To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

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

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

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

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

User’s Manual

μPD780344, 780354, 780344Y, 780354Y Subseries

8-Bit Single-Chip Microcontrollers

μPD780343	μPD780343Y
μPD780344	μPD780344Y
μPD780353	μPD780353Y
μPD780354	μPD780354Y
μPD78F0354	μPD78F0354Y
μPD78F0354A	μPD78F0354AY

Document No. U15798EJ2V1UD00 (2nd edition)
Date Published August 2005 N CP(K)

[MEMO]

NOTES FOR CMOS DEVICES

① VOLTAGE APPLICATION WAVEFORM AT INPUT PIN

Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (MAX) and V_{IH} (MIN) due to noise, etc., the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (MAX) and V_{IH} (MIN).

② HANDLING OF UNUSED INPUT PINS

Unconnected CMOS device inputs can be cause of malfunction. If an input pin is unconnected, it is possible that an internal input level may be generated due to noise, etc., causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using pull-up or pull-down circuitry. Each unused pin should be connected to V_{DD} or GND via a resistor if there is a possibility that it will be an output pin. All handling related to unused pins must be judged separately for each device and according to related specifications governing the device.

③ PRECAUTION AGAINST ESD

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

④ STATUS BEFORE INITIALIZATION

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

⑤ POWER ON/OFF SEQUENCE

In the case of a device that uses different power supplies for the internal operation and external interface, as a rule, switch on the external power supply after switching on the internal power supply. When switching the power supply off, as a rule, switch off the external power supply and then the internal power supply. Use of the reverse power on/off sequences may result in the application of an overvoltage to the internal elements of the device, causing malfunction and degradation of internal elements due to the passage of an abnormal current.

The correct power on/off sequence must be judged separately for each device and according to related specifications governing the device.

⑥ INPUT OF SIGNAL DURING POWER OFF STATE

Do not input signals or an I/O pull-up power supply while the device is not powered. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements.

Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.

FIP, EEPROM, and IEBus are trademarks of NEC Electronics Corporation.

Windows and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

PC/AT is a trademark of International Business Machines Corporation.

HP9000 Series 700 and HP-UX are trademarks of Hewlett-Packard Company.

SPARCstation is a trademark of SPARC International, Inc.

Solaris and SunOS are trademarks of Sun Microsystems, Inc.

TRON is an abbreviation of The Realtime Operating system Nucleus.

ITRON is an abbreviation of Industrial TRON.

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

- **The information in this document is current as of August, 2005. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC Electronics sales representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC Electronics does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.
- NEC Electronics products are classified into the following three quality grades: "Standard", "Special" and "Specific".
  The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of an NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product before using it in a particular application.
  "Standard": Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots.
  "Special": Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support).
  "Specific": Aircraft, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems and medical equipment for life support, etc.

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

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

M8E 02. 11-1

Regional Information

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

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

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

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

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

NEC Electronics (Europe) GmbH
Duesseldorf, Germany
Tel: 0211-65030

- **Sucursal en España**
  Madrid, Spain
  Tel: 091-504 27 87

- **Succursale Française**
  Vélizy-Villacoublay, France
  Tel: 01-30-67 58 00

- **Filiale Italiana**
  Milano, Italy
  Tel: 02-66 75 41

- **Branch The Netherlands**
  Eindhoven, The Netherlands
  Tel: 040-265 40 10

- **Tyskland Filial**
  Taeby, Sweden
  Tel: 08-63 87 200

- **United Kingdom Branch**
  Milton Keynes, UK
  Tel: 01908-691-133

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

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

NEC Electronics Shanghai Ltd.
Shanghai, P.R. China
Tel: 021-5888-5400

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

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

J05.6

Major Revisions in This Edition (1/3)

Page	Description
U15798EJ1V0UD00 → U15798EJ2V0UD00	
Throughout	Deletion of indication “under development” for all target products
	AV_{REF} pin → AV_{DD} pin
	A/D converter operation enable voltage AV_{REF} = 2.7 to 5.5 V → AV_{DD} = 2.2 to 5.5 V
p.34	Change of 113-pin plastic FBGA package in **1.4 Pin Configuration (Top View)**
p.53	Modification of **Table 2-1 Pin I/O Circuit Types**
p.62	Addition of description on program area in **3.1.2 (1) Internal high-speed RAM** and **(2) Internal expansion RAM**
pp.68, 69	Change of **Figure 3-10 Data To Be Saved to Stack Memory** and **Figure 3-11 Data To Be Restored from Stack Memory**
p.82	Modification of **[Description example]** in **3.4.4 Short direct addressing**
pp.85 to 87	Addition of **[Illustration]** in **3.4.7 Based addressing, 3.4.8 Based indexed addressing**, and **3.4.9 Stack addressing**
p.89	Modification of description of port 1 and port 4 in **Table 4-1 Port Functions**
p.93	Modification of **Figure 4-4 P10 to P17 Block Diagram**
p.94	Addition of **Caution** in **4.2.3 Port 2**
p.108	Modification of **Note** in **Figure 4-18 Format of Port Mode Registers (PM0, PM2 to PM4, PM7 to PM11)**
p.111	Modification of setting and addition of **Caution 2** in **Figure 4-21 Format of Pin Function Switching Registers (PF8 to PF11)**
p.124	Addition of description in **5.5.1 Main system clock operations**
p.129	Modification of **Figure 6-1 Block Diagram of 16-Bit Timer/Event Counter 0**
p.141	Addition of **Figure 6-11 Configuration of PPG Output** and **Figure 6-12 PPG Output Operation Timing**
pp.153, 156	Modification of **6.6 (4) Capture register data retention timing** and addition of **(13) STOP mode and main system clock stop mode settings**
p.150 in 1st edition	Deletion of <1> in **6.6 (7) Conflicting operations** in 1st edition
p.166	Modification of **Figure 7-6 Format of Carrier Generator Output Control Register B0**
p.180	Addition of input frequency from TMIB0 pin in **Table 7-7 Square-Wave Output Range with 16-Bit Resolution**
p.191	Addition of description in **8.3 (2) 8-bit timer compare register 5n (CR5n: n = 0, 1)**
p.199	Addition of **[Setting]** in **8.5.2 External event counter operation**
p.200	Addition of description on frequencies in **[Setting]** in **8.5.3 Square-wave output operation**
p.201	Modification of description of **[Setting]** in **8.5.4 PWM output operation**
p.208	Correction of **Figure 9-2 Format of Watch Timer Operation Mode Register 0 (WTNM0)**
p.225	Correction of **12.2 (3) Sample & hold circuit** and **(4) Voltage comparator**
pp.228, 229	Modification of description of **Note 3** in **Figure 12-2 Format of A/D Converter Mode Register 0 (ADM0)**, and addition of **Table 12-2 Settings of ADCS0 and ADCE0** and **Figure 12-3 Timing Chart When Boost Reference Voltage Generator Is Used**
p.240	Modification of **Figure 12-12 Analog Input Pin Connection**
 p.240 p.243	Addition of the followings in **12.6 A/D Converter Cautions** **(6) Input impedance of ANI0 to ANI7 pins** **(14) AV_{DD} pin**

Major Revisions in This Edition (2/3)

Page	Description
p.243	Change of **Figure 12-16 Example of Connecting Capacitor to V_{DD1} and AV_{REF} Pins** in 1st edition to **Figure 12-16 Example of Connecting Capacitor to AV_{DD} Pin**
p.244	Modification of **Table 12-3 Resistances and Capacitances of Equivalent Circuit (Reference Values)**
p.246	Addition and modification of description in **13.2 (2) A/D conversion result register 0 (ADCR0)**, **(3) Sample & hold circuit**, and **(4) Voltage comparator**
pp.248, 249	Modification of description of **Note 3** in **Figure 13-2 Format of A/D Converter Mode Register 0 (ADM0)**, and addition of **Table 13-2 Settings of ADCS0 and ADCE0** and **Figure 13-3 Timing Chart When Boost Reference Voltage Generator Is Used**
p.261	Modification of **Figure 13-16 Analog Input Pin Connection**
 p.261 p.264	Addition of the followings in **13.6 A/D Converter Cautions** **(6) Input impedance of ANI0 to ANI7 pins** **(14) AV_{DD} pin**
p.264	Change of **Figure 13-20 Example of Connecting Capacitor to V_{DD1} and AV_{REF} Pins** in 1st edition to **Figure 13-20 Example of Connecting Capacitor to AV_{DD} Pin**
p.265	Modification of **Table 13-3 Resistances and Capacitances of Equivalent Circuit (Reference Values)**
p.269	Modification of description of MODE flag in **Figure 14-2 Format of Serial Operation Mode Register 3 (CSIM3)**
p.277	Modification of **Caution 1** in **Figure 15-3 Format of Serial Clock Select Register 1 (CSIC1)**
p.283 in 1st edition	Deletion of **15.4.2 (6) SCK1 pin** and **(7) SO1 pin** in 1st edition
p.291	Change of **Caution** in **Figure 16-3 Format of Asynchronous Serial Interface Mode Register 0 (ASIM0)**
p.293	Addition of baud rate calculation in **Remarks** in **Figure 16-5 Format of Baud Rate Generator Control Register 0 (BRGC0)**
p.305	Modification of description in **16.4.2 (2) (d) Reception**
p.305	Change of **Caution** in **Figure 16-9 Timing of Asynchronous Serial Interface Receive Completion Interrupt Request**
p.306	Modification of **Caution 2** in **Figure 16-10 Receive Error Timing**
p.310	Combination of **17.2 (1) IIC shift register 0 (IIC0)**, **(2) Slave address register 0 (SVA0)**, and **17.3 (5) IIC shift register 0 (IIC0)**, **(6) Slave address register 0 (SVA0)** in 1st edition
pp.312, 315, 318	Correction of address value in **Figure 17-3 Format of IIC Control Register 0 (IICC0)**, **Figure 17-4 Format of IIC Status Register 0 (IICS0)**, and **Figure 17-5 Format of IIC Transfer Clock Select Register 0 (IICCL0)**
p.325	Addition of description on **“Transfer lines”** in **Figure 17-14 Wait Signal**
p.337	Correction of **17.5.7 (3) (d) (ii) When WTIM0 = 1 (after restart, does not match with address (= not extension code))**
p.345	Addition of description in **Notes 1** and **2** in **Table 17-2 INTIIC0 Timing and Wait Control**
pp.355, 356	Correction of **Figure 17-21 Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (1) Start condition ~ address** and **(2) Data**
pp.358 to 360	Correction of **Figure 17-22 Example of Slave to Master Communication (When 9-Clock Wait Is Selected for Both Master and Slave)**
p.363	Correction of **Figure 18-1 LCD Controller/Driver Block Diagram**
p.367	Modification of **Note** in **Table 18-4 Frame Frequency**
p.368	Modification of description of GAIN bit in **Figure 18-6 Format of LCD Gain Adjust Register 0 (VLCG0)**

Major Revisions in This Edition (3/3)

Page	Description
p.369	Modification of **Figure 18-7 Format of Static/Dynamic Display Switching Register 3 (SDSEL3)**
p.370	Modification of **Figure 18-8 Format of Pin Function Switching Registers (PF8 to PF11)** and addition of **Caution 2**.
pp.371, 372	Replace **18.4 LCD Controller/Driver Settings** and **18.5 LCD Display RAM** of 1st edition
p.371 in 1st edition	Deletion of **Table 18-7 LCD Drive Voltages** of 1st edition
pp.374 to 376	Standardization of symbols • V_{LC0} pin output voltage: V_{LCD0} • V_{LC1} pin output voltage: V_{LCD1} • V_{LC2} pin output voltage: V_{LCD2}
p.376	Change of **Table 18-6 Output Voltages of V_{LC0} to V_{LC2} Pins**
p.377	Addition of description in **18.8.1 Static display example**
 p.378 p.381 p.384	Change of LCD panel connection examples • **Figure 18-14 Static LCD Panel Connection Example (SDSEL3n = 1: n = 0, 1)** • **Figure 18-17 3-Time-Division LCD Panel Connection Example (SDSEL3n = 0: n = 0 to 2)** • **Figure 18-20 4-Time-Division LCD Panel Connection Example (SDSEL3n = 0, n = 0 to 2)**
p.390	Correction of **Figure 19-1 Basic Configuration of Interrupt Function (E) Software interrupt**
p.395	Addition of **Caution** in **Figure 19-5 Format of External Interrupt Rising Edge Enable Register (EGP)**, **External Interrupt Falling Edge Enable Register (EGN)**
p.397	Addition of description and **Remark** in **19.4.1 Non-maskable interrupt request acknowledgment operation**
p.400	Addition of description in **19.4.2 Maskable interrupt request acknowledgment operation**
p.403	Addition of items in **Table 19-4 Interrupt Request Enabled for Nesting During Interrupt Servicing**
pp.411, 412	Addition of **Caution** and **Table 20-3 HALT Mode Release Condition and Necessity of NOP Instruction Setting When Subclock Multiplied by 4 Is Used (μPD78F0354, 78F0354Y Only)** in **20.2.1 (2) HALT mode release**
p.420	Addition of description on flash memory in **CHAPTER 22 ROM CORRECTION**
p.432	Correction of **Figure 23-1 Format of Memory Size Switching Register (IMS)**
p.435	Modification of **Table 23-3 Communication Mode List**
pp.436, 437	Change of pin names and signal names in **Figure 23-5 Example of Connection with Dedicated Flash Programmer** and **Table 23-4 Pin Connection List**
 p.441 p.442 p.443 p.444	Correction of flash writing adapter wiring examples • **Figure 23-10 Wiring Example for Flash Writing Adapter with 3-Wire Serial I/O (SIO3)** • **Figure 23-11 Wiring Example for Flash Writing Adapter with 3-Wire Serial I/O (SIO3) with Handshake** • **Figure 23-12 Wiring Example for Flash Writing Adapter with 3-Wire Serial I/O (CSI1)** • **Figure 23-13 Wiring Example for Flash Writing Adapter with UART (UART0)**
p.460	Revision of **CHAPTER 25 ELECTRICAL SPECIFICATIONS**
p.483	Addition of 113-pin plastic FBGA package in **CHAPTER 27 PACKAGE DRAWINGS**
p.484	Addition of **CHAPTER 28 RECOMMENDED SOLDERING CONDITIONS**
p.491	Addition of emulation probe NP-113F1-DA3 and conversion sockets LSPACK113A1110N01 and CSSOCKET113A1110N01 in **A.5 Debugging Tools (Hardware)**
p.495	Addition of **APPENDIX B NOTES ON TARGET SYSTEM DESIGN**
p.503	Addition of **APPENDIX D REVISION HISTORY**
U15798EJ2V0UD00 → U15798EJ2V1UD00	
Throughout	Addition of μPD78F0354A, 78F0354AY
p.32	Modification of **1.3 Ordering Information**
p.484	Modification of **CHAPTER 28 RECOMMENDED SOLDERING CONDITIONS**

The mark ★ shows major revised points.

INTRODUCTION

Readers

This manual is intended for user engineers who wish to understand the functions of the μPD780344, 780354, 780344Y, and 780354Y Subseries and to design and develop application systems and programs for these devices.

μPD780344 Subseries:	μPD780343, 780344
μPD780354 Subseries:	μPD780353, 780354, 78F0354, 78F0354A
μPD780344Y Subseries:	μPD780343Y, 780344Y
μPD780354Y Subseries:	μPD780353Y, 780354Y, 78F0354Y, 78F0354AY

Purpose

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

Organization

The μPD780344, 780354, 780344Y, and 780354Y Subseries manual is separated into two parts: this manual and the instructions edition (common to the 78K/0 Series).

μPD780344, 780354, 780344Y, 780354Y Subseries User's Manual (This Manual)	**78K/0 Series User's Manual Instructions**
• Pin functions • Internal block functions • Interrupt • 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 reader of this manual has general knowledge in the fields of electrical engineering, logic circuits, and microcontrollers.

- To gain a general understanding of functions:
  → Read this manual in the order of the contents.
- How to interpret the register format:
  → For a bit number enclosed in square, the bit name is defined as a reserved word in RA78K0, and is defined in the header file sfrbit.h in CC78K0.
- To check the details of a register when you know the register name:
  → Refer to **APPENDIX C REGISTER INDEX.**
- To understand the instruction functions of the 78K/0 Series:
  → Refer to the **78K/0 Series Instructions User's Manual (U12326E).**
- To understand the electrical specifications of the μPD780344, 780354, 780344Y, and 780354Y Subseries:
  → Refer to **CHAPTER 25 ELECTRICAL SPECIFICATIONS.**

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

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

Documents Related to Devices

Document Name	Document No.
★ μPD780344, 780354, 780344Y, 780354Y Subseries User's Manual	This document
78K/0 Series Instructions User's Manual	U12326E

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

Document Name		Document No.
RA78K0 Assembler Package	Operation	U14445E
	Language	U14446E
	Structured Assembly Language	U11789E
CC78K0 C Compiler	Operation	U14297E
	Language	U14298E
★ SM78K Series System Simulator Ver. 2.30 or Later	Operation (Windows™ Based)	U15373E
★	External Part User Open Interface Specifications	U15802E
★ ID78K Series Integrated Debugger Ver. 2.30 or Later	Operation (Windows Based)	U15185E
RX78K0 Real-time OS	Fundamentals	U11537E
	Installation	U11536E
Project Manager Ver. 3.12 or Later (Windows Based)		U14610E

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

Document Name	Document No.
IE-78K0-NS In-Circuit Emulator	U13731E
IE-78K0-NS-A In-Circuit Emulator	U14889E
IE-780354-NS-EM1 Emulation Board	To be prepared

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

Documents Related to Flash Memory Writing

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

Other Related Documents

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

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

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

CONTENTS

LIST OF FIGURES (1/8)

LIST OF FIGURES (2/8)

LIST OF FIGURES (3/8)

LIST OF FIGURES (4/8)

LIST OF FIGURES (5/8)

LIST OF FIGURES (6/8)

LIST OF FIGURES (7/8)

Figure No. Title Page

LIST OF FIGURES (8/8)

LIST OF TABLES (1/3)

LIST OF TABLES (2/3)

LIST OF TABLES (3/3)

CHAPTER 1 OUTLINE

1.1 Features

- Internal memory

Type	Program Memory (ROM/Flash Memory)	Data Memory		LCD Display RAM
Part Number		High-Speed RAM	Expansion RAM	
µPD780343, 780353, 780343Y, 780353Y	24 KB	512 bytes	512 bytes	40 × 8 bits
µPD780344, 780354, 780344Y, 780354Y	32 KB			
µPD78F0354, 78F0354A, 78F0354Y, 78F0354AY	32 KB[Note]	1,024 bytes		

Note The capacity of on-chip flash memory can be changed by means of the memory size switching register (IMS).

- Minimum instruction execution time changeable from high speed (0.2 µs: @10 MHz operation with main system clock) to ultra-low speed (122 µs: @32.768 kHz operation with subsystem clock)
- A circuit to multiply the subsystem clock by 4 is selectable (30.52 µs: @131 kHz operation = subsystem clock 32.768 kHz operation × 4)
- I/O port: 66 (N-ch open-drain: 6 (middle voltage: 4))
- 10-bit resolution A/D converter: 8 channels (µPD780353, 780354, 78F0354, 78F0354A, 780353Y, 780354Y, 78F0354Y, 78F0354AY only)
- 8-bit resolution A/D converter: 8 channels (µPD780343, 780344, 780343Y, 780344Y only)
- LCD controller/driver
  - Segment signal output: 40 max., common signal output: 4 max.
  - LCD reference voltage generator: booster type (×3 only)
  - Blinking display possible (blinking interval can be selected: 0.5 s or 1 s)
- Serial interface
  - 3-wire serial I/O mode (SIO3): 1 channel
  - 3-wire serial I/O mode (CSI1): 1 channel
  - UART: 1 channel
  - I^{2}C bus: 1 channel (µPD780344Y, 780354Y Subseries only)
- Timer
  - 16-bit timer/event counter: 1 channel
  - 8-bit timer/event counter: 3 channels
  - 8-bit timer: 1 channel
  - Watch timer: 1 channel
  - Watchdog timer: 1 channel
- ROM correction
- Vectored interrupt sources: 26
- Power supply voltage: V_{DD} = 1.8 to 5.5 V

1.2 Applications

APS cameras, digital cameras, AV systems, household appliances, etc.

★ 1.3 Ordering Information

Part Number	Package	Internal ROM
μPD780343GC-×××-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM
μPD780343GC-×××-8EU-A	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM
μPD780343F1-×××-DA3	113-pin plastic FBGA (10 × 10)	Mask ROM
μPD780344GC-×××-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM
μPD780344GC-×××-8EU-A	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM
μPD780344F1-×××-DA3	113-pin plastic FBGA (10 × 10)	Mask ROM
μPD780353GC-×××-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM
μPD780353GC-×××-8EU-A	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM
μPD780353F1-×××-DA3	113-pin plastic FBGA (10 × 10)	Mask ROM
μPD780354GC-×××-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM
μPD780354GC-×××-8EU-A	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM
μPD780353F1-×××-DA3	113-pin plastic FBGA (10 × 10)	Mask ROM
μPD78F0354GC-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Flash memory
μPD78F0354GC-8EU-A	100-pin plastic LQFP (fine pitch) (14 × 14)	Flash memory
μPD78F0354F1-DA3	113-pin plastic FBGA (10 × 10)	Flash memory
μPD78F0354AGC-8EU-A[Note]	100-pin plastic LQFP (fine pitch) (14 × 14)	Flash memory
μPD78F0354AF1-DA3[Note]	113-pin plastic FBGA (10 × 10)	Flash memory
μPD780343YGC-×××-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM
μPD780343YGC-×××-8EU-A	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM
μPD780343YF1-×××-DA3	113-pin plastic FBGA (10 × 10)	Mask ROM
μPD780344YGC-×××-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM
μPD780344YGC-×××-8EU-A	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM
μPD780344YF1-×××-DA3	113-pin plastic FBGA (10 × 10)	Mask ROM
μPD780353YGC-×××-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM
μPD780353YGC-×××-8EU-A	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM
μPD780353YF1-×××-DA3	113-pin plastic FBGA (10 × 10)	Mask ROM
μPD780354YGC-×××-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM
μPD780354YGC-×××-8EU-A	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM
μPD780353YF1-×××-DA3	113-pin plastic FBGA (10 × 10)	Mask ROM
μPD78F0354YGC-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Flash memory
μPD78F0354YGC-8EU-A	100-pin plastic LQFP (fine pitch) (14 × 14)	Flash memory
μPD78F0354YF1-DA3	113-pin plastic FBGA (10 × 10)	Flash memory
μPD78F0354AYGC-8EU-A[Note]	100-pin plastic LQFP (fine pitch) (14 × 14)	Flash memory
μPD78F0354AYF1-DA3[Note]	113-pin plastic FBGA (10 × 10)	Flash memory

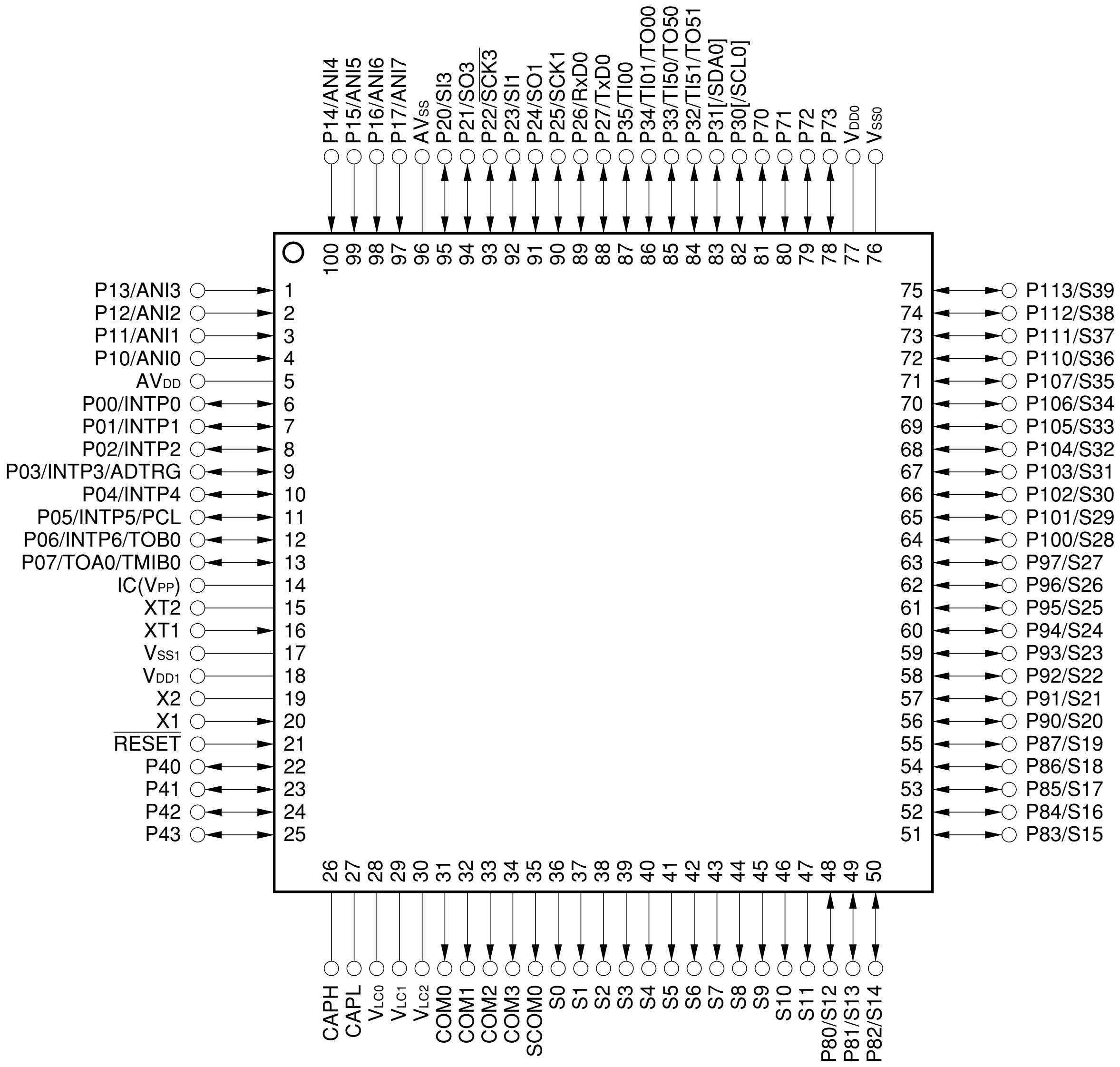
Note Under development.

Remarks 1. ××× indicates ROM code suffix.

2. Products that have the part numbers suffixed by “-A” are lead-free products.

1.4 Pin Configuration (Top View)

- **100-pin plastic LQFP (fine pitch) (14 × 14)**



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

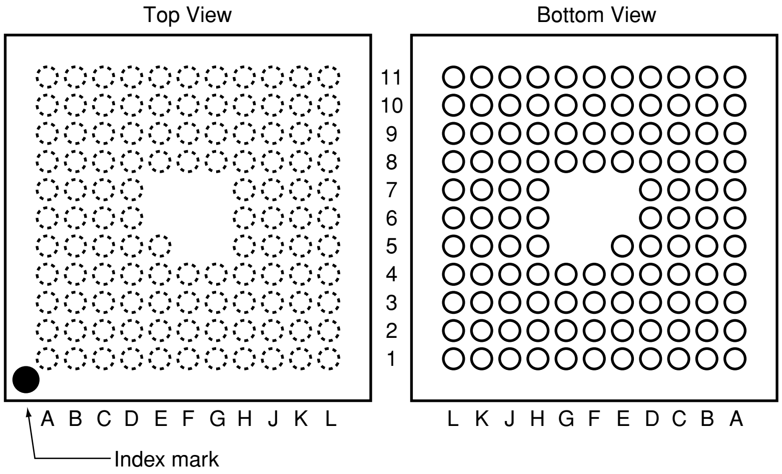
2. Connect the AV_{SS} pin to V_{SS0}.

Remarks 1. ( ): μPD78F0354, 78F0354A, 78F0354Y, 78F0354AY

2. [ ]: μPD780344Y, 780354Y Subseries

3. When the μPD780344, 780354, 780344Y, 780354Y Subseries is used in application fields that require reduction of the noise generated from inside the microcontroller, the implementation of noise reduction measures, such as supplying to V_{DD0} and V_{DD1} individually and connecting V_{SS0} and V_{SS1} to different ground lines, is recommended.

★ • **113-pin plastic FBGA (10 × 10)**



Pin No.	Name	Pin No.	Name	Pin No.	Name	Pin No.	Name	Pin No.	Name	Pin No.	Name
A1	NC	C1	NC	E1	P02/INTP2	G1	XT2	J1	NC	L1	NC
A2	P14/ANI4	C2	P11/ANI1	E2	P01/INTP1	G2	XT1	J2	P42	L2	CAPL
A3	NC	C3	P16/ANI6	E3	P04/INTP4	G3	$\overline{\text{RESET}}$	J3	P41	L3	NC
A4	P21/SO3	C4	P17/ANI7	E4	P07/TOA0/TMIB0	G4	V_{DD1}	J4	COM3	L4	COM0
A5	P25/SCK1	C5	AV_{SS}	E5	NC	G5	–	J5	SCOM0	L5	COM2
A6	P35/TI00	C6	P23/SI1	E6	–	G6	–	J6	S6	L6	S1
A7	P31 [/SDA0]	C7	P33/TI50/TO50	E7	–	G7	–	J7	S10	L7	S4
A8	P70	C8	P32/TI51/TO51	E8	P104/S32	G8	P97/S27	J8	S11	L8	S8
A9	NC	C9	P111/S37	E9	P107/S35	G9	P94/S24	J9	P80/S12	L9	NC
A10	V_{DD0}	C10	P112/S38	E10	P102/S30	G10	P91/S21	J10	P85/S17	L10	P82/S14
A11	NC	C11	NC	E11	P101/S29	G11	P92/S22	J11	NC	L11	NC
B1	P12/ANI2	D1	P00/INTP0	F1	P06/INTP6/TOB0	H1	X2	K1	P43		
B2	P13/ANI3	D2	P10/ANI0	F2	P05/INTP5/PCL	H2	X1	K2	CAPH		
B3	P15/ANI5	D3	P03/INTP3/ADTRG	F3	V_{SS1}	H3	P40	K3	V_{LC0}		
B4	P20/SI3	D4	AV_{DD}	F4	IC (V_{PP})	H4	V_{LC2}	K4	V_{LC1}		
B5	P24/SO1	D5	P22/SCK3	F5	–	H5	S2	K5	COM1		
B6	P34/TI01/TO00	D6	P26/RxD0	F6	–	H6	S3	K6	S0		
B7	P30 [/SCL0]	D7	P27/TxD0	F7	–	H7	S7	K7	S5		
B8	P72	D8	P71	F8	P100/S28	H8	P87/S19	K8	S9		
B9	P73	D9	P110/S36	F9	P103/S31	H9	P93/S23	K9	P81/S13		
B10	V_{SS0}	D10	P106/S34	F10	P95/S25	H10	P86/S18	K10	P83/S15		
B11	P113/S39	D11	P105/S33	F11	P96/S26	H11	P90/S20	K11	P84/S16		

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

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

Remarks 1. ( ): μPD78F0354, 78F0354A, 78F0354Y, 78F0354AY

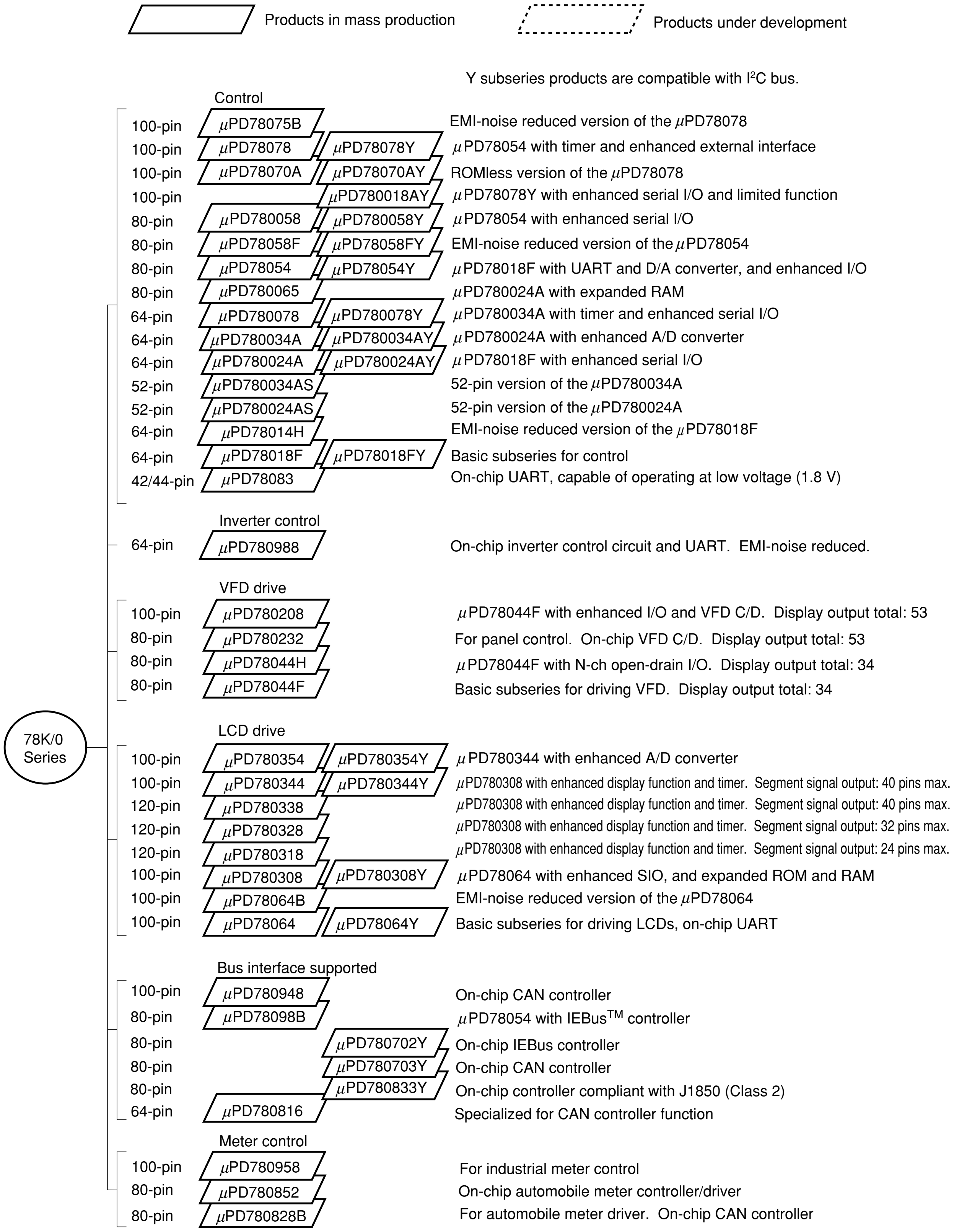
2. [ ]: μPD780344Y, 780354Y Subseries

3. When the μPD780344, 780354, 780344Y, 780354Y Subseries is used in application fields that require reduction of the noise generated from inside the microcontroller, the implementation of noise reduction measures, such as supplying to V_{DD0} and V_{DD1} individually and connecting V_{SS0} and V_{SS1} to different ground lines, is recommended.

ADTRG:	AD trigger input
ANI0 to ANI7:	Analog input
★ AV_{DD}:	Analog power supply
AV_{SS}:	Analog ground
CAPH, CAPL:	Capacitor for LCD
COM0 to COM3:	Common output for dynamic display
IC:	Internally connected
INTP0 to INTP6:	External interrupt input
P00 to P07:	Port 0
P10 to P17:	Port 1
P20 to P27:	Port 2
P30 to P35:	Port 3
P40 to P43:	Port 4
P70 to P73:	Port 7
P80 to P87:	Port 8
P90 to P97:	Port 9
★ P100 to P107:	Port 10
P110 to P113:	Port 11
PCL:	Programmable clock
$\overline{\text{RESET}}$:	Reset
R_XD0:	Receive data
S0 to S39:	Segment output
SCK1, $\overline{\text{SCK3}}$, SCL0:	Serial clock
SCOM0:	Common output for static display
SDA0:	Serial data
SI1, SI3:	Serial input
SO1, SO3:	Serial output
TI00, TI01, TMIB0, TI50, TI51:	Timer input
TO0, TOA0, TOB0, TO50, TO51:	Timer output
T_XD0:	Transmit data
V_{DD0}, V_{DD1}:	Power supply
V_{LC0} to V_{LC2}:	LCD power supply
V_{PP}:	Programming power supply
V_{SS0}, V_{SS1}:	Ground
X1, X2:	Crystal (main system clock)
XT1, XT2:	Crystal (subsystem clock)

★ 1.5 78K/0 Series Lineup

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



▱ Products in mass production

⏥ Products under development

Y subseries products are compatible with I²C bus.

78K/0 Series

	Control		
100-pin	μPD78075B		EMI-noise reduced version of the μPD78078
100-pin	μPD78078	μPD78078Y	μPD78054 with timer and enhanced external interface
100-pin	μPD78070A	μPD78070AY	ROMless version of the μPD78078
100-pin		μPD780018AY	μPD78078Y with enhanced serial I/O and limited function
80-pin	μPD780058	μPD780058Y	μPD78054 with enhanced serial I/O
80-pin	μPD78058F	μPD78058FY	EMI-noise reduced version of the μPD78054
80-pin	μPD78054	μPD78054Y	μPD78018F with UART and D/A converter, and enhanced I/O
80-pin	μPD780065		μPD780024A with expanded RAM
64-pin	μPD780078	μPD780078Y	μPD780034A with timer and enhanced serial I/O
64-pin	μPD780034A	μPD780034AY	μPD780024A with enhanced A/D converter
64-pin	μPD780024A	μPD780024AY	μPD78018F with enhanced serial I/O
52-pin	μPD780034AS		52-pin version of the μPD780034A
52-pin	μPD780024AS		52-pin version of the μPD780024A
64-pin	μPD78014H		EMI-noise reduced version of the μPD78018F
64-pin	μPD78018F	μPD78018FY	Basic subseries for control
42/44-pin	μPD78083		On-chip UART, capable of operating at low voltage (1.8 V)
	Inverter control		
64-pin	μPD780988		On-chip inverter control circuit and UART. EMI-noise reduced.
	VFD drive		
100-pin	μPD780208		μPD78044F with enhanced I/O and VFD C/D. Display output total: 53
80-pin	μPD780232		For panel control. On-chip VFD C/D. Display output total: 53
80-pin	μPD78044H		μPD78044F with N-ch open-drain I/O. Display output total: 34
80-pin	μPD78044F		Basic subseries for driving VFD. Display output total: 34
	LCD drive		
100-pin	μPD780354	μPD780354Y	μPD780344 with enhanced A/D converter
100-pin	μPD780344	μPD780344Y	μPD780308 with enhanced display function and timer. Segment signal output: 40 pins max.
120-pin	μPD780338		μPD780308 with enhanced display function and timer. Segment signal output: 40 pins max.
120-pin	μPD780328		μPD780308 with enhanced display function and timer. Segment signal output: 32 pins max.
120-pin	μPD780318		μPD780308 with enhanced display function and timer. Segment signal output: 24 pins max.
100-pin	μPD780308	μPD780308Y	μPD78064 with enhanced SIO, and expanded ROM and RAM
100-pin	μPD78064B		EMI-noise reduced version of the μPD78064
100-pin	μPD78064	μPD78064Y	Basic subseries for driving LCDs, on-chip UART
	Bus interface supported		
100-pin	μPD780948		On-chip CAN controller
80-pin	μPD78098B		μPD78054 with IEBus™ controller
80-pin		μPD780702Y	On-chip IEBus controller
80-pin		μPD780703Y	On-chip CAN controller
80-pin		μPD780833Y	On-chip controller compliant with J1850 (Class 2)
64-pin	μPD780816		Specialized for CAN controller function
	Meter control		
100-pin	μPD780958		For industrial meter control
80-pin	μPD780852		On-chip automobile meter controller/driver
80-pin	μPD780828B		For automobile meter driver. On-chip CAN controller

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

The major functional differences among the subseries are listed below.

