$ SxxxxSM78K0  
 $\mu$ SxxxxID78K0-NS

xxxx	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	3.5-inch 2HD FD
BB13		Windows (English version)	
AB17		Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	

## APPENDIX B EMBEDDED SOFTWARE

The following embedded products are available for efficient development and maintenance of the 78K0/KB1 Series.

### Real-Time OS

RX78K/0 Real-time OS	The RX78K/0 is a real-time OS conforming to the $\mu$ TRON specifications. A tool (configurator) for generating the nucleus of the RX78K/0 and multiple information tables is supplied. Used in combination with an assembler package (RA78K/0) and device file (DF780103) (both sold separately). <b>&lt;Precaution when using RX78K/0 in PC environment&gt;</b> The real-time OS is a DOS-based application. It should be used in the DOS prompt when using it in Windows.
Part number: $\mu$ SxxxxRX78013- $\Delta\Delta\Delta\Delta$	

**Caution** To purchase the RX78K/0, first fill in the purchase application form and sign the user agreement.

**Remark** xxxx and  $\Delta\Delta\Delta\Delta$  in the part number differ depending on the host machine and OS used.

$\mu$ SxxxxRX78013- $\Delta\Delta\Delta\Delta$

$\Delta\Delta\Delta\Delta$	Product Outline	Maximum Number for Use in Mass Production
001	Evaluation object	Do not use for mass-produced product.
100K	Mass-production object	0.1 million units
001M		1 million units
010M		10 million units
S01	Source program	Object source program for mass production

$\Delta\Delta\Delta\Delta$	Host Machine	OS	Supply Medium
AA13	PC-9800 series	Windows (Japanese version) <sup>Note</sup>	3.5-inch 2HD FD
AB13	IBM PC/AT compatibles	Windows (Japanese version) <sup>Note</sup>	3.5-inch 2HD FD
BB13		Windows (English version) <sup>Note</sup>	
3P16	HP9000 series 700	HP-UX (Rel. 10.10)	DAT
3K13	SPARCstation	SunOS (Rel. 4.1.4),	3.5-inch 2HD FD
3K15		Solaris (Rel. 2.5.1)	1/4-inch CGMT

**Note** Can also be operated in DOS environment.

## APPENDIX C REGISTER INDEX

### C.1 Register Index (In Alphabetical Order with Respect to Register Names)

#### [A]

A/D conversion result register (ADCR) ... 174  
A/D converter mode register (ADM) ... 176  
Analog input channel specification register (ADS) ... 178  
Asynchronous serial interface control register 6 (ASICL6) ... 228  
Asynchronous serial interface operation mode register 0 (ASIM0) ... 196  
Asynchronous serial interface operation mode register 6 (ASIM6) ... 222  
Asynchronous serial interface reception error status register 0 (ASIS0) ... 198  
Asynchronous serial interface reception error status register 6 (ASIS6) ... 224  
Asynchronous serial interface transmission status register 6 (ASIF6) ... 225

#### [B]

Baud rate generator control register 0 (BRGC0) ... 199  
Baud rate generator control register 6 (BRGC6) ... 227

#### [C]

Capture/compare control register 00 (CRC00) ... 117  
Clock monitor mode register (CLM) ... 305  
Clock selection register 6 (CKSR6) ... 226

#### [E]

8-bit timer compare register 50 (CR50) ... 138  
8-bit timer counter 50 (TM50) ... 138  
8-bit timer H compare register 00 (CMP00) ... 150  
8-bit timer H compare register 01 (CMP01) ... 150  
8-bit timer H compare register 10 (CMP10) ... 150  
8-bit timer H compare register 11 (CMP11) ... 150  
8-bit timer H mode register 0 (TMHMD0) ... 151  
8-bit timer H mode register 1 (TMHMD1) ... 151  
8-bit timer mode control register 50 (TMC50) ... 140  
External interrupt falling edge enable register (EGN) ... 279  
External interrupt rising edge enable register (EGP) ... 279