- Non-Y subseries

Function		ROM Capacity	Timer				8-Bit	10-Bit	8-Bit	Serial Interface	I/O	V_{DD} MIN.	External
Subseries Name		(Bytes)	8-Bit	16-Bit	Watch	WDT	A/D	A/D	D/A			Value	Expansion
Control	μPD78075B	32 K to 40 K	4 ch	1 ch	1 ch	1 ch	8 ch	–	2 ch	3 ch (UART: 1 ch)	88	1.8 V	√
	μPD78078	48 K to 60 K											
	μPD78070A	–									61	2.7 V	
	μPD780058	24 K to 60 K	2 ch							3 ch (time-division UART: 1 ch)	68	1.8 V	
	μPD78058F	48 K to 60 K								3 ch (UART: 1 ch)	69	2.7 V	
	μPD78054	16 K to 60 K										2.0 V	
	μPD780065	40 K to 48 K							–	4 ch (UART: 1 ch)	60	2.7 V	
	μPD780078	48 K to 60 K		2 ch			–	8 ch		3 ch (UART: 2 ch)	52	1.8 V	
	μPD780034A	8 K to 32 K		1 ch						3 ch (UART: 1 ch)	51		
	μPD780024A						8 ch	–					
	μPD780034AS						–	4 ch			39		–
	μPD780024AS						4 ch	–					
	μPD78014H						8 ch			2 ch	53		√
	μPD78018F	8 K to 60 K											
	μPD78083	8 K to 16 K		–	–					1 ch (UART: 1 ch)	33		–
Inverter control	μPD780988	16 K to 60 K	3 ch	**Note**	–	1 ch	–	8 ch	–	3 ch (UART: 2 ch)	47	4.0 V	√
VFD drive	μPD780208	32 K to 60 K	2 ch	1 ch	1 ch	1 ch	8 ch	–	–	2 ch	74	2.7 V	–
	μPD780232	16 K to 24 K	3 ch	–	–		4 ch				40	4.5 V	
	μPD78044H	32 K to 48 K	2 ch	1 ch	1 ch		8 ch			1 ch	68	2.7 V	
	μPD78044F	16 K to 40 K								2 ch			
LCD drive	μPD780354	24 K to 32 K	4 ch	1 ch	1 ch	1 ch	–	8 ch	–	3 ch (UART: 1 ch)	66	1.8 V	–
	μPD780344						8 ch	–					
	μPD780338	48 K to 60 K	3 ch	2 ch			–	10 ch	1 ch	2 ch (UART: 1 ch)	54		
	μPD780328										62		
	μPD780318										70		
	μPD780308	48 K to 60 K	2 ch	1 ch			8 ch	–	–	3 ch (time-division UART: 1 ch)	57	2.0 V	
	μPD78064B	32 K								2 ch (UART: 1 ch)			
	μPD78064	16 K to 32 K											
Bus interface supported	μPD780948	60 K	2 ch	2 ch	1 ch	1 ch	8 ch	–	–	3 ch (UART: 1 ch)	79	4.0 V	√
	μPD78098B	40 K to 60 K		1 ch					2 ch		69	2.7 V	–
	μPD780816	32 K to 60 K		2 ch			12 ch		–	2 ch (UART: 1 ch)	46	4.0 V	
Meter control	μPD780958	48 K to 60 K	4 ch	2 ch	–	1 ch	–	–	–	2 ch (UART: 1 ch)	69	2.2 V	–
Dash-board control	μPD780852	32 K to 40 K	3 ch	1 ch	1 ch	1 ch	5 ch	–	–	3 ch (UART: 1 ch)	56	4.0 V	–
	μPD780828B	32 K to 60 K									59		

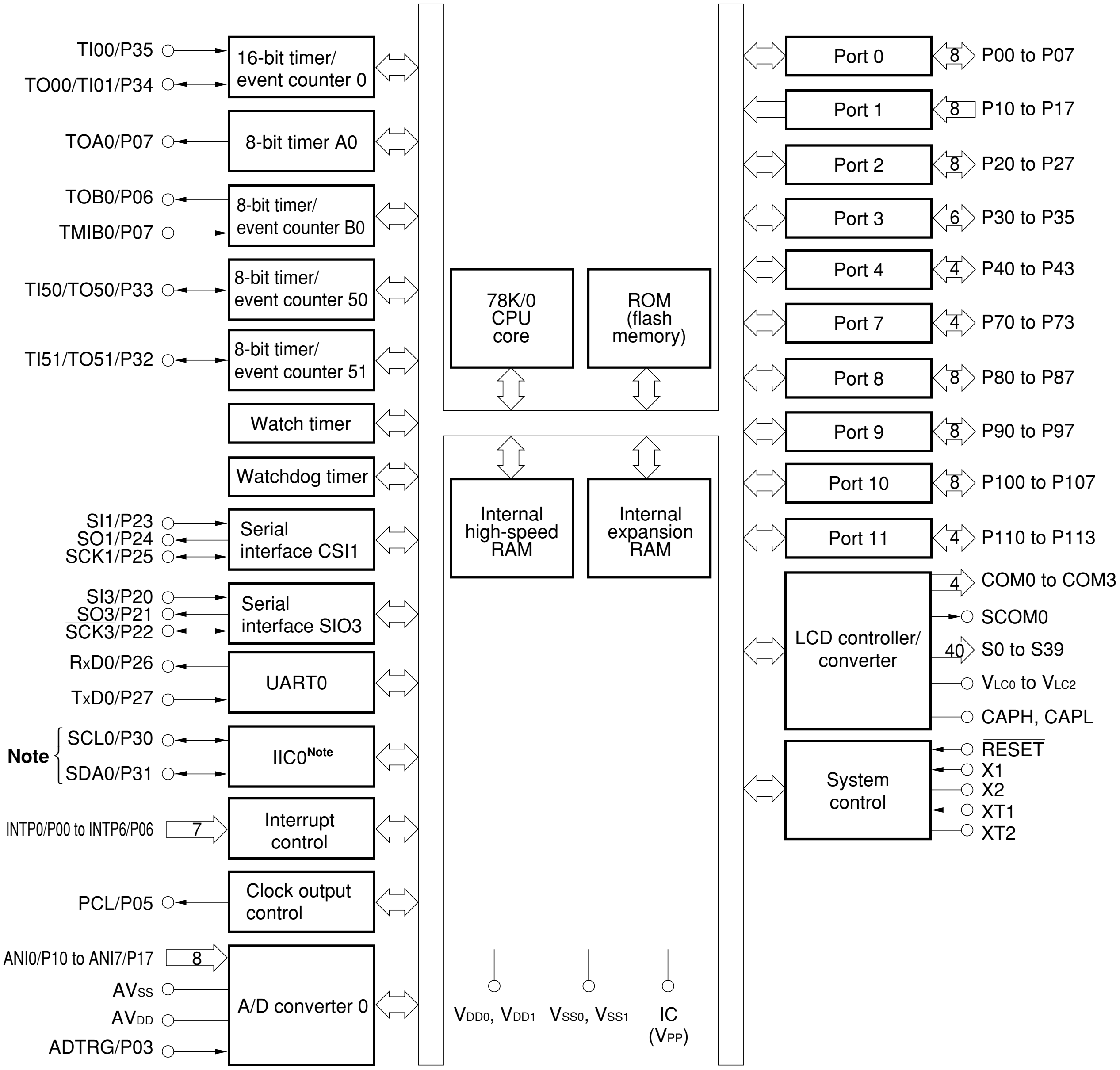
Note 16-bit timer: 2 channels
10-bit timer: 1 channel

- Y subseries

Subseries Name / Function		ROM Capacity (Bytes)	Timer				8-Bit A/D	10-Bit A/D	8-Bit D/A	Serial Interface	I/O	V$_{DD}$ MIN. Value	External Expansion
			8-Bit	16-Bit	Watch	WDT							
Control	μPD78078Y	48 K to 60 K	4 ch	1 ch	1 ch	1 ch	8 ch	–	2 ch	3 ch (UART: 1 ch, I^{2}C: 1 ch)	88	1.8 V	√
	μPD78070AY	–									61	2.7 V	
	μPD780018AY	48 K to 60 K							–	3 ch (I^{2}C: 1 ch)	88		
	μPD780058Y	24 K to 60 K	2 ch						2 ch	3 ch (time-division UART: 1 ch, I^{2}C: 1 ch)	68	1.8 V	
	μPD78058FY	48 K to 60 K								3 ch (UART: 1 ch, I^{2}C: 1 ch)	69	2.7 V	
	μPD78054Y	16 K to 60 K										2.0 V	
	μPD780078Y	48 K to 60 K		2 ch			–	8 ch	–	4 ch (UART: 2 ch, I^{2}C: 1 ch)	52	1.8 V	
	μPD780034AY	8 K to 32 K		1 ch						3 ch (UART: 1 ch, I^{2}C: 1 ch)	51		
	μPD780024AY						8 ch	–					
	μPD78018FY	8 K to 60 K								2 ch (I^{2}C: 1 ch)	53		
LCD drive	μPD780354Y	24 K to 32 K	4 ch	1 ch	1 ch	1 ch	–	8 ch	–	4 ch (UART: 1 ch, I^{2}C: 1 ch)	66	1.8 V	–
	μPD780344Y						8 ch	–					
	μPD780308Y	48 K to 60 K	2 ch							3 ch (time-division UART: 1 ch, I^{2}C: 1 ch)	57	2.0 V	
	μPD78064Y	16 K to 32 K								2 ch (UART: 1 ch, I^{2}C: 1 ch)			
Bus interface supported	μPD780701Y	60 K	3 ch	2 ch	1 ch	1 ch	16 ch	–	–	4 ch (UART: 1 ch, I^{2}C: 1 ch)	67	3.5 V	–
	μPD780703Y												
	μPD780833Y										65	4.5 V	

Remark Functions other than the serial interface are common to both the Y and non-Y subseries.

1.6 Block Diagram



★

Note μPD780344Y, 780354Y Subseries only

Remark The internal ROM and RAM capacities depend on the product.

1.7 Outline of Function

(1/2)

Item \ Part Number		μPD780343 μPD780343Y	μPD780344 μPD780344Y	μPD780353 μPD780353Y	μPD780354 μPD780354Y	μPD78F0354 μPD78F0354A μPD78F0354Y μPD78F0354AY
Internal memory	ROM	24 KB (mask ROM)	32 KB (mask ROM)	24 KB (mask ROM)	32 KB (mask ROM)	32 KB[Note 1] (flash memory)
	High-speed RAM	512 bytes				1,024 bytes
	Expansion RAM	512 bytes				
	LCD display RAM	40 × 8 bits				
Memory space		64 KB				
General-purpose registers		8 bits × 32 registers (8 bits × 8 registers × 4 banks)				
Minimum instruction execution time		Function to change minimum instruction execution time provided				
	When main system clock selected	0.2 μs/0.4 μs/0.8 μs/1.6 μs/3.2 μs (@V_{DD} = 5 V, f_X = 10 MHz)				
	When subsystem clock selected	122 μs (@f_{XT} = 32.768 kHz)				
		30.52 μs (@f_{XT} = 32.768 kHz (131 kHz) with ×4 multiplication clock)				
Subsystem clock multiplication function		×4 multiplication circuit (operating supply voltage: V_{DD} = 2.7 to 5.5 V)[Note 2]				
Instruction set		• 16-bit operation • Multiply/divide (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulate (set, reset, test, and Boolean operation) • BCD adjust, etc.				
I/O port		66				
	CMOS input	8				
	CMOS I/O	52[Note 3]				
	N-ch open-drain I/O	6 (Middle withstand voltage: 4)				
★ A/D converter		• 8-bit resolution × 8 channels • Voltage operation: AV_{DD} = 2.2 to 5.5 V		• 10-bit resolution × 8 channels • Voltage operation: AV_{DD} = 2.2 to 5.5 V		
LCD controller/driver		• LCD reference voltage generator: booster type (×3 only) • Blinking display possible (blinking interval can be selected: 0.5 s or 1 s) • Static display and dynamic display (1/3 bias only) can be used simultaneously (Static display up to 12 segments)				
	Segment signal outputs	40 max.[Note 3]				
	Common signal outputs	4 max. (dynamic display), 1 (static display)				
Serial interface		• 3-wire serial I/O mode (SIO3): 1 channel • 3-wire serial I/O mode (CSI1): 1 channel • UART: 1 channel • I^{2}C bus: 1 channel (μPD780344Y, 780354Y Subseries only)				

Notes 1. The capacity of the on-chip flash memory can be changed by means of the memory size switching register (IMS).

2. Whether a circuit to multiply the clock by 4 is used or not is selected by the subclock select register (SSCK).

3. 28 pins are used either as a port function or LCD segment output selected by the pin function switching register.

(2/2)

Item / Part Number		µPD780343 µPD780343Y	µPD780344 µPD780344Y	µPD780353 µPD780353Y	µPD780354 µPD780354Y	µPD78F0354 µPD78F0354A µPD78F0354Y µPD78F0354AY
Timer		• 16-bit timer/event counter: 1 channel • 8-bit timer/event counter: 3 channels • 8-bit timer: 1 channel • Watch timer: 1 channel • Watchdog timer: 1 channel				
Timer output		5 (8-bit PWM output possible: 3)				
Clock output		• 78.1 kHz, 156 kHz, 313 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5 MHz, 10 MHz (@10 MHz operation with main system clock) • 32.768 kHz (@32.768 kHz operation with subsystem clock)				
Vectored interrupt sources	Maskable	Internal: 16, external: 8				
	Non-maskable	Internal: 1				
	Software	1				
ROM correction		Provided				
Power supply voltage		V_{DD} = 1.8 to 5.5 V				
Operating ambient temperature		T_A = –40 to +85°C				
Package		• 100-pin plastic LQFP (fine pitch) (14 × 14) • 113-pin plastic FBGA (10 × 10)				

The following table outlines the timer/event counters (for details, refer to **CHAPTER 6 16-BIT TIMER/EVENT COUNTER 0**, **CHAPTER 7 8-BIT TIMERS A0, B0, CHAPTER 8 8-BIT TIMER/EVENT COUNTERS 50, 51**, **CHAPTER 9 WATCH TIMER**, and **CHAPTER 10 WATCHDOG TIMER**):

		16-Bit Timer/ Event Counter 0	8-Bit Timer A0, B0		8-Bit Timer/ Event Counters 50, 51	Watch Timer	Watchdog Timer
			A0	B0			
Operation mode	Interval timer	1 channel	2 channels		2 channels	1 channel[Note 1]	1 channel[Note 2]
	External event counter	○	—	○	○	—	—
Function	Timer output	○	○		○	—	—
	PPG output	○	—		—	—	—
	PWM output	—	—	○	○	—	—
	Pulse width measurement	○	—		—	—	—
	Square wave output	○	○		○	—	—
	Interrupt request	○	○		○	○	○

Notes 1. The watch timer can be used both as a watch timer and an interval timer at the same time.

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

1.8 Mask Options

The mask ROM versions (μPD780343, 780344, 780353, 780354, 780343Y, 780344Y, 780353Y, and 780354Y) provide pull-up resistor mask options which allow users to specify whether to connect a pull-up resistor to a specific port pin when the user places an order for device production. Using the mask option when pull-up resistors are required reduces the number of components to add to the device, resulting in board space saving.

The mask options provided in the μPD780344, 780354 Subseries are shown in Table 1-1 and the mask options provided in the μPD780344Y, 780354Y Subseries are shown in Table 1-2.

Table 1-1. Mask Options of Mask ROM Versions of μPD780344, 780354 Subseries

Pin Names	Mask Option
P30, P31, P70 to P73	Pull-up resistor connection can be specified in 1-bit units.

Table 1-2. Mask Options of Mask ROM Versions of μPD780344Y, 780354Y Subseries

Pin Names	Mask Option
P70 to P73	Pull-up resistor connection can be specified in 1-bit units.

CHAPTER 2 PIN FUNCTIONS

2.1 Pin Function List

(1) Port pins (1/2)

Pin Name	I/O	Function		After Reset	Alternate Function
P00	I/O	Port 0 8-bit I/O port Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be used by setting software.		Input	INTP0
P01					INTP1
P02					INTP2
P03					INTP3/ADTRG
P04					INTP4
P05					INTP5/PCL
P06					INTP6/TOB0
P07					TOA0/TMIB0
P10 to P17	Input	Port 1 8-bit input only port.		Input	ANI0 to ANI7
P20	I/O	Port 2 8-bit I/O port Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be used by setting software.		Input	SI3
P21					SO3
P22					$\overline{\text{SCK3}}$
P23					SI1
P24					SO1
P25					SCK1
P26					RxD0
P27					TxD0
P30	I/O	Port 3 6-bit I/O port Input/output mode can be specified in 1-bit units.	5 V N-ch open-drain I/O port. LEDs can be driven directly. On-chip pull-up resistor can be specified by mask option (mask ROM version of the μPD780344, 780354 Subseries only)Note 1.	Input	SCL0Note 2
P31					SDA0Note 2
P32			An on-chip pull-up resistor can be used by setting software.		TI51/TO51
P33					TI50/TO50
P34					TI01/TO00
P35					TI00
P40 to P43	I/O	Port 4 4-bit I/O port Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be used by setting software. Interrupt request flag (KRIF) is set to 1 by falling edge detection.		Input	—

Notes 1. μPD780344Y, 780354Y Subseries and flash memory version cannot use an on-chip pull-up resistor.

2. μPD780344Y, 780354Y Subseries only.

(1) Port pins (2/2)

Pin Name	I/O	Function	After Reset	Alternate Function
P70 to P73	I/O	Port 7 4-bit middle-voltage N-ch open-drain I/O port Input/output mode can be specified in 1-bit units. LEDs can be driven directly. On-chip pull-up resistor can be specified by mask option (mask ROM version only).	Input	—
P80 to P87[Note]	I/O	Port 8 8-bit I/O port	Input	S12 to S19[Note]
P90 to P97[Note]	I/O	Port 9 8-bit I/O port	Input	S20 to S27[Note]
P100 to P107[Note]	I/O	Port 10 8-bit I/O port	Input	S28 to S35[Note]
P110 to P113[Note]	I/O	Port 11 4-bit I/O port	Input	S36 to S39[Note]

Note These pins can be used as a port function or LCD segment output by setting the pin function switching register.

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

Pin Name	I/O	Function	After Reset	Alternate Function
INTP0	Input	External interrupt request input with specifiable valid edges (rising edge, falling edge, both rising and falling edges)	Input	P00
INTP1				P01
INTP2				P02
INTP3				P03/ADTRG
INTP4				P04
INTP5				P05/PCL
INTP6				P06/TOB0
SI1	Input	Serial interface serial data input	Input	P23
SI3				P20
SO1	Output	Serial interface serial data output	Input	P24
SO3				P21
SCK1	I/O	Serial interface serial clock input/output	Input	P25
$\overline{\text{SCK3}}$				P22
RxD0	Input	Asynchronous serial interface serial data input	Input	P26
TxD0	Output	Asynchronous serial interface serial data output	Input	P27
SCL0	I/O	Serial interface serial clock input/output (μPD780344Y, 780354Y Subseries only)	Input	P30
SDA0		Serial interface serial data input/output (μPD780344Y, 780354Y Subseries only)	Input	P31
TI00	Input	External count clock input to 16-bit timer/event counter 0 Capture trigger input to capture registers (CR00, CR01) of 16-bit timer/event counter 0	Input	P35
TI01		Capture trigger input to capture register (CR00) of 16-bit timer/event counter 0		P34/TO00
TO00	Output	16-bit timer/event counter 0 output		P34/TI01
TMIB0	Input	External count clock input to 8-bit timer/event counter B0	Input	P07/TOA0
TOA0	Output	8-bit timer A0 output	Input	P07/TMIB0
TOB0		8-bit timer/event counter B0 output		P06/INTP6
TI50	Input	External count clock input to 8-bit timer/event counter 50	Input	P33/TO50
TI51		External count clock input to 8-bit timer/event counter 51		P34/TO51
TO50	Output	8-bit timer/event counter 50 output	Input	P33/TI50
TO51		8-bit timer/event counter 51 output		P34/TI51
PCL	Output	Clock output (for main system clock, subsystem clock trimming)	Input	P05/INTP5
ANI0 to ANI7	Input	Analog input of A/D converter	Input	P10 to P17
ADTRG	Input	Trigger signal input of A/D converter	Input	P03/INTP3
★ AV$_{DD}$	Input	A/D converter analog power supply. Supply the same potential as that of V$_{DD0}$ or V$_{DD1}$.	—	—
AV$_{SS}$	—	Ground potential for A/D converter. Supply the same potential as that of V$_{SS0}$ or V$_{SS1}$.	—	—

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

Pin Name	I/O	Function	After Reset	Alternate Function
S0 to S11	Output	LCD controller/driver segment signal output (Static and dynamic display can be selected)	Output	—
S12 to S19		LCD controller/driver segment signal output (for dynamic display)		P80 to P87
S20 to S27				P90 to P97
S28 to S35				P100 to P107
S36 to S39				P110 to P113
COM0 to COM3	Output	LCD controller/driver common signal output (for dynamic display)	Output	—
SCOM0	Output	LCD controller/driver common signal output (for static display)	Output	—
V_{LC0} to V_{LC2}	—	LCD driving voltage • V_{LC0}: Three times V_{LC2} output voltage • V_{LC1}: Two times V_{LC2} output voltage • V_{LC2}: Reference voltage	—	—
CAPH, CAPL	—	Booster capacitor connection for LCD drive voltage	—	—
$\overline{\text{RESET}}$	Input	System reset input	—	—
X1	Input	Crystal connection for main system clock oscillation	—	—
X2	—		—	—
XT1	Input	Crystal connection for subsystem clock oscillation	—	—
XT2	—		—	—
V_{DD0}	—	Positive power supply for ports	—	—
V_{DD1}	—	Positive power supply other than ports	—	—
V_{SS0}	—	Ground potential for ports	—	—
V_{SS1}	—	Ground potential other than ports	—	—
IC	—	Internally connected. Connect directly to V_{SS0} or V_{SS1}.	—	—
V_{PP}	—	High-voltage application for program write/verify	—	—

2.2 Description of Pin Functions

2.2.1 P00 to P07 (Port 0)

These are 8-bit I/O ports. Besides serving as I/O ports, they function as an external interrupt request input, A/D converter external trigger input, clock output, and timer I/O function.

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

(1) Port mode

These ports function as 8-bit I/O ports. They can be specified as input or output ports in 1-bit units with port mode register 0 (PM0). On-chip pull-up resistors can be used by setting pull-up resistor option register 0 (PU0).

(2) Control mode

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

(a) INTP0 to INTP6

INTP0 to INTP6 are external interrupt request input pins for which valid edges (rising edge, falling edge, and both rising and falling edges) can be specified.

(b) ADTRG

A/D converter external trigger input pin.

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

(c) PCL

Clock output pin.

(d) TOA0, TOB0

Timer output pins of 8-bit timers A0 and B0.

(e) TMIB0

External count clock input pin to 8-bit timer/event counter B0.

2.2.2 P10 to P17 (Port 1)

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

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

(1) Port mode

These ports function as 8-bit input only ports.

(2) Control mode

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

2.2.3 P20 to P27 (Port 2)

These are 8-bit I/O ports. Besides serving as I/O ports, they function as serial interface data I/O and clock I/O. The following operation modes can be specified in 1-bit units.

(1) Port mode

These ports function as 8-bit I/O ports. They can be specified as input or output ports in 1-bit units with port mode register 2 (PM2). On-chip pull-up resistors can be used by setting pull-up resistor option register 2 (PU2).

(2) Control mode

These ports function as serial interface data I/O and clock I/O functions.

(a) SI1, SI3, SO1, and SO3

Serial interface serial data I/O pins.

(b) SCK1 and $\overline{\text{SCK3}}$

Serial interface serial clock I/O pins.

(c) RxD0 and TxD0

Asynchronous serial interface serial data I/O pins.

2.2.4 P30 to P35 (Port 3)

These are 6-bit I/O ports. Besides serving as I/O ports, they function as serial interface data I/O, clock I/O, and timer I/O.

(1) Port mode

These ports function as 6-bit I/O ports. They can be specified as input or output ports in 1-bit units with port mode register 3 (PM3).

P30 and P31 are 5 V N-ch open-drain. They can drive LEDs directly. μPD780343, 780344, 780353, and 780354 can use on-chip pull-up resistors by mask option.

P32 to P35 can use on-chip pull-up resistors by setting pull-up resistor option register 3 (PU3).

(2) Control mode

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

(a) SDA0

Serial interface serial data I/O pin.

(b) SCL0

Serial interface serial clock I/O pin.

(c) TI00

External count clock input pin to 16-bit timer/event counter 0 and capture trigger signal input pin to capture registers (CR00 and CR01) of the 16-bit timer/event counter 0.

(d) TI01

Capture trigger signal input pin to capture register (CR00) of the 16-bit timer/event counter 0.

(e) TI50 and TI51

External count clock input pins to 8-bit timer/event counters 50 and 51.

(f) TO00, TO50, and TO51

Timer output pins.

2.2.5 P40 to P43 (Port 4)

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

Interrupt request flag (KRIF) can be set to 1 by detecting the falling edge. The number of ports to detect the falling edge is four.

Caution Be sure to set memory expansion mode register (MEM) to 01H when using falling edge detection interrupt (INTKR).

2.2.6 P70 to P73 (Port 7)

These are 4-bit I/O ports. They can be specified as input or output ports in 1-bit units with port mode register 7 (PM7). Port 7 can drive LEDs directly.

P70 to P73 are middle-voltage N-ch open-drain. On-chip pull-up resistors can be used by mask option with the mask ROM versions.

2.2.7 P80 to P87 (Port 8)

These are 8-bit I/O ports. Besides serving as I/O ports, they function as segment signal output (for dynamic display) of the LCD controller/driver. Either the I/O port or segment signal output function can be selected by setting the pin function switching register 8 (PF8).

(1) Port mode

These ports function as 8-bit I/O ports. They can be specified as input or output ports in 1-bit units with port mode register 8 (PM8).

(2) Control mode

These ports function as segment signal output pins (S12 to S19) (for dynamic display) of the LCD controller/driver.

2.2.8 P90 to P97 (Port 9)

These are 8-bit I/O ports. Besides serving as I/O ports, they function as segment signal output (for dynamic display) of the LCD controller/driver. Either the I/O port or segment signal output function can be selected by setting the pin function switching register 9 (PF9).

(1) Port mode

These ports function as 8-bit I/O ports. They can be specified as input or output ports in 1-bit units with port mode register 9 (PM9).

(2) Control mode

These ports function as segment signal output pins (S20 to S27) (for dynamic display) of the LCD controller/driver.

2.2.9 P100 to P107 (Port 10)

These are 8-bit I/O ports. Besides serving as I/O ports, they function as segment signal output (for dynamic display) of the LCD controller/driver. Either the I/O port or segment signal output function can be selected by setting the pin function switching register 10 (PF10).

(1) Port mode

These ports function as 8-bit I/O ports. They can be specified as input or output ports in 1-bit units with port mode register 10 (PM10).

(2) Control mode

These ports function as segment signal output pins (S28 to S35) (for dynamic display) of the LCD controller/driver.

2.2.10 P110 to P113 (Port 11)

These are 4-bit I/O ports. Besides serving as I/O ports, they function as segment signal output (for dynamic display) of the LCD controller/driver. Either the I/O port or segment signal output function can be selected by setting the pin function switching register 11 (PF11).

(1) Port mode

These ports function as 4-bit I/O ports. They can be specified as input or output ports in 1-bit units with port mode register 11 (PM11).

(2) Control mode

These ports function as segment signal output pins (S36 to S39) (for dynamic display) of the LCD controller/driver.

★ 2.2.11 AV_{DD}

This is the A/D converter analog power supply pin. Even when the A/D converter is not used, supply the same potential as that of the V_{DD0} or V_{DD1} pin.

2.2.12 AV_{SS}

This is the ground potential pin of A/D converter. Always use the same potential as that of the V_{SS0} or V_{SS1} pin even when an A/D converter is not used.

2.2.13 S0 to S39[Note]

These are the segment signal output pins of the LCD controller/driver.

S0 to S11: Static or dynamic display can be switched
S12 to S39: For dynamic display

Note S12 to S19 and P80 to P87, S20 to S27 and P90 to P97, S28 to S35 and P100 to P107, and S36 to S39 and P110 to P113 function alternately. These functions can be switched to a port in 1-bit units by using the pin function switching register.

2.2.14 COM0 to COM3

These are the common signal output pins (for dynamic display) of the LCD controller/driver.

2.2.15 SCOM0

This is a common signal output pin (for static display) of the LCD controller/driver.

2.2.16 V_{LC0} to V_{LC2}

These are the LCD driving voltage pins. Individually connect to capacitors (recommended value: 0.47 μF) externally between V_{LC0} and GND, V_{LC1} and GND, V_{LC2} and GND to supply the LCD driving voltage corresponding to each bias to the inside of the V_{LC0} to V_{LC2} pins.

- V_{LC0}: Three times the V_{LC2} output voltage
- V_{LC1}: Two times the V_{LC2} output voltage
- V_{LC2}: Reference voltage

2.2.17 CAPH and CAPL

These are booster capacitor connection pins for the LCD drive voltage. These pins are used for connecting capacitors (recommended value: 0.47 μF) between CAPH and CAPL.

2.2.18 $\overline{\text{RESET}}$

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

2.2.19 X1 and X2

Crystal resonator connection pins for main system clock oscillation.

For external clock supply, input the clock signal to X1 and its inverted signal to X2.

2.2.20 XT1 and XT2

Crystal resonator connection pins for subsystem clock oscillation.

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

2.2.21 V_{DD0} and V_{DD1}

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

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

2.2.22 V_{SS0} and V_{SS1}

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

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

2.2.23 V_{PP} (flash memory versions only)

High-voltage application pin for flash memory programming mode setting and program write/verify.

Connect this pin in either of the following ways.

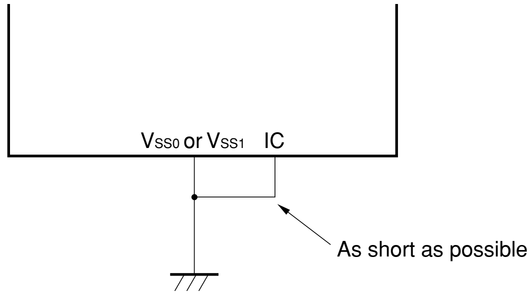
- Connect independently to a 10 kΩ pull-down resistor.
- By using a jumper on the board, connect directly to the dedicated flash programmer in the programming mode or to V_{SS0} in the normal operation mode.

2.2.24 IC (mask ROM version only)

The IC (Internally Connected) pin is provided to set the test mode to check the μPD780344, 780354, 780344Y, 780354Y Subseries at delivery. Connect it directly to the V_{SS0} or V_{SS1} pin with the shortest possible wire in the normal operation mode.

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

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



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 circuit of each type.

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

Pin Name		I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00/INTP0 to P02/INTP2		8-C	I/O	Input: Independently connect to V_{SS0} or V_{SS1} via a resistor. Output: Leave open.
P03/INTP3/ADTRG				
P04/INTP4				
P05/INTP5/PCL				
P06/INTP6/TOB0				
P07/TOA0/TMIB0				
P10/ANI0 to P17/ANI7		25	Input	Directly connect to V_{DD0} or V_{SS0}.
P20/SI3		8-C	I/O	Input: Independently connect to V_{DD0}, V_{DD1}, V_{SS0} or V_{SS1} via a resistor. Output: Leave open.
P21/SO3		5-H		
P22/$\overline{SCK3}$		8-C		
P23/SI1				
P24/SO1		5-H		
P25/SCK1		8-C		
P26/RxD0				
P27/TxD0		5-H		
P30, P31	µPD780343, 780344, 780353, 780354 only	13-S		Input: Directly connect to V_{SS0} or V_{SS1}. Output: Leave open at low level.
	µPD78F0354, 78F0354A only	13-R		
P30/SCL0	µPD780344Y, 780354Y Subseries only			
P31/SDA0				
P32/TI51/TO51		8-C		Input: Independently connect to V_{DD0}, V_{DD1}, V_{SS0}, or V_{SS1} via a resistor. Output: Leave open.
P33/TO50/TI50				
P34/TI01/TO00				
P35/TI00				
P40 to P43		5-H		
P70 to P73	Mask ROM version	13-Q		Input: Directly connect to V_{SS0} or V_{SS1}. Output: Leave open at low level.
	Flash memory version	13-P		
P80/S12 to P87/S19		17-G		Input: Independently connect to V_{DD0}, V_{DD1}, V_{SS0}, or V_{SS1} via a resistor. Output: Leave open.
P90/S20 to P97/S27				
P100/S28 to P107/S35				
P110/S36 to P113/S39				
S0 to S11		17-D	Output	Leave open.

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

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
COM0 to COM3	18-B	Output	Leave open.
SCOM0			
V_{LC0} to V_{LC2}	—	—	
CAPH, CAPL			
$\overline{\text{RESET}}$	2	Input	—
XT1	16	Input	Directly connect to V_{DD0} or V_{DD1}.
XT2		—	Leave open.
AV_{DD}	—	Input	Directly connect to V_{DD0} or V_{DD1}.
AV_{SS}		—	Directly connect to V_{SS0} or V_{SS1}.
IC			
V_{PP}			Independently connect 10 kΩ pull-down resistor, or directly connect to V_{SS0} or V_{SS1}.

Figure 2-1. Pin I/O Circuit List (1/2)

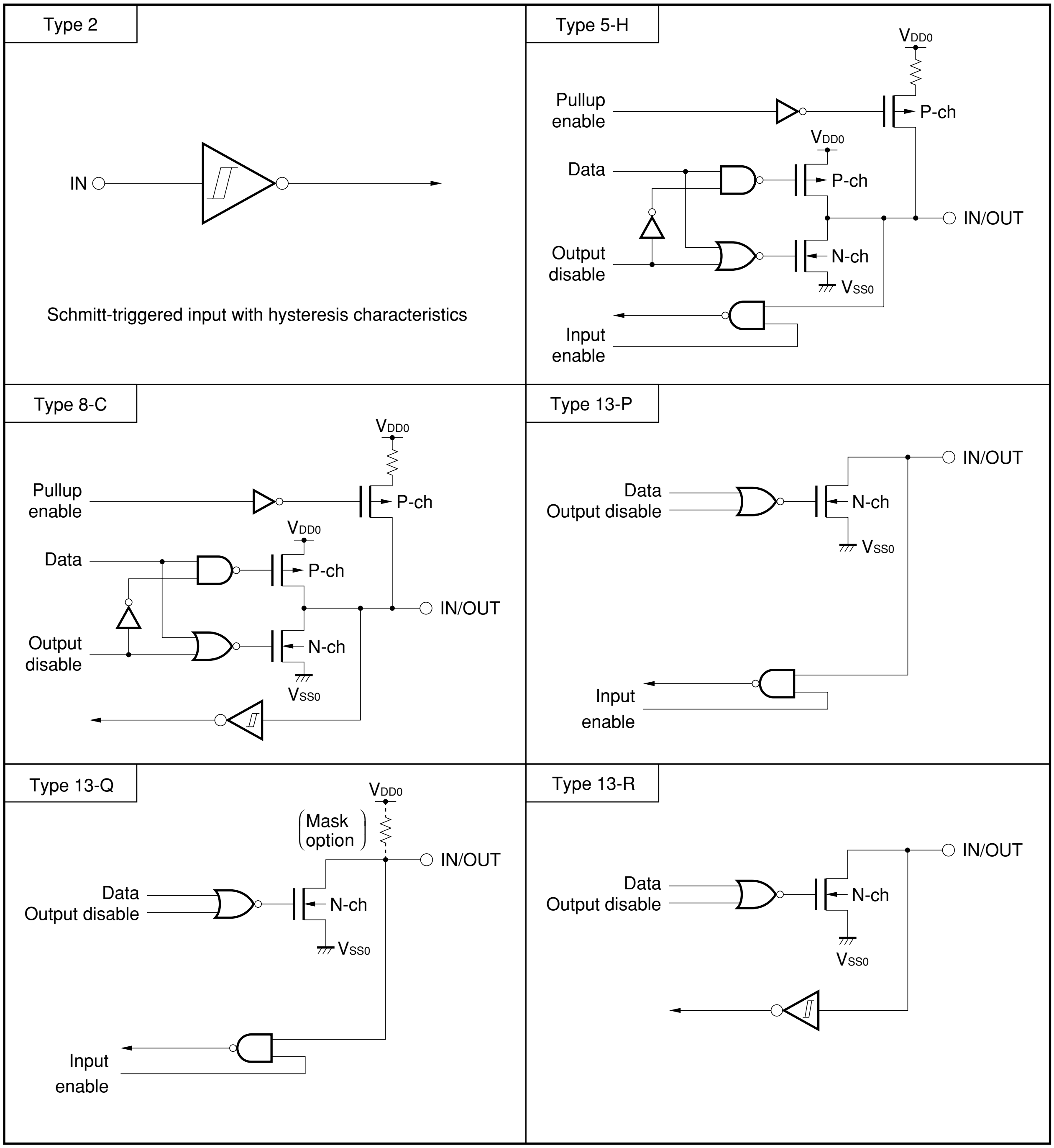
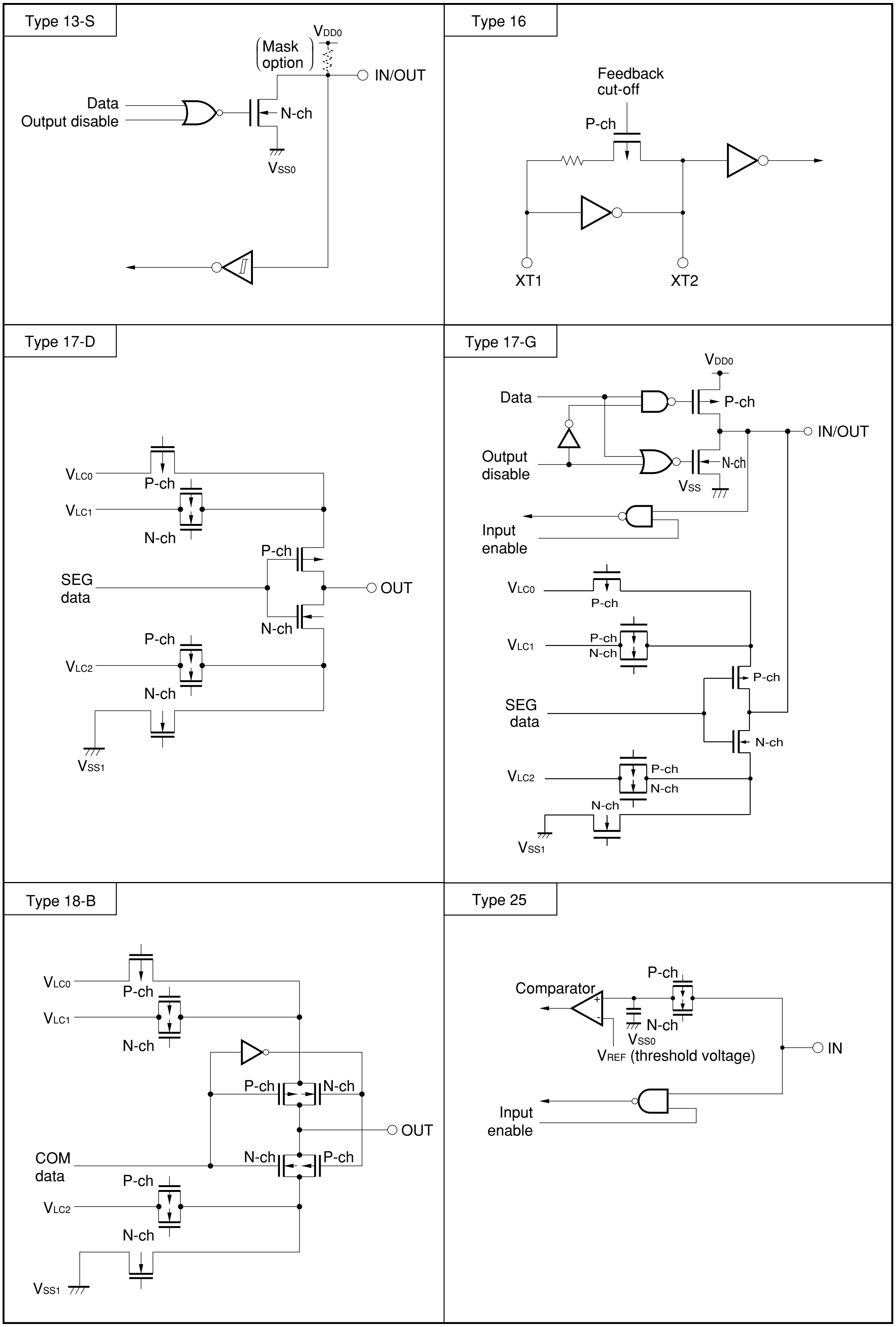


Figure 2-1. Pin I/O Circuit List (2/2)



CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Spaces

The μPD780344, 780354, 780344Y, 780354Y Subseries can each access a 64 KB memory space.
Figures 3-1 to 3-3 show the memory maps.

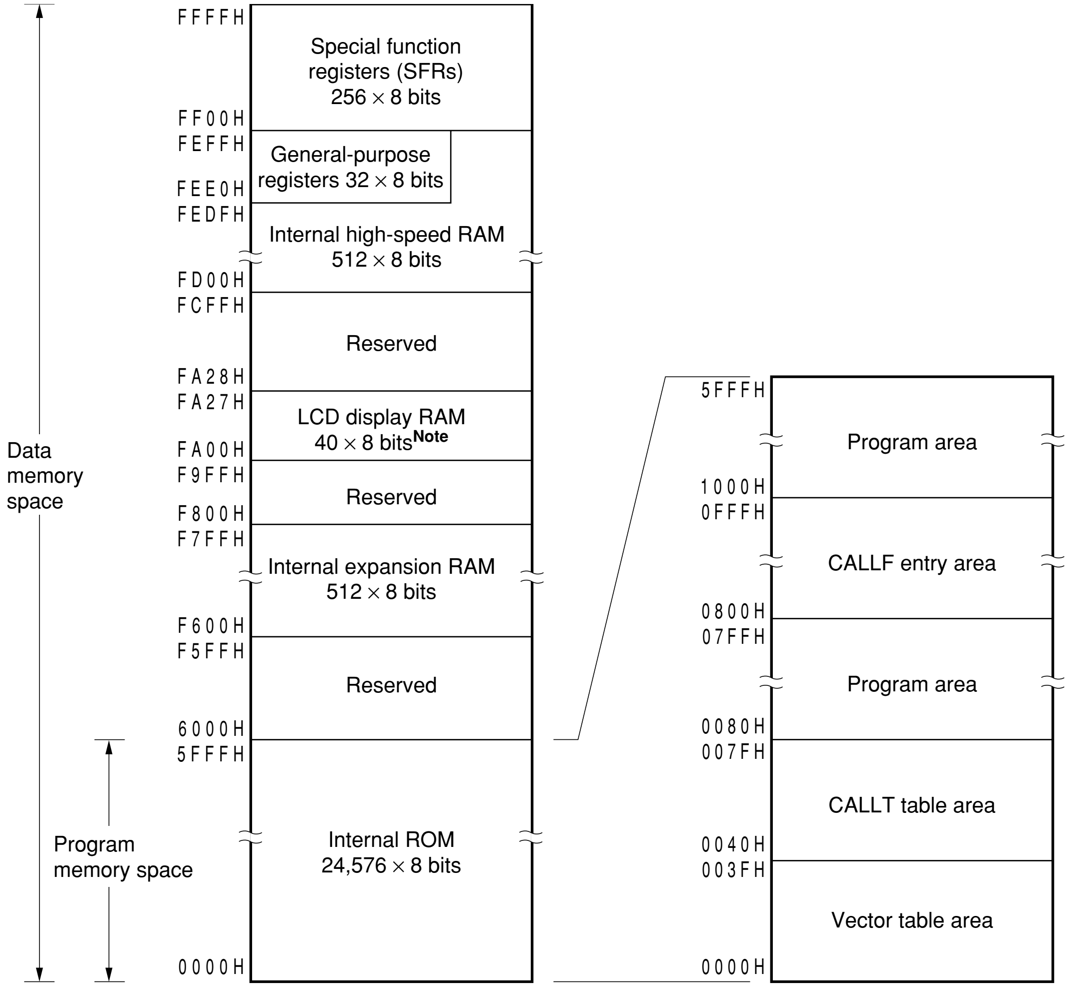
Caution In the case of the internal memory capacity, the initial values of the memory size switching register (IMS) and internal expansion RAM size switching register (IXS) of all products (μPD780344, 780354, 780344Y, and 780354Y Subseries) are fixed (IMS = CFH, IXS = 0CH). Therefore, set the value corresponding to each product indicated below.

	Set Value of IMS	Set Value of IXS
μPD780343, 780353, 780343Y, 780353Y	46H	0BH
μPD780344, 780354, 780344Y, 780354Y	48H	
μPD78F0354, 78F0354A, 78F0354Y, 78F0354AY	C8H or Value corresponding to mask ROM version	

(1) μPD780343, 780353, 780343Y, 780353Y

Set the value of the memory size switching register (IMS) to 46H, and the value of the internal expansion RAM size switching register (IXS) to 0BH (default setting: IMS = CFH, IXS = 0CH).

Figure 3-1. Memory Map (μPD780343, 780353, 780343Y, 780353Y)

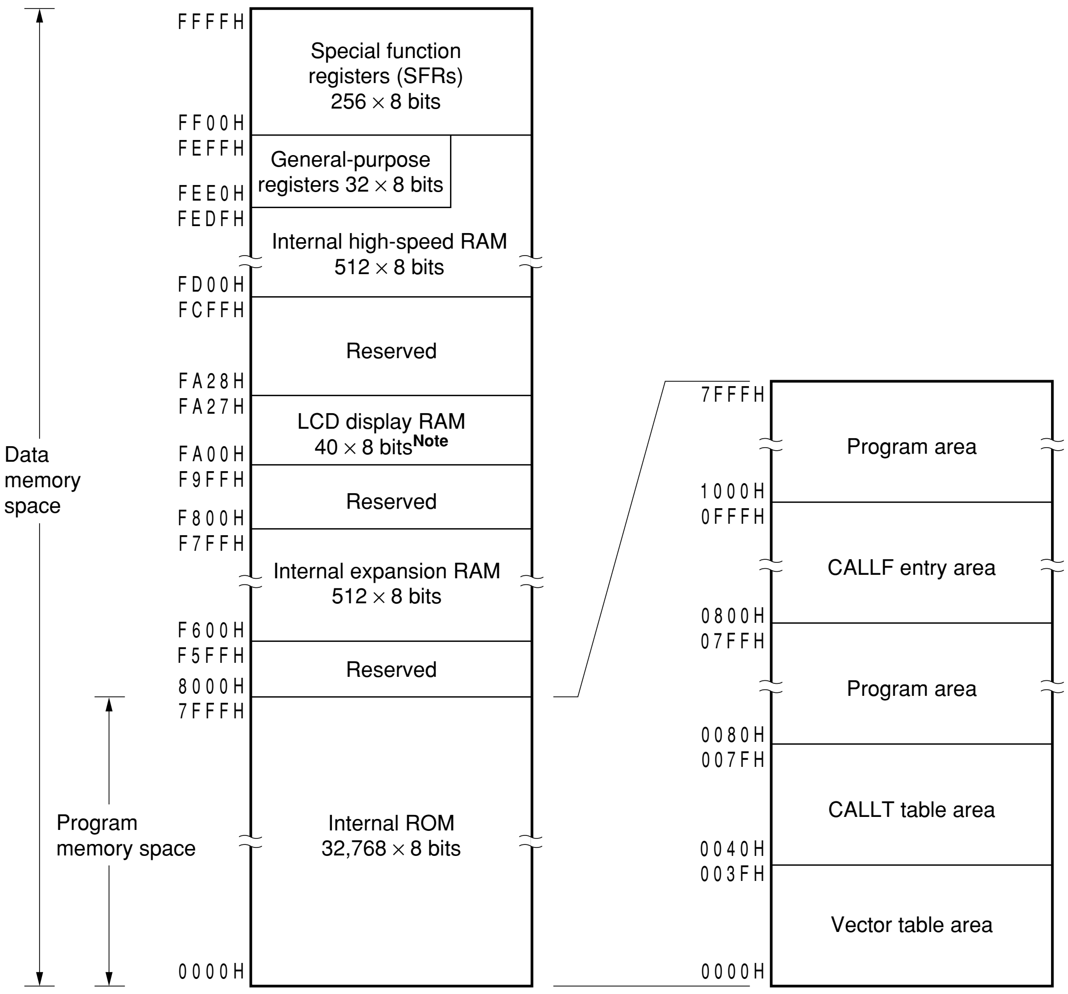


Note The area not used for LCD display data can be used as normal RAM.

(2) μPD780344, 780354, 780344Y, 780354Y

Set the value of the memory size switching register (IMS) to 48H, and the value of the internal expansion RAM size switching register (IXS) to 0BH (default setting: IMS = CFH, IXS = 0CH).

Figure 3-2. Memory Map (μPD780344, 780354, 780344Y, 780354Y)

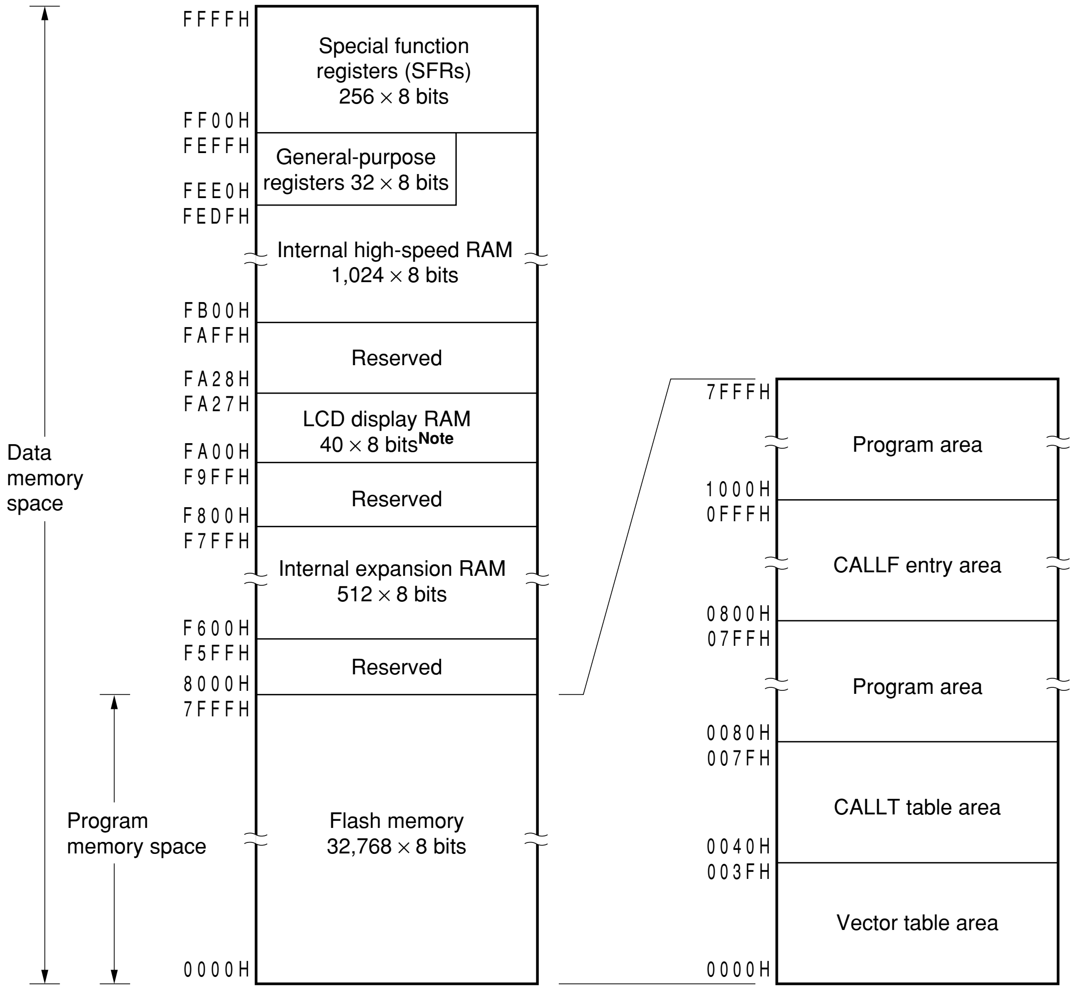


Note The area not used for LCD display data can be used as normal RAM.

(3) μPD78F0354, 78F0354A, 78F0354Y, 78F0354AY

Set the value of the memory size switching register (IMS) to C8H or the value corresponding to the mask ROM version, and the value of the internal expansion RAM size switching register (IXS) to 0BH (default setting: IMS = CFH, IXS = 0CH).

Figure 3-3. Memory Map (μPD78F0354, 78F0354A, 78F0354Y, 78F0354AY)



Note The area not used for LCD display data can be used as normal RAM.

3.1.1 Internal program memory space

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

The μPD780344, 780354, 780344Y, and 780354Y Subseries products incorporate an internal ROM (or flash memory), as listed below.

Table 3-1. Internal Memory Capacity

Part Number	Structure	Capacity
μPD780343, 780353, 780343Y, 780353Y	Mask ROM	24,576 × 8 bits (0000H to 5FFFH)
μPD780344, 780354, 780344Y, 780354Y		32,768 × 8 bits (0000H to 7FFFH)
μPD78F0354, 78F0354A, 78F0354Y, 78F0354AY	Flash memory	

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

(1) Vector table area

The 64-byte area 0000H to 003FH is reserved as a vector table area. The program start addresses for branch upon $\overline{\text{RESET}}$ input or generation of an 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 3-2. Vector Table

Vector Table Address	Interrupt Source	Vector Table Address	Interrupt Source
0000H	$\overline{\text{RESET}}$ input	001CH	INTCSI1
0004H	INTWDT	001EH	INTCSI3
0006H	INTP0	0020H	INTIIC0[Note]
0008H	INTP1	0022H	INTWTNI0
000AH	INTP2	0024H	INTTM00
000CH	INTP3	0026H	INTTM01
000EH	INTP4	0028H	INTTMA0
0010H	INTP5	002AH	INTTMB0
0012H	INTP6	002CH	INTTM50
0014H	INTKR	002EH	INTTM51
0016H	INTSER0	0030H	INTAD0
0018H	INTSR0	0032H	INTWTN0
001AH	INTST0	003EH	BRK

Note μPD780344Y, 780354Y Subseries only

(2) CALLT instruction table area

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

(3) CALLF instruction entry area

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

3.1.2 Internal data memory space

The μPD780344, 780354, 780344Y, and 780354Y Subseries products incorporate the following RAM.

(1) Internal high-speed RAM

The internal high-speed RAM is assigned to the area FD00H to FEFFH (512 bytes) of the mask ROM version and to the area FB00H to FEFFH (1,024 bytes) of the flash memory version.

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

★ The internal high-speed RAM cannot be used as a program area in which instructions are written and executed.

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

(2) Internal expansion RAM

The area F600H to F7FFH (512 bytes) is assigned to the internal expansion RAM.

★ The internal expansion RAM can be used as a normal data area in the same way as the internal high-speed RAM, and can be used as a program area in which instructions are written and executed.

(3) LCD display RAM

The area FA00H to FA27H (40 bytes) is assigned to the LCD display RAM.

LCD display RAM can also be used as normal RAM. Therefore, the area not used as LCD display data can be used as normal RAM.

3.1.3 Special function register (SFR) area

An on-chip peripheral hardware special function register (SFR) is allocated in the area FF00H to FFFFH (refer to **3.2.3 Special function register (SFR) Table 3-3 Special Function Register List**).

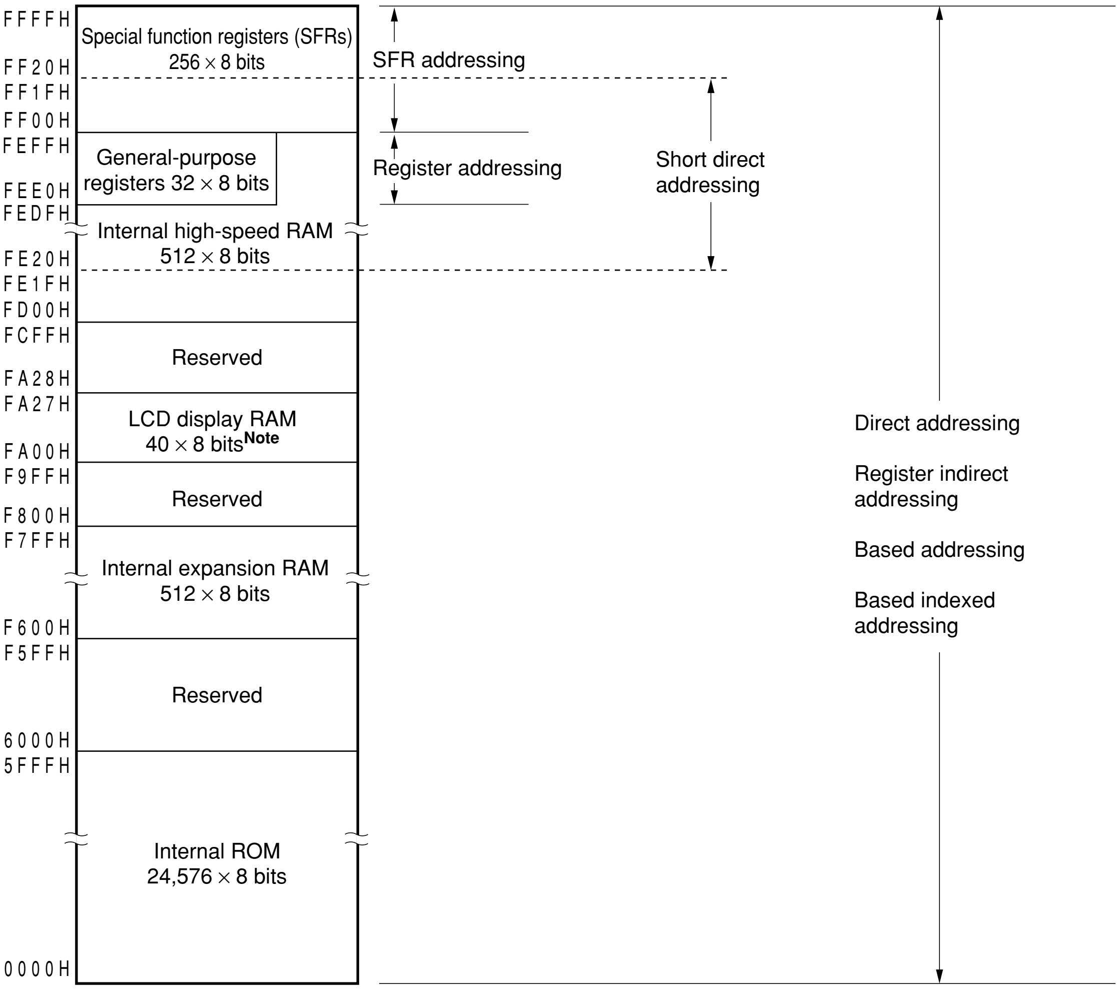
Caution Do not access addresses where SFRs are not assigned.

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.

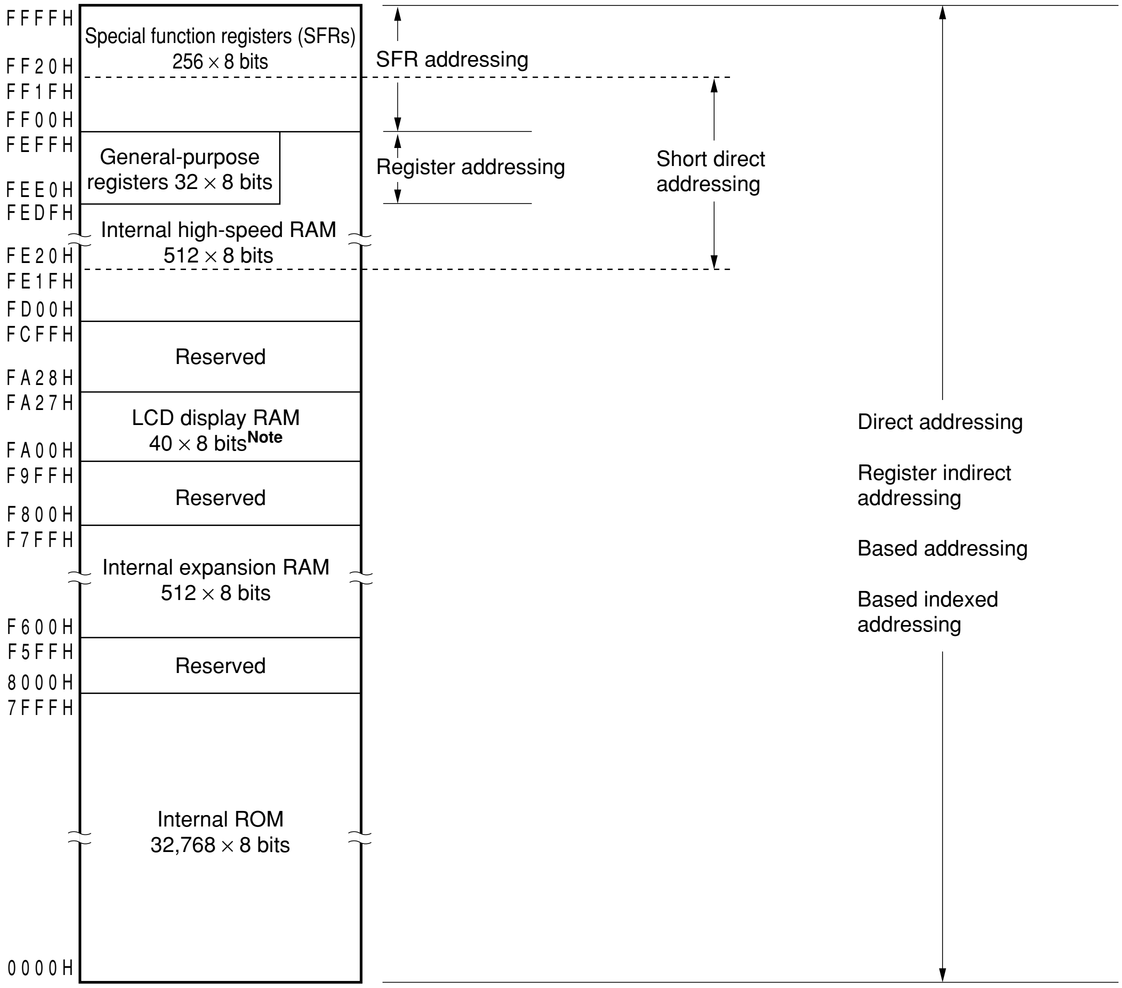
Several addressing modes are provided for addressing the memory relevant to the execution of instructions for the μPD780344, 780354, 780344Y, and 780354Y Subseries, based on operability and other considerations. For areas containing data memory in particular, special addressing methods designed for the functions of special function registers (SFR) and general-purpose registers are available for use. The correspondence between the data memory and the addressing mode is illustrated in Figures 3-4 to 3-6. For details of each addressing mode, see **3.4 Operand Address Addressing**.

Figure 3-4. Correspondence Between Data Memory and Addressing (μPD780343, 780353, 780343Y, 780353Y)



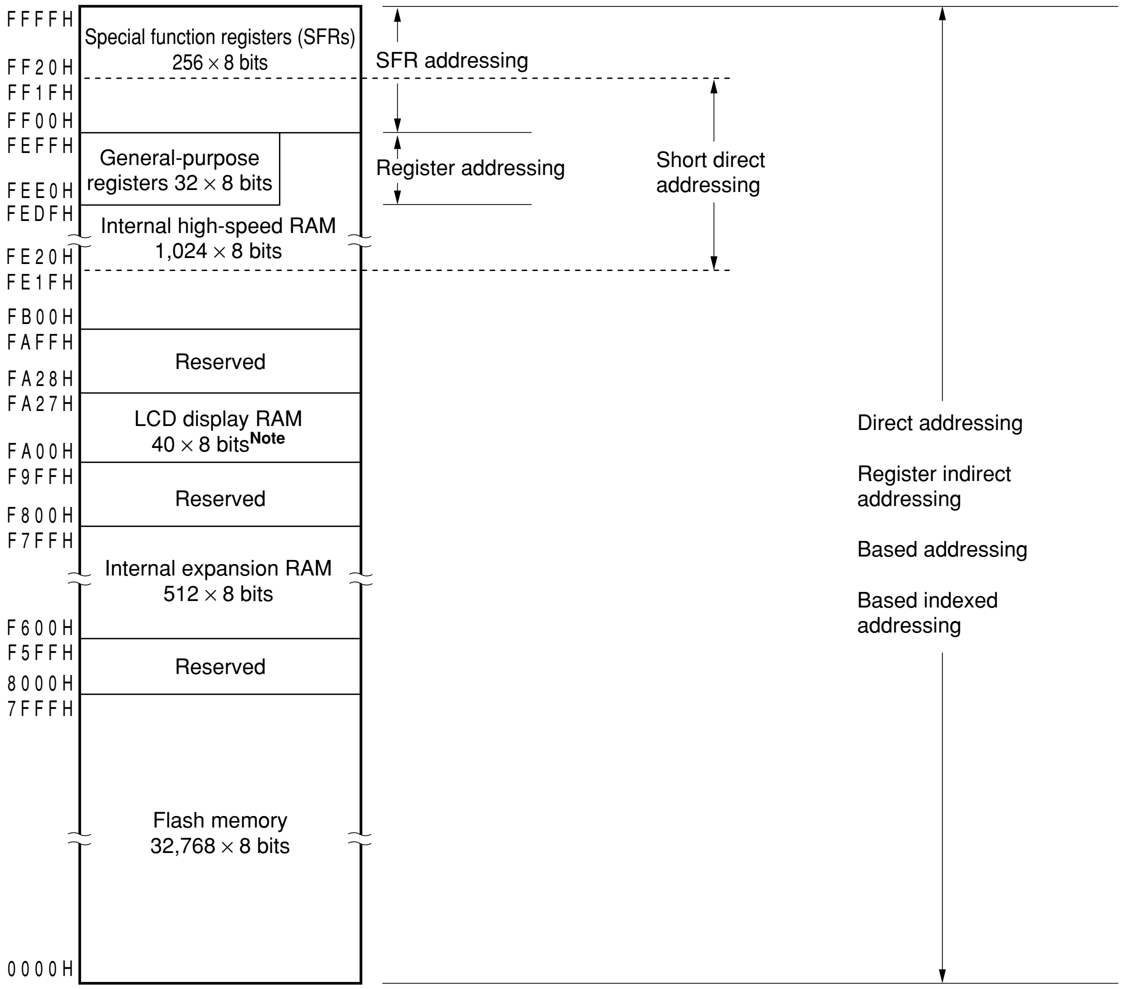
Note The area not used as LCD display data can be used as normal RAM.

Figure 3-5. Correspondence Between Data Memory and Addressing (μPD780344, 780354, 780344Y, 780354Y)



Note The area not used as LCD display data can be used as normal RAM.

Figure 3-6. Correspondence Between Data Memory and Addressing (μPD78F0354, 78F0354A, 78F0354Y, 78F0354AY)



Note The area not used as LCD display data can be used as normal RAM.

3.2 Processor Registers

The μPD780344, 780354, 780344Y, and 780354Y Subseries products incorporate the following processor registers.

3.2.1 Control registers

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

(1) Program counter (PC)

The program counter is a 16-bit register which holds the address information of the next program to be executed. In normal operation, the PC is automatically incremented according to the number of bytes of the instruction to be fetched. When a branch instruction is executed, immediate data and register contents are set.

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

Figure 3-7. Program Counter Format

	15															0
PC	PC15	PC14	PC13	PC12	PC11	PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0

(2) Program status word (PSW)

The program status word is an 8-bit register consisting of various flags to be set/reset by instruction execution. Program status word contents are automatically stacked upon interrupt request generation or PUSH PSW instruction execution and are automatically reset upon execution of the RETB, RETI, and POP PSW instructions.

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

Figure 3-8. Program Status Word Format

	7							0
PSW	IE	Z	RBS1	AC	RBS0	0	ISP	CY

(a) Interrupt enable flag (IE)

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

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

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

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

(b) Zero flag (Z)

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

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

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

In these flags, the 2-bit information which indicates the register bank selected by the 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 interrupt requests specified by a priority specification flag register (PR0L, PR0H, PR1L) (refer to **18.3 (3) Priority specification flag registers (PR0L, PR0H, PR1L)**) are disabled for acknowledgement. Actual request acknowledgement is controlled by the interrupt enable flag (IE).

(f) Carry flag (CY)

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

(3) Stack pointer (SP)

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

Figure 3-9. Stack Pointer Format

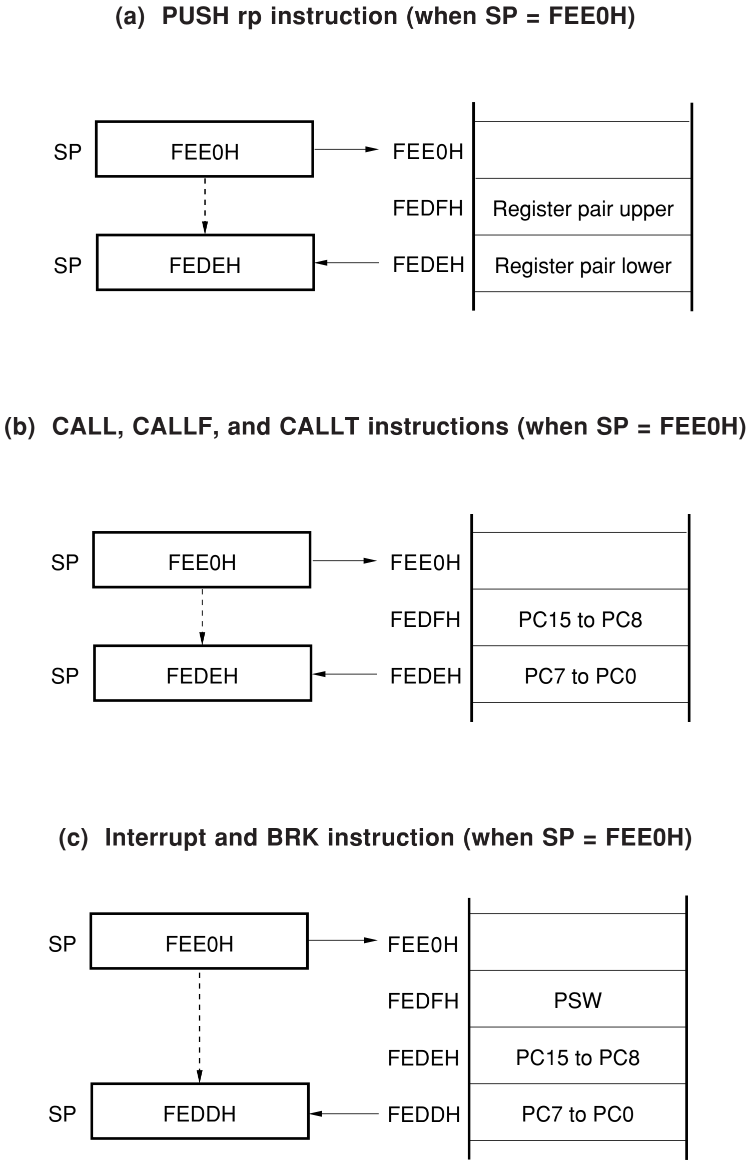
	15															0
SP	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0

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

Each stack operation saves/resets data as shown in Figures 3-10 and 3-11.

★ **Caution Since $\overline{\text{RESET}}$ input makes SP contents undefined, be sure to initialize the SP before using the stack.**

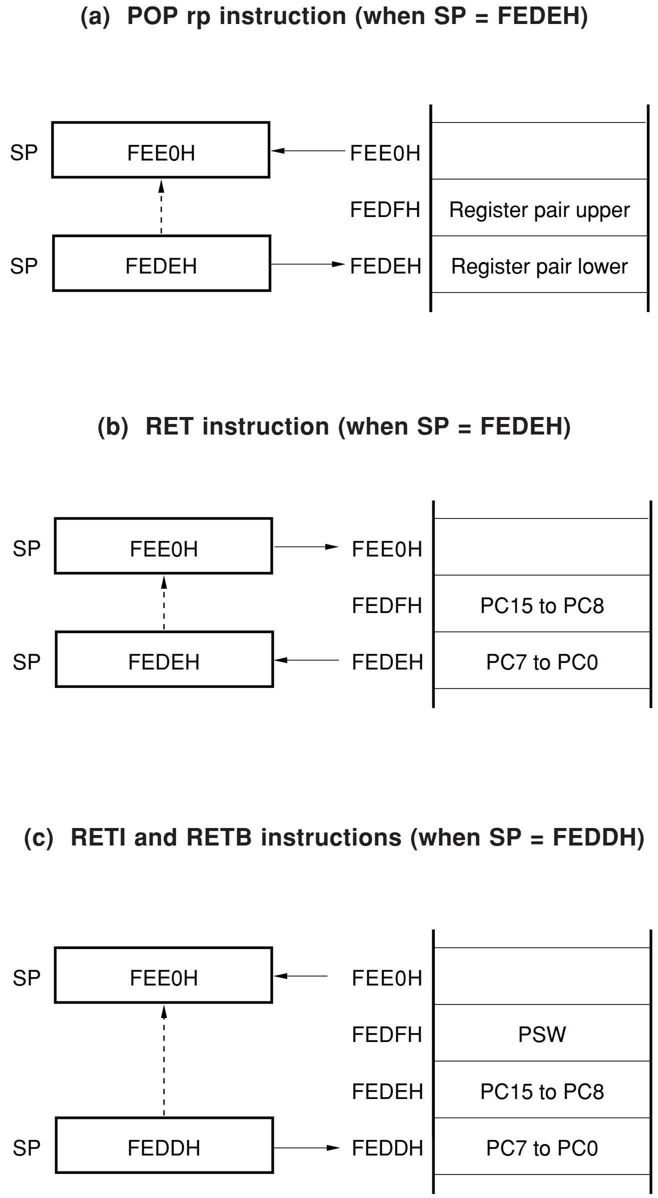
★ Figure 3-10. Data To Be Saved to Stack Memory


(a) PUSH rp instruction (when SP = FEE0H)

(b) CALL, CALLF, and CALLT instructions (when SP = FEE0H)

(c) Interrupt and BRK instruction (when SP = FEE0H)

★ **Figure 3-11. Data To Be Restored from Stack Memory**



(a) POP rp instruction (when SP = FEDEH)

(b) RET instruction (when SP = FEDEH)

(c) RETI and RETB instructions (when SP = FEDDH)

3.2.2 General-purpose registers

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

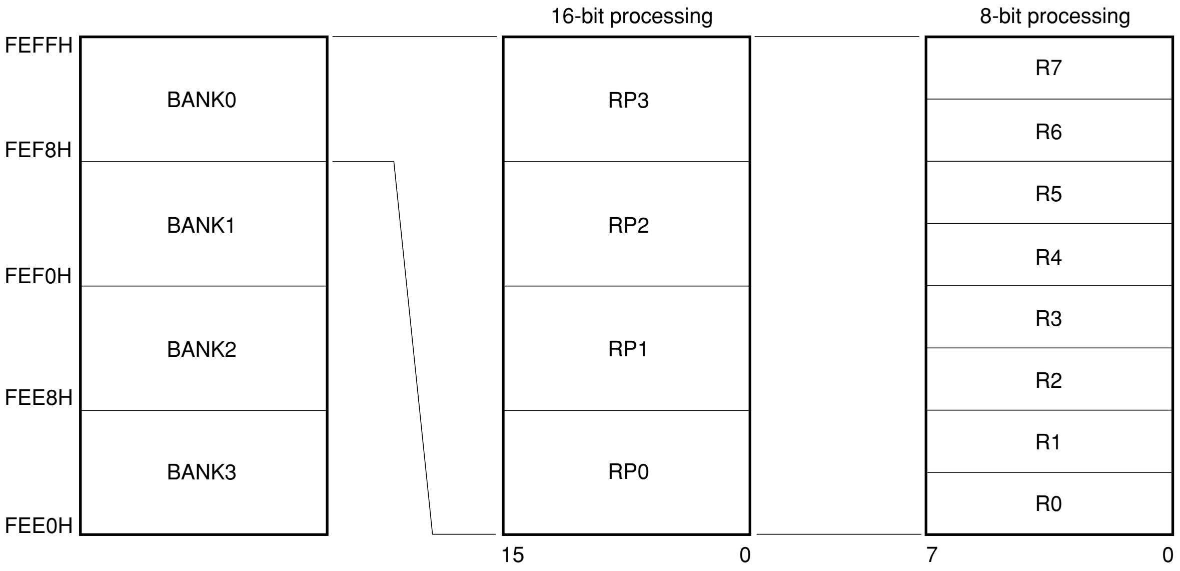
Each register can be used as an 8-bit register, and two 8-bit registers can be used in pairs as a 16-bit register (AX, BC, DE, and HL).

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

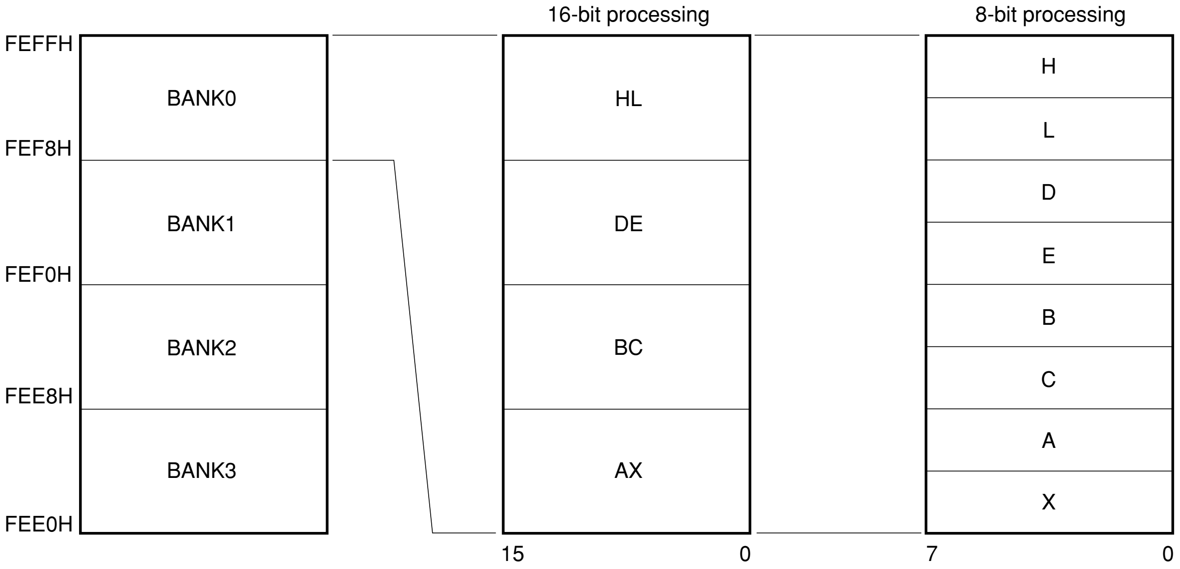
Register banks to be used for instruction execution are set with the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interrupts for each bank.

Figure 3-12. General-Purpose Register Configuration

(a) Absolute name



(b) Function name



3.2.3 Special-function registers (SFR)

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

They are allocated to the FF00H to FFFFH area.

The special function registers can be manipulated like general-purpose registers, with 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 in the assembler for the 1-bit manipulation instruction operand (sfr.bit).
  This manipulation can also be specified with an address.
- 8-bit manipulation
  Describe the symbol reserved in the assembler for the 8-bit manipulation instruction operand (sfr).
  This manipulation can also be specified with an address.
- 16-bit manipulation
  Describe the symbol reserved in the assembler for the 16-bit manipulation instruction operand (sfrp).
  When addressing an address, describe an even address.

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

- Symbol
  Symbol indicating the address of a special function register. It is a reserved word in the RA78K0, and is defined via the header file “sfrbit.h” in the CC78K0. When using the RA78K0, ID78K0-NS, 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 3-3. Special Function Register List (1/3)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset
				1 Bit	8 Bits	16 Bits	
FF00H	Port 0	P0	R/W	√	√	—	00H
FF01H	Port 1	P1	R	√	√	—	00H
FF02H	Port 2	P2	R/W	√	√	—	00H
FF03H	Port 3	P3	R/W	√	√	—	00H
FF04H	Port 4	P4	R/W	√	√	—	00H
FF07H	Port 7	P7	R/W	√	√	—	00H
FF08H	Port 8[Note 1]	P8	R/W	√	√	—	00H
FF09H	Port 9[Note 1]	P9	R/W	√	√	—	00H
FF0AH	Port 10[Note 1]	P10	R/W	√	√	—	00H
FF0BH	Port 11[Note 1]	P11	R/W	√	√	—	00H
FF0CH	Transmit buffer register 1	SOTB1	R/W	—	√	—	Undefined
FF0DH	Serial I/O shift register 1	SIO1	R	—	√	—	Undefined
FF0EH	A/D conversion result register 0[Note 2]	ADCR0[Note 2]	R	—	—	√	0000H
FF0FH	A/D conversion result register 1[Note 3]	ADCR1[Note 3]	R	—	√	—	00H
FF10H	16-bit timer capture/compare register 00	CR00	R/W	—	—	√	Undefined
FF11H							
FF12H	16-bit timer capture/compare register 01	CR01	R/W	—	—	√	Undefined
FF13H							
FF14H	16-bit timer counter 0	TM0	R	—	—	√	0000H
FF15H							
FF16H	8-bit timer compare register 50	CR50	R/W	—	√	—	Undefined
FF17H	8-bit timer compare register 51	CR51	R/W	—	√	—	Undefined
FF18H	8-bit timer counter 50	TM50 TM5	R	—	√	√	00H
FF19H	8-bit timer counter 51	TM51	R	—	√		00H
FF1AH	Serial I/O shift register 3	SIO3	R/W	—	√	—	Undefined
FF1BH	Transmit shift register 0	TXS0	W	—	√	—	FFH
	Receive buffer register 0	RXB0	R	—	√	—	FFH
FF1FH	IIC shift register 0[Note 4]	IIC0	R/W	—	√	—	00H
FF20H	Port mode register 0	PM0	R/W	√	√	—	FFH
FF22H	Port mode register 2	PM2	R/W	√	√	—	FFH
FF23H	Port mode register 3	PM3	R/W	√	√	—	FFH
FF24H	Port mode register 4	PM4	R/W	√	√	—	FFH
FF27H	Port mode register 7	PM7	R/W	√	√	—	FFH
FF28H	Port mode register 8[Note 1]	PM8	R/W	√	√	—	FFH
FF29H	Port mode register 9[Note 1]	PM9	R/W	√	√	—	FFH
FF2AH	Port mode register 10[Note 1]	PM10	R/W	√	√	—	FFH
FF2BH	Port mode register 11[Note 1]	PM11	R/W	√	√	—	FFH

Notes **1.** Only the pin whose port function is selected by the pin function switching register (PF8 to PF11) is valid.

2. μPD780354, 780354Y Subseries only

3. μPD780344, 780344Y Subseries only

★ **4.** μPD780344Y, 780354Y Subseries only

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset
					1 Bit	8 Bits	16 Bits	
FF30H	Pull-up resistor option register 0	PU0		R/W	√	√	—	00H
FF32H	Pull-up resistor option register 2	PU2		R/W	√	√	—	00H
FF33H	Pull-up resistor option register 3	PU3		R/W	√	√	—	00H
FF34H	Pull-up resistor option register 4	PU4		R/W	√	√	—	00H
FF38H	Correction address register 0	CORAD0		R/W	—	—	√	0000H
FF39H								
FF3AH	Correction address register 1	CORAD1		R/W	—	—	√	0000H
FF3BH								
FF40H	Clock output select register	CKS		R/W	√	√	—	00H
FF41H	Watch timer operation mode register 0	WTNM0		R/W	√	√	—	00H
FF42H	Watchdog timer clock select register	WDCS		R/W	—	√	—	00H
FF47H	Memory expansion mode register	MEM		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
FF58H	Pin function switching register 8	PF8		R/W	√	√	—	00H
FF59H	Pin function switching register 9	PF9		R/W	√	√	—	00H
FF5AH	Pin function switching register 10	PF10		R/W	√	√	—	00H
FF5BH	Pin function switching register 11	PF11		R/W	√	√	—	00H
FF60H	16-bit timer mode control register 0	TMC0		R/W	√	√	—	00H
FF61H	Prescaler mode register 0	PRM0		R/W	—	√	—	00H
FF62H	Capture/compare control register 0	CRC0		R/W	√	√	—	00H
FF63H	16-bit timer output control register 0	TOC0		R/W	√	√	—	00H
FF68H	8-bit timer counter B0	TMB0	TMB	R	—	√	√	00H
FF69H	8-bit timer counter A0	TMA0		R	—	√		00H
FF6AH	8-bit compare register B0	CRB0	CRB	W	—	√	√	Undefined
FF6BH	8-bit compare register A0	CRA0		W	—	√		Undefined
FF6CH	8-bit H width compare register B0	CRHB0		W	—	√	—	Undefined
FF6DH	8-bit timer mode control register A0	TMCA0		R/W	√	√	—	00H
FF6EH	8-bit timer mode control register B0	TMCB0		R/W	√	√	—	00H
FF6FH	Carrier generator output control register B0	TCAB0		R/W	—	√	—	00H
FF70H	8-bit timer mode control register 50	TMC50		R/W	√	√	—	00H
FF71H	Timer clock select register 50	TCL50		R/W	—	√	—	00H
FF73H	8-bit timer mode control register 51	TMC51		R/W	√	√	—	00H
FF74H	Timer clock select register 51	TCL51		R/W	—	√	—	00H
FF78H	Subclock select register	SSCK		R/W	√	√	—	00H
FF79H	Watch timer interrupt time select register	WTIM		R/W	√	√	—	00H
FF80H	A/D converter mode register 0	ADM0		R/W	√	√	—	00H
FF81H	Analog input channel specification register 0	ADS0		R/W	—	√	—	00H
FF8AH	Correction control register	CORCN		R/W	√	√	—	00H

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset
					1 Bit	8 Bits	16 Bits	
FF90H	LCD display mode register 3	LCDM3		R/W	√	√	—	00H
FF91H	LCD clock control register 3	LCDC3		R/W	—	√	—	00H
FF92H	Static/dynamic display switching register 3	SDSEL3		R/W	—	√	—	00H
★ FF94H	LCD gain adjust register 0	VLCG0		R/W	√	√	—	00H
FFA0H	Asynchronous serial interface mode register 0	ASIM0		R/W	√	√	—	00H
FFA1H	Asynchronous serial interface status register 0	ASIS0		R	—	√	—	00H
FFA2H	Baud rate generator control register 0	BRGC0		R/W	—	√	—	00H
FFA4H	IIC control register 0[Note 1]	IICC0		R/W	√	√	—	00H
FFA5H	IIC status register 0[Note 1]	IICS0		R	√	√	—	
FFA6H	IIC transfer clock select register 0[Note 1]	IICCL0		R/W	√	√	—	
FFA7H	Slave address register 0[Note 1]	SVA0			—	√	—	
FFA8H	IIC function expansion register 0[Note 1]	IICX0			√	√	—	
FFAFH	Serial operation mode register 3	CSIM3		R/W	√	√	—	00H
FFB0H	Serial operation mode register 1	CSIM1		R/W	√	√	—	00H
FFB1H	Serial clock select register 1	CSIC1		R/W	√	√	—	10H
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	IF1L		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	Memory size switching register[Note 2]	IMS		R/W	—	√	—	CFH
★ FFF4H	Internal expansion RAM size switching register[Note 3]	IXS		R/W	—	√	—	0CH
FFF9H	Watchdog timer mode register	WDTM		R/W	√	√	—	00H
FFFAH	Oscillation stabilization time select register	OSTS		R/W	—	√	—	04H
FFFBH	Processor clock control register	PCC		R/W	√	√	—	04H

★ **Notes 1.** μPD780344Y, 780354Y Subseries only

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

μPD780343, 780353, 780343Y, 780353Y: 46H

μPD780344, 780354, 780344Y, 780354Y: 48H

μPD78F0354, 78F0354A, 78F0354Y, 78F0354AY: C8H or value for mask ROM version

3. Although the default value of this register is 0CH, use this register with a setting of 0BH.

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 Series Instructions User's Manual (U12326E)**).

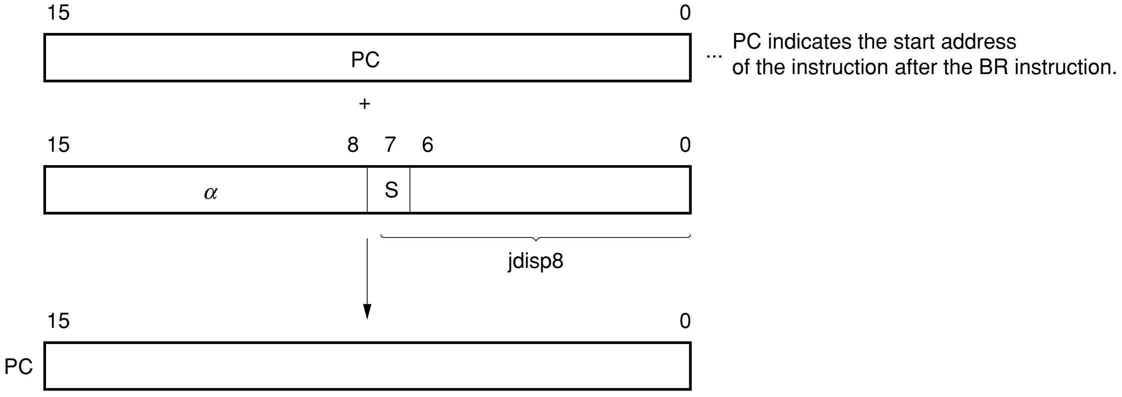
3.3.1 Relative addressing

[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 in relative branching from the start address of the following instruction to the –128 to +127 range.

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

[Illustration]


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

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

3.3.2 Immediate addressing

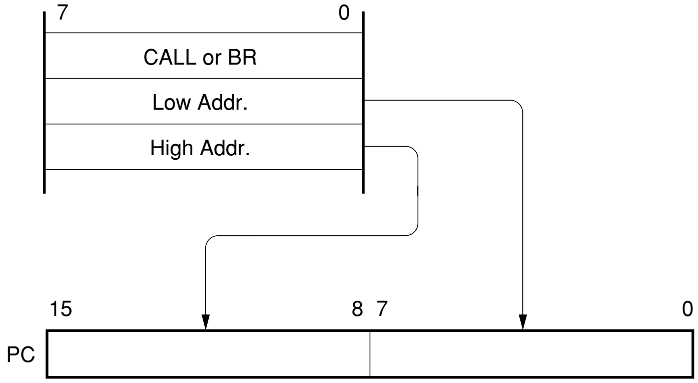
[Function]

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

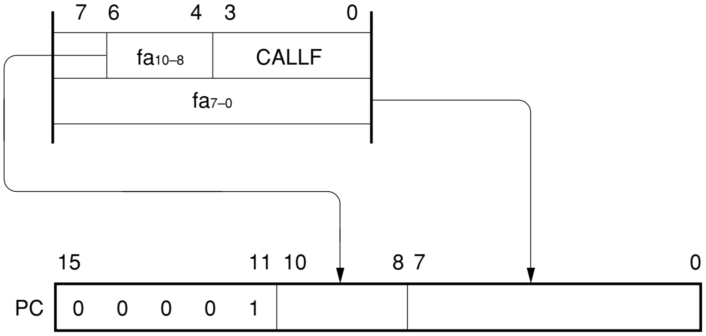
This function is carried out when the CALL !addr16 or BR !addr16 or CALLF !addr11 instruction is executed. CALL !addr16 and BR !addr16 instructions can be branched to the entire memory space. The CALLF !addr11 instruction is branched to the 0800H to 0FFFH area.