#### [I]

Input switch control register (ISC) ... 91  
Internal memory size switching register (IMS) ... 328  
Interrupt mask flag register 0H (MK0H) ... 277  
Interrupt mask flag register 0L (MK0L) ... 277  
Interrupt mask flag register 1L (MK1L) ... 277  
Interrupt request flag register 0H (IF0H) ... 276  
Interrupt request flag register 0L (IF0L) ... 276  
Interrupt request flag register 1L (IF1L) ... 276

**[L]**

Low-voltage detection level selection register (LVIS) ... 317

Low-voltage detection register (LVIM) ... 315

**[M]**

Main clock mode register (MCM) ... 97

Main OSC control register (MOC) ... 98

**[O]**

Oscillation stabilization time counter status register (OSTC) ... 98, 290

Oscillation stabilization time select register (OSTS) ... 99, 291

**[P]**

Port 0 (P0) ... 75

Port 1 (P1) ... 78

Port 12 (P12) ... 86

Port 13 (P13) ... 87

Port 2 (P2) ... 84

Port 3 (P3) ... 85

Port mode register 0 (PM0) ... 87, 120

Port mode register 1 (PM1) ... 87, 141

Port mode register 12 (PM12) ... 87

Port mode register 3 (PM3) ... 87

Power-fail comparison mode register (PFM) ... 178

Power-fail comparison threshold register (PFT) ... 179

Prescaler mode register 00 (PRM00) ... 119

Priority specification flag register 0H (PR0H) ... 278

Priority specification flag register 0L (PR0L) ... 278

Priority specification flag register 1L (PR1L) ... 278

Processor clock control register (PCC) ... 95

Pull-up resistor option register 0 (PU0) ... 90

Pull-up resistor option register 1 (PU1) ... 90

Pull-up resistor option register 12 (PU12) ... 90

Pull-up resistor option register 3 (PU3) ... 90

**[R]**

Receive buffer register 0 (RXB0) ... 195

Receive buffer register 6 (RXB6) ... 221

Reset control flag register (RESF) ... 303

Ring-OSC mode register (RCM) ... 96

**[S]**

Serial clock selection register 10 (CSIC10) ... 261

Serial operation mode register 10 (CSIM10) ... 260

Serial I/O shift register 10 (SIO10) ... 259

16-bit timer capture/compare register 000 (CR000) ... 113

16-bit timer capture/compare register 010 (CR010) ... 114

16-bit timer counter 00 (TM00) ... 113

16-bit timer mode control register 00 (TMC00) ... 115  
16-bit timer output control register 00 (TOC00) ... 118

**[T]**

Timer clock selection register 50 (TCL50) ... 139  
Transmit buffer register 10 (SOTB10) ... 259  
Transmit buffer register 6 (TXB6) ... 221  
Transmit shift register 0 (TXS0) ... 195

**[W]**

Watchdog timer enable register (WDTE) ... 167  
Watchdog timer mode register (WDTM) ... 165

**C.2 Register Index (In Alphabetical Order with Respect to Register Symbol)****[A]**

ADCR:	A/D conversion result register ... 174
ADM:	A/D converter mode register ... 176
ADS:	Analog input channel specification register ... 178
ASICL6:	Asynchronous serial interface control register 6 ... 228
ASIF6:	Asynchronous serial interface transmission status register 6 ... 225
ASIM0:	Asynchronous serial interface operation mode register 0 ... 196
ASIM6:	Asynchronous serial interface operation mode register 6 ... 222
ASIS0:	Asynchronous serial interface reception error status register 0 ... 198
ASIS6:	Asynchronous serial interface reception error status register 6 ... 224

**[B]**

BRGC0:	Baud rate generator control register 0 ... 199
BRGC6:	Baud rate generator control register 6 ... 227