[Illustration]

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



In the case of CALLF !addr11 instruction



3.3.3 Table indirect addressing

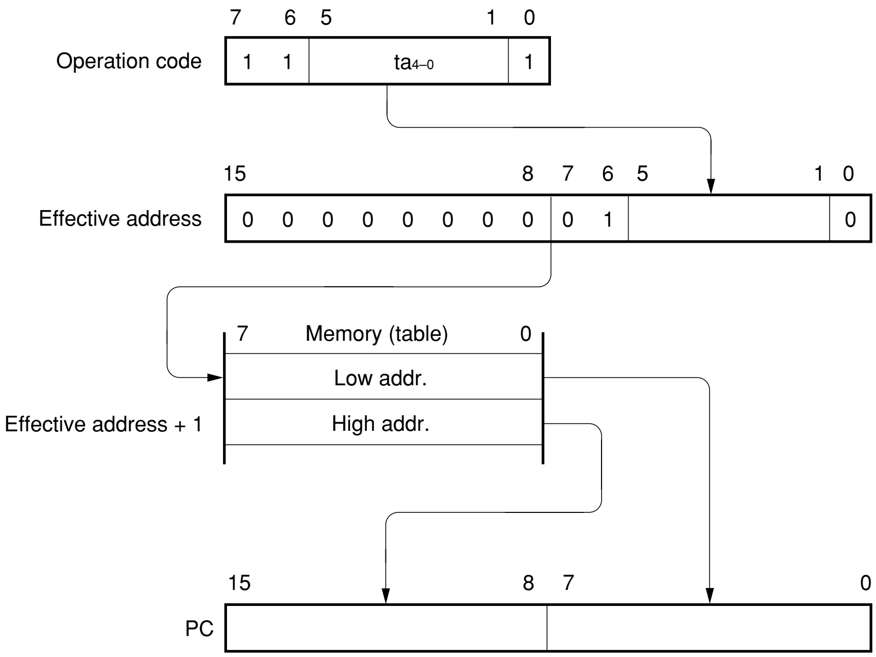
[Function]

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]



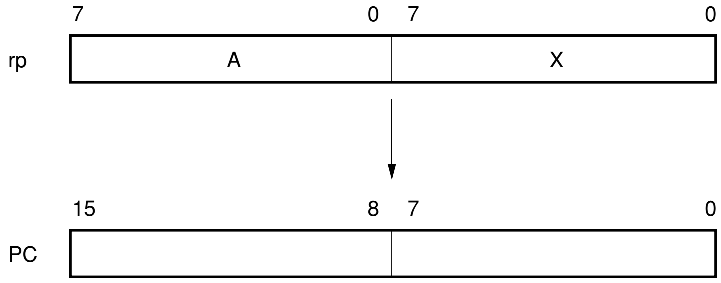
3.3.4 Register addressing

[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) which undergo manipulation during instruction execution.

3.4.1 Implied addressing

[Function]

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

Of the μPD780344, 780354, 780344Y, and 780354Y Subseries instruction words, the following instructions employ implied addressing.

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

[Operand format]

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

[Description example]

In the case of MULU X

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

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 and RBS1). Register addressing is carried out when an instruction with the following operand format is executed. When an 8-bit register is specified, one of the eight registers is specified with 3 bits in the operation code.

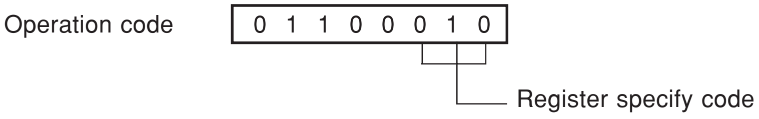
[Operand format]

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

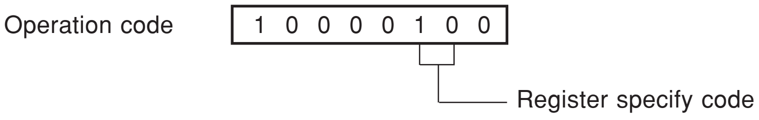
'r' and 'rp' can be described with absolute names (R0 to R7 and RP0 to RP3) as well as function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL).

[Description example]

MOV A, C; when selecting C register as r



INCW DE; when selecting DE register pair as rp



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]

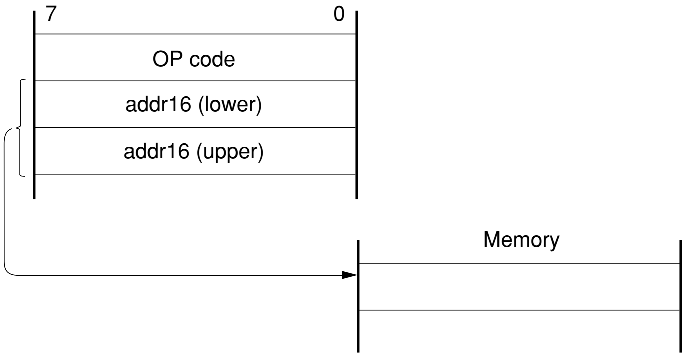
Identifier	Description
addr16	Label or 16-bit immediate data

[Description example]

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

Operation code	1 0 0 0 1 1 1 0	OP code
	0 0 0 0 0 0 0 0	00H
	1 1 1 1 1 1 1 0	FEH

[Illustration]



3.4.4 Short direct addressing

[Function]

The memory to be manipulated in the fixed space is directly addressed with 8-bit data in an instruction word. This addressing is applied to the 256-byte space FE20H to FF1FH. An internal RAM and a special function register (SFR) are mapped at FE20H to FEFFH and FF00H to FF1FH, respectively.

If the SFR area (FF00H to FF1FH) where short direct addressing is applied, ports which are frequently accessed in a program and a compare register of the timer/event counter and a capture register of the timer/event counter are mapped and these SFRs can be manipulated with a small number of bytes and clocks.

When 8-bit immediate data is at 20H to FFH, bit 8 of an effective address is set to 0. When it is at 00H to 1FH, bit 8 is set to 1. Refer to the **[Illustration]** below.

[Operand format]

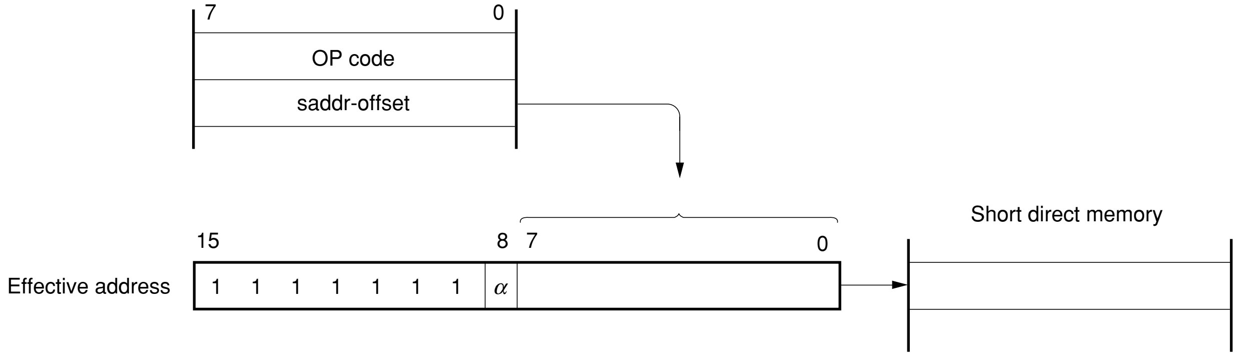
Identifier	Description
saddr	Label or immediate data indicating FE20H to FF1FH
saddrp	Label or immediate data (even address only) indicating FE20H to FF1FH

★ **[Description example]**

MOV 0FE30H, A; when transferring the value of the A register to saddr (FE30H)

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

[Illustration]



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

3.4.5 Special function register (SFR) addressing

[Function]

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

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

[Operand format]

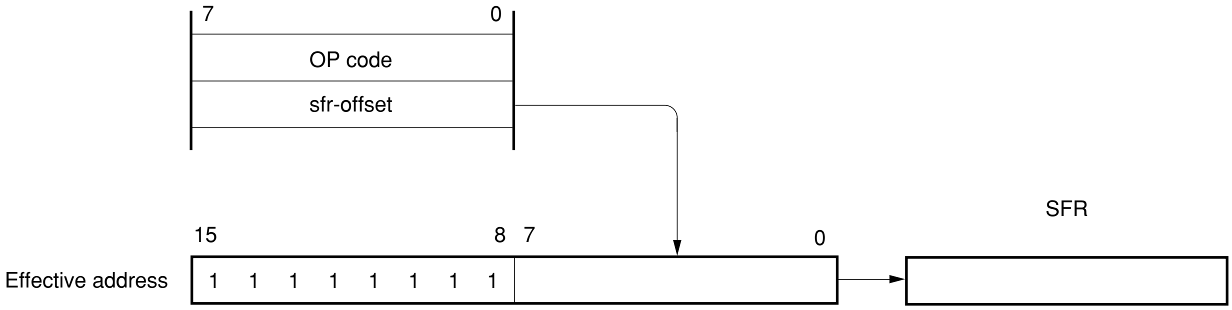
Identifier	Description
sfr	Special function register name
sfrp	16-bit manipulatable special function register name (even address only)

[Description example]

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

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

[Illustration]



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]

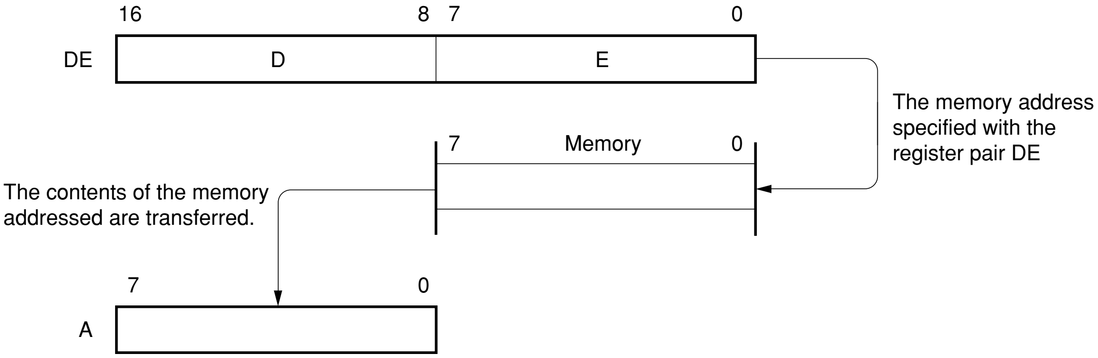
Identifier	Description
—	[DE], [HL]

[Description example]

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

Operation code 1 0 0 0 0 1 0 1

[Illustration]



The memory address specified with the register pair DE

The contents of the memory addressed are transferred.

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]

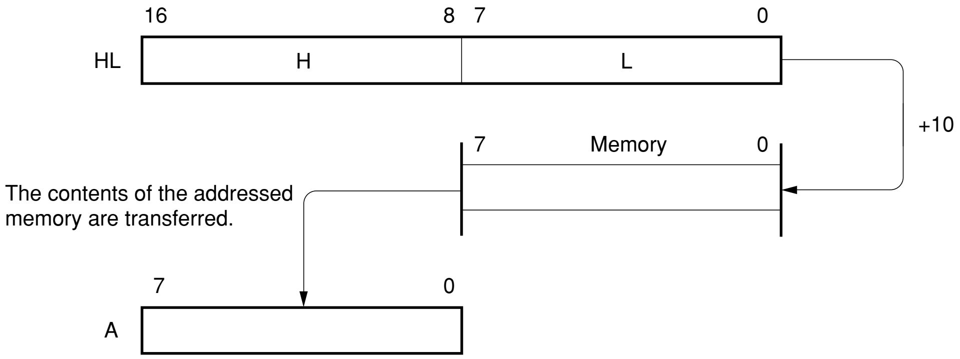
[Description example]

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

★ **[Illustration]**



The contents of the addressed memory are transferred.

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]

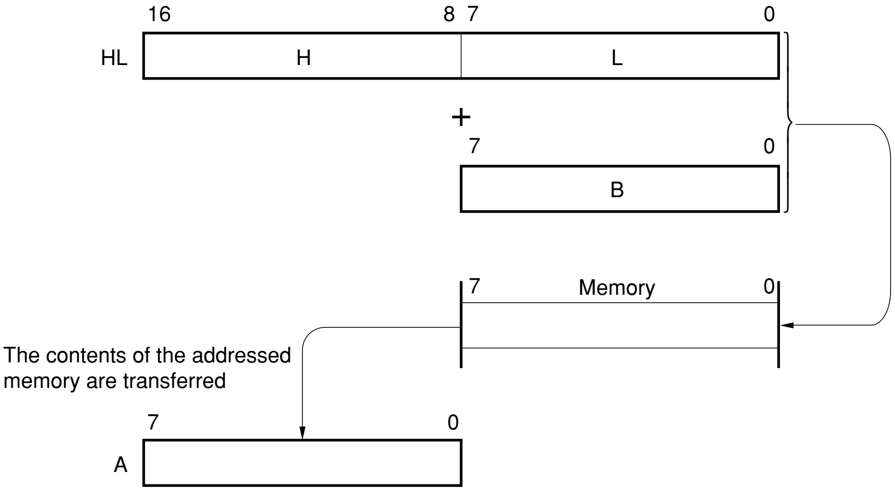
Identifier	Description
—	[HL + B], [HL + C]

[Description example]

In the case of MOV A, [HL + B] (when selecting B register)

Operation code | 1 0 1 0 1 0 1 1 |

★ **[Illustration]**



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.

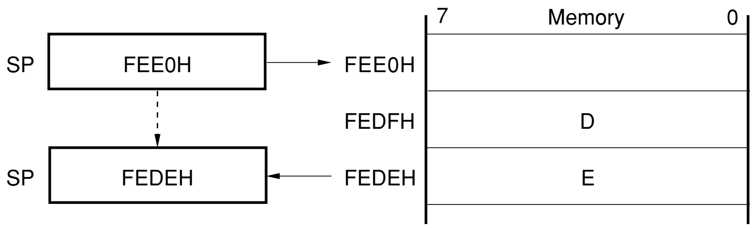
Stack addressing enables to address the internal high-speed RAM area only.

[Description example]

In the case of PUSH DE (when saving DE register)

Operation code	1 0 1 1 0 1 0 1

★ **[Illustration]**



CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

The μPD780344, 780354, 780344Y, and 780354Y Subseries products incorporate ports as shown in Figure 4-1. Every port is capable of 1-bit and 8-bit manipulations and can carry out considerably varied control operations. Besides port functions, the ports can also serve as on-chip hardware I/O pins.

Figure 4-1. Port Types

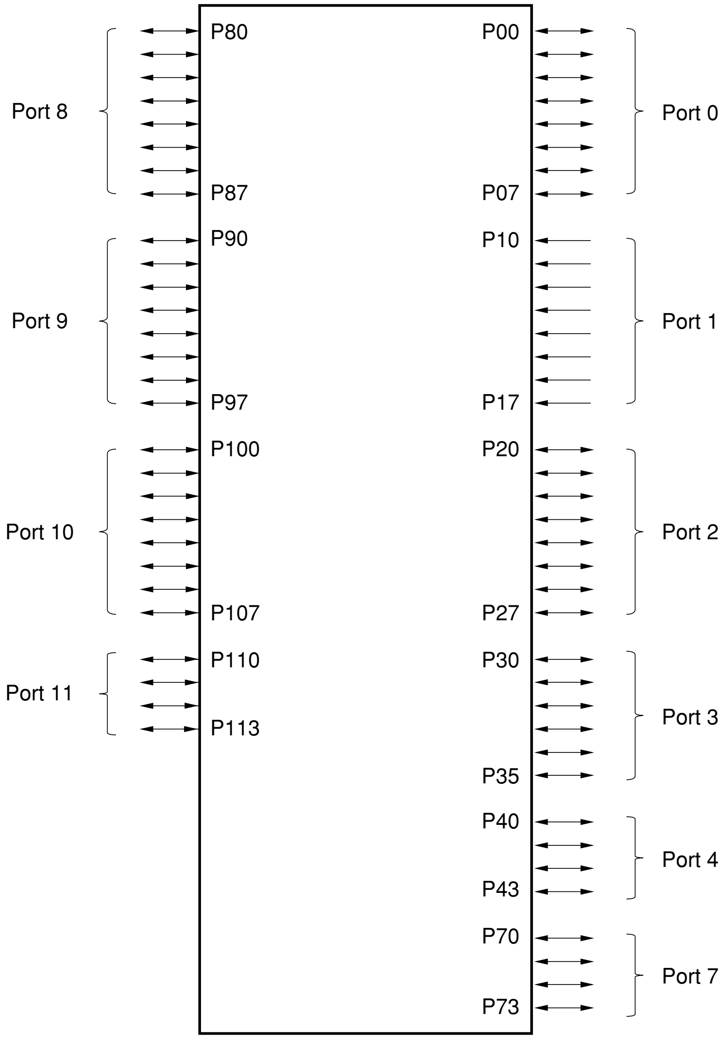


Table 4-1. Port Functions

Name	Pin Name	Function
Port 0	P00 to P07	I/O port. Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be used in 1-bit units by setting pull-up resistor option register 0 (PU0).
★ Port 1	P10 to P17	Input-only port
Port 2	P20 to P27	I/O port. Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be used in 1-bit units by setting pull-up resistor option register 2 (PU2).
Port 3	P30 to P35	I/O port. Input/output mode can be specified in 1-bit units. P30 and P31 are 5 V N-ch open-drain I/O port. Mask ROM version of the µPD780344, 780354 Subseries can specify an on-chip pull-up resistor in 1-bit units by mask option. P32 to P35 can use an on-chip pull-up resistor in 1-bit units by setting pull-up resistor option register 3 (PU3).
Port 4	P40 to P43	I/O port. Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be used in 1-bit units by setting pull-up resistor option register 4 (PU4). ★ Sets the interrupt request flag (KRIF) to 1 by detecting a rising edge.
Port 7	P70 to P73	Middle-voltage N-ch open-drain I/O port. Input/output mode can be specified in 1-bit units. Mask ROM version can specify an on-chip pull-up resistor in 1-bit units by mask option.
Port 8[Note]	P80 to P87	I/O port. (Only pin whose I/O port is selected by pin function switching register 8 (PF8))
Port 9[Note]	P90 to P97	I/O port. (Only pin whose I/O port is selected by pin function switching register 9 (PF9))
Port 10[Note]	P100 to P107	I/O port. (Only pin whose I/O port is selected by pin function switching register 10 (PF10))
Port 11[Note]	P110 to P113	I/O port. (Only pin whose I/O port is selected by pin function switching register 11 (PF11))

Note Whether these pins are used as I/O port pins or segment output pins can be selected in 1-bit units by using the pin function switching register.

4.2 Port Configuration

A port consists of the following hardware.

Table 4-2. Port Configuration

Item	Configuration
Control registers	Port mode register (PMm: m = 0, 2 to 4, 8 to 11) Pull-up resistor option register (PUm: m = 0, 2 to 4) Memory expansion register (MEM) Pin function switching registers (PF8 to PF11)
Port	Total: 66 (input: 8, I/O: 58)
Pull-up resistors	• μPD780343, 780344, 780353, 780354 Total: 30 (software control: 24, mask option: 6) • μPD780343Y, 780344Y, 780353Y, 780354Y Total: 28 (software control: 24, mask option: 4) • μPD78F0354, 78F0354A, 78F0354Y, 78F0354AY Total: 24 (software control: 24)

4.2.1 Port 0

Port 0 is an 8-bit I/O port with an output latch. The P00 to P07 pins can be set to input mode/output mode in 1-bit units using port mode register 0 (PM0). An on-chip pull-up resistor can be used for the P00 to P07 pins in 1-bit units using pull-up resistor option register 0 (PU0).

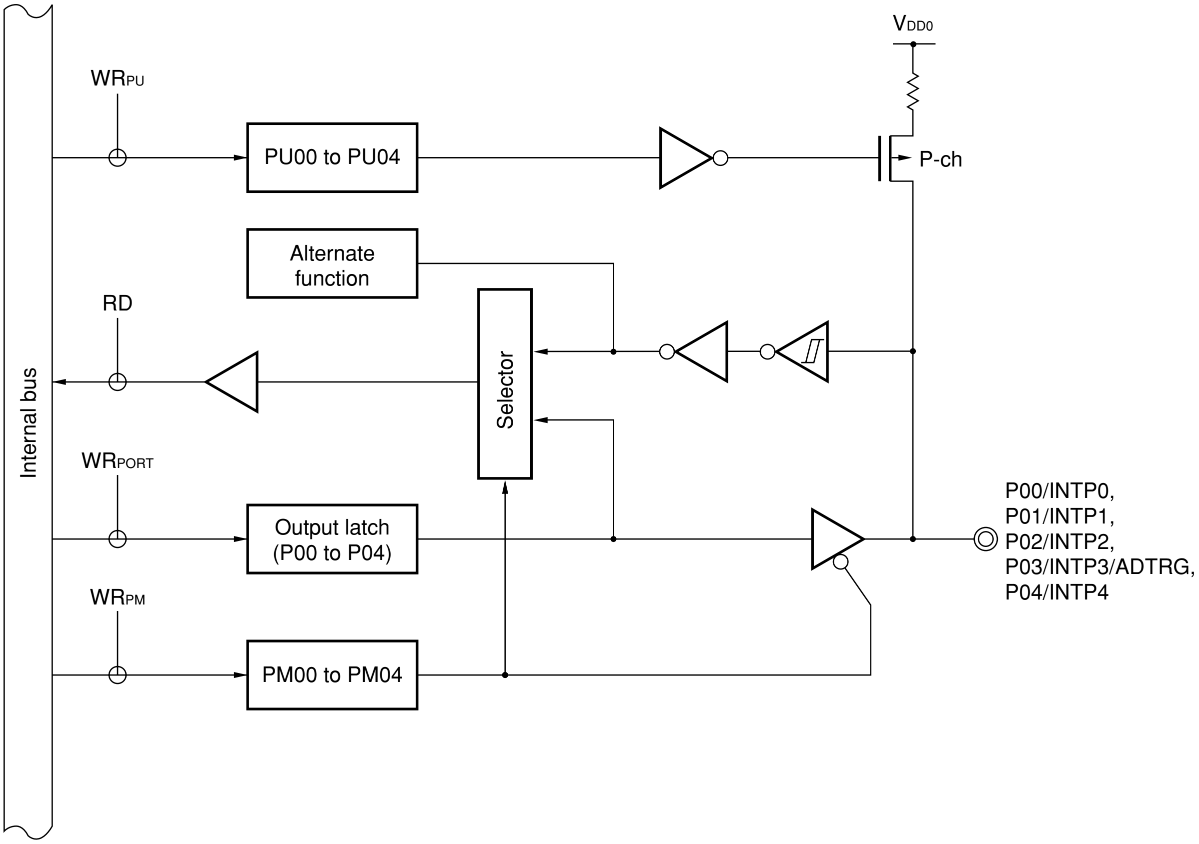
This port can also be used for external interrupt request input, A/D converter external trigger input, clock output, and timer I/O.

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

Figures 4-2 and 4-3 show block diagrams of port 0.

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

Figure 4-2. Block Diagram of P00 to P04



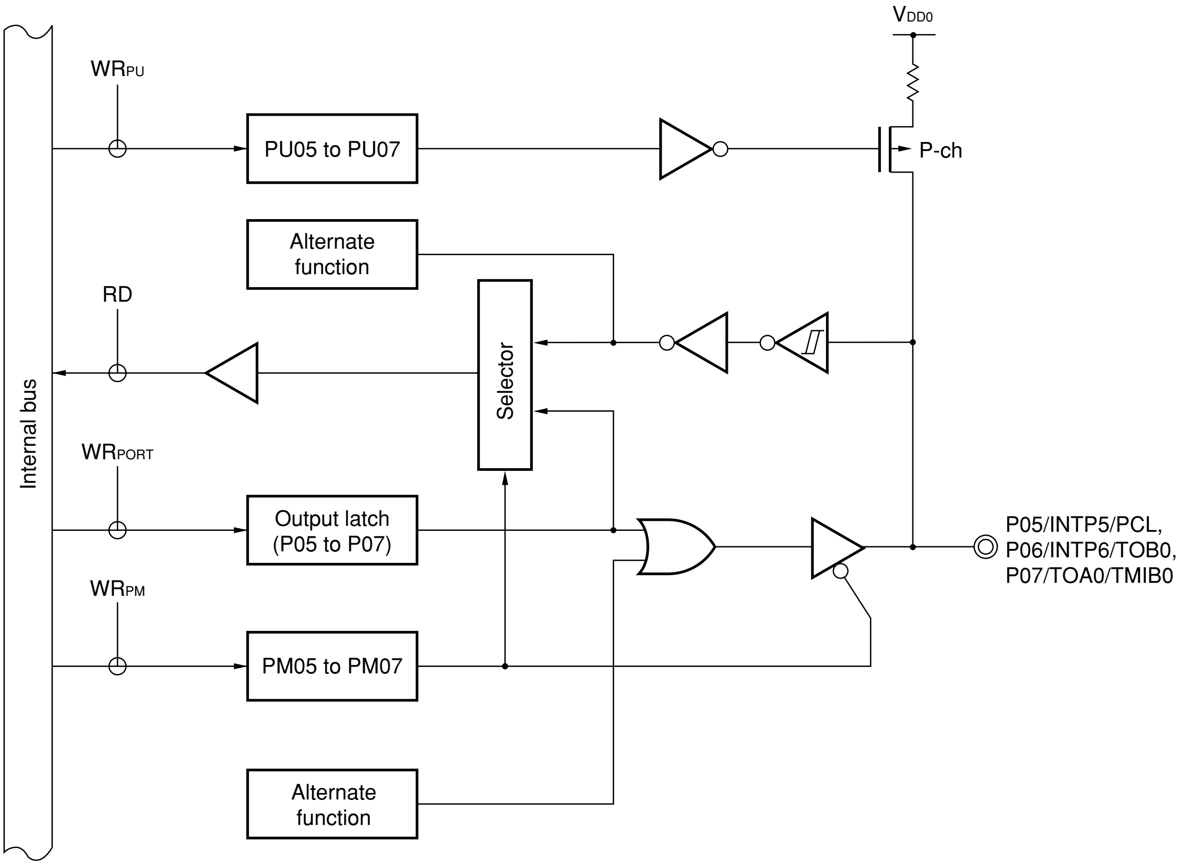
PU: Pull-up resistor option register

PM: Port mode register

RD: Port 0 read signal

WR: Port 0 write signal

Figure 4-3. Block Diagram of P05 to P07



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

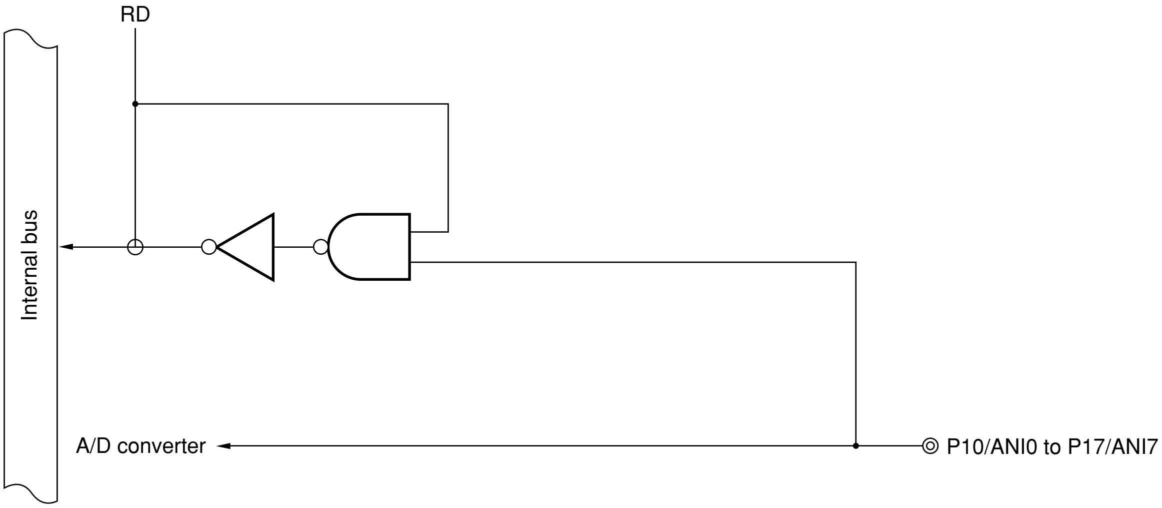
4.2.2 Port 1

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

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

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

★ **Figure 4-4. Block Diagram of P10 to P17**



RD: Port 1 read signal

4.2.3 Port 2

Port 2 is an 8-bit I/O port with an output latch. The P20 to P27 pins can be set to input mode/output mode in 1-bit units using port mode register 2 (PM2). An on-chip pull-up resistor can be used for the P20 to P27 pins in 1-bit units using pull-up resistor option register 2 (PU2).

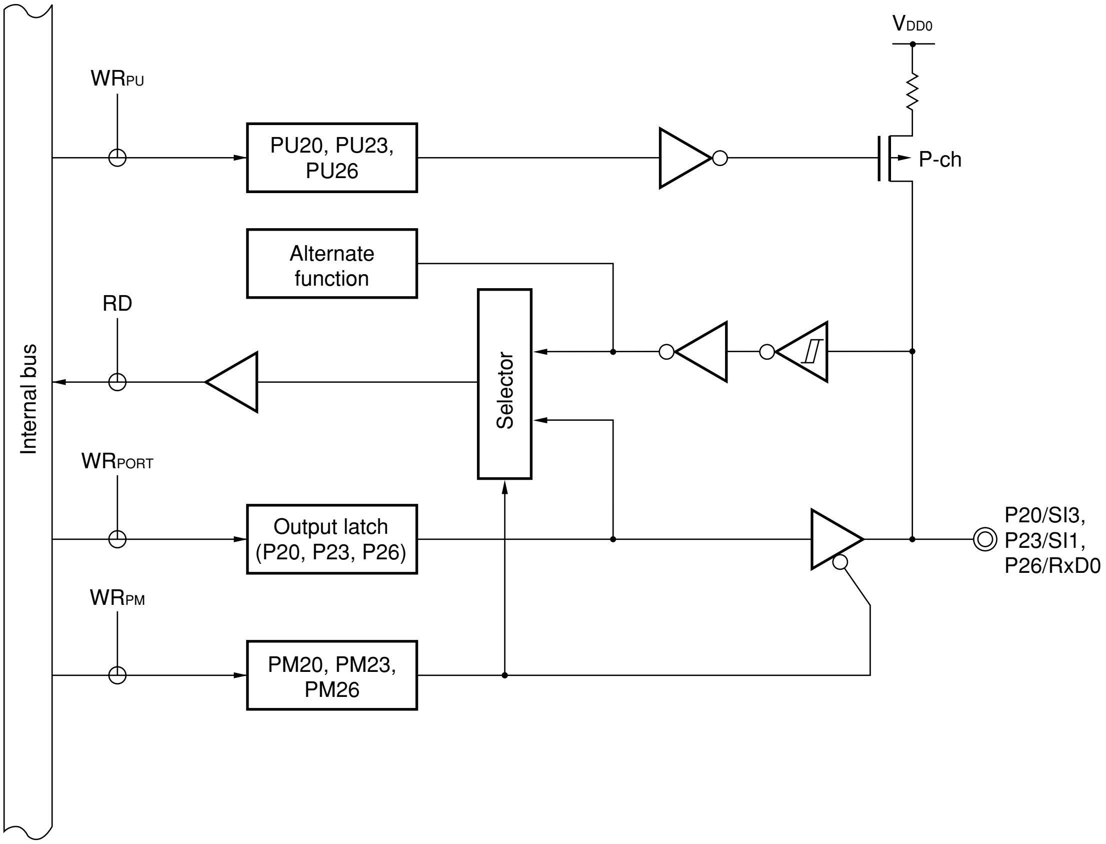
This port can also be used for serial interface data I/O and clock I/O.

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

Figures 4-5 to 4-7 show block diagrams of port 2.

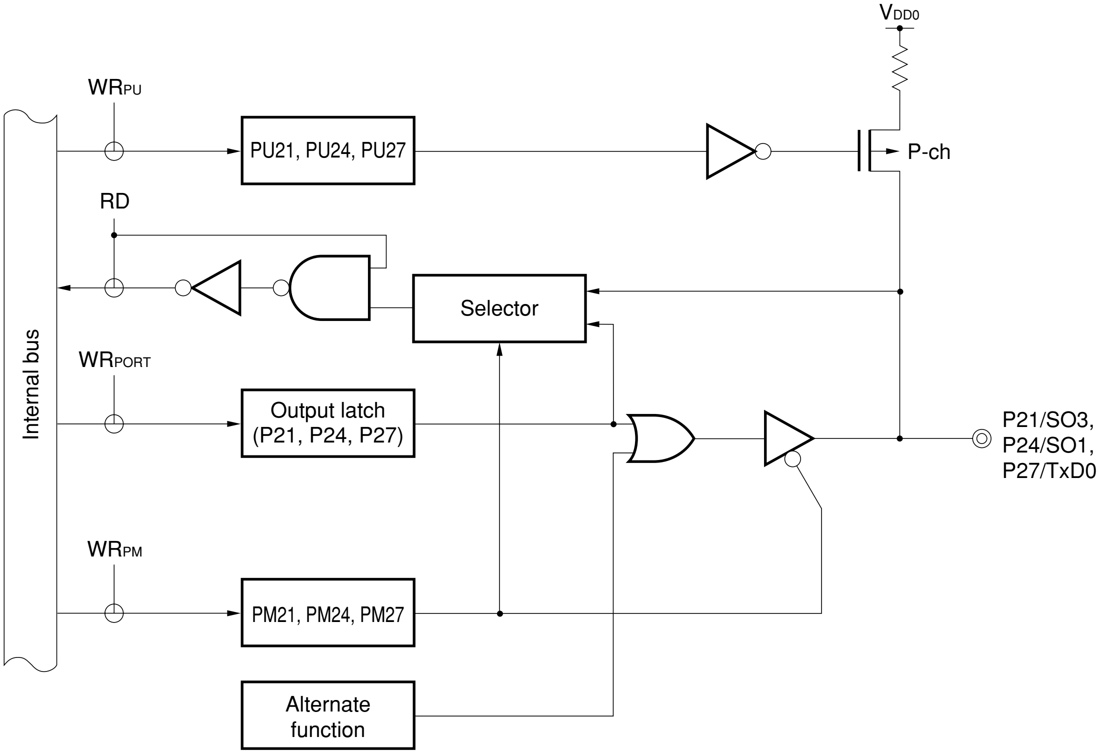
★ **Caution When P23/SI1, P24/SO1, and P25/SCK1 are used as general-purpose ports, do not write to the serial clock select register (CSIC1).**

Figure 4-5. Block Diagram of P20, P23, P26



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

Figure 4-6. Block Diagram of P21, P24, P27



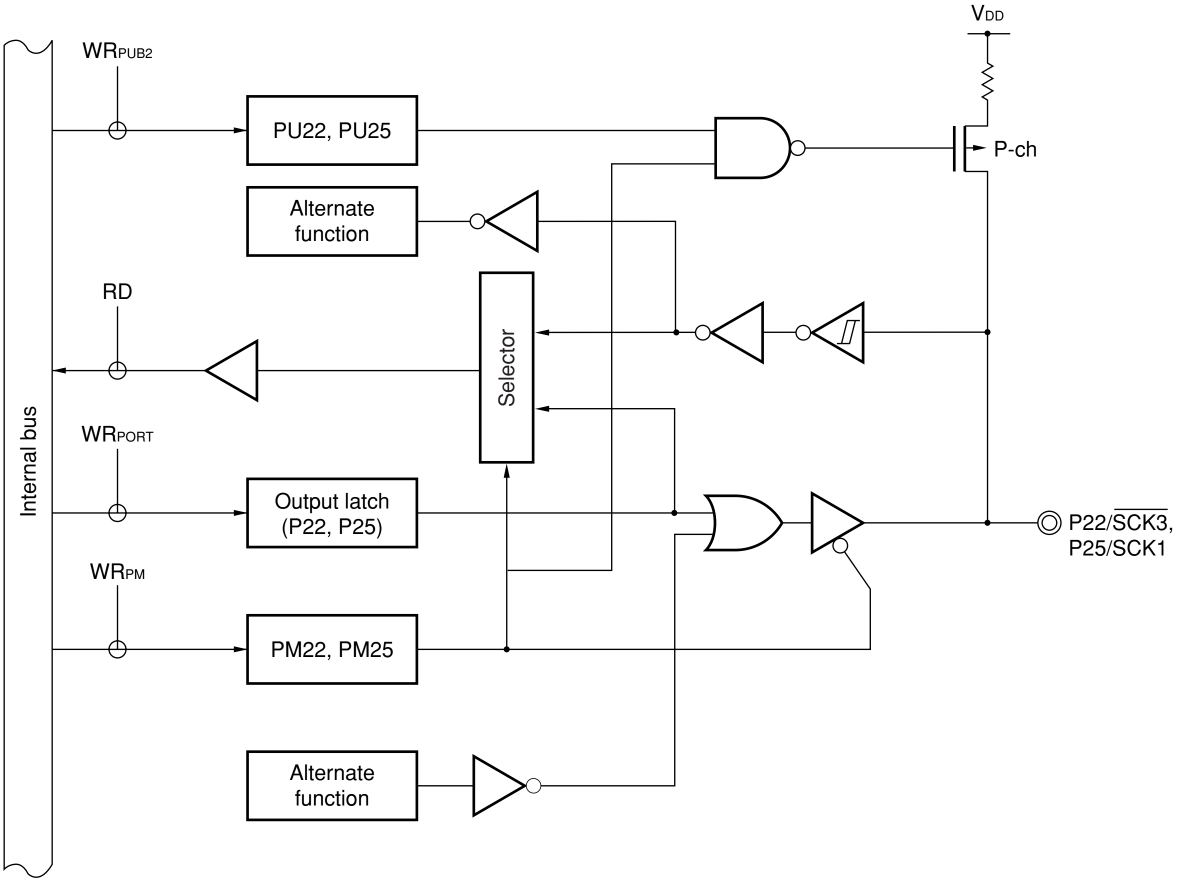
PU: Pull-up resistor option register

PM: Port mode register

RD: Port 2 read signal

WR: Port 2 write signal

Figure 4-7. Block Diagram of P22, P25



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

4.2.4 Port 3

Port 3 is a 6-bit I/O port with an output latch. The P30 to P35 pins can be set to input mode/output mode in 1-bit units using port mode register 3 (PM3).

The P30 and P31 pins are 5 V N-ch open-drain. These pins can drive LEDs directly. In the μPD780344Y, 780354Y Subseries, these pins have alternate functions as serial interface data I/O and clock I/O.

An on-chip pull-up resistor can be used for the P32 to P35 pins in 1-bit units using pull-up resistor option register 3 (PU3). These pins can also be used for timer I/O.

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

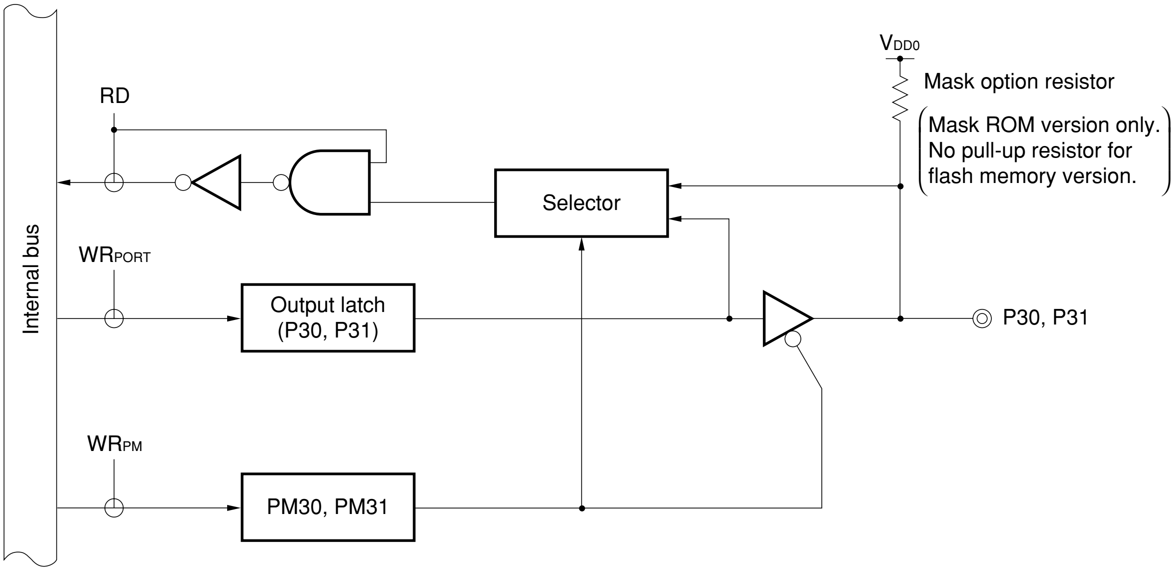
Table 4-3 lists the port 3 pin function of each product and Figures 4-8 to 4-10 show block diagrams of port 3.