**[C]**

CKSR6:	Clock selection register 6 ... 226
CLM:	Clock monitor mode register ... 305
CMP00:	8-bit timer H compare register 00 ... 150
CMP01:	8-bit timer H compare register 01 ... 150
CMP10:	8-bit timer H compare register 10 ... 150
CMP11:	8-bit timer H compare register 11 ... 150
CR000:	16-bit timer capture/compare register 000 ... 113
CR010:	16-bit timer capture/compare register 010 ... 114
CR50:	8-bit timer compare register 50 ... 138
CRC00:	Capture/compare control register 00 ... 117
CSIC10:	Serial clock selection register 10 ... 261
CSIM10:	Serial operation mode register 10 ... 260

**[E]**

EGN:	External interrupt falling edge enable register ... 279
EGP:	External interrupt rising edge enable register ... 279

**[I]**

IF0H:	Interrupt request flag register 0H ... 276
IF0L:	Interrupt request flag register 0L ... 276
IF1L:	Interrupt request flag register 1L ... 276
IMS:	Internal memory size switching register ... 328
ISC:	Input switch control register ... 91

**[L]**

LVIM:	Low-voltage detection register ... 315
LVIS:	Low-voltage detection level selection register ... 317

**[M]**

MCM:	Main clock mode register ... 97
------	---------------------------------

MK0H:	Interrupt mask flag register 0H ... 277
MK0L:	Interrupt mask flag register 0L ... 277
MK1L:	Interrupt mask flag register 1L ... 277
MOC:	Main OSC control register ... 98
<b>[O]</b>	
OSTC:	Oscillation stabilization time counter status register ... 98, 290
OSTS:	Oscillation stabilization time select register ... 99, 291
<b>[P]</b>	
P0:	Port 0 ... 75
P1:	Port 1 ... 78
P12:	Port 12 ... 86
P13:	Port 13 ... 87
P2:	Port 2 ... 84
P3:	Port 3 ... 85
PCC:	Processor clock control register ... 95
PFM:	Power-fail comparison mode register ... 178
PFT:	Power-fail comparison threshold register ... 179
PM0:	Port mode register 0 ... 87, 120
PM1:	Port mode register 1 ... 87, 141
PM12:	Port mode register 12 ... 87
PM3:	Port mode register 3 ... 87
PR0H:	Priority specification flag register 0H ... 278
PR0L:	Priority specification flag register 0L ... 278
PR1L:	Priority specification flag register 1L ... 278
PRM00:	Prescaler mode register 00 ... 119
PU0:	Pull-up resistor option register 0 ... 90
PU1:	Pull-up resistor option register 1 ... 90
PU12:	Pull-up resistor option register 12 ... 90
PU3:	Pull-up resistor option register 3 ... 90
<b>[R]</b>	
RCM:	Ring-OSC mode register ... 96
RESF:	Reset control flag register ... 303
RXB0:	Receive buffer register 0 ... 195
RXB6:	Receive buffer register 6 ... 221
<b>[S]</b>	
SIO10:	Serial I/O shift register 10 ... 259
SOTB10:	Transmit buffer register 10 ... 259
<b>[T]</b>	
TCL50:	Timer clock selection register 50 ... 139
TM00:	16-bit timer counter 00 ... 113
TM50:	8-bit timer counter 50 ... 138
TMC00:	16-bit timer mode control register 00 ... 115
TMC50:	8-bit timer mode control register 50 ... 140

TMHMD0: 8-bit timer H mode register 0 ... 151  
TMHMD1: 8-bit timer H mode register 1 ... 151  
TOC00: 16-bit timer output control register 00 ... 118  
TXB6: Transmit buffer register 6 ... 221  
TXS0: Transmit shift register 0 ... 195

**[W]**

WDTE: Watchdog timer enable register ... 167  
WDTM: Watchdog timer mode register ... 165

## APPENDIX D REVISION HISTORY

A history of the revisions up to this edition is shown below. “Applied to:” indicates the chapters to which the revision was applied.

(1/2)

Edition	Description	Applied to:
2nd	X1 input clock oscillation stabilization time $2^{12}/f_x, 2^{14}/f_x, 2^{15}/f_x, 2^{16}/f_x, 2^{17}/f_x \rightarrow 2^{11}/f_x, 2^{13}/f_x, 2^{14}/f_x, 2^{15}/f_x, 2^{16}/f_x$	Throughout
	Modification of <b>Figure 4-5 Block Diagram of P10</b>	CHAPTER 4 PORT FUNCTIONS
	Modification of <b>Table 4-3 Settings of Port Mode Register and Output Latch When Alternate-Function Is Used</b>	
	Modification of <b>Figure 5-6 Format of Oscillation Stabilization Time Counter Status Register (OSTC)</b>	CHAPTER 5 CLOCK GENERATOR
	Modification of <b>Figure 5-7 Format of Oscillation Stabilization Time Select Register (OSTS)</b>	
	Addition of <b>5.7 Clock Selection Flowchart and Register Settings</b>	
	Addition of <b>Remark to 12.1 Functions of Serial Interface UART6</b>	CHAPTER 12 SERIAL INTERFACE UART6
	Addition of Reset to <b>Table 14-1 Interrupt Source List</b>	CHAPTER 14 INTERRUPT FUNCTIONS
	Modification of <b>Figure 15-2 Format of Oscillation Stabilization Time Counter Status Register (OSTC)</b>	CHAPTER 15 STANDBY FUNCTION
	Modification of <b>Figure 15-3 Format of Oscillation Stabilization Time Select Register (OSTS)</b>	
	Addition of <b>CHAPTER 25 RETRY</b>	CHAPTER 25 RETRY
2nd (corrected edition)	Modification of reset value of the following register in <b>Table 3-5 Special Function Register List</b> <ul style="list-style-type: none"> <li>• Serial I/O shift register 10 (SIO10)</li> </ul>	CHAPTER 3 CPU ARCHITECTURE
	Modification of manipulatable bit unit of the following registers in <b>Table 3-5 Special Function Register List</b> <ul style="list-style-type: none"> <li>• Oscillation stabilization time counter status register (OSTC)</li> <li>• Interrupt request flag register 1L (IF1L)</li> <li>• Interrupt mask flag register 1L (MK1L)</li> <li>• Priority specification flag register 1L (PR1L)</li> </ul>	
	Modification of manipulatable bit unit in <b>5.3 (5) Oscillation stabilization time counter status register (OSTC)</b>	CHAPTER 5 CLOCK GENERATOR
	Modification of <b>Figure 5-11 Status Transition Diagram</b>	
	Modification of <b>Table 5-3 Relationship Between Operation Clocks in Each Operation Status</b>	
	Modification of <b>Table 5-4 Oscillation Control Flags and Clock Oscillation Status</b>	
	Modification of <b>Table 5-6 Clock and Register Settings</b>	
	Modification of reset value in <b>6.2 (2) 16-bit timer capture/compare register 000 (CR000)</b> and <b>(3) 16-bit timer capture/compare register 010 (CR010)</b>	CHAPTER 6 16-BIT TIMER/EVENT COUNTER 00
	Modification of manipulatable bit unit in <b>6.3 (4) Prescaler mode register 00 (PRM00)</b>	