Table 4-3. Port 3 Function of Each Product

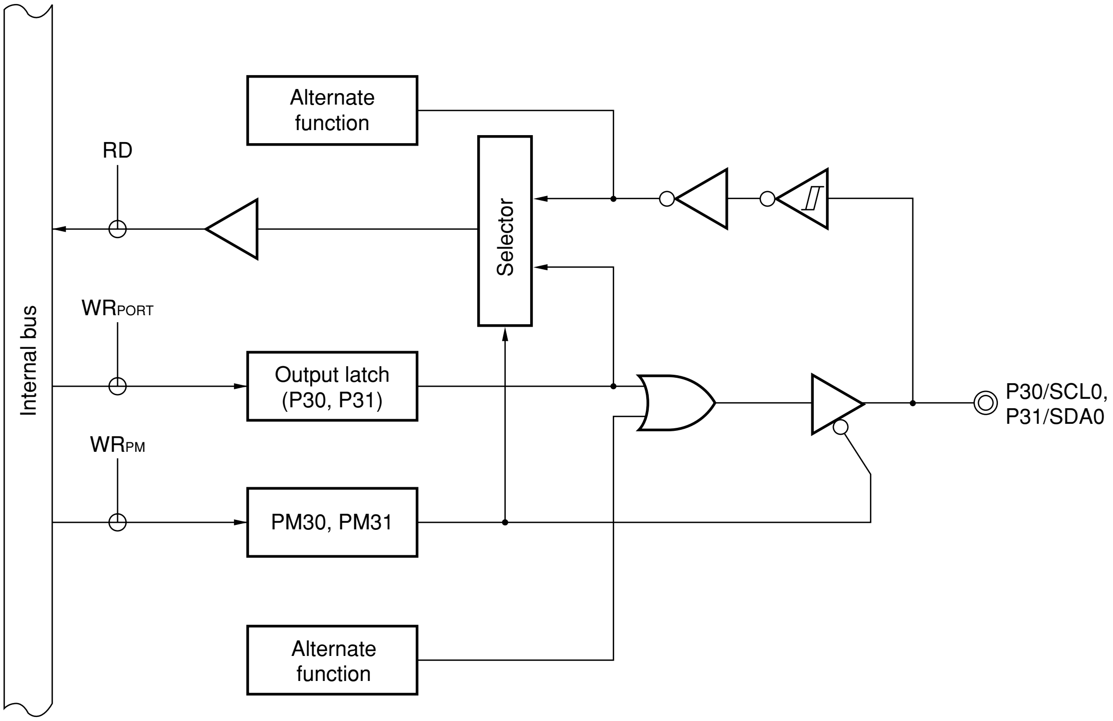
<table>
<tr><th>Product Name
Pin Name</th><th>μPD780343, 780344,
780353, 780354</th><th>μPD78F0354,
78F0354A</th><th>μPD780343Y, 780344Y,
780353Y, 780354Y</th><th>μPD78F0354Y,
78F0354AY</th></tr>
<tr><td rowspan="3">P30, P31</td><td colspan="4">• N-ch open-drain I/O
• Can directly drive LED.</td></tr>
<tr><td colspan="2">No alternate function</td><td colspan="2">Function alternately as data I/O of serial interface and clock I/O</td></tr>
<tr><td>Pull-up resistor can be connected in 1-bit units by mask option.</td><td colspan="3">No pull-up resistor</td></tr>
<tr><td>P32 to P35</td><td colspan="4">• CMOS I/O
• Function alternately as timer I/O
• Pull-up resistor can be connected in 1-bit units by PU3.</td></tr>
</table>

Figure 4-8. Block Diagram of P30, P31

(a) μPD780344, 780354 Subseries

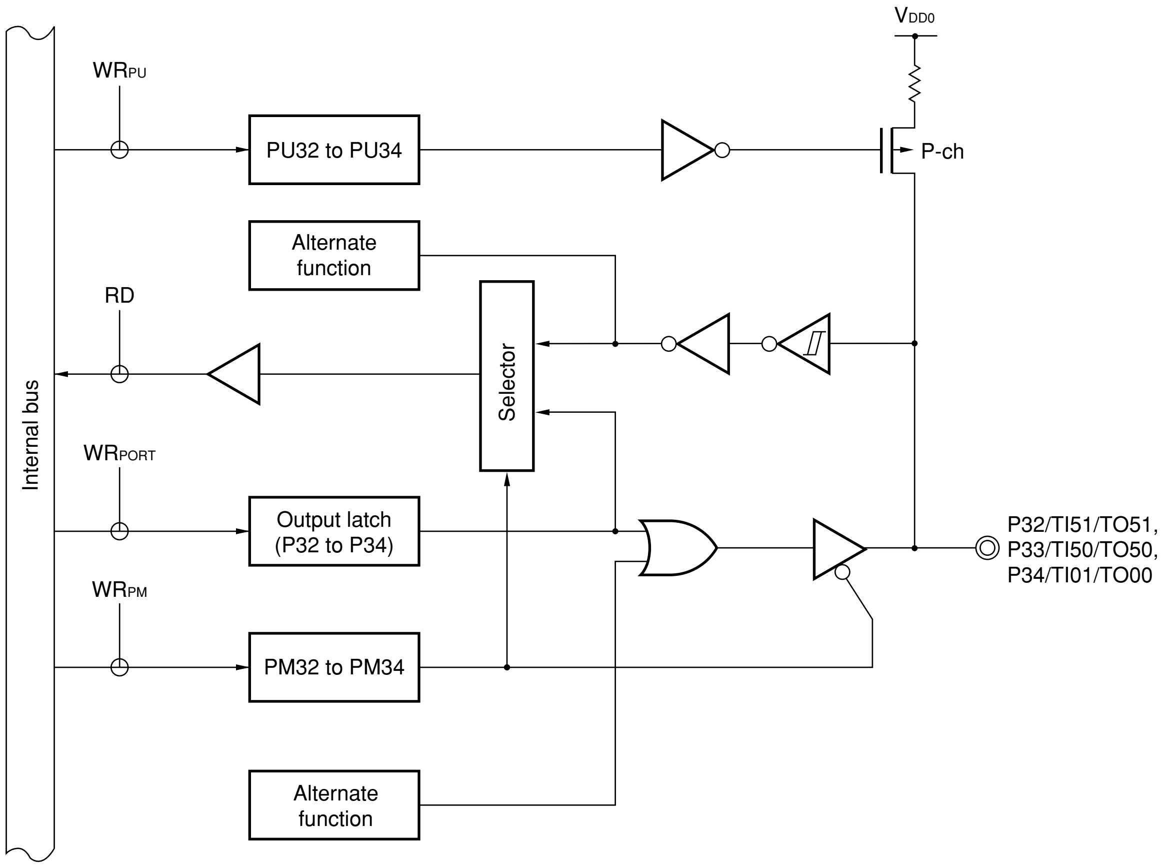


(b) μPD780344Y, 780354Y Subseries



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

Figure 4-9. Block Diagram of P32 to P34



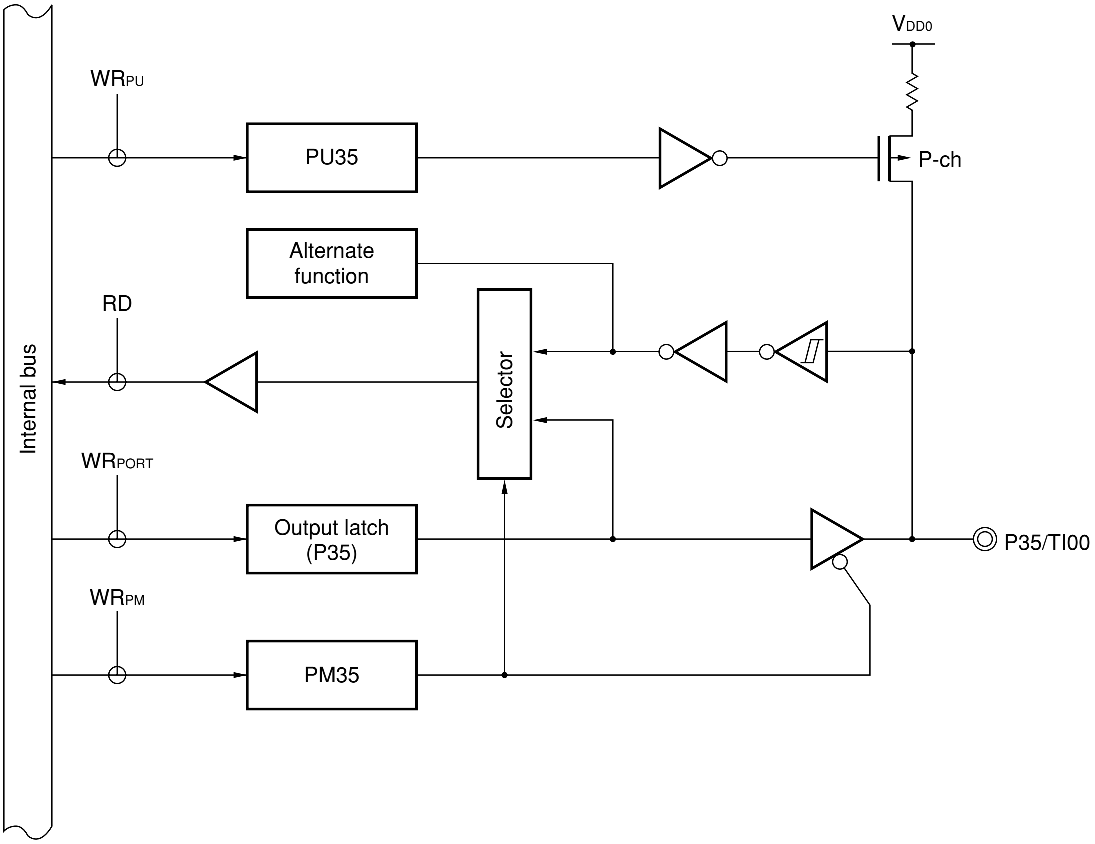
PU: Pull-up resistor option register

PM: Port mode register

RD: Port 3 read signal

WR: Port 3 write signal

Figure 4-10. Block Diagram of P35



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

4.2.5 Port 4

Port 4 is a 4-bit I/O port with an output latch. The P40 to P43 pins can be set to input mode/output mode in 1-bit units using port mode register 4 (PM4).

At the falling edge of any of the P40 to P43 pins, the interrupt request flag (KRIF) can be set to 1.

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

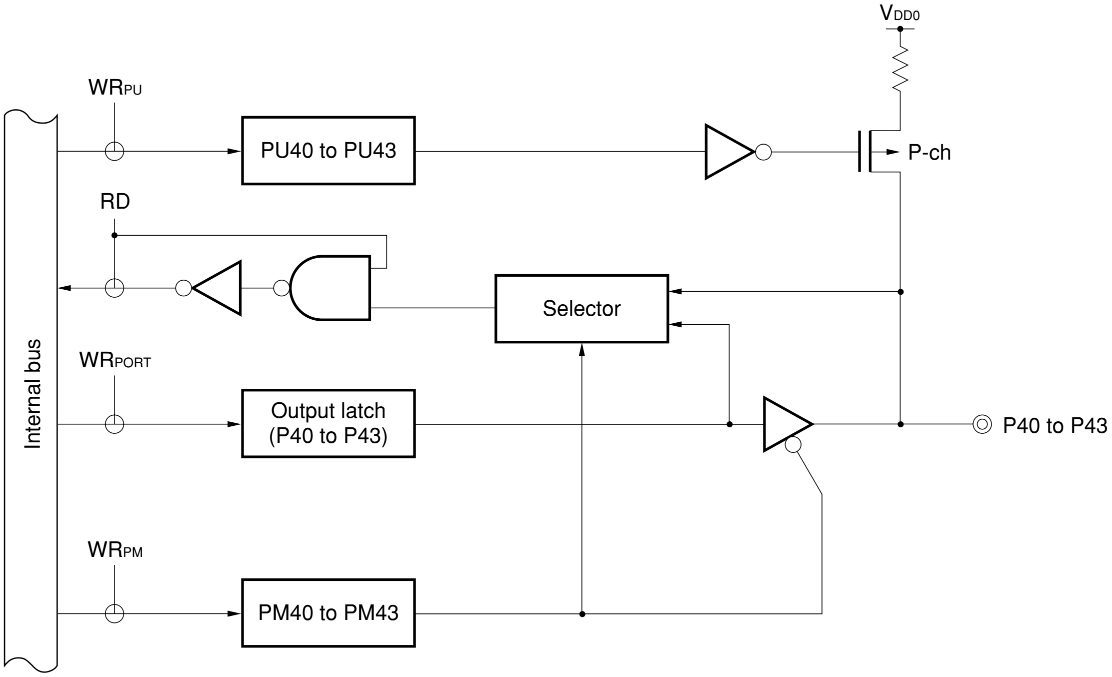
Figure 4-11 shows a block diagram of port 4 and Figure 4-12 shows a block diagram of the falling edge detector, respectively.

Cautions 1. When using the falling edge detection interrupt (INTKR), be sure to set the memory expansion mode register (MEM) to 01H.

2. The falling edge can be detected only when a falling edge occurs while all the P40 to P43 pins are high.

The falling edge of another pin cannot be detected while even one of the P40 to P43 pins is low.

Figure 4-11. Block Diagram of P40 to P43



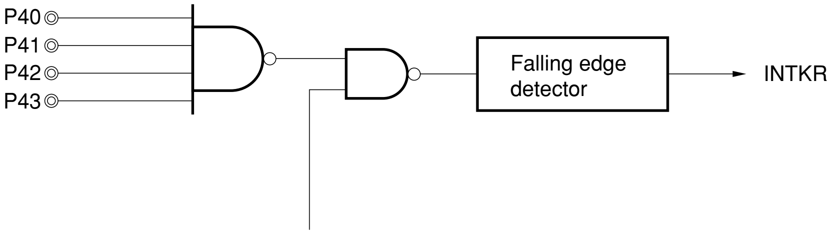
PU: Pull-up resistor option register

PM: Port mode register

RD: Port 4 read signal

WR: Port 4 write signal

Figure 4-12. Block Diagram of Falling Edge Detector



“1” when MEM = 01H

4.2.6 Port 7

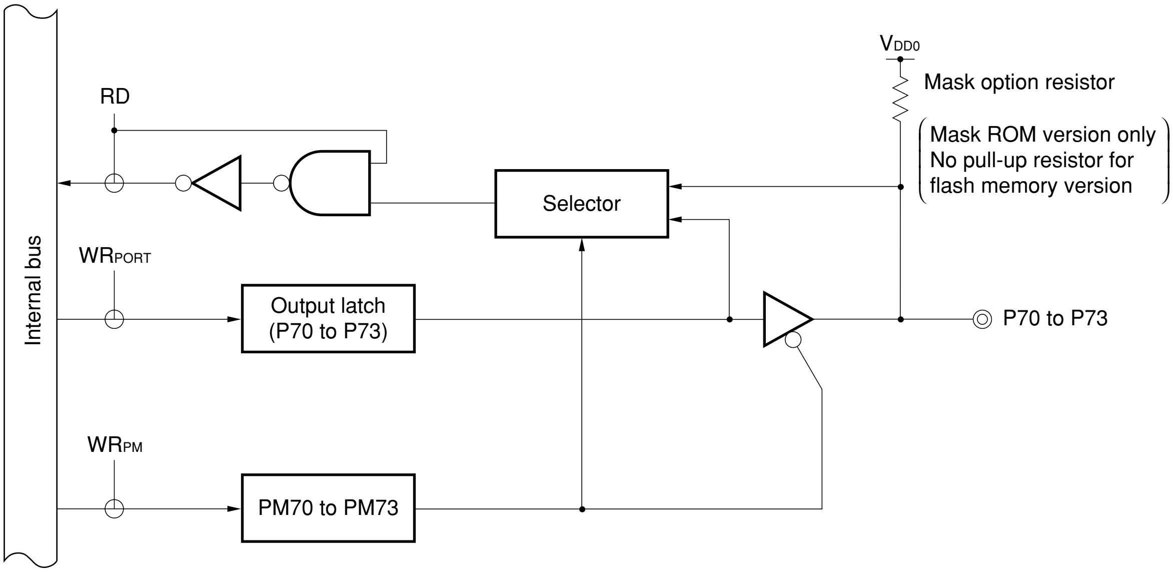
Port 7 is a 4-bit I/O port with an output latch. The P70 to P73 pins can be set to input mode/output mode in 1-bit units using port mode register 7 (PM7). In mask ROM versions use of a pull-up resistor can be set by mask option.

The P70 to P73 pins can drive LEDs directly.

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

Figure 4-13 shows a block diagram of port 7.

Figure 4-13. Block Diagram of P70 to P73



PM: Port mode register

RD: Port 7 read signal

WR: Port 7 write signal

4.2.7 Port 8

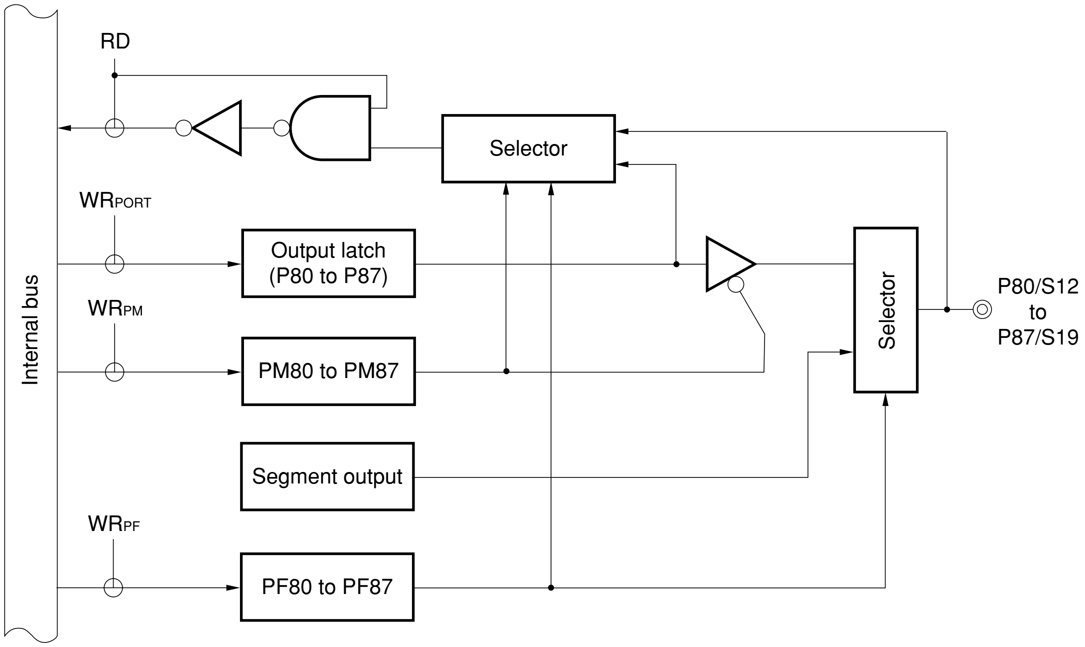
Port 8 is an 8-bit I/O port with an output latch. The P80 to P87 pins can be set to input mode/output mode in 1-bit units using port mode register 8 (PM8).

This port can also be used for LCD controller/driver segment output. This port can be switched between an I/O port and a segment output port in 1-bit units by pin function switching register 8 (PF8).

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

Figure 4-14 shows a block diagram of port 8.

Figure 4-14. Block Diagram of P80 to P87



PF: Pin function switching register
PM: Port mode register
RD: Port 8 read signal
WR: Port 8 write signal

4.2.8 Port 9

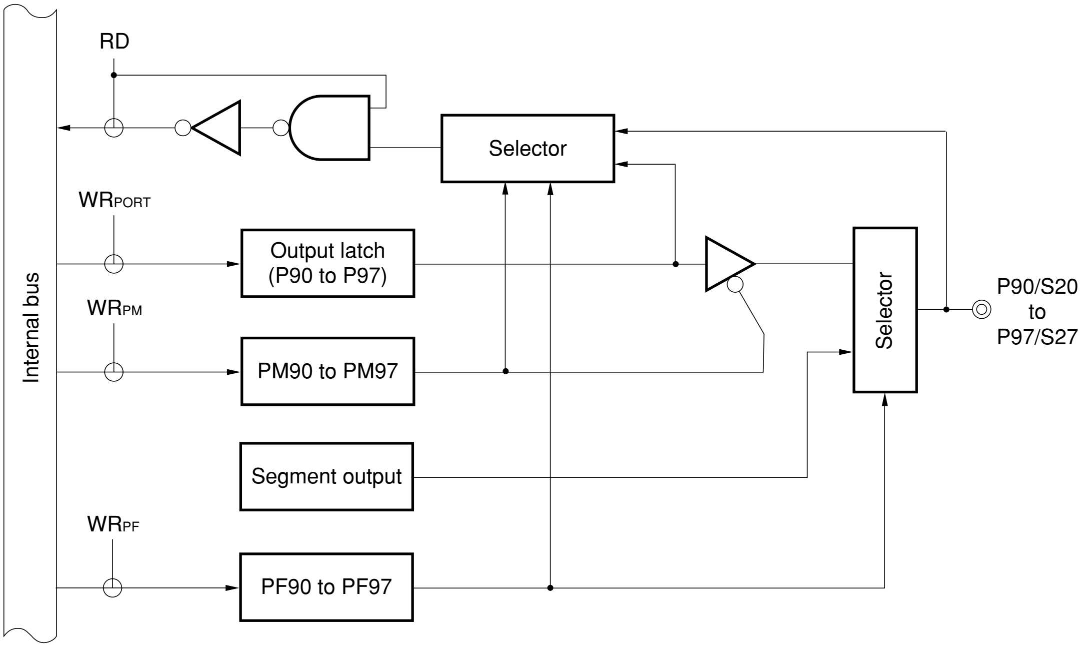
Port 9 is an 8-bit I/O port with an output latch. The P90 to P97 pins can be set to input mode/output mode in 1-bit units using port mode register 9 (PM9).

This port can also be used for LCD controller/driver segment output. This port can be switched between an I/O port and a segment output port in 1-bit units by pin function switching register 9 (PF9).

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

Figure 4-15 shows a block diagram of port 9.

Figure 4-15. Block Diagram of P90 to P97



PF:	Pin function switching register
PM:	Port mode register
RD:	Port 9 read signal
WR:	Port 9 write signal

4.2.9 Port 10

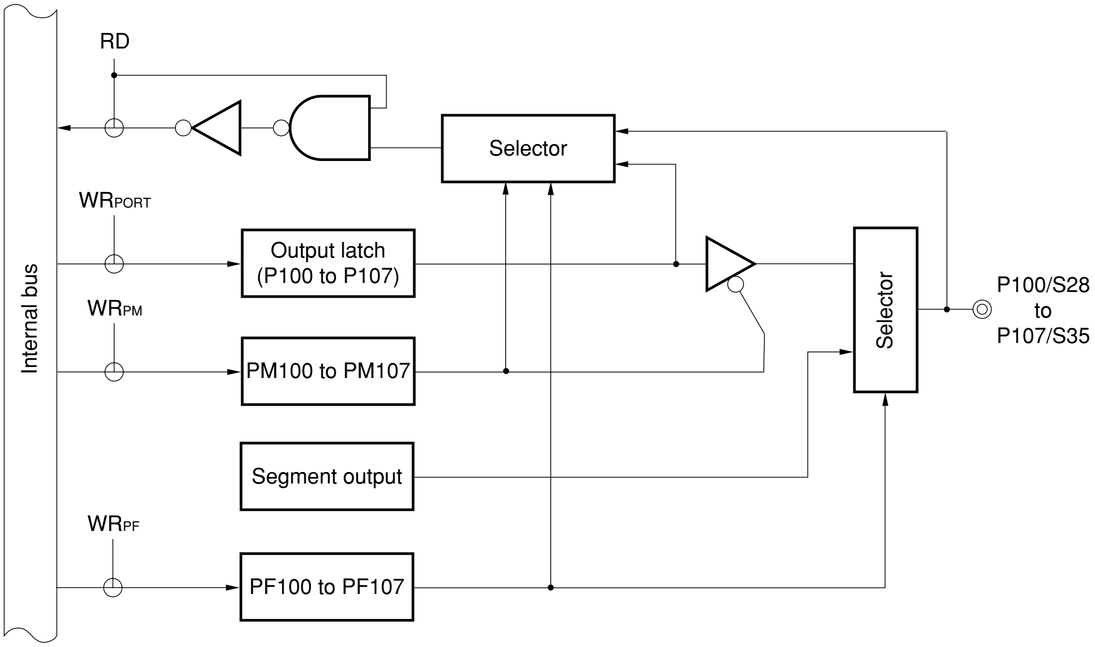
Port 10 is an 8-bit I/O port with an output latch. The P100 to P107 pins can be set to input mode/output mode in 1-bit units using port mode register 10 (PM10).

This port can also be used for LCD controller/driver segment output. This port can switched between an I/O port and a segment output port in 1-bit units by pin function switching register 10 (PF10).

$\overline{\text{RESET}}$ input sets port 10 to input mode.

Figure 4-16 shows a block diagram of port 10.

Figure 4-16. Block Diagram of P100 to P107



PF: Pin function switching register
PM: Port mode register
RD: Port 10 read signal
WR: Port 10 write signal

4.2.10 Port 11

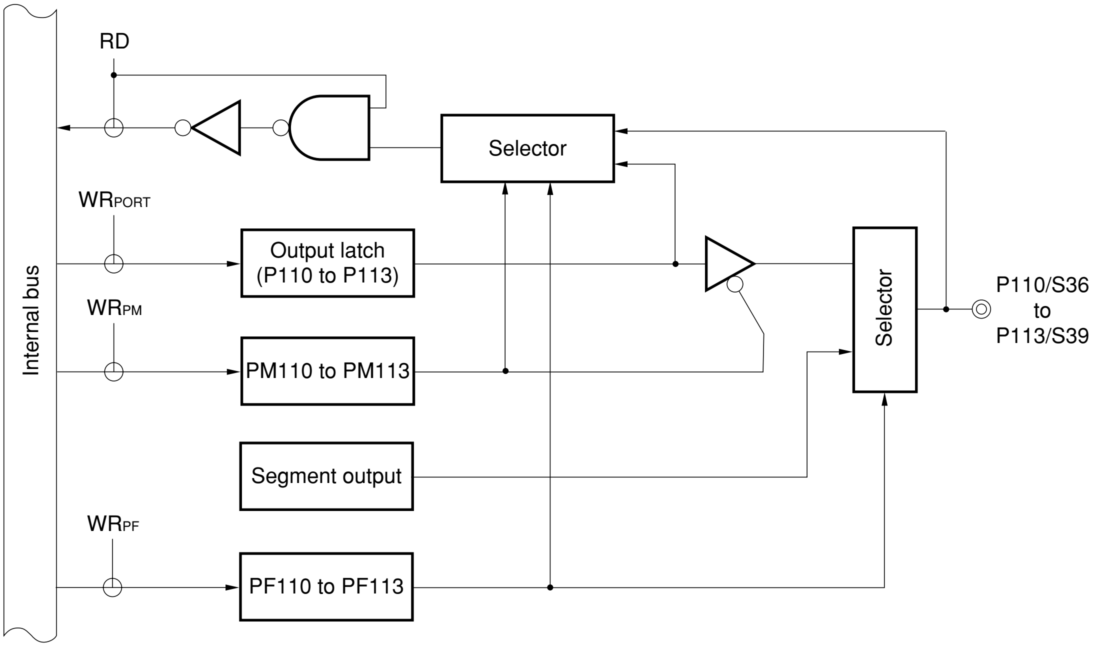
Port 11 is a 4-bit I/O port with an output latch. P110 to P113 pins can be set to input mode/output mode in 1-bit units using port mode register 11 (PM11).

This port can also be used for LCD controller/driver segment output. This port can be switched between an I/O port and a segment output port in 1-bit units by pin function switching register 11 (PF11).

$\overline{\text{RESET}}$ input sets port 11 to input mode.

Figure 4-17 shows a block diagram of port 11.

Figure 4-17. Block Diagram of P110 to P113



PF: Pin function switching register
PM: Port mode register
RD: Port 11 read signal
WR: Port 11 write signal

4.3 Port Function Control Registers

The following four types of registers control the ports.

- Port mode registers (PM0, PM2 to PM4, PM7 to PM11)
- Pull-up resistor option registers (PU0, PU2 to PU4)
- Memory expansion register (MEM)
- Pin function switching registers (PF8 to PF11)

(1) Port mode registers (PM0, PM2 to PM4, PM7 to PM11)

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

PM0, PM2 to PM4, and PM7 to PM11 are independently set by a 1-bit or 8-bit memory manipulation instruction.

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

Cautions 1. The P10 to P17 pins are input-only pins.

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

3. If a pin of ports 0, 2, and 3 is used as an alternate output function, set the output latches (P0, P2, and P3) to 0.

Figure 4-18. Format of Port Mode Registers (PM0, PM2 to PM4, PM7 to PM11)

Address: FF20H After reset: FFH R/W

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

Address: FF22H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20

Address: FF23H After reset: FFH R/W

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

Address: FF24H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM4	1	1	1	1	PM43	PM42	PM41	PM40

Address: FF27H After reset: FFH R/W

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

Address: FF28H After reset: FFH R/W

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

Address: FF29H After reset: FFH R/W

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

Address: FF2AH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM10^{Note}	PM107	PM106	PM105	PM104	PM103	PM102	PM101	PM100

Address: FF2BH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM11^{Note}	1	1	1	1	PM113	PM112	PM111	PM110

PMmn	Pmn pin input/output mode selection (m = 0, 2 to 4, 7 to 11: n = 0 to 7)
0	Output mode (Output buffer ON)
1	Input mode (Output buffer OFF)

★ **Note** When ports 8 to 11 are used as port pins, set the corresponding bits of the pin function switching registers (PF8 to PF11) to 0.

(2) Pull-up resistor option registers (PU0, PU2 to PU4)

These registers are used to set whether to use an on-chip pull-up resistor at each port or not in 1-bit units. By setting PU0 and PU2 to PU4, the on-chip pull-up resistors of the port pins corresponding to the bits in PU0 and PU2 to PU4 can be used.

PU0 and PU2 to PU4 are set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the values of these registers to 00H.

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

2. The P70 to P73 pins can be used with a pull-up resistor by setting a mask option only for mask ROM versions.

3. The P30 and P31 pins can be used with a pull-up resistor by setting a mask option only for mask ROM versions of the μPD780344, 780354 Subseries.

4. When PUm is set to 1, an on-chip pull-up resistor is connected irrespective of the input/output mode. When using the port in output mode, therefore, set the corresponding bits of PUm to 0 (m = 0, 2 to 4).

Figure 4-19. Format of Pull-Up Resistor Option Registers (PU0, PU2 to PU4)

Address: FF30H After reset: 00H R/W

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

Address: FF32H After reset: 00H R/W

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

Address: FF33H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU3	0	0	PU35	PU34	PU33	PU32	0	0

Address: FF34H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU4	0	0	0	0	PU43	PU42	PU41	PU40

PUmn	Pmn pin internal pull-up resistor selection (m = 0, 2 to 4: n = 0 to 7)
0	On-chip pull-up resistor not used
1	On-chip pull-up resistor used

(3) Memory expansion mode register (MEM)

This register is used to set whether port 4 is used as port pins or key input pins.

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

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

Figure 4-20. Format of Memory Expansion Mode Register (MEM)

Address: FF47H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MEM	0	0	0	0	0	MM2	MM1	MM0

MM2	MM1	MM0	Single-chip/key return mode selection
0	0	0	Single-chip mode (used as port pin)
0	0	1	Key return mode (used as key input pin)
Other than above			Setting prohibited

Caution Be sure to set MM1 and MM2 to 0.

(4) Pin function switching registers (PF8 to PF11)

These registers are used to select if ports 8 to 11 are used as port pins or segment pins in 1-bit units.

PF8 to PF11 are set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the values of these registers to 00H.

Figure 4-21. Format of Pin Function Switching Registers (PF8 to PF11)

Address: FF58H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PF8	PF87	PF86	PF85	PF84	PF83	PF82	PF81	PF80

Address: FF59H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PF9	PF97	PF96	PF95	PF94	PF93	PF92	PF91	PF90

Address: FF5AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PF10	PF107	PF106	PF105	PF104	PF103	PF102	PF101	PF100

Address: FF5BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PF11	0	0	0	0	PF113	PF112	PF111	PF110

PFmn	Pin settings
0	★ I/O port
1	★ Segment output

Cautions 1. The pins specified as segment output pins by PF8 to PF11 can output their signals regardless of the value of the corresponding port mode register (PM8 to PM11).

★ **2. PF8 to PF11 can be set only once after a reset. To change the settings, a reset must be performed beforehand.**

Remark m = 8 to 11, n = 0 to 7

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.

★ The output latch is cleared after a reset.

(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.

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.

★ The output latch is cleared after a reset.

(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.

4.5 Selection of Mask Option

The following mask option is provided in the mask ROM version. The flash memory versions have no mask options. Note that the mask option differs between the μPD780344 and 780354 Subseries, and the μPD780344Y and 780354Y Subseries.

Table 4-4. Comparison Between Mask ROM Version and Flash Memory Version

Pin Name	Mask ROM Version of μPD780344, 780354 Subseries	Mask ROM Version of μPD780344Y, 780354Y Subseries	Flash Memory Version
P30, P31 pins	On-chip pull-up resistors can be specified in 1-bit units	Cannot specify an on-chip pull-up resistor	Cannot specify an on-chip pull-up resistor
P70 to P73 pins	On-chip pull-up resistors can be specified in 1-bit units	On-chip pull-up resistors can be specified in 1-bit units	Cannot specify an on-chip pull-up resistor

CHAPTER 5 CLOCK GENERATOR

5.1 Clock Generator Functions

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

(1) Main system clock oscillator

★ This circuit oscillates at frequencies of 2 to 10 MHz. Oscillation can be stopped by executing the STOP instruction or setting the processor clock control register (PCC).

(2) Subsystem clock oscillator

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

The ×4 subclock circuit can also be selected by the subclock select register (SSCK).

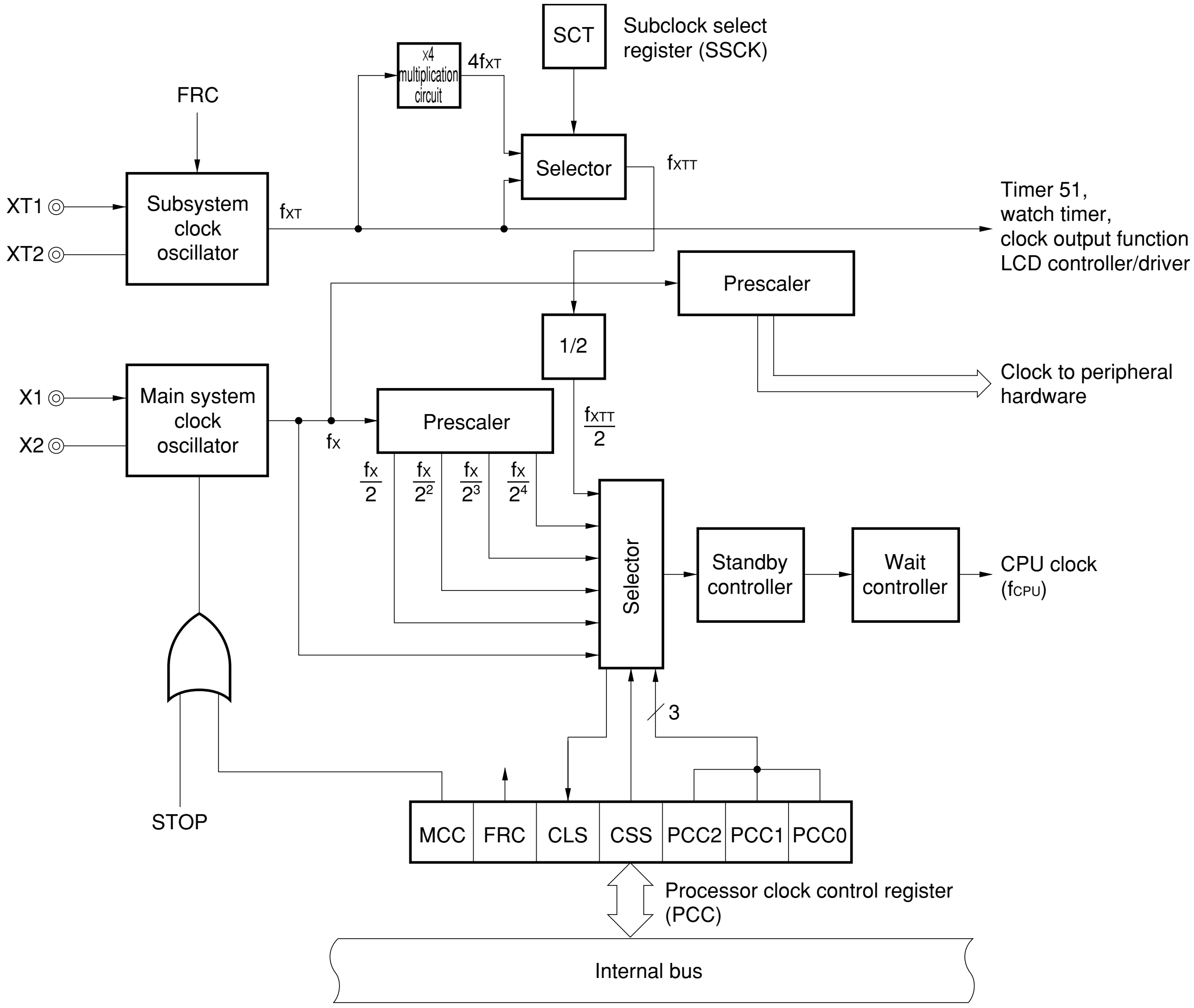
5.2 Clock Generator Configuration

The clock generator consists of the following hardware.

Table 5-1. Clock Generator Configuration

Item	Configuration
Control registers	Processor clock control register (PCC) Subclock select register (SSCK)
Oscillators	Main system clock oscillator Subsystem clock oscillator

Figure 5-1. Clock Generator Block Diagram


SCT
Subclock select register (SSCK)
x4 multiplication circuit
4fXT
FRC
Selector
fXTT
XT1
XT2
Subsystem clock oscillator
fXT
Timer 51, watch timer, clock output function LCD controller/driver
Prescaler
1/2
Clock to peripheral hardware
X1
X2
Main system clock oscillator
Prescaler
fX
fXTT/2
fX/2
fX/2^2
fX/2^3
fX/2^4
Selector
Standby controller
Wait controller
CPU clock (fCPU)
3
STOP
MCC
FRC
CLS
CSS
PCC2
PCC1
PCC0
Processor clock control register (PCC)
Internal bus

Remark f_{XTT}: f_{XT} or $4f_{XT}$

5.3 Clock Generator Control Registers

The clock generator is controlled by the following two registers.

- Processor clock control register (PCC)
- Subclock select register (SSCK)

(1) Processor clock control register (PCC)

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

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

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

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

★ **Note** The feedback resistor is necessary for adjusting the bias point of the oscillation waveform close to the medium level of the supply voltage. The current consumption in the STOP mode can be further suppressed by setting bit 6 (FRC) of PCC to 1 only when the subsystem clock is not used.

Figure 5-2. Subsystem Clock Feedback Resistor

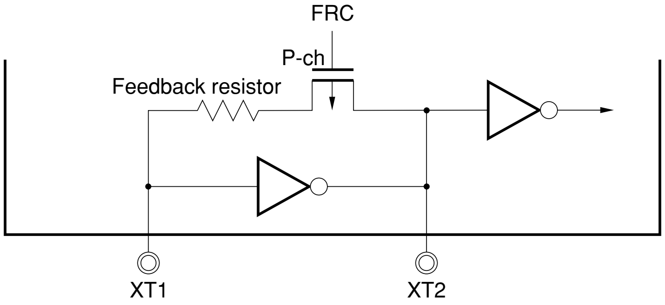


Figure 5-3. Format of Processor Clock Control Register (PCC)

Address: FFFBH After reset: 04H R/W[Note 1]

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

MCC	Main system clock oscillation control[Note 2]
0	Oscillation possible
1	Oscillation stopped

FRC	Subsystem clock feedback resistor selection
0	Internal feedback resistor used
1	Internal feedback resistor not used[Note 3]

CLS	CPU clock status
0	Main system clock
1	Subsystem clock

CSS	PCC2	PCC1	PCC0	CPU clock (f_{CPU}) selection
0	0	0	0	f_X
	0	0	1	$f_X/2$
	0	1	0	$f_X/2^2$
	0	1	1	$f_X/2^3$
	1	0	0	$f_X/2^4$
1	0	0	0	$f_{XT}/2$ $2f_{XT}$ (when ×4 circuit is used)
	0	0	1	
	0	1	0	
	0	1	1	
	1	0	0	
Other than above				Setting prohibited

Notes 1. Bit 5 is a read-only bit.

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

★ **3.** FRC can be set to 1 only when the subsystem clock is not used.

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

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

3. If the clock must be switched over between the subsystem clock being used in ×4 mode and the main system clock, be sure to set the CPU clock frequency when the main system clock is used to 280 kHz or more.

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

The fastest instructions of μPD780344, 780354, 780344Y, and 780354Y Subseries are carried out in two CPU clocks. The relationship of the CPU clock (f_{CPU}) and the minimum instruction execution time is shown in Table 5-2.

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

CPU Clock (f_{CPU})	Min. Instruction Execution Time: 2/(f_{CPU})
f_X	0.2 μs
$f_X/2$	0.4 μs
$f_X/2^2$	0.8 μs
$f_X/2^3$	1.6 μs
$f_X/2^4$	3.2 μs
$f_{XT}/2$	122 μs
$2f_{XT}$ (when ×4 circuit is used)	30.5 μs

f_X = 10 MHz, f_{XT} = 32.768 kHz

f_X: Main system clock oscillation frequency

f_{XT}: Subsystem clock oscillation frequency

(2) Subclock select register (SSCK)

This register is used to control the operation of the ×4 subsystem clock multiplication circuit.

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

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

Figure 5-4. Format of Subclock Select Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
SSCK	0	0	0	0	0	0	0	SCT	FF78H	Retained[Note]	R/W

SCT	Control of ×4 subsystem clock multiplication circuit
0	Operation stopped (subsystem clock source (32.768 kHz) supplied to the CPU)
1	Operation enabled (clock that is the subsystem clock multiplied by 4 (131 kHz) supplied to the CPU)

Note The register is set to 00H only by $\overline{\text{RESET}}$ input.

Cautions 1. Always set bits 1 to 7 to 0.

2. Write to the SCT flag prior to setting the CSS flag to 1 following the release of reset. Write operations following the first operation are invalid (input the $\overline{\text{RESET}}$ signal to rewrite).

3. The ×4 circuit is stopped during the HALT period to lower the power consumption, even while its operation is enabled by the SCT flag.
After the HALT mode has been released, the device waits for the duration of one source clock of the subsystem clock and then starts operating on the ×4 clock.

5.4 System Clock Oscillator

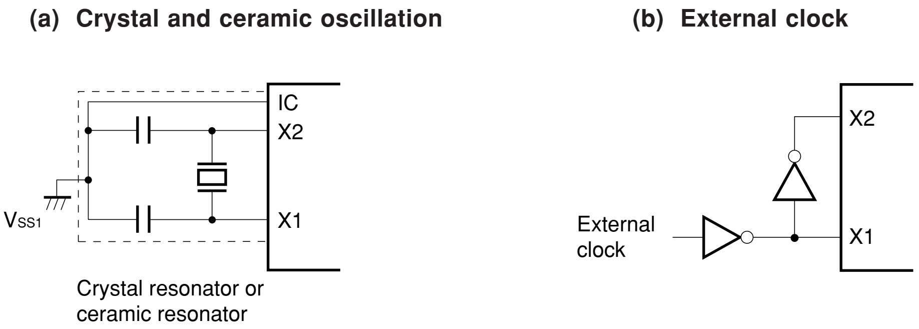
5.4.1 Main system clock oscillator

The main system clock oscillator oscillates with a crystal resonator or a ceramic resonator (10 MHz TYP.) connected to the X1 and X2 pins.

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

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

Figure 5-5. External Circuit of Main System Clock Oscillator



(a) Crystal and ceramic oscillation

(b) External clock

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

2. When using the main system clock oscillator and subsystem clock oscillator, wire as follows in the area enclosed by the broken lines in the Figures 5-5 and 5-6 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 and do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as V_{SS1}. Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

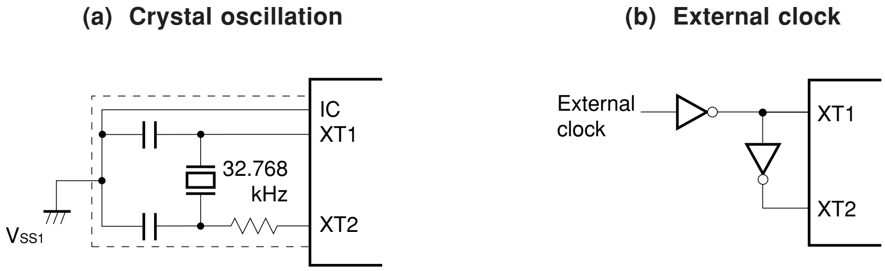
5.4.2 Subsystem clock oscillator

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

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

Figure 5-6 shows an external circuit of the subsystem clock oscillator.

Figure 5-6. External Circuit of Subsystem Clock Oscillator



(a) Crystal oscillation

(b) External clock

Caution When using the main system clock oscillator and subsystem clock oscillator, wire as follows in the area enclosed by the broken lines in the Figures 5-5 and 5-6 to avoid an adverse effect from wiring capacitance.