Edition	Description	Applied to:
2nd (corrected edition)	Modification of <b>Caution in 9.4.2 Watchdog timer operation when “Ring-OSC can be stopped by software” is selected by mask option</b>	CHAPTER 9 WATCHDOG TIMER
	Modification of <b>9.4.3 Watchdog timer operation in STOP mode (when “Ring-OSC can be stopped by software” is selected by mask option)</b>	
	Addition of <b>9.4.4 Watchdog timer operation in HALT mode (when “Ring-OSC can be stopped by software” is selected by mask option)</b>	
	Addition of <b>(11) A/D converter sampling time and A/D conversion start delay time in 10.6 Cautions for A/D Converter</b>	CHAPTER 10 A/D CONVERTER
	Modification of reset value in <b>13.2 (2) Serial I/O shift register 10 (SIO10)</b>	CHAPTER 13 SERIAL INTERFACE CSI10
	Modification of manipulatable bit unit in <b>15.1.2 (1) Oscillation stabilization time counter status register (OSTC)</b>	CHAPTER 15 STANDBY FUNCTION
	Modification of A/D converter item in <b>Table 15-2 Operating Statuses in HALT mode</b>	
	Addition of <b>18.4 Cautions for Power-on-Clear Circuit</b>	CHAPTER 18 POWER-ON- CLEAR CIRCUIT
	Modification of <b>Figure 19-3 Format of Low-Voltage Detection Level Selection Register (LVIS)</b>	CHAPTER 19 LOW-VOLTAGE DETECTOR
	Addition of <b>19.5 Cautions for Low-Voltage Detector</b>	
	Modification of the following contents in <b>CHAPTER 23 ELECTRICAL SPECIFICATIONS (TARGET VALUES)</b> <ul style="list-style-type: none"> <li>• <b>Absolute Maximum Ratings</b></li> <li>• <b>X1 Oscillator Characteristics</b></li> <li>• <b>DC Characteristics</b></li> <li>• <b>A/D Converter Characteristics</b></li> <li>• <b>POC Circuit Characteristics</b></li> <li>• <b>LVI Circuit Characteristics</b></li> <li>• <b>Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics</b> (deletion of data retention supply current)</li> <li>• <b>Deletion of Ring-OSC Characteristics</b></li> <li>• <b>Flash Memory Programming Characteristics</b></li> </ul>	CHAPTER 23 ELECTRICAL SPECIFICATIONS (TARGET VALUES)
	Modification from <b>CHAPTER 25 RETRY</b> to <b>CHAPTER 25 CAUTIONS FOR WAIT</b>	CHAPTER 25 CAUTIONS FOR WAIT

## Facsimile Message

Although NEC has taken all possible steps to ensure that the documentation supplied to our customers is complete, bug free and up-to-date, we readily accept that errors may occur. Despite all the care and precautions we've taken, you may encounter problems in the documentation. Please complete this form whenever you'd like to report errors or suggest improvements to us.

From:

Name

Company

Tel.

FAX

Address

**Thank you for your kind support.**

### North America

NEC Electronics Inc.  
Corporate Communications Dept.  
Fax: +1-800-729-9288  
+1-408-588-6130

### Hong Kong, Philippines, Oceania

NEC Electronics Hong Kong Ltd.  
Fax: +852-2886-9022/9044

### Taiwan

NEC Electronics Taiwan Ltd.  
Fax: +886-2-2719-5951

### Europe

NEC Electronics (Europe) GmbH  
Market Communication Dept.  
Fax: +49-211-6503-274

### Korea

NEC Electronics Hong Kong Ltd.  
Seoul Branch  
Fax: +82-2-528-4411

### Asian Nations except Philippines

NEC Electronics Singapore Pte. Ltd.  
Fax: +65-250-3583

### South America

NEC do Brasil S.A.  
Fax: +55-11-6462-6829

### P.R. China

NEC Electronics Shanghai, Ltd.  
Fax: +86-21-6841-1137

### Japan

NEC Semiconductor Technical Hotline  
Fax: +81- 44-435-9608

I would like to report the following error/make the following suggestion:

Document title: \_\_\_\_\_

Document number: \_\_\_\_\_ Page number: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

If possible, please fax the referenced page or drawing.

Document Rating	Excellent	Good	Acceptable	Poor
Clarity	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Technical Accuracy	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Organization	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>