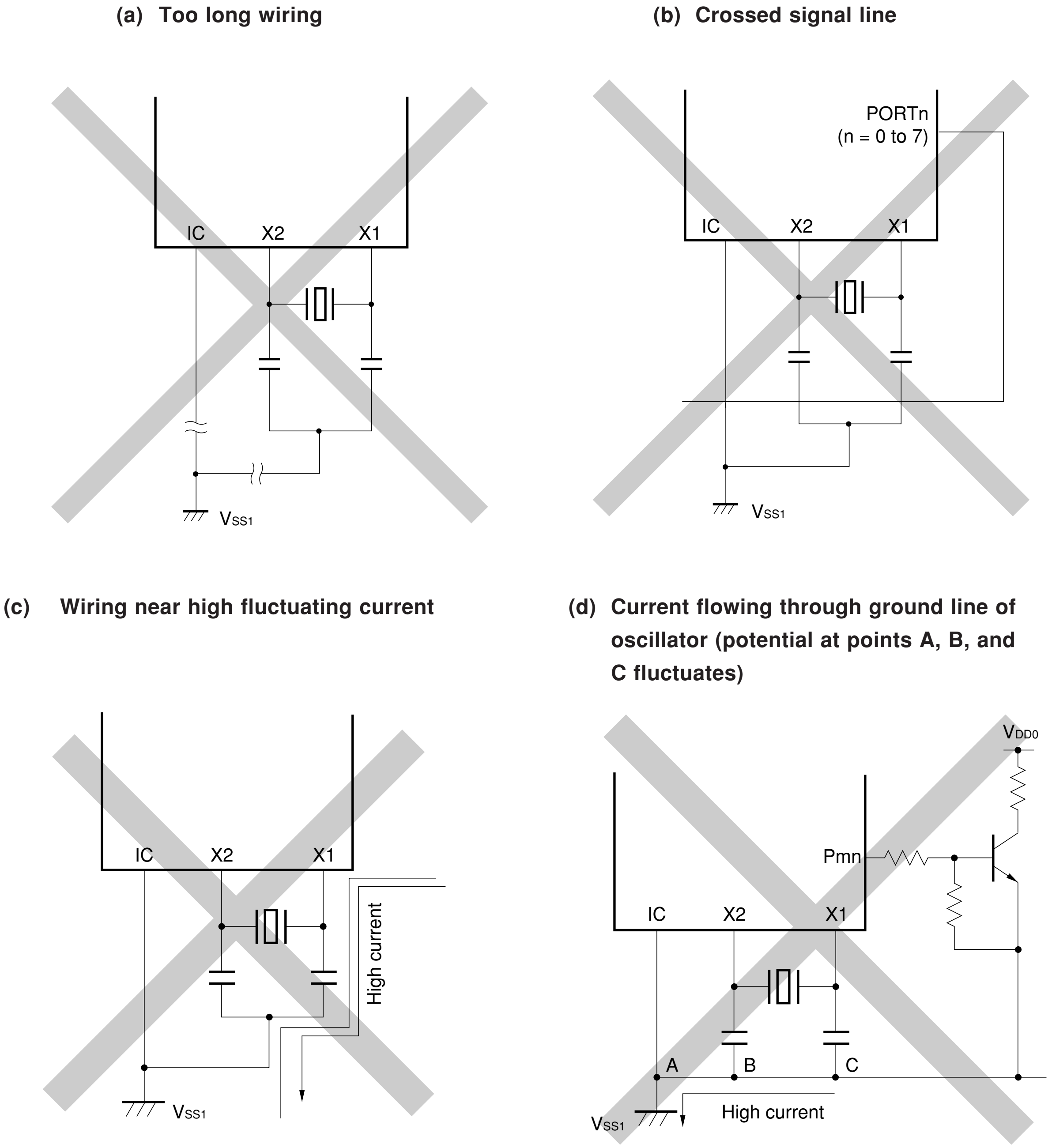
- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines and do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as V_{SS1}. Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

Note that the subsystem clock oscillator is designed as a low-amplitude circuit for reducing power consumption.

5.4.3 Examples of incorrect resonator connection

Figure 5-7 shows examples of incorrect resonator connection.

Figure 5-7. Examples of Incorrect Resonator Connection (1/2)



(a) Too long wiring

(b) Crossed signal line

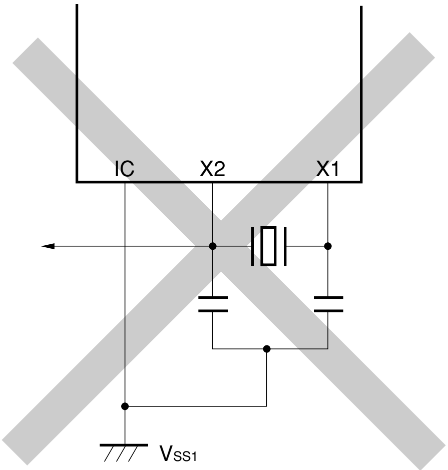
(c) Wiring near high fluctuating current

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

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

Figure 5-7. Examples of Incorrect Resonator Connection (2/2)

(e) Signals are fetched



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

Caution When X2 and XT1 are wired in parallel, the crosstalk noise of X2 may increase with XT1, resulting in malfunction.

To prevent that from occurring, it is recommended to wire X2 and XT1 so that they are not in parallel.

5.4.4 Divider

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

5.4.5 When no subsystem clocks are used

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

XT1: Connect to V_{DD0} or V_{DD1}
XT2: Open

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

5.5 Clock Generator Operations

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

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

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

(a) Upon generation of the $\overline{\text{RESET}}$ signal, the lowest speed mode of the main system clock (3.2 μs @10 MHz operation) is selected (PCC = 04H). Main system clock oscillation stops while a low level is applied to the $\overline{\text{RESET}}$ pin.

★ (b) With the main system clock selected, one of the five minimum instruction execution times (0.2 μs, 0.4 μs, 0.8 μs, 1.6 μs, 3.2 μs, @10 MHz operation) can be selected by setting PCC.

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

(d) The PCC can be used to select the subsystem clock and to operate the system with low power consumption (122 μs @32.768 kHz operation). Also, use of a circuit to multiply the subsystem clock by 4 can be selected via the subclock select register (SSCK) (15.3 μs when a circuit to multiply the subsystem clock by 4 is used).

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

(f) The main system clock is divided and supplied to the peripheral hardware. The subsystem clock is supplied to 8-bit timer 51, the watch timer, the clock output functions, and the LCD controller/driver only. Thus 8-bit timer 51, the watch function, the clock output function, and the LCD controller/driver can also be continued in the standby state. However, since all other peripheral hardware operate with the main system clock, the peripheral hardware also stops if the main system clock is stopped (except when a clock is input externally).

5.5.1 Main system clock operations

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

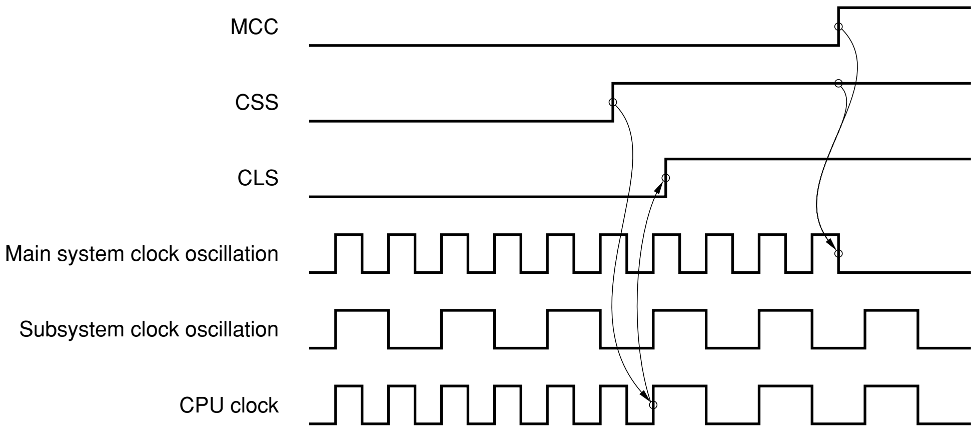
(a) Because the operation guaranteed instruction execution speed depends on the power supply voltage, the minimum instruction execution time can be changed by bits 0 to 2 (PCC0 to PCC2) of PCC.

★ (b) While the clock generator is operating on the main system clock, main system clock oscillation is stopped (see **Figure 5-8 (1)**) if bit 4 (CSS) of PCC is set to 1, the subsystem clock operation is started (CLS = 1), and then bit 7 (MCC) of PCC is set to 1.

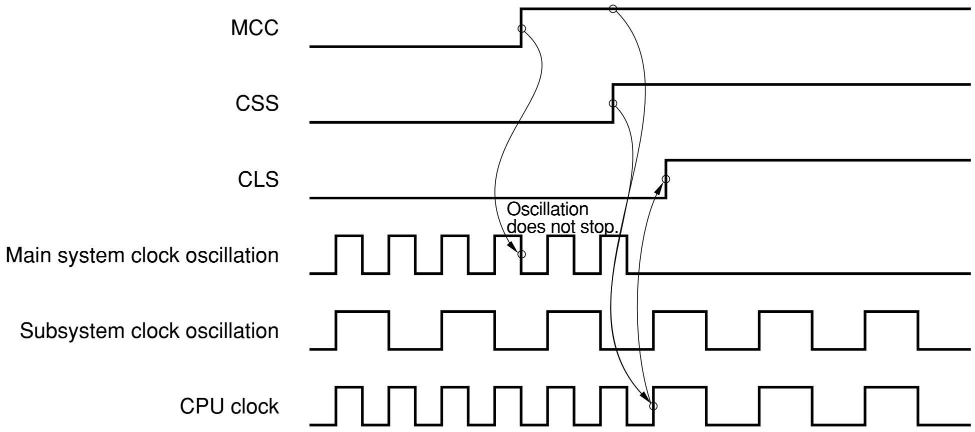
(c) If bit 7 (MCC) of the PCC is set to 1 when operated with the main system clock, the main system clock oscillation does not stop. When bit 4 (CSS) of the PCC is set to 1 and the operation is switched to subsystem clock operation (CLS = 1) after that, the main system clock oscillation stops (see **Figure 5-8 (2)**).

Figure 5-8. Main System Clock Stop Function

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



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


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

5.5.2 Subsystem clock operations

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

(a) The minimum instruction execution time is either of the following, depending on the setting of the subclock select register (SSCK).
- 122 μs: at 32.768 kHz operation
- 30.5 μs: with 32.768 kHz multiplied by 4

The setting does not depend on the bits 0 to 2 (PCC0 to PCC2) of PCC.

(b) Watchdog timer counting stops.

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

5.6 Changing System Clock and CPU Clock Settings

5.6.1 Time required for switchover between system clock and CPU clock

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

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

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

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

<table>
<tr><th colspan="4">Set Value Before Switchover</th><th colspan="24">Set Value After Switchover</th></tr>
<tr><td rowspan="2">CSS</td><td rowspan="2">PCC2</td><td rowspan="2">PCC1</td><td rowspan="2">PCC0</td><td>CSS</td><td>PCC2</td><td>PCC1</td><td>PCC0</td><td>CSS</td><td>PCC2</td><td>PCC1</td><td>PCC0</td><td>CSS</td><td>PCC2</td><td>PCC1</td><td>PCC0</td><td>CSS</td><td>PCC2</td><td>PCC1</td><td>PCC0</td><td>CSS</td><td>PCC2</td><td>PCC1</td><td>PCC0</td><td>CSS</td><td>PCC2</td><td>PCC1</td><td>PCC0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>×</td><td>×</td><td>×</td></tr>
<tr><td rowspan="5">0</td><td>0</td><td>0</td><td>0</td><td colspan="4"></td><td colspan="4">16 instructions</td><td colspan="4">16 instructions</td><td colspan="4">16 instructions</td><td colspan="4">16 instructions</td><td colspan="4">$f_X/2f_{XT}$ instruction
(153 instructions)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td colspan="4">8 instructions</td><td colspan="4"></td><td colspan="4">8 instructions</td><td colspan="4">8 instructions</td><td colspan="4">8 instructions</td><td colspan="4">$f_X/4f_{XT}$ instruction
(77 instructions)</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td colspan="4">4 instructions</td><td colspan="4">4 instructions</td><td colspan="4"></td><td colspan="4">4 instructions</td><td colspan="4">4 instructions</td><td colspan="4">$f_X/8f_{XT}$ instruction
(39 instructions)</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td colspan="4">2 instructions</td><td colspan="4">2 instructions</td><td colspan="4">2 instructions</td><td colspan="4"></td><td colspan="4">2 instructions</td><td colspan="4">$f_X/16f_{XT}$ instruction
(20 instructions)</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td colspan="4">1 instruction</td><td colspan="4">1 instruction</td><td colspan="4">1 instruction</td><td colspan="4">1 instruction</td><td colspan="4"></td><td colspan="4">$f_X/32f_{XT}$ instruction
(10 instructions)</td></tr>
<tr><td>1</td><td>×</td><td>×</td><td>×</td><td colspan="4">1 instruction</td><td colspan="4">1 instruction</td><td colspan="4">1 instruction</td><td colspan="4">1 instruction</td><td colspan="4">1 instruction</td><td colspan="4"></td></tr>
</table>

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

2. Figures in parentheses are for operation with f_X = 10 MHz and f_{XT} = 32.768 kHz.

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

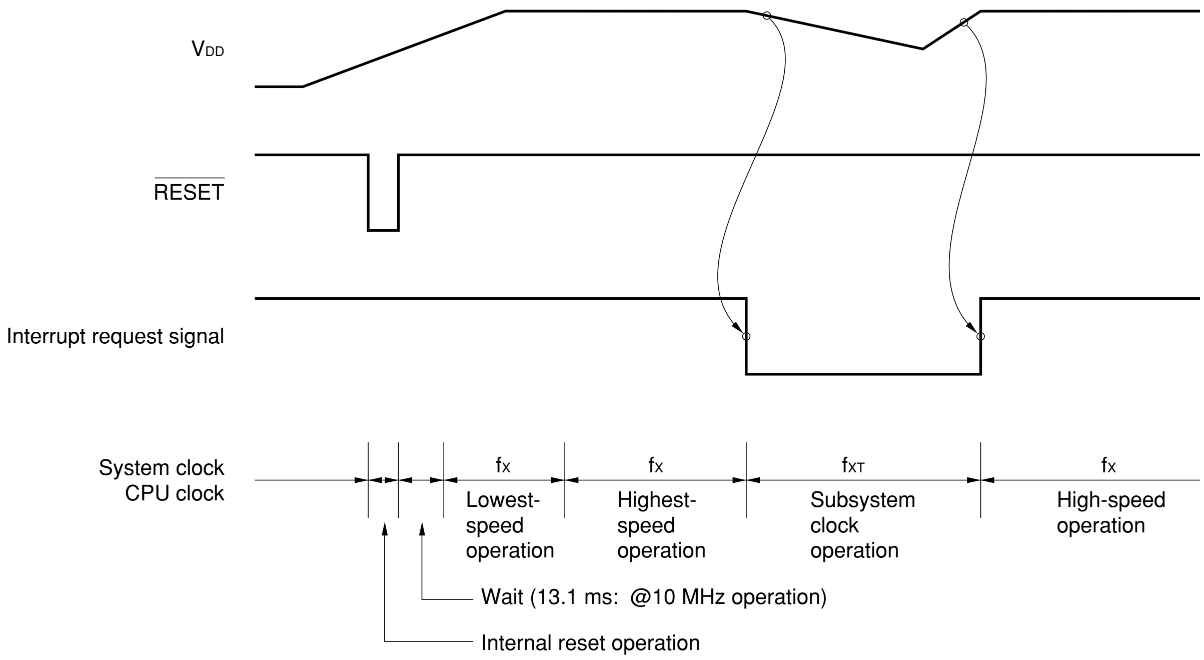
Simultaneous setting is possible, however, for selection of the CPU clock cycle division ratio (PCC0 to PCC2) and switchover from the subsystem clock to the main system clock (changing CSS from 1 to 0).

2. If the clock must be switched over between the subsystem clock being used in ×4 mode and the main system clock, be sure to set the CPU clock frequency when the main system clock is used to 280 kHz or more.

5.6.2 System clock and CPU clock switching procedure

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

Figure 5-9. System Clock and CPU Clock Switching



<1> The CPU is reset by setting the $\overline{\text{RESET}}$ signal to low level after power-on. After that, when reset is released by setting the $\overline{\text{RESET}}$ signal to high level, main system clock starts oscillation. At this time, oscillation stabilization time ($2^{17}/f_X$) is secured automatically.
After that, the CPU starts executing instructions at the minimum speed of the main system clock (3.2 μs @10 MHz operation).

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

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

<4> Upon detection of V_{DD} voltage reset due to an interrupt, bit 7 (MCC) of PCC is set to 0 and oscillation of the main system clock is started. After the lapse of time required for stabilization of oscillation, PCC is rewritten and the maximum-speed operation is resumed.

Caution When the main system clock is stopped and the device is operating on the subsystem clock, wait until the oscillation stabilization time has been secured by the program before switching back to the main system clock.

CHAPTER 6 16-BIT TIMER/EVENT COUNTER 0

6.1 Outline of 16-Bit Timer/Event Counter 0

16-bit timer/event counter 0 can be used as an interval timer, PPG output, pulse width measurement (infrared ray remote control receive function), external event counter, or square wave output of any frequency.

6.2 Functions of 16-Bit Timer/Event Counter 0

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

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

(1) Interval timer
Generates an interrupt request at the preset time interval.

(2) PPG output
Can output a square wave whose frequency and output pulse can be set freely.

(3) Pulse width measurement
Can measure the pulse width of an externally input signal.

(4) External event counter
Can measure the number of pulses of an externally input signal.

(5) Square-wave output
Can output a square wave with any selected frequency.

6.3 Configuration of 16-Bit Timer/Event Counter 0

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

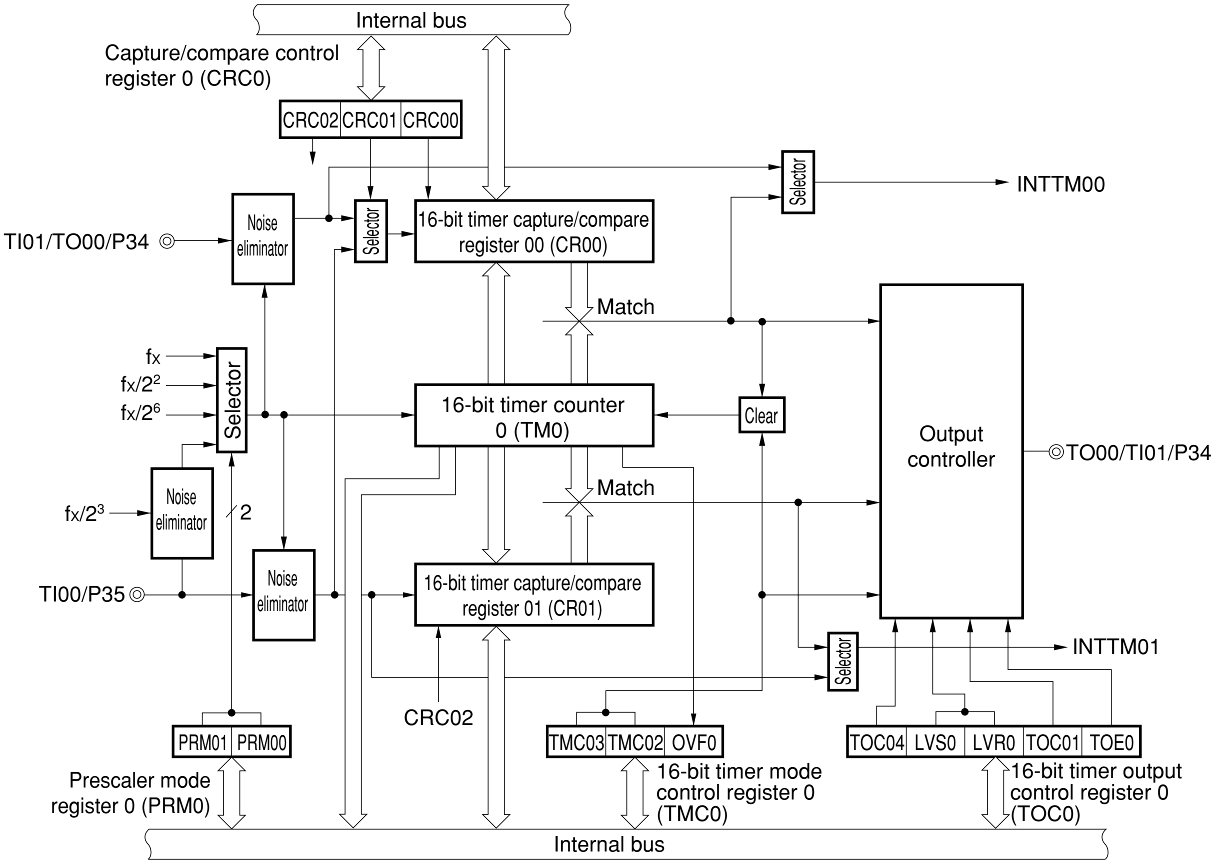
Table 6-1. Configuration of 16-Bit Timer/Event Counter 0

Item	Configuration
Timer/counter	16 bits × 1 (TM0)
Register	16-bit timer capture/compare register: 16 bits × 2 (CR00, CR01)
Timer output	1 (TO00)
Control registers	16-bit timer mode control register 0 (TMC0) Capture/compare control register 0 (CRC0) 16-bit timer output control register 0 (TOC0) Prescaler mode register 0 (PRM0) Port mode register 3 (PM3)$^{\text{Note}}$

Note Refer to **Figure 4-9 Block Diagram of P32 to P34** and **Figure 4-10 P35 Block Diagram.**

Figure 6-1 shows a block diagram.

★ **Figure 6-1. Block Diagram of 16-Bit Timer/Event Counter 0**


Internal bus
Capture/compare control register 0 (CRC0)
CRC02
CRC01
CRC00
Selector
INTTM00
TI01/TO00/P34
Noise eliminator
Selector
16-bit timer capture/compare register 00 (CR00)
Match
fx
fx/2²
fx/2⁶
Selector
16-bit timer counter 0 (TM0)
Clear
Output controller
TO00/TI01/P34
fx/2³
Noise eliminator
2
Match
TI00/P35
Noise eliminator
16-bit timer capture/compare register 01 (CR01)
Selector
INTTM01
CRC02
PRM01
PRM00
TMC03
TMC02
OVF0
TOC04
LVS0
LVR0
TOC01
TOE0
Prescaler mode register 0 (PRM0)
16-bit timer mode control register 0 (TMC0)
16-bit timer output control register 0 (TOC0)
Internal bus

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

TM0 is a 16-bit read-only register that counts count pulses.

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

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

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

CR00 is a 16-bit register which has the functions of both a capture register and a compare register. Whether it is used as a capture register or as a compare register is set by bit 0 (CRC00) of capture/compare control register 0 (CRC0).

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

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

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

  It is possible to select the valid edge of the TI00/P35 pin or the TI01/TO00/P34 pin as the capture trigger. Setting of the TI00 or TI01 valid edge is performed by means of prescaler mode register 0 (PRM0).

  If CR00 is specified as a capture register and capture trigger is specified to be the valid edge of the TI00/P35 pin, the situation is as shown in Table 6-2. On the other hand, when capture trigger is specified to be the valid edge of the TI01/TO00/P34 pin, the situation is as shown in Table 6-3.

Table 6-2. TI00/P35 Pin Valid Edge and CR00, CR01 Capture Trigger

ES01	ES00	TI00/P35 Pin Valid Edge	CR00 Capture Trigger	CR01 Capture Trigger
0	0	Falling edge	Rising edge	Falling edge
0	1	Rising edge	Falling edge	Rising edge
1	0	Setting prohibited	Setting prohibited	Setting prohibited
1	1	Both rising and falling edges	No capture operation	Both rising and falling edges

Table 6-3. TI01/TO00/P34 Pin Valid Edge and CR00 Capture Trigger

ES11	ES10	TI01/TO00/P34 Pin Valid Edge	CR00 Capture Trigger
0	0	Falling edge	Falling edge
0	1	Rising edge	Rising edge
1	0	Setting prohibited	Setting prohibited
1	1	Both rising and falling edges	Both rising and falling edges

CR00 is set by a 16-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input makes the value of this register undefined.

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

2. If the new value of CR00 is less than the value of 16-bit timer counter 0 (TM0), TM0 continues counting, overflows, and then start counting from 0 again. If the new value of CR00 is less than the old value, therefore, the timer must be reset and restarted after the value of CR00 is changed.

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

CR01 is a 16-bit register which has the functions of both a capture register and a compare register. Whether it is used as a capture register or a compare register is set by bit 2 (CRC02) of capture/compare control register 0 (CRC0).

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

- **When CR01 is used as a capture register**
  It is possible to select the valid edge of the TI00/P35 pin as the capture trigger. The TI00/P35 valid edge is set by means of prescaler mode register 0 (PRM0). Table 6-2 shows the setting when the valid edge of the TI00/P35 pin is specified as the capture trigger.

CR01 is set by a 16-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input makes the value of this register undefined.

Caution In the clear & start mode entered on a match between TM0 and CR00, set a value other than 0000H in CR01. However, in the free-running mode and in the clear mode using the valid edge of TI00, if 0000H is set to CR01, an interrupt request (INTTM01) is generated following overflow (FFFFH).

6.4 Registers to Control 16-Bit Timer/Event Counter 0

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

- 16-bit timer mode control register 0 (TMC0)
- Capture/compare control register 0 (CRC0)
- 16-bit timer output control register 0 (TOC0)
- Prescaler mode register 0 (PRM0)
- Port mode register 3 (PM3)

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

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

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

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

Caution 16-bit timer counter 0 (TM0) starts operation at the moment a value other than 0, 0 (operation stop mode) is set in TMC02 to TMC03, respectively. Set 0, 0 in TMC02 to TMC03 to stop the operation.

Figure 6-2. Format of 16-Bit Timer Mode Control Register 0 (TMC0)

Address: FF60H After reset: 00H R/W

Symbol	7	6	5	4	$\boxed{3}$	$\boxed{2}$	1	$\boxed{0}$
TMC0	0	0	0	0	TMC03	TMC02	0	OVF0

TMC03	TMC02	Operation mode and clear mode selection	TO00 output timing selection	Interrupt request generation
0	0	Operation stop (TM0 cleared to 0)	No change	Not generated
0	1	Free-running mode	Match between TM0 and CR00 or match between TM0 and CR01	Generated on match between TM0 and CR00, or match between TM0 and CR01
1	0	Clear & start on TI00 valid edge	—	
1	1	Clear & start on match between TM0 and CR00	Match between TM0 and CR00 or match between TM0 and CR01	

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

Cautions 1. Be sure to stop timer operation before writing to bits other than the OVF0 flag.

2. Set the valid edge of the TI00/P35 pin with prescaler mode register 0 (PRM0).

3. If clear & start mode on entered a match between TM0 and CR00 is selected, when the set value of CR00 is FFFFH and the TM0 value changes from FFFFH to 0000H, the OVF0 flag is set to 1.

Remarks 1. TO00: 16-bit timer/event counter 0 output pin

2. TI00: 16-bit timer/event counter 0 input pin

3. TM0: 16-bit timer counter 0

4. CR00: 16-bit timer capture/compare register 00

5. CR01: 16-bit timer capture/compare register 01

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

This register controls the operation of 16-bit timer capture/compare registers 00 and 01 (CR00, CR01).

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

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

Figure 6-3. Format of Capture/Compare Control Register 0 (CRC0)

Address: FF62H After reset: 00H R/W

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

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

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

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

Cautions 1. Be sure to stop timer operation before setting CRC0.

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

3. If both the rising and falling edges have been selected as the valid edges of TI00, capture is not performed.

★ **4. To surely perform the capture operation, the capture trigger requires a pulse longer than two of the count clock cycles selected by prescaler mode register 0 (PRM0).**

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

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

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

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

Figure 6-4 shows the TOC0 format.

Figure 6-4. Format of 16-Bit Timer Output Control Register 0 (TOC0)

Address: FF63H After reset: 00H R/W

Symbol	7	6	5	4	$\boxed{3}$	$\boxed{2}$	1	$\boxed{0}$
TOC0	0	0	0	TOC04	LVS0	LVR0	TOC01	TOE0

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

LVS0	LVR0	16-bit timer/event counter 0 timer output F/F status setting
0	0	No change
0	1	Timer output F/F reset (0)
1	0	Timer output F/F set (1)
1	1	Setting prohibited

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

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

Cautions 1. Be sure to stop timer operation before setting TOC0.

2. If LVS0 and LVR0 are read after data is set, they will be 0.

3. Be sure to set bits 5, 6 and 7 to 0.

(4) Prescaler mode register 0 (PRM0)

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

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

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

Figure 6-5. Format of Prescaler Mode Register 0 (PRM0)

Address: FF61H After reset: 00H R/W

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

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

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

PRM01	PRM00	Count clock selection
0	0	f_X (10 MHz)
0	1	$f_X/2^2$ (2.5 MHz)
1	0	$f_X/2^6$ (156.25 kHz)
1	1	TI00 valid edge**Note**

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

Cautions 1. If the valid edge of TI00 is to be set as the count clock, do not set the clear & start mode and the capture trigger at the valid edge of TI00.

2. Be sure to stop timer operation before setting data to PRM0.

3. If the TI00 or TI01 pin is high level immediately after system reset, the rising edge is immediately detected after the rising edge or both the rising and falling edges are set as the valid edge(s) of the TI00 pin or TI01 pin to enable the operation of 16-bit timer counter 0 (TM0). Please be careful when pulling up the TI00 pin or the TI01 pin. However, when re-enabling operation after the operation has been stopped once, the rising edge is not detected.

Remarks 1. f_X: Main system clock oscillation frequency

2. TI00, TI01: 16-bit timer/event counter 0 input pin

3. Figures in parentheses are for operation with f_X = 10 MHz.

(5) Port mode register 3 (PM3)

This register sets port 3 input/output in 1-bit units.

When using the P34/TI01/TO00 pin for timer output, set PM30 and the output latch of P30 to 0.

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

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

Figure 6-6. Format of Port Mode Register 3 (PM3)

Address: FF23H After reset: FFH R/W

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

PM3n	P3n pin I/O mode selection (n = 0 to 5)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

6.5 Operations of 16-Bit Timer/Event Counter 0

6.5.1 Interval timer operations

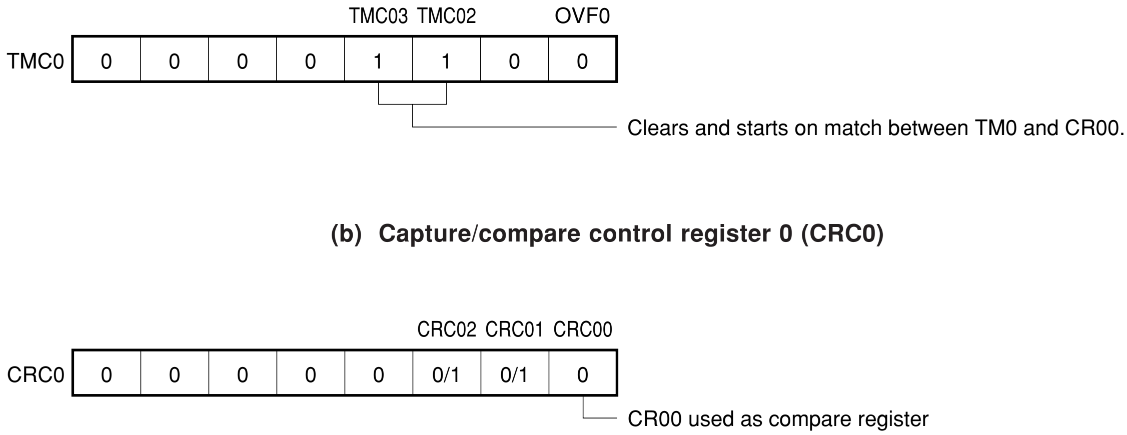
Setting 16-bit timer mode control register 0 (TMC0) and capture/compare control register 0 (CRC0) as shown in Figure 6-7 allows operation as an interval timer. Interrupt requests are generated repeatedly using the count value set in 16-bit timer capture/compare register 00 (CR00) beforehand as the interval.

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

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

See **6.6 16-Bit Timer/Event Counter 0 Cautions (2)** about the operation when the compare register value is changed during timer count operation.

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



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

					TMC03	TMC02		OVF0
TMC0	0	0	0	0	1	1	0	0

Clears and starts on match between TM0 and CR00.

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

						CRC02	CRC01	CRC00
CRC0	0	0	0	0	0	0/1	0/1	0

CR00 used as compare register

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

Figure 6-8. Interval Timer Configuration Diagram

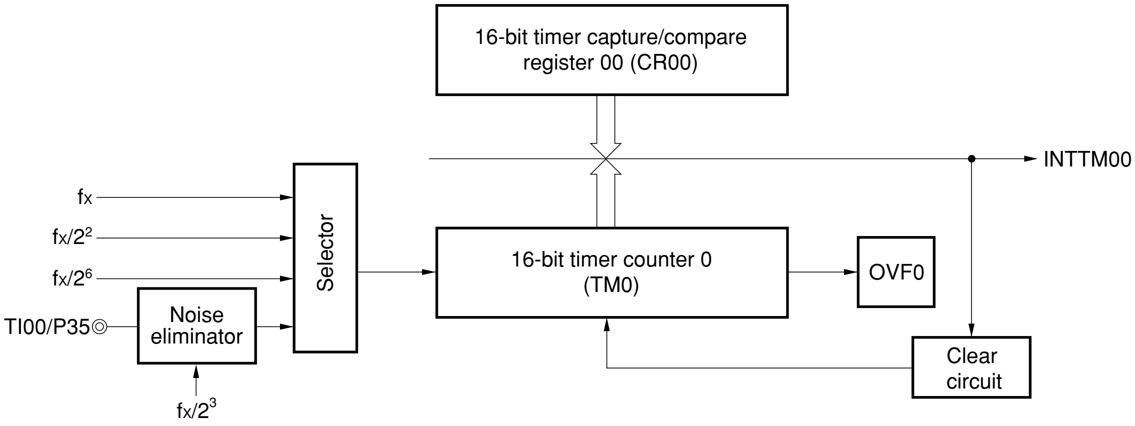
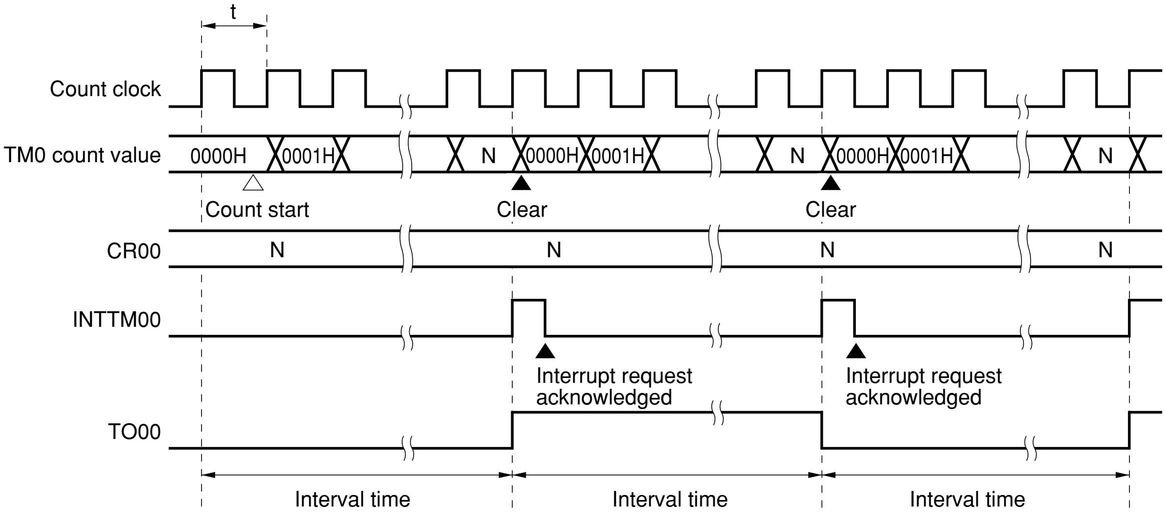

16-bit timer capture/compare register 00 (CR00)
INTTM00
fx
fx/2²
fx/2⁶
TI00/P35◎
Noise eliminator
fx/2³
Selector
16-bit timer counter 0 (TM0)
OVF0
Clear circuit

Figure 6-9. Timing of Interval Timer Operation



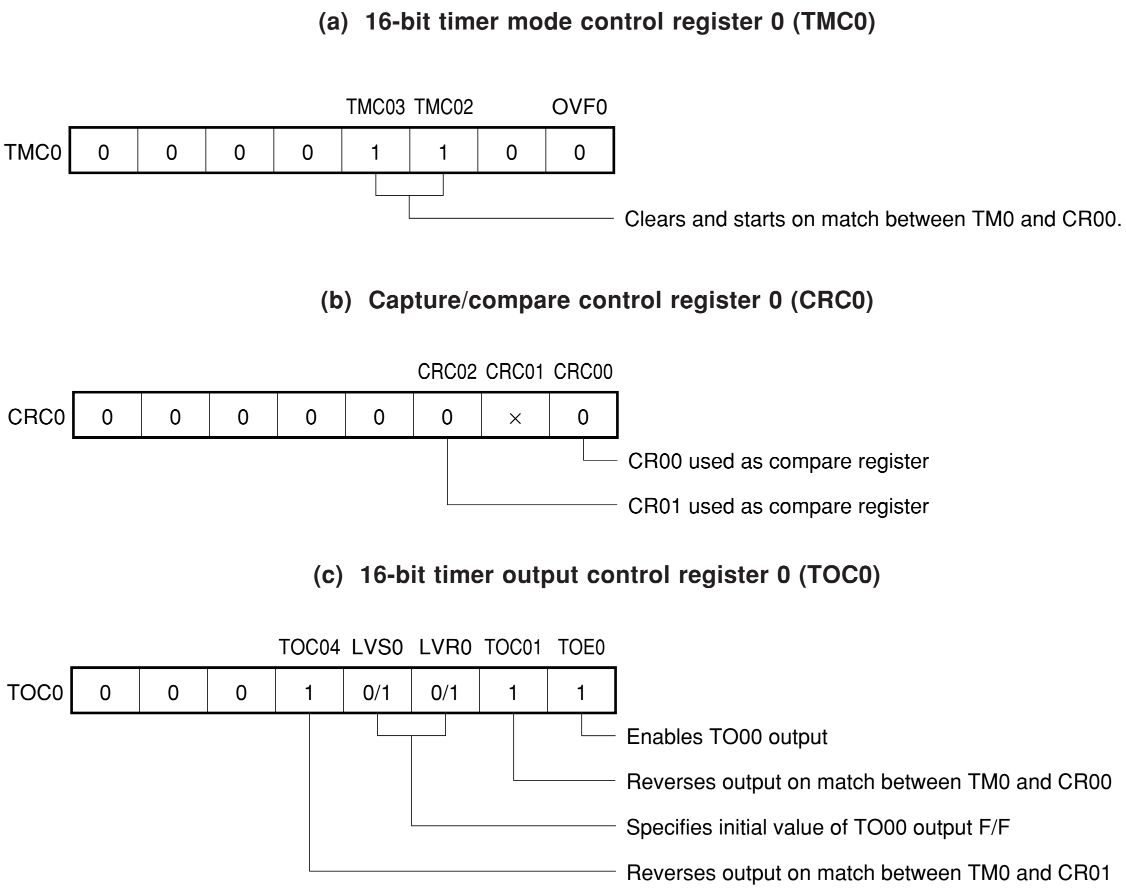
Remark Interval time = (N + 1) × t
N = 0001H to FFFFH

6.5.2 PPG output operations

Setting 16-bit timer mode control register 0 (TMC0) and capture/compare control register 0 (CRC0) as shown in Figure 6-10 allows operation as PPG (Programmable Pulse Generator) output.

In the PPG output operation, square waves are output from the TO00/TI01/P34 pin with the pulse width and the cycle that correspond to the count values set beforehand in 16-bit timer capture/compare register 01 (CR01) and in 16-bit timer capture/compare register 00 (CR00), respectively.

Figure 6-10. Control Register Settings for PPG Output Operation



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

					TMC03	TMC02		OVF0
TMC0	0	0	0	0	1	1	0	0

Clears and starts on match between TM0 and CR00.

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

						CRC02	CRC01	CRC00
CRC0	0	0	0	0	0	0	×	0

CR00 used as compare register
CR01 used as compare register

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

				TOC04	LVS0	LVR0	TOC01	TOE0
TOC0	0	0	0	1	0/1	0/1	1	1

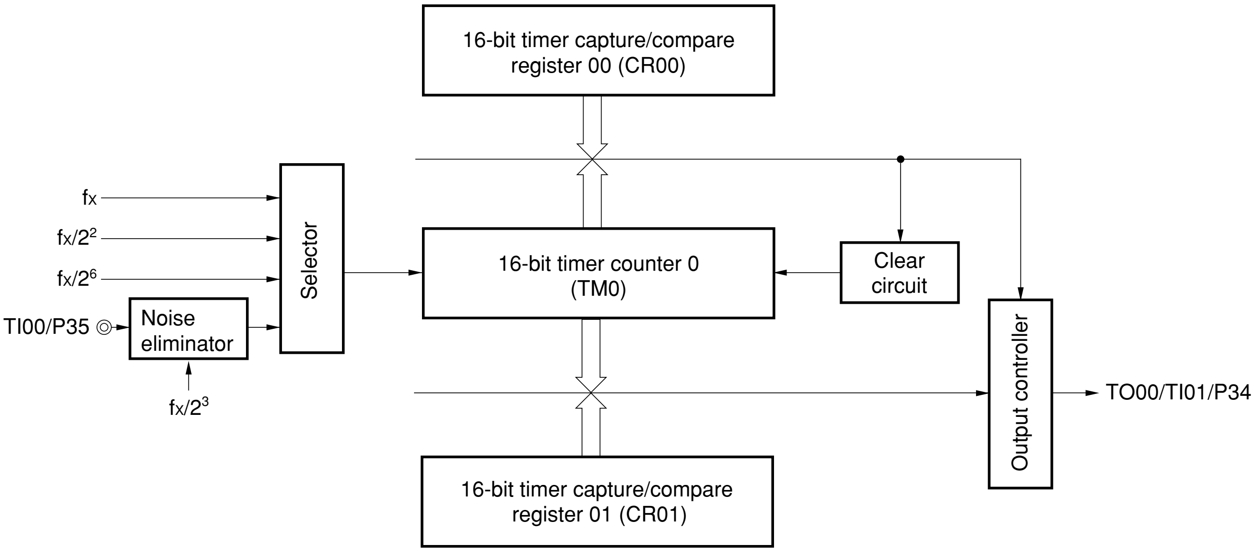
Enables TO00 output
Reverses output on match between TM0 and CR00
Specifies initial value of TO00 output F/F
Reverses output on match between TM0 and CR01

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

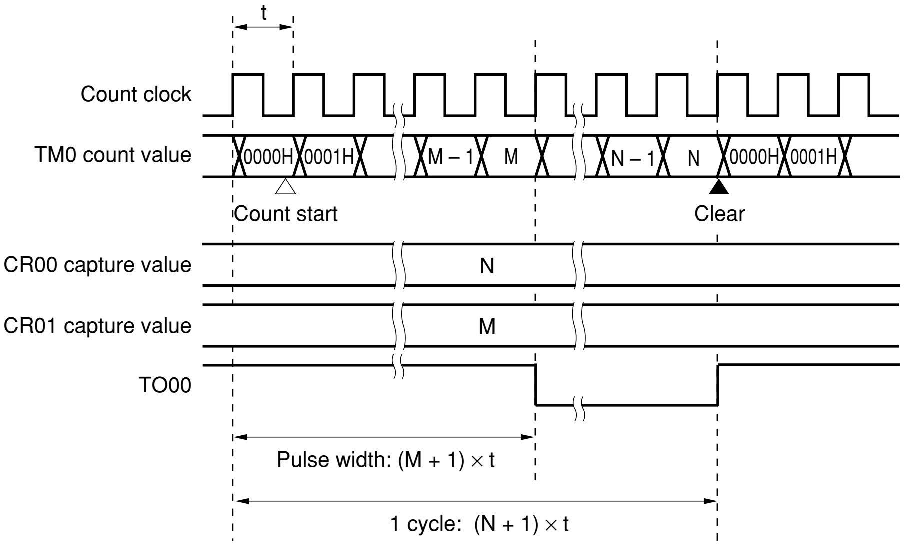
2. The cycle of the pulse generated through PPG output (CR00 setting value + 1) has a duty of (CR01 setting value + 1)/(CR00 setting value + 1).

Remark ×: don’t care

★ **Figure 6-11. Configuration of PPG Output**


16-bit timer capture/compare register 00 (CR00)
fx
fx/2²
fx/2⁶
TI00/P35
Noise eliminator
fx/2³
Selector
16-bit timer counter 0 (TM0)
Clear circuit
Output controller
TO00/TI01/P34
16-bit timer capture/compare register 01 (CR01)

★ **Figure 6-12. PPG Output Operation Timing**



Remark 0000H < M < N ≤ FFFFH

6.5.3 Pulse width measurement operations

It is possible to measure the pulse width of the signals input to the TI00/P35 pin and TI01/TO00/P34 pin using 16-bit timer counter 0 (TM0).

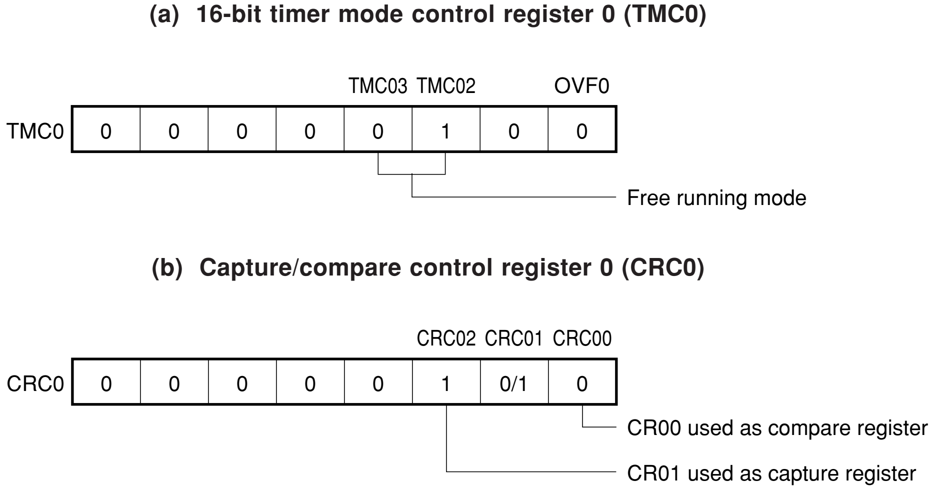
There are two measurement methods: measuring with TM0 used in free-running mode, and measuring by restarting the timer in synchronization with the edge of the signal input to the TI00/P35 pin.

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

When 16-bit timer counter 0 (TM0) is operated in free-running mode (see register settings in **Figure 6-13**), and the edge specified by prescaler mode register 0 (PRM0) is input to the TI00/P35 pin, the value of TM0 is taken into 16-bit timer capture/compare register 01 (CR01) and an external interrupt request signal (INTTM01) is set. Any of three edges can be selected—rising, falling, or both edges—specified by means of bits 4 and 5 (ES00 and ES01) of PRM0.

Sampling is performed at the count clock selected by PRM0, and a capture operation is only performed when a valid level of the TI00/P35 pin or TI01/TO00/P34 pin is detected twice, thus eliminating noise with a short pulse width.

Figure 6-13. Control Register Settings for Pulse Width Measurement with Free-Running Counter and One Capture Register



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

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

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

Figure 6-14. Configuration Diagram for Pulse Width Measurement by Free-Running Counter

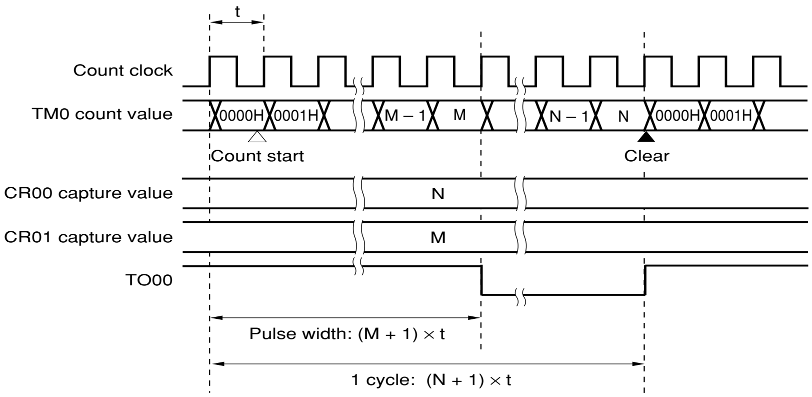
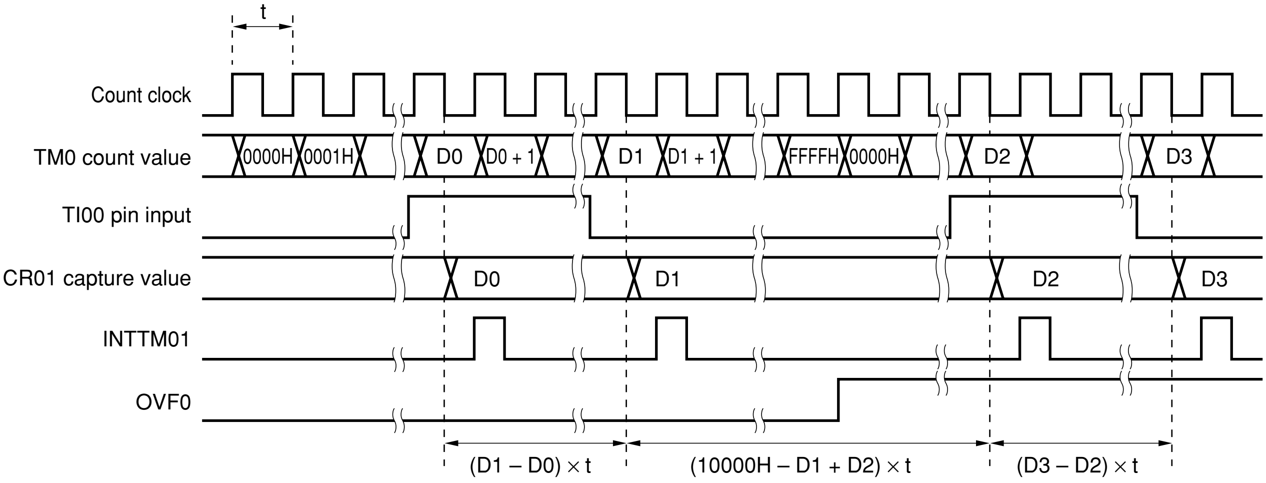


Figure 6-15. Timing of Pulse Width Measurement Operation by Free-Running Counter and One Capture Register (with Both Edges Specified)



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

When 16-bit timer counter 0 (TM0) is operated in free-running mode (see register settings in **Figure 6-16**), it is possible to simultaneously measure the pulse widths of the two signals input to the TI00/P35 pin and the TI01/TO00/P34 pin.

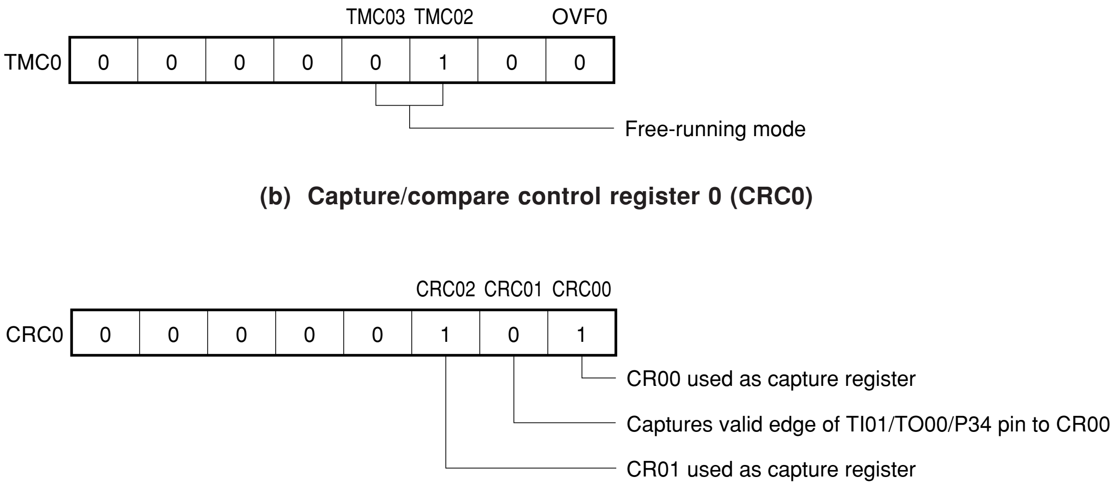
When the edge specified by bits 4 and 5 (ES00 and ES01) of prescaler mode register 0 (PRM0) is input to the TI00/P35 pin, the value of TM0 is taken into 16-bit timer capture/compare register 01 (CR01) and an interrupt request signal (INTTM01) is set.

Also, when the edge specified by bits 6 and 7 (ES10 and ES11) of PRM0 is input to the TI01/TO00/P34 pin, the value of TM0 is taken into 16-bit timer capture/compare register 00 (CR00) and an external interrupt request signal (INTTM00) is set.

Any of three edges can be selected—rising, falling, or both edges—as the valid edges for the TI00/P35 pin and the TI01/TO00/P34 pin specified by means of bits 4 and 5 (ES00 and ES01) and bits 6 and 7 (ES10 and ES11) of PRM0, respectively.

Sampling is performed at the interval selected by means of prescaler mode register 0 (PRM0), and a capture operation is only performed when a valid level of the TI00/P35 pin or TI01/TO00/P34 pin is detected twice, thus eliminating noise with a short pulse width.

Figure 6-16. Control Register Settings for Measurement of Two Pulse Widths with Free-Running Counter



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

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

TMC03 TMC02
OVF0
TMC0
0 0 0 0 0 1 0 0
Free-running mode
CRC02 CRC01 CRC00
CRC0
0 0 0 0 0 1 0 1
CR00 used as capture register
Captures valid edge of TI01/TO00/P34 pin to CR00
CR01 used as capture register

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

  Capture register operation in capture trigger input is shown.

Figure 6-17. Capture Operation of CR01 with Rising Edge Specified

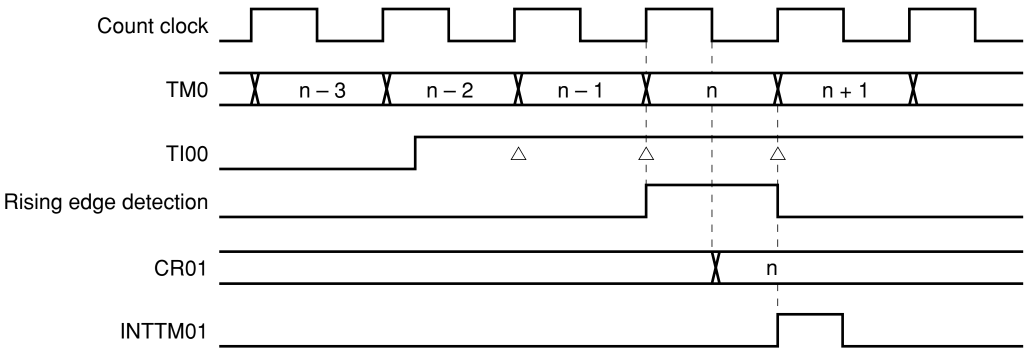
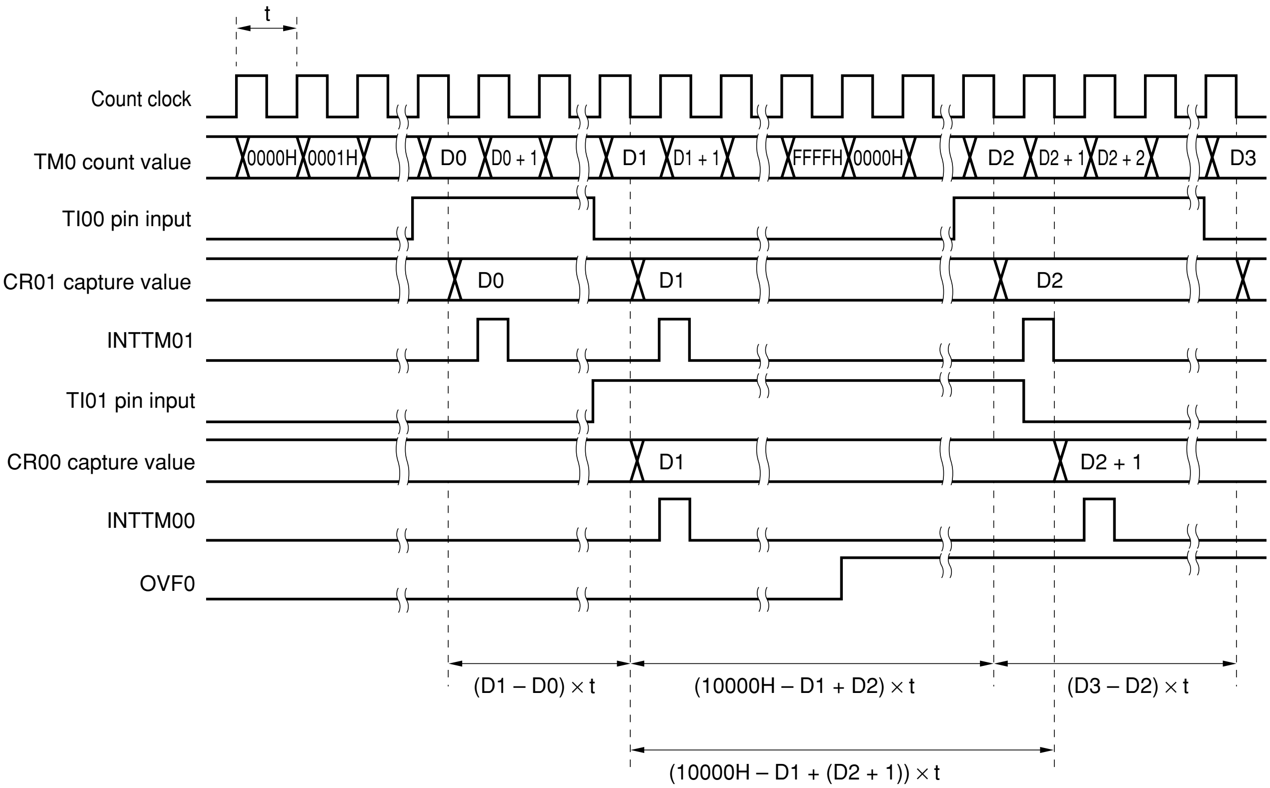


Figure 6-18. Timing of Pulse Width Measurement Operation with Free-Running Counter (with Both Edges Specified)



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

When 16-bit timer counter 0 (TM0) is operated in free-running mode (see register settings in **Figure 6-19**), it is possible to measure the pulse width of the signal input to the TI00/P35 pin.

When the edge specified by bits 4 and 5 (ES00 and ES01) of prescaler mode register 0 (PRM0) is input to the TI00/P35 pin, the value of TM0 is taken into 16-bit timer capture/compare register 01 (CR01) and an interrupt request signal (INTTM01) is set.

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

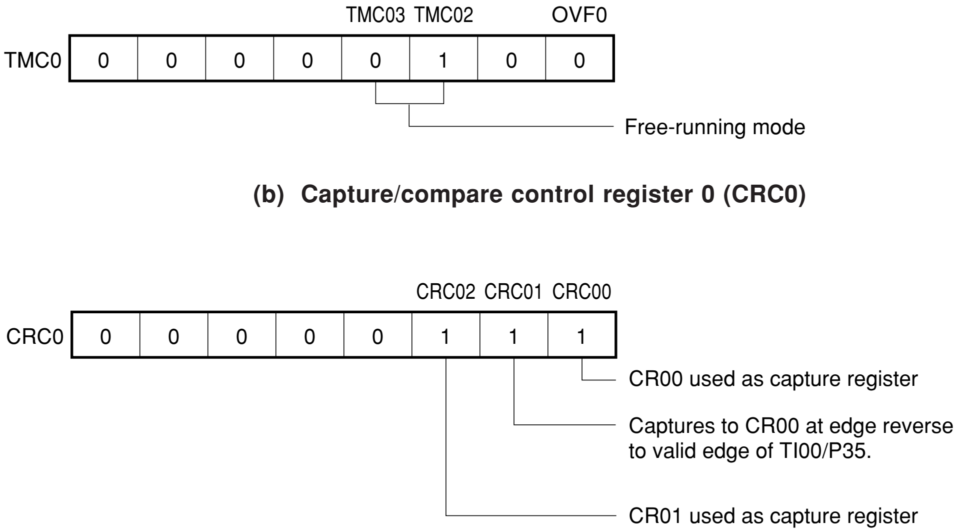
Either of two edges can be selected—rising or falling—as the valid edge for the TI00/P35 pin specified by means of bits 4 and 5 (ES00 and ES01) of prescaler mode register 0 (PRM0).

Sampling is performed at the interval selected by means of prescaler mode register 0 (PRM0), and a capture operation is only performed when a valid level of the TI00/P35 pin is detected twice, thus eliminating noise with a short pulse width.

Caution If the valid edge of the TI00/P35 pin is specified to be both the rising and falling edges, 16-bit timer capture/compare register 00 (CR00) cannot perform the capture operation.

Figure 6-19. Control Register Settings for Pulse Width Measurement with Free-Running Counter and Two Capture Registers

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

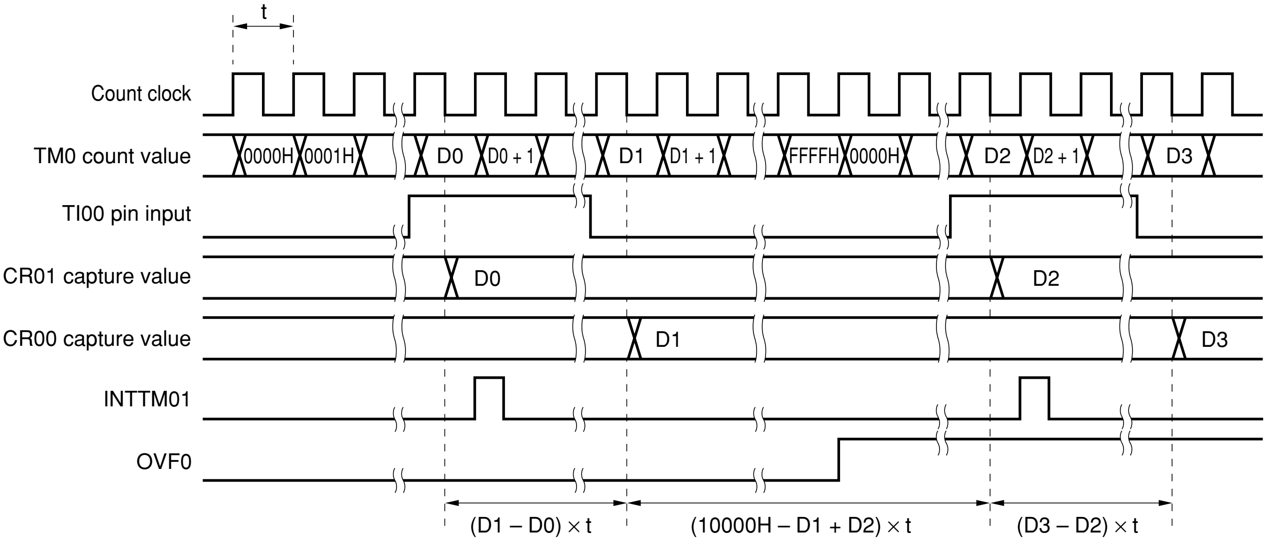


TMC03 TMC02
OVF0
TMC0 0 0 0 0 0 1 0 0
Free-running mode

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

CRC02 CRC01 CRC00
CRC0 0 0 0 0 0 1 1 1
CR00 used as capture register
Captures to CR00 at edge reverse to valid edge of TI00/P35.
CR01 used as capture register

Figure 6-20. 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

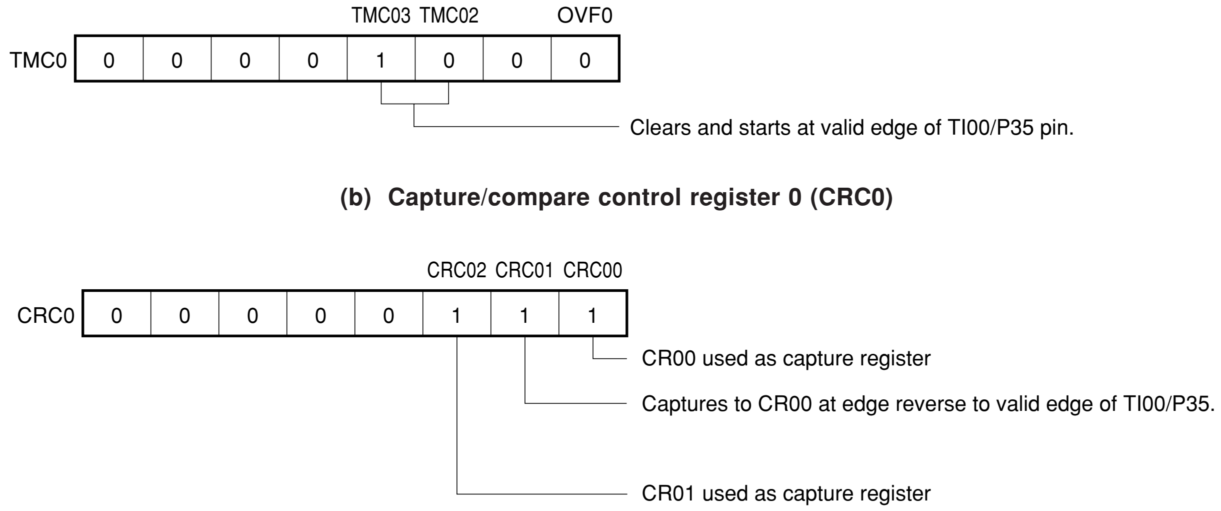
When input of a valid edge to the TI00/P35 pin is detected, the count value of 16-bit timer counter 0 (TM0) is taken into 16-bit timer capture/compare register 01 (CR01), and then the pulse width of the signal input to the TI00/P35 pin is measured by clearing TM0 and restarting the count (see register settings in **Figure 6-21**).

The edge specification can be selected from two types—rising and falling edges—by bits 4 and 5 (ES00 and ES01) of prescaler mode register 0 (PRM0).

In a valid edge detection, the sampling is performed by a cycle selected by prescaler mode register 0 (PRM0) 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 the TI00/P35 pin is specified to be both the rising and falling edges, 16-bit timer capture/compare register 00 (CR00) cannot perform the capture operation.

Figure 6-21. Control Register Settings for Pulse Width Measurement by Means of Restart



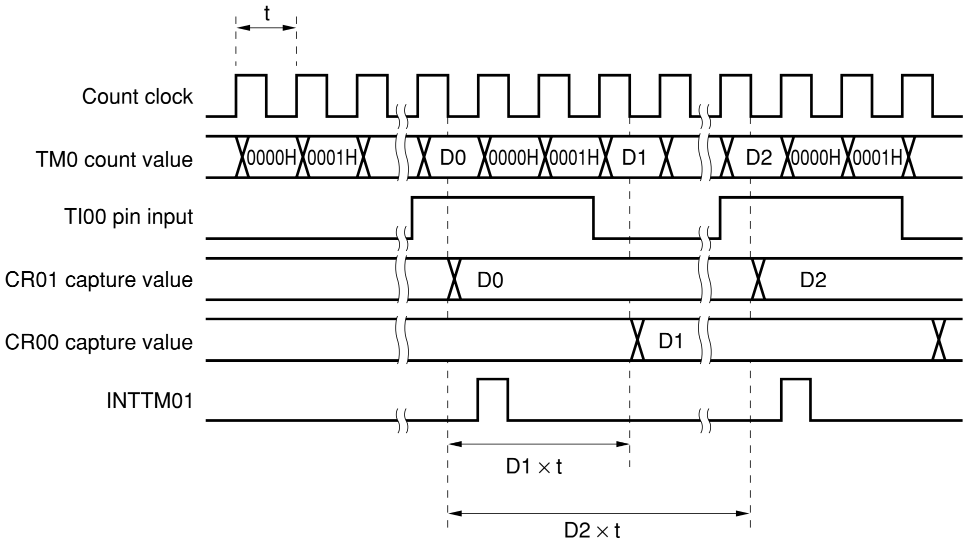
(a) 16-bit timer mode control register 0 (TMC0)

TMC03 TMC02
OVF0
TMC0
0 0 0 0 1 0 0 0
Clears and starts at valid edge of TI00/P35 pin.

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

CRC02 CRC01 CRC00
CRC0
0 0 0 0 0 1 1 1
CR00 used as capture register
Captures to CR00 at edge reverse to valid edge of TI00/P35.
CR01 used as capture register

Figure 6-22. Timing of Pulse Width Measurement Operation by Means of Restart (with Rising Edge Specified)



6.5.4 External event counter operation

The external event counter counts the number of external clock pulses to be input to the TI00/P35 pin with 16-bit timer counter 0 (TM0).

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

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

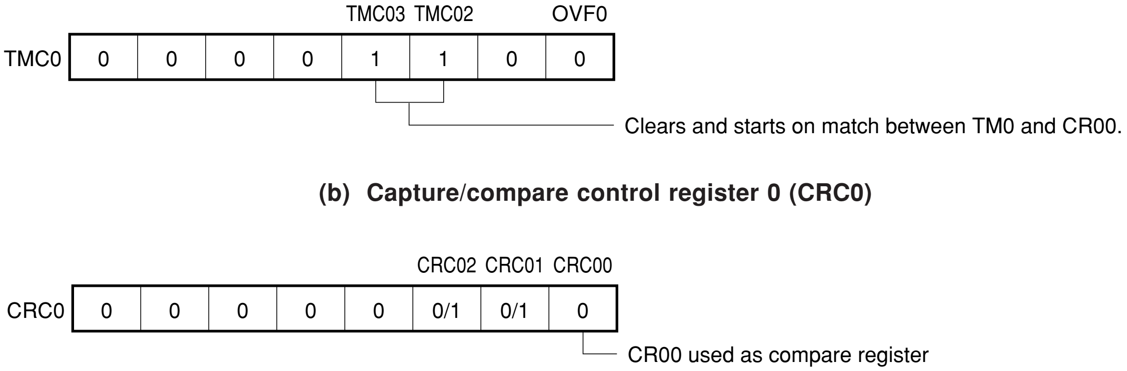
Input any value except 0000H to CR00. (Count operation with a pulse cannot be carried out.)

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

Because capture operation is carried out only after the valid edge of the TI00/P35 pin is detected twice by sampling with the internal clock (fx/2^3), noise with a short pulse width can be eliminated.

Figure 6-23. Control Register Settings in External Event Counter Mode

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


TMC03 TMC02
OVF0
TMC0
0 0 0 0 1 1 0 0
Clears and starts on match between TM0 and CR00.

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

CRC02 CRC01 CRC00
CRC0
0 0 0 0 0 0/1 0/1 0
CR00 used as compare register

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

Figure 6-24. External Event Counter Configuration Diagram

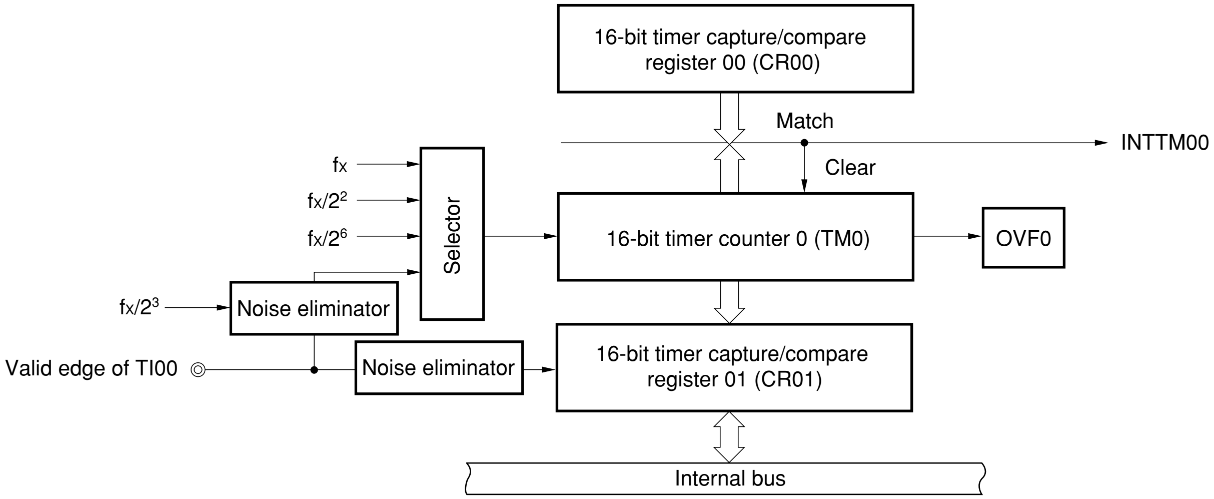
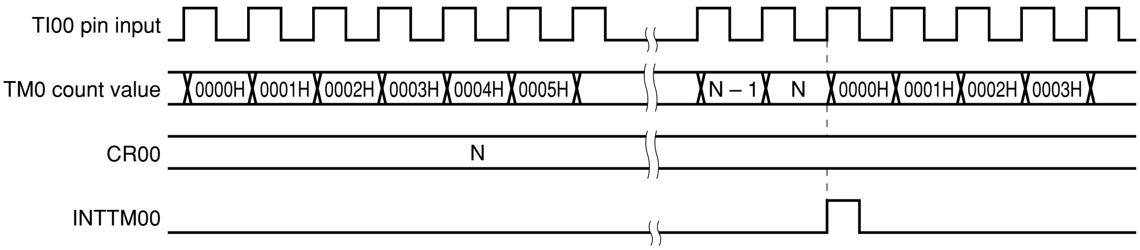

16-bit timer capture/compare register 00 (CR00)
Match
INTTM00
Clear
fX
fX/2^2
fX/2^6
Selector
16-bit timer counter 0 (TM0)
OVF0
fX/2^3
Noise eliminator
Valid edge of TI00
Noise eliminator
16-bit timer capture/compare register 01 (CR01)
Internal bus

Figure 6-25. External Event Counter Operation Timing (with Rising Edge Specified)



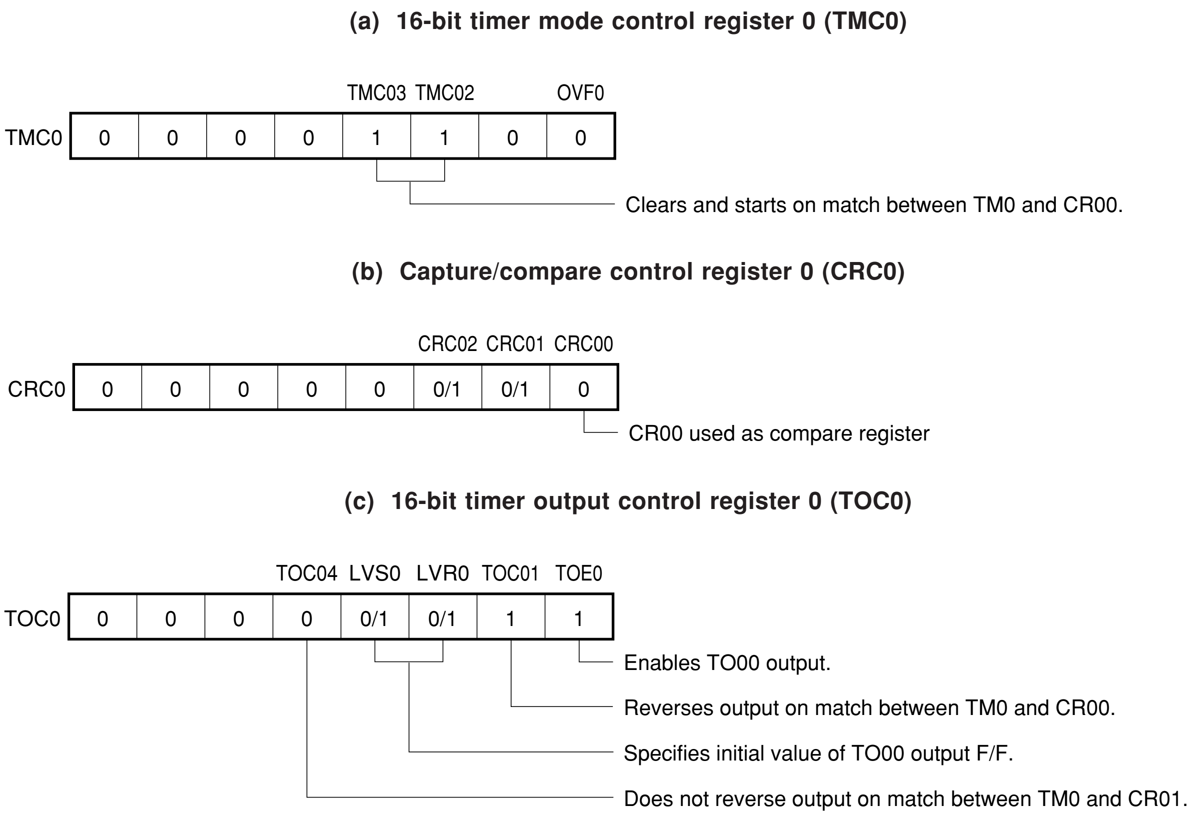
Caution When reading the external event counter count value, TM0 should be read.

6.5.5 Square-wave output operation

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

The TO00 pin output status is reversed at intervals of the count value preset to CR00 by setting bit 0 (TOE0) and bit 1 (TOC01) of 16-bit timer output control register 0 (TOC0) to 1. This enables a square wave with any selected frequency to be output.

Figure 6-26. Control Register Settings in Square-Wave Output Mode


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

TMC03 TMC02
OVF0
TMC0
0 0 0 0 1 1 0 0
Clears and starts on match between TM0 and CR00.

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

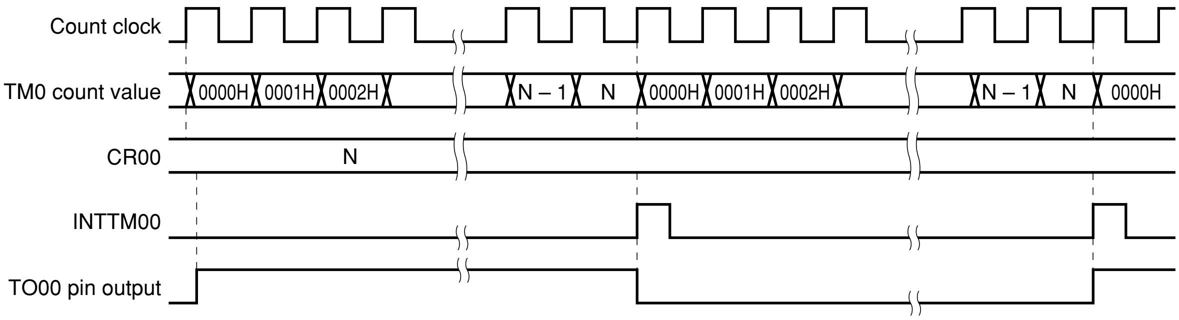
CRC02 CRC01 CRC00
CRC0
0 0 0 0 0 0/1 0/1 0
CR00 used as compare register

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

TOC04 LVS0 LVR0 TOC01 TOE0
TOC0
0 0 0 0 0/1 0/1 1 1
Enables TO00 output.
Reverses output on match between TM0 and CR00.
Specifies initial value of TO00 output F/F.
Does not reverse output on match between TM0 and CR01.

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

Figure 6-27. Square-Wave Output Operation Timing

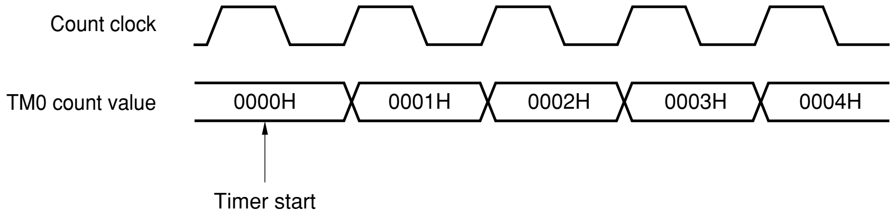


6.6 16-Bit Timer/Event Counter 0 Cautions

(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 0 (TM0) is started asynchronously to the count clock.

Figure 6-28. Start Timing of 16-Bit Timer Counter 0 (TM0)



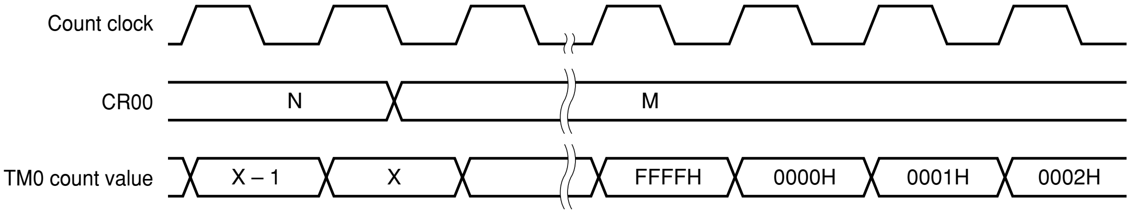
(2) 16-bit timer compare register setting (in the clear & start mode on match between TM0 and CR00)

Set 16-bit timer capture/compare registers 00, 01 (CR00, CR01) to other than 0000H. This means a 1-pulse count operation cannot be performed when the timer 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 00 (CR00) is changed is smaller than that of 16-bit timer counter 0 (TM0), TM0 continues counting, overflows and then restarts counting from 0. Thus, if the value (M) after CR00 has changed is smaller than that (N) before the change, it is necessary to reset and restart the timer after changing CR00.

Figure 6-29. Timing After Change of Compare Register During Timer Count Operation

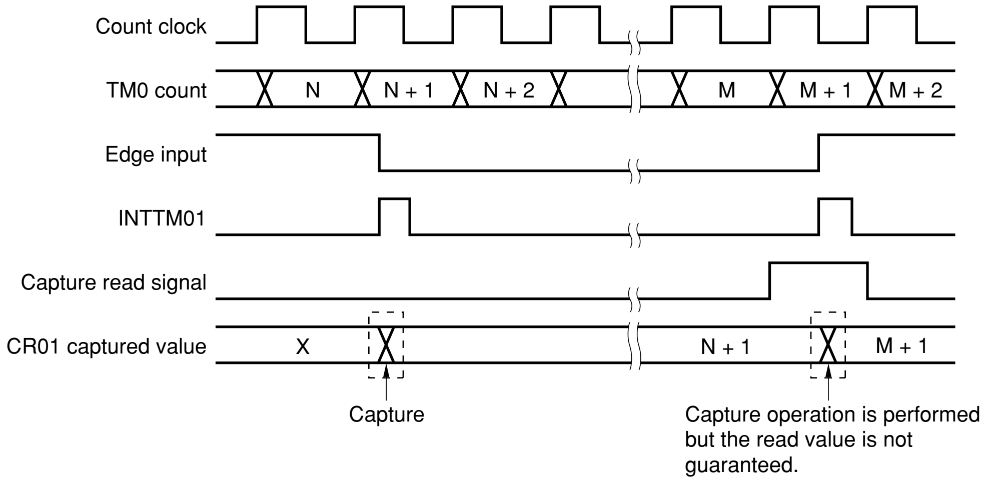


Remark N > X > M

★ **(4) Capture register data retention timings**

If the valid edge of the TI00/P35 pin is input during 16-bit timer capture/compare register 01 (CR01) read, CR01 carries out a capture operation but the read value at this time is not guaranteed. However, the interrupt request signal (INTTM01) is generated upon detection of the valid edge.

★ **Figure 6-30. Capture Register Data Retention Timing**



(5) Valid edge setting

Set the valid edge of the TI00/P35 pin after setting bits 2 and 3 (TMC02 and TMC03) of 16-bit timer mode control register 0 (TMC0) to 0, 0, respectively, and then stopping timer operation. The valid edge is set with bits 4 and 5 (ES00 and ES01) of prescaler mode register 0 (PRM0).

(6) Operation of OVF0 flag

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

Select any of the clear & start mode entered on a match between TM0 and CR00, the mode in which the timer is cleared and started by the valid edge of TI00, and the free-running mode.

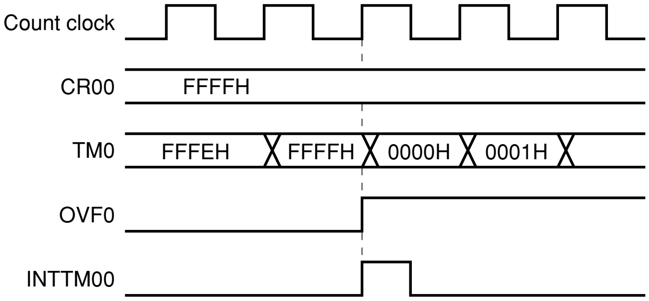
↓

CR00 is set to FFFFH.

↓

When TM0 is counted up from FFFFH to 0000H.

Figure 6-31. Operation Timing of OVF0 Flag



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

(7) Conflicting operations

Conflict between the write period of the 16-bit timer capture/compare register (CR00/CR01) and match with 16-bit timer counter 0 (TM0) (CR00/CR01 used as a compare register)

The match judgment is not performed normally. Do not write any data to CR00/CR01 near the match timing.

(8) Timer operation

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

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

(9) Capture operation

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

<2> If both the rising and falling edges are selected as the valid edges of TI00, capture is not performed.

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

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

(10) Compare operation

<1> INTTM0 may not be generated if the set value of 16-bit timer capture registers 00, 01 (CR00, CR01) and the count value of 16-bit timer counter 0 (TM0) match and CR00 and CR01 are overwritten at the timing of INTTM0 generation. Therefore, do not overwrite CR00 and CR01 frequently even if overwriting the same value.

<2> Capture operation may not be performed for CR00/CR01 set in compare mode even if a capture trigger has been input.

(11) Edge detection

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

<2> The sampling clock used to eliminate noise differs when the TI00 pin valid edge is used as the count clock and when it is used as a capture trigger. In the former case, the sampling clock is $f_X/2^3$, and because the main system clock is used, the sampling clock can only be used when the main system clock is operating. In the latter case, sampling is performed with the count clock selected by prescaler mode register 0 (PRM0). The capture operation is only started after a valid edge is detected twice by sampling, therefore noise with a short pulse width can be eliminated.

(12) TI01 input and TO00 output

Because the TI01 input pin and TO00 output pin are assigned to the same pin (P34/TI01/TO00), these pins cannot be used at the same time.

★ (13) STOP mode and main system clock stop mode settings

Except when the TI00 and TI01 inputs are selected, be sure to stop the timer operation before setting STOP or main system clock stop mode. Otherwise, the timer may malfunction when the main system clock is started.

CHAPTER 7 8-BIT TIMERS A0, B0

7.1 8-Bit Timer A0, B0 Functions

The μPD780344, 780354, 780344Y, 780354Y Subseries have 8-bit timer A0 and 8-bit timer/event counter B0. The operation modes listed in the following table can be set via mode register settings.

Table 7-1. Operation Modes

Channel Mode	Timer A0	Timer B0
8-bit timer counter mode (Discrete mode)	Available	Available
16-bit timer counter mode (Cascade connection mode)	Available	
Carrier generator mode	Available	
PWM output mode	Not available	Available

(1) 8-bit timer counter mode (discrete mode)

The following functions can be used in this mode.

- Interval timer with 8-bit resolution
- External event counter with 8-bit resolution (timer B0 only)
- Square wave output with 8-bit resolution

(2) 16-bit timer counter mode (cascade connection mode)

Operation as a 16-bit timer/event counter is enabled during cascade connection mode.
The following functions can be used in this mode.

- Interval timer with 16-bit resolution
- External event counter with 16-bit resolution
- Square wave output with 16-bit resolution

(3) Carrier generator mode

The carrier clock generated by timer B0 is output in cycles set by timer A0.

(4) PWM output mode (timer B0 only)

Pulses are output using any duty factor set by timer B0.

7.2 8-Bit Timer A0, B0 Configuration

8-bit timer A0 and B0 consist of the following hardware.

Table 7-2. Configuration of 8-Bit Timer A0, B0

Item	Configuration
Timer counters	8 bits × 2 (TMA0, TMB0)
Registers	Compare registers: 8 bits × 3 (CRA0, CRB0, CRHB0)
Timer input	1 (TMIB0)
Timer output	2 (TOA0, TOB0)
Control registers	8-bit timer mode control register A0 (TMCA0) 8-bit timer mode control register B0 (TMCB0) Carrier generator output control register B0 (TCAB0) Port mode register 0 (PM0)

Figure 7-1. Block Diagram of Timer A0

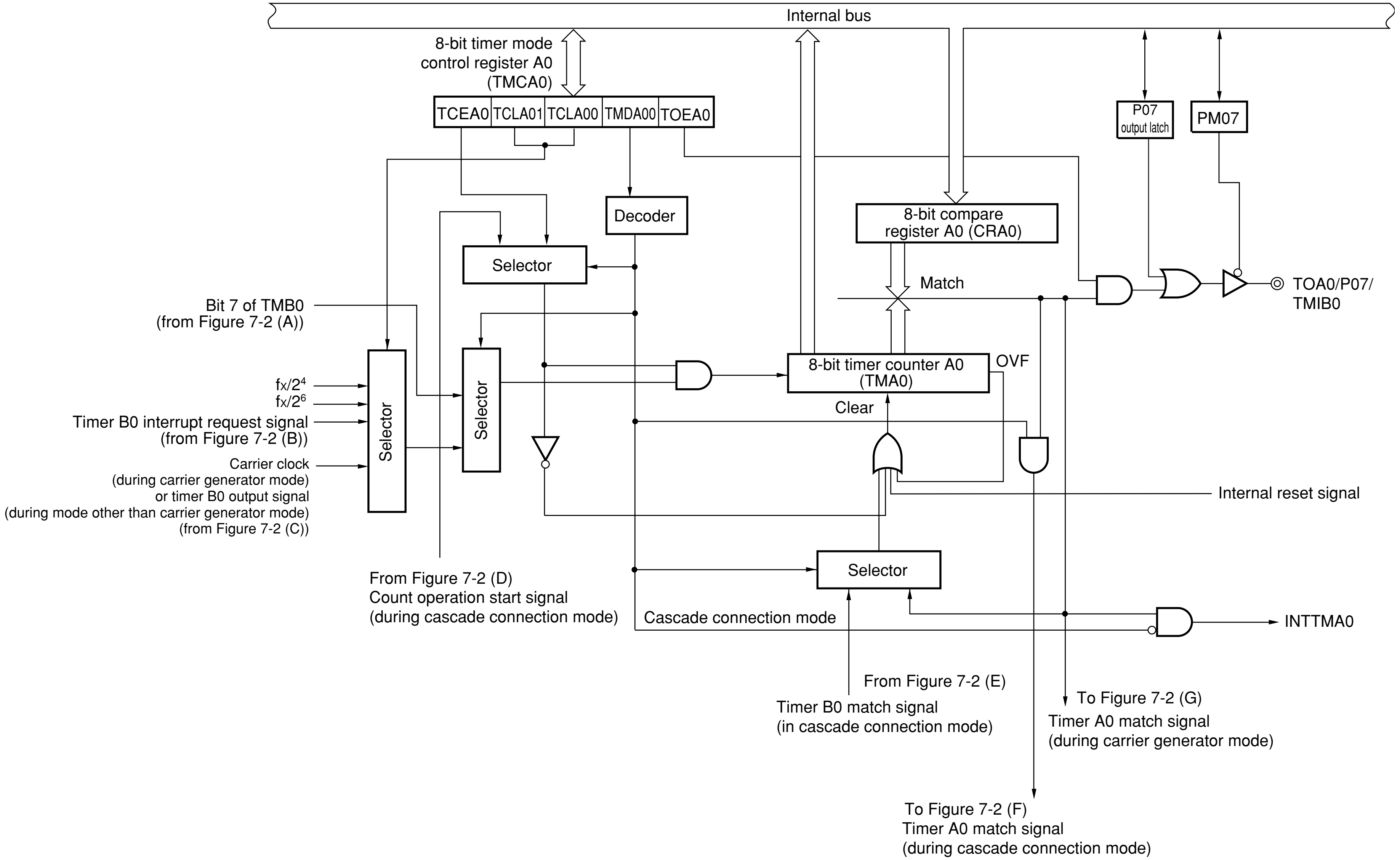
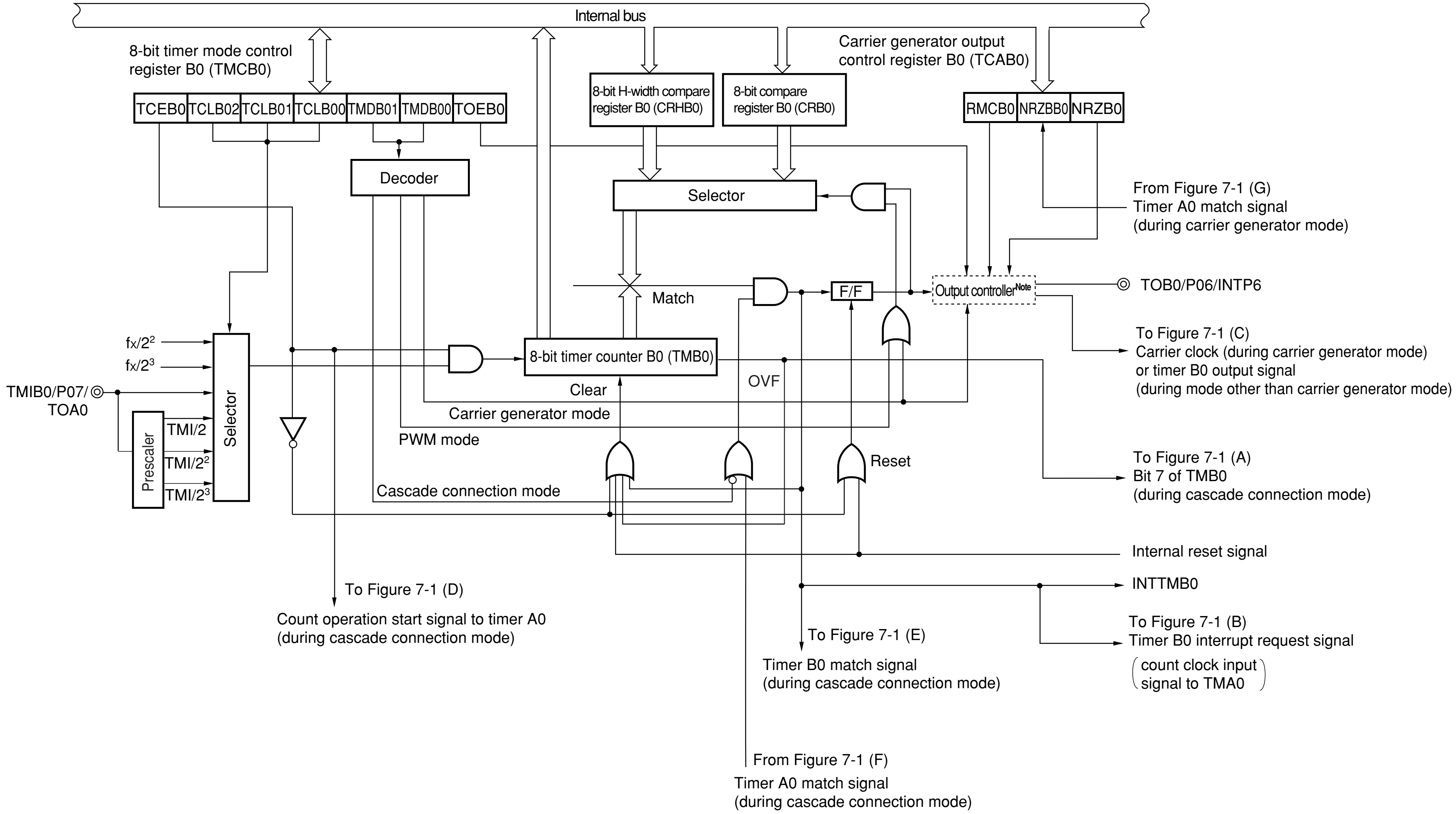
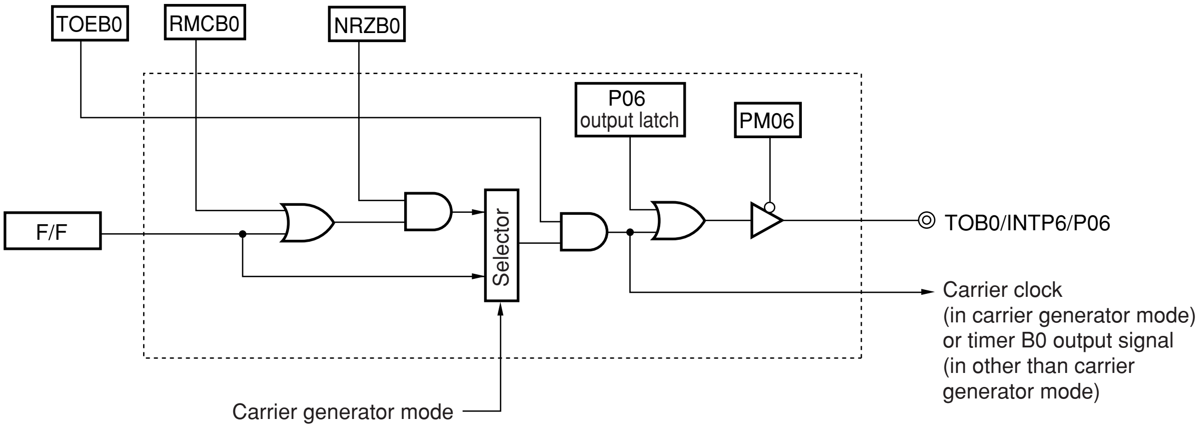


Figure 7-2. Block Diagram of Timer B0


Internal bus
8-bit timer mode control register B0 (TMCB0)
TCEB0
TCLB02
TCLB01
TCLB00
TMDB01
TMDB00
TOEB0
8-bit H-width compare register B0 (CRHB0)
8-bit compare register B0 (CRB0)
Carrier generator output control register B0 (TCAB0)
RMCB0
NRZBB0
NRZB0
Decoder
Selector
From Figure 7-1 (G)
Timer A0 match signal
(during carrier generator mode)
Match
F/F
Output controller Note
TOB0/P06/INTP6
To Figure 7-1 (C)
Carrier clock (during carrier generator mode)
or timer B0 output signal
(during mode other than carrier generator mode)
fx/2²
fx/2³
TMIB0/P07/TOA0
Prescaler
TMI/2
TMI/2²
TMI/2³
Selector
8-bit timer counter B0 (TMB0)
Clear
OVF
Carrier generator mode
PWM mode
Reset
Cascade connection mode
To Figure 7-1 (A)
Bit 7 of TMB0
(during cascade connection mode)
Internal reset signal
INTTMB0
To Figure 7-1 (D)
Count operation start signal to timer A0
(during cascade connection mode)
To Figure 7-1 (E)
Timer B0 match signal
(during cascade connection mode)
To Figure 7-1 (B)
Timer B0 interrupt request signal
(count clock input signal to TMA0)
From Figure 7-1 (F)
Timer A0 match signal
(during cascade connection mode)

Note For details, see **Figure 7-3**.

Figure 7-3. Block Diagram of Output Controller (Timer B0)



(1) 8-bit compare register A0 (CRA0)

This 8-bit register is used to continually compare the value set to CRA0 with the count value in 8-bit timer counter A0 (TMA0) and to generate an interrupt request (INTTMA0) when a match occurs.

★ CRA0 is written by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes CRA0 undefined.

Caution CRA0 cannot be used in PWM output mode.

(2) 8-bit compare register B0 (CRB0)

This 8-bit register is used to continually compare the value set to CRB0 with the count value in 8-bit timer counter B0 (TMB0) and to generate an interrupt request (INTTMB0) when a match occurs. When connected to TMA0 via a cascade connection and used as a 16-bit timer, the interrupt request (INTTMB0) occurs only when matches occur simultaneously between CRA0 and TMA0 and between CRB0 and TMB0 (INTTMA0 does not occur).

★ CRB0 is written by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes CRB0 undefined.

(3) 8-bit H width compare register B0 (CRHB0)

During carrier generator mode or PWM output mode, the high-level width of timer output is set by writing a value to CRHB0.

★ CRHB0 is written by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes CRHB0 undefined.

(4) 8-bit timer counters A0 and B0 (TMA0 and TMB0)

These are 8-bit registers that are used to count the count pulse.

TMA0 and TMB0 are read by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TMA0 and TMB0 to 00H.

TMA0 and TMB0 are cleared to 00H under the following conditions.

(a) Discrete mode

(i) TMA0

- After reset
- When TCEA0 (bit 7 of 8-bit timer mode control register A0 (TMCA0)) is cleared to 0
- When a match occurs between TMA0 and CRA0
- When the TMA0 count value overflows

(ii) TMB0

- After reset
- When TCEB0 (bit 7 of 8-bit timer mode control register B0 (TMCB0)) is cleared to 0
- When a match occurs between TMB0 and CRB0
- When the TMB0 count value overflows

(b) Cascade connection mode (TMA0 and TMB0 are simultaneously cleared to 00H)

- After reset
- When the TCEB0 flag is cleared to 0
- When matches occur simultaneously between TMA0 and CRA0 and between TMB0 and CRB0
- When the TMA0 and TMB0 count values overflow simultaneously

(c) Carrier generator mode/PWM output mode (TMB0 only)

- After reset
- When the TCEB0 flag is cleared to 0
- When a match occurs between TMB0 and CRB0
- When a match occurs between TMB0 and CRHB0
- When the TMB0 count value overflows

7.3 Registers to Control 8-Bit Timer A0, B0

8-bit timer A0 and B0 are controlled by the following four registers.

- 8-bit timer mode control register A0 (TMCA0)
- 8-bit timer mode control register B0 (TMCB0)
- Carrier generator output control register B0 (TCAB0)
- Port mode register 0 (PM0)

(1) 8-bit timer mode control register A0 (TMCA0)

8-bit timer mode control register A0 (TMCA0) is used to control the timer A0 count clock setting and the operation mode setting.

TMCA0 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TMCA0 to 00H.

Figure 7-4. Format of 8-Bit Timer Mode Control Register A0

Symbol	$\boxed{7}$	6	5	4	3	2	1	0	Address	After reset	R/W
TMCA0	TCEA0	0	0	TCLA01	TCLA00	0	TMDA00	TOEA0	FF6DH	00H	R/W

TCEA0	Control of TMA0 count operation[Note 1]
0	Clears TMA0 count value and stops operation
1	Starts count operation

TCLA01	TCLA00	Selection of timer A0 count clock
0	0	$f_X/2^4$ (625 kHz)
0	1	$f_X/2^6$ (156 kHz)
1	0	Timer B0 match signal
1	1	Carrier clock (in carrier generator mode) or timer B0 output signal (in other than carrier generator mode)

TMDA00	TMDB01	TMDB00	Selection of operation mode for timer A0 and timer B0[Note 2]
0	0	0	8-bit timer counter mode (discrete mode)
1	0	1	16-bit timer counter mode (cascade connection mode)
0	1	1	Carrier generator mode
0	1	0	Timer B0: PWM output mode Timer A0: 8-bit timer counter mode
Other than above			Setting prohibited

TOEA0	Control of timer output
0	Output disabled (port mode)
1	Output enabled

Notes 1. Since the count operation is controlled by TCEB0 (bit 7 of TMCB0) in cascade connection mode, any setting for TCEA0 is ignored.

2. The operation mode selection is set to both the TMCA0 register and TMCB0 register.

Caution In cascade connection mode, the timer B0 output signal is forcibly selected for the count clock.

Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 10.0 MHz.

(2) 8-bit timer mode control register B0 (TMCB0)

8-bit timer mode control register B0 (TMCB0) is used to control the timer B0 count clock setting and the operation mode setting.

TMCB0 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TMCB0 to 00H.

Figure 7-5. Format of 8-Bit Timer Mode Control Register B0

Symbol	$\boxed{7}$	6	5	4	3	2	1	$\boxed{0}$	Address	After reset	R/W
TMCB0	TCEB0	0	TCLB02	TCLB01	TCLB00	TMDB01	TMDB00	TOEB0	FF6EH	00H	R/W

TCEB0	Control of TMB0 count operation[Note 1]
0	Clears TMB0 count value and stops operation (the count value is also cleared for TMA0 during cascade connection mode)
1	Starts count operation (the count operation is also started for TMA0 during cascade connection mode)

TCLB02	TCLB01	TCLB00	Selection of timer B0 count clock
0	0	0	$f_X/2^2$ (2.5 MHz)
0	0	1	$f_X/2^3$ (1.25 MHz)
0	1	0	f_{TMI}
0	1	1	$f_{TMI}/2$
1	0	0	$f_{TMI}/2^2$
1	0	1	$f_{TMI}/2^3$
Other than above			Setting prohibited

TMDA00	TMDB01	TMDB00	Selection of operation mode for timer A0 and timer B0[Note 2]
0	0	0	8-bit timer counter mode (discrete mode)
1	0	1	16-bit timer counter mode (cascade connection mode)
0	1	1	Carrier generator mode
0	1	0	Timer B0: PWM output mode Timer A0: 8-bit timer counter mode
Other than above			Setting prohibited

TOEB0	Control of timer output
0	Output disabled (port mode)
1	Output enabled

Notes 1. Since the count operation is controlled by TCEB0 in cascade connection mode, any setting for TCEA0 (bit 7 of TMCA0) is ignored.

2. The operation mode selection is set to both the TMCA0 register and TMCB0 register.

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{TMI}: Input frequency from TMIB0 pin

3. The parenthesized values apply to operation at f_X = 10.0 MHz.

(3) Carrier generator output control register B0 (TCAB0)

This register is used to set the timer output data during carrier generator mode.

TCAB0 is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TCAB0 to 00H.

★ **Figure 7-6. Format of Carrier Generator Output Control Register B0**

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TCAB0	0	0	0	0	0	RMCB0	NRZBB0	NRZB0	FF6FH	00H	R/W

RMCB0	Control of remote control output
0	When NRZB0 = 1, a carrier clock is output to TOB0/INTP6/P06 pin
1	When NRZB0 = 1, high-level signal is output to TOB0/INTP6/P06 pin

NRZBB0	This is the bit that stores the next data to be output to NRZB0. Data is transferred to NRZB0 at the rising edge of the timer A0 match signal. Input the necessary value in NRZBB0 in advance by program.

NRZB0	No return zero data
0	A low-level signal is output to TOB0/INTP6/P06 pin (carrier clock is stopped)
1	A carrier clock or high-level signal is output to TOB0/INTP6/P06 pin

Cautions 1. TCAB0 cannot be set by a 1-bit memory manipulation instruction. Be sure to use an 8-bit memory manipulation instruction to set TCAB0.

2. The NRZB0 flag can be written only when carrier generator output is stopped (TOEB0 = 0). The data cannot be overwritten when TOEB0 = 1.

(4) Port mode register 0 (PM0)

This register is used to set the input/output mode of port 0 in 1-bit units.

When using the P06/INTP6/TOB0 pin as a timer B0 output, set the PM06 and P06 output latch to 0.

When using the P07/TOA0/TMIB0 pin as a timer B0 input, set the PM07 to 1, and when using as a timer A0 output, set the PM07 and P07 output latch to 0.

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

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

Figure 7-7. Format of Port Mode Register 0

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00	FF20H	FFH	R/W

PM0n	I/O mode of P0n pin (n = 0 to 7)
0	Output mode (output buffer is on)
1	Input mode (output buffer is off)

7.4 8-Bit Timer A0, B0 Operation

7.4.1 Operation as 8-bit timer counter

Timer A0 and timer B0 can independently be used as an 8-bit timer counter.

The following modes can be used for the 8-bit timer counter.

- Interval timer with 8-bit resolution
- External event counter with 8-bit resolution (timer B0 only)
- Square wave output with 8-bit resolution

(1) Operation as interval timer with 8-bit resolution

The interval timer with 8-bit resolution repeatedly generates an interrupt at a time interval specified by the count value preset in 8-bit compare register n0 (CRn0).

To operate 8-bit timer n0 as an interval timer, settings must be made in the following sequence.

<1> Disable operation of 8-bit timer counter n0 (TMn0) (TCEn0 = 0).
<2> Disable timer output of TOn0 (TOEn0 = 0).
★ <3> Set the count clock for timer n0 (see **Tables 7-3** and **7-4**).
<4> Set the operation mode of timer n0 to 8-bit timer counter mode (see **Figures 7-4** and **7-5**).
★ <5> Set a count value in CRn0.
<6> Enable the operation of TMn0 (TCEn0 = 1).

When the count value of 8-bit timer counter n0 (TMn0) matches the value set in CRn0, TMn0 is cleared to 00H and continues counting. At the same time, an interrupt request signal (INTTMn0) is generated.

Tables 7-3 and 7-4 show interval time, and Figures 7-8 to 7-13 show the timing of the interval timer operation.

Caution Be sure to stop the timer operation before overwriting the count clock with different data.

Remark n = A, B

Table 7-3. Interval Time of Timer A0

TCLA01	TCLA00	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	2^4/fx (1.6 μs)	2^{12}/fx (410 μs)	2^4/fx (1.6 μs)
0	1	2^6/fx (6.4 μs)	2^{14}/fx (1.64 ms)	2^6/fx (6.4 μs)
1	0	Input cycle of timer B0 match signal	Input cycle of timer B0 match signal × 2^8	Input cycle of timer B0 match signal
1	1	Input cycle of timer B0 output	Input cycle of timer B0 output × 2^8	Input cycle of timer B0

Remarks 1. fx: Main system clock oscillation frequency

2. The parenthesized values apply to operation at fx = 10.0 MHz

Table 7-4. Interval Time of Timer B0

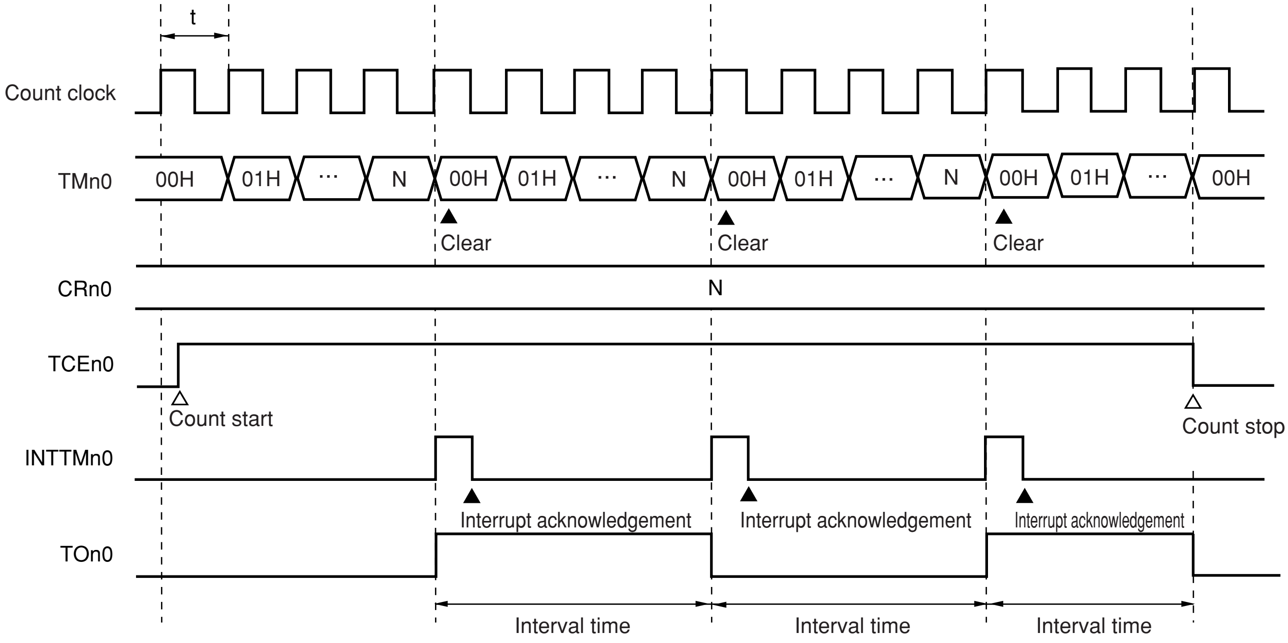
TCLB02	TCLB01	TCLB00	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	0	2^2/fx (0.4 μs)	2^{10}/fx (102 ms)	2^2/fx (0.4 μs)
0	0	1	2^3/fx (0.8 μs)	2^{11}/fx (205 ms)	2^3/fx (0.8 μs)
0	1	0	f_{TMI} input cycle	f_{TMI} input cycle × 2^8	f_{TMI} input cycle
0	1	1	$f_{TMI}/2$ input cycle	$f_{TMI}/2$ input cycle × 2^8	$f_{TMI}/2$ input cycle
1	0	0	$f_{TMI}/2^2$ input cycle	$f_{TMI}/2^2$ input cycle × 2^8	$f_{TMI}/2^2$ input cycle
1	0	1	$f_{TMI}/2^3$ input cycle	$f_{TMI}/2^3$ input cycle × 2^8	$f_{TMI}/2^3$ input cycle

Remarks 1. fx: Main system clock oscillation frequency

2. f_{TMI}: Input frequency from TMIB0 pin

3. The parenthesized values apply to operation at fx = 10.0 MHz

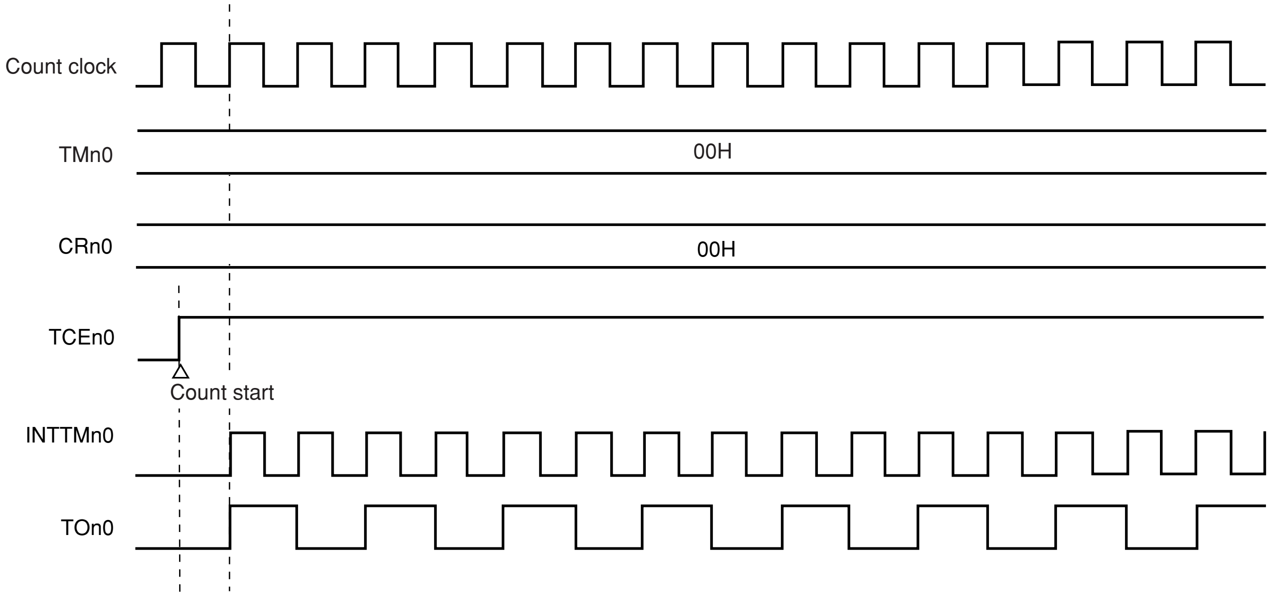
Figure 7-8. Timing of Interval Timer Operation with 8-Bit Resolution (Basic Operation)



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

2. n = A, B

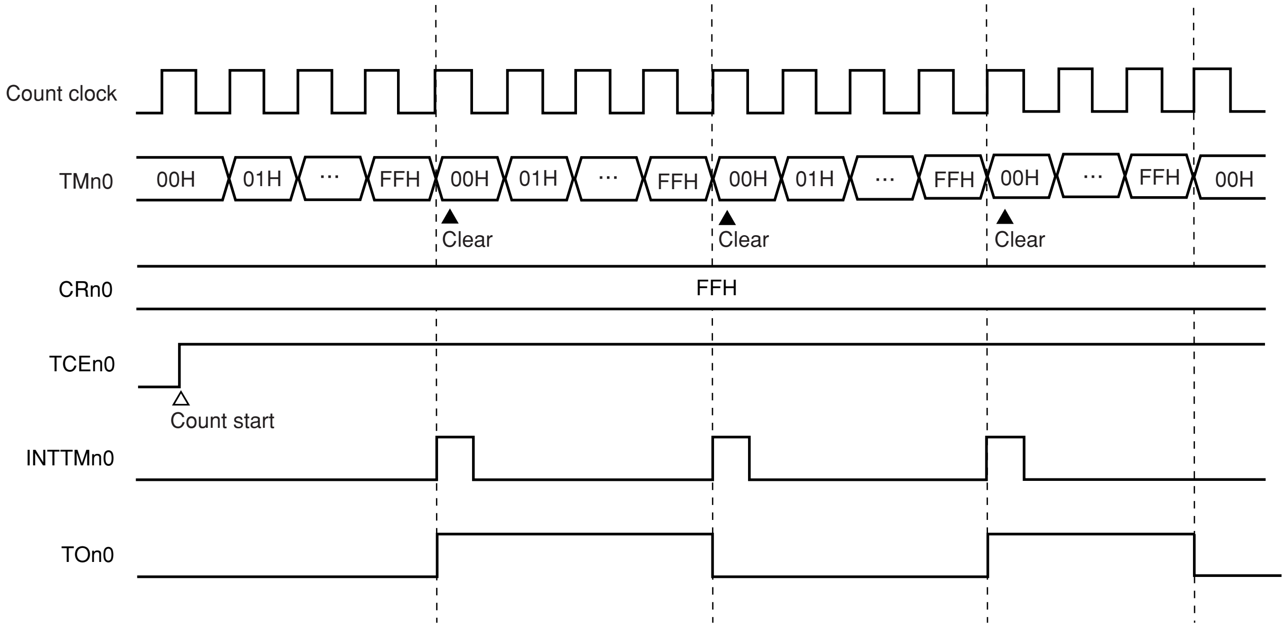
Figure 7-9. Timing of Interval Timer Operation with 8-Bit Resolution (When CRn0 Is Set to 00H)



Remark n = A, B

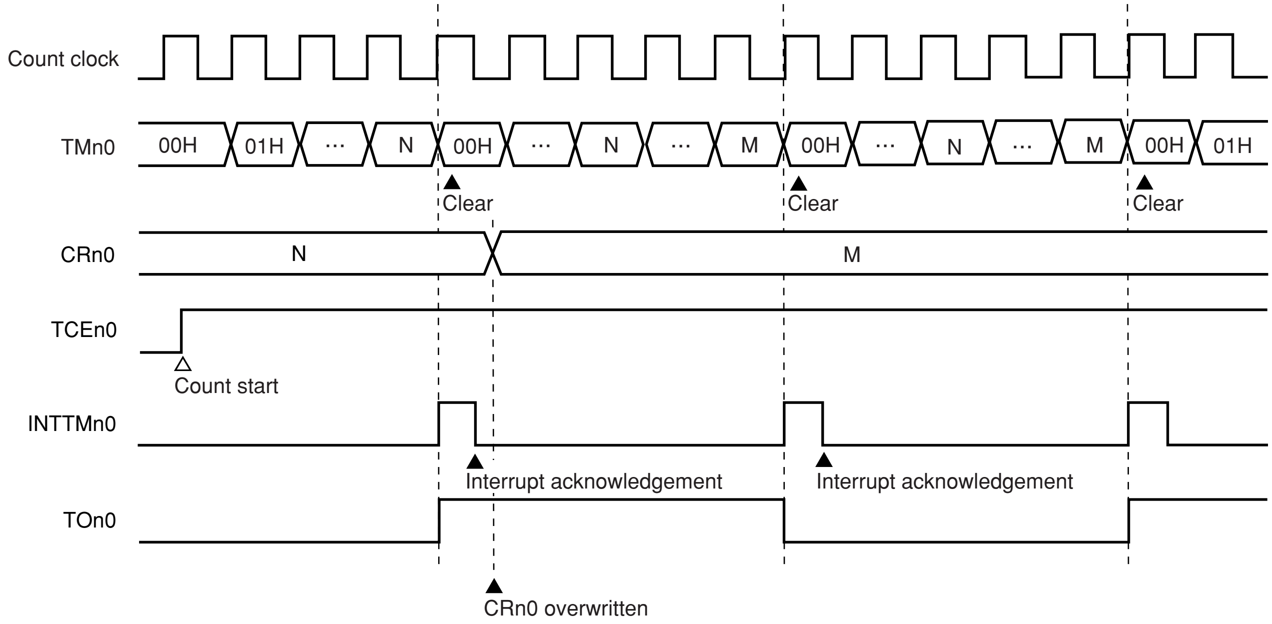
Figure 7-10. Timing of Interval Timer Operation with 8-Bit Resolution (When CRn0 Is Set to FFH)

★



Remark n = A, B

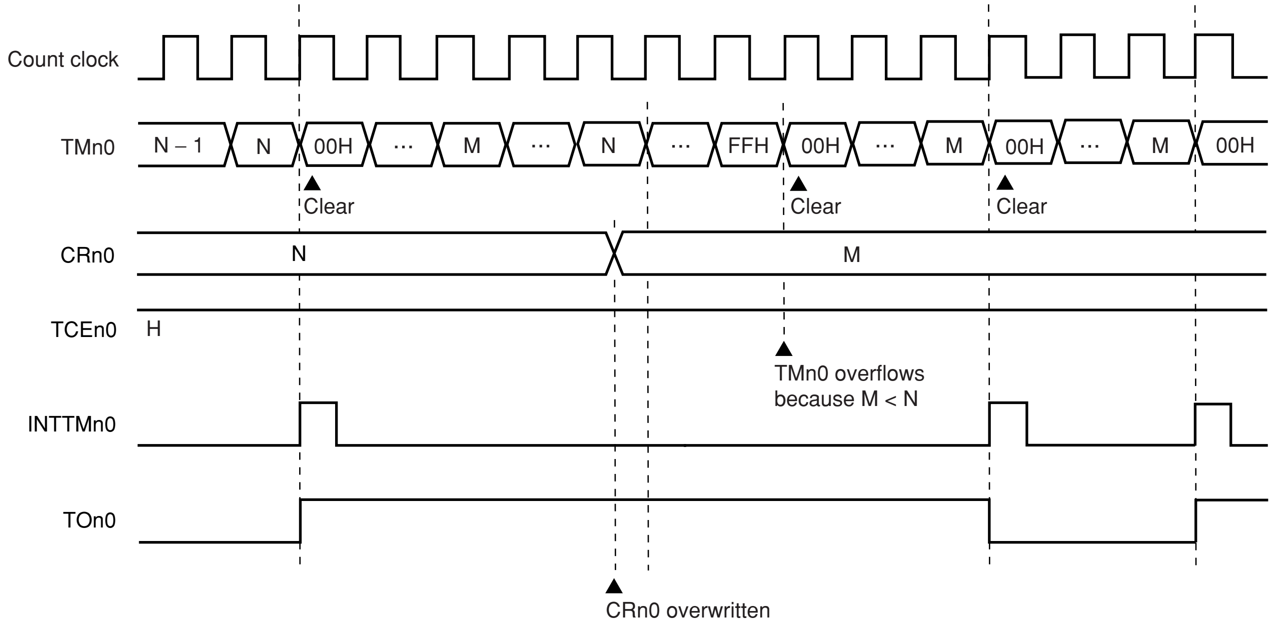
Figure 7-11. Timing of Interval Timer Operation with 8-Bit Resolution (When CRn0 Changes from N to M (N < M))



Remarks 1. n = A, B

2. 00H ≤ N < M ≤ FFH

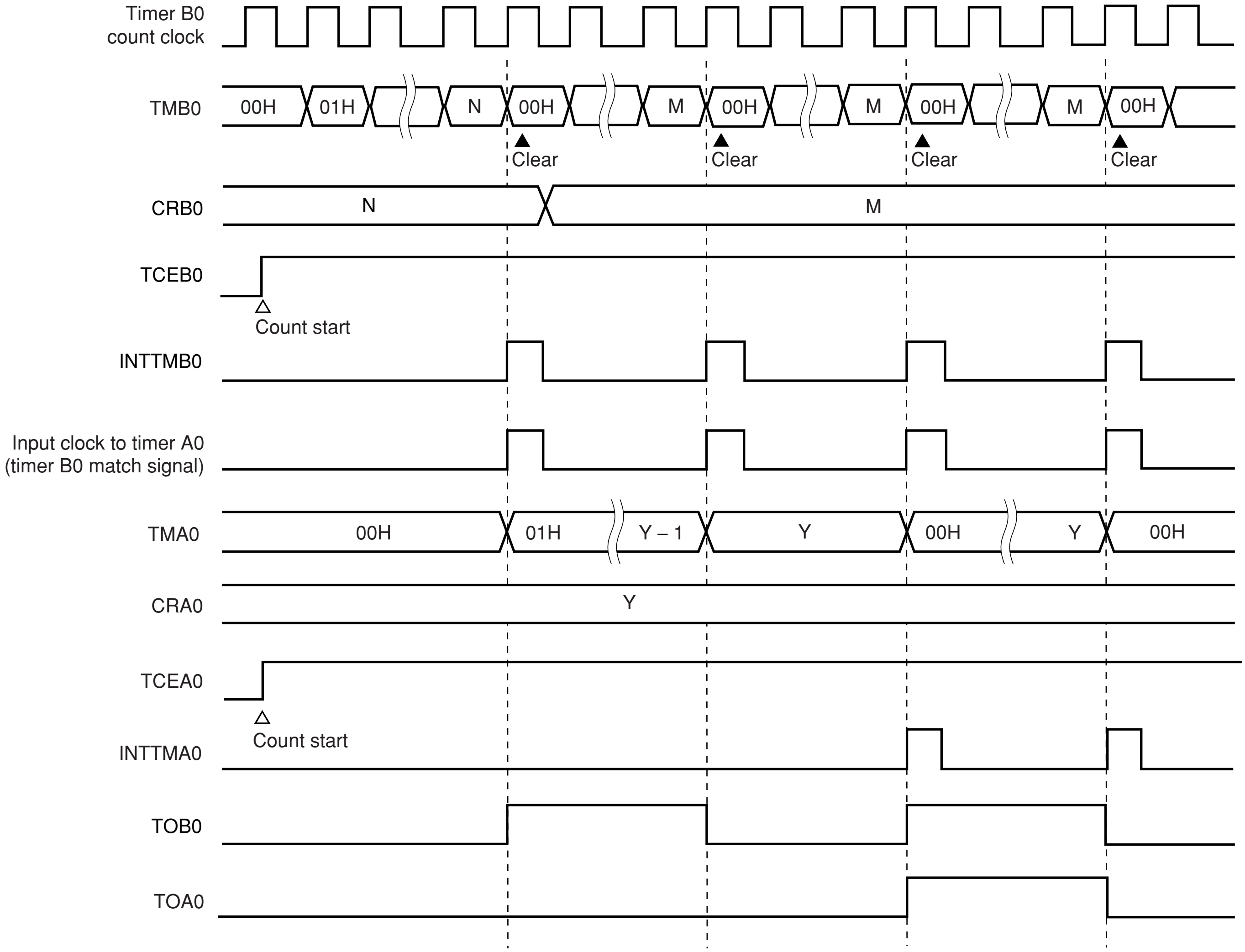
Figure 7-12. Timing of Interval Timer Operation with 8-Bit Resolution (When CRn0 Changes from N to M (N > M))



Remarks 1. n = A, B

2. 00H ≤ M < N ≤ FFH

Figure 7-13. Timing of Interval Timer Operation with 8-Bit Resolution (When Timer B0 Match Signal Is Selected for Timer A0 Count Clock)



Remarks 1. n = A, B

2. N, M, Y = 00H to FFH

(2) Operation as external event counter with 8-bit resolution (timer B0 only)

The external event counter counts the number of external clock pulses input to the TMIB0/P07/TOA0 pin by using 8-bit timer counter B0 (TMB0).

To operate timer B0 as an external event counter, settings must be made in the following sequence.

<1> Disable operation of 8-bit timer counter B0 (TMB0) (TCEB0 = 0).
<2> Disable timer output of TOB0 (TOEB0 = 0).
<3> Set P07 to input mode (PM07 = 1).
<4> Select the external input clock for timer B0 (see **Table 7-4**).
<5> Set the operation mode of timer B0 to 8-bit timer counter mode (see **Figures 7-4** and **7-5**).
<6> Set a count value in CRB0.
<7> Enable the operation of TMB0 (TCEB0 = 1).

Caution This operation only applies to timer B0. Timer A0 cannot be used as an external event counter because it does not have a timer input pin.

Each time the valid edge is input, the value of TMB0 is incremented.

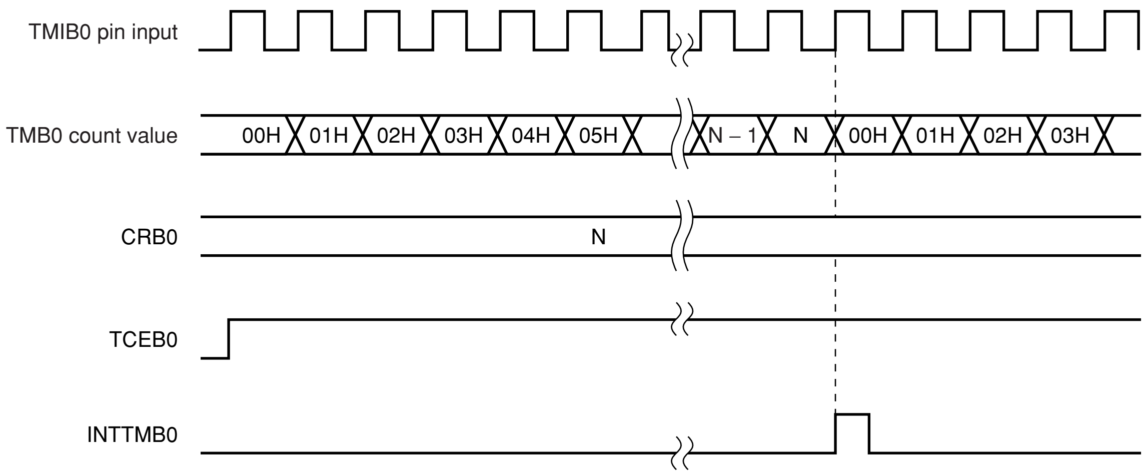
When the count value of TMB0 matches the value set in CRB0, TMB0 is cleared to 00H and continues counting.

At the same time, an interrupt request signal (INTTMB0) is generated.

Figure 7-14 shows the timing of the external event counter operation.

Caution Be sure to stop the timer operation before overwriting the count clock with different data.

Figure 7-14. Timing of Operation of External Event Counter with 8-Bit Resolution



Remark N = 00H to FFH

(3) Operation as square-wave output with 8-bit resolution

Square waves of any frequency can be output at an interval specified by the value preset in 8-bit compare register n0 (CRn0).

To operate timer n0 for square-wave output, settings must be made in the following sequence.

<1> Disable operation of 8-bit timer counter n0 (TMn0) (TCEn0 = 0).
★ <2> Disable timer output of TOn0 (TOEn0 = 0).
★ <3> Set the count clock for timer n0 (see **Table 7-5**).
<4> Set the operation mode of timer n0 to 8-bit timer counter mode (see **Figures 7-4** and **7-5**).
★ <5> Set a count value in CRn0.
<6> Enable output of TOn0 (TOEn0 = 1).
<7> In the case of timer A0, set P07 to output mode (PM07 = 0).
In the case of timer B0, set P06 to output mode (PM06 = 0).
<8> In the case of timer A0, set the output latches of P07 to 0.
In the case of timer B0, set the output latches of P06 to 0.
<9> Enable the operation of TMn0 (TCEn0 = 1).

When the count value of TMn0 matches the value set in CRn0, the TOn0 pin output status will be inverted. Through application of this mechanism, square waves of any frequency can be output. As soon as a match occurs, TMn0 is cleared to 00H and continues counting. At the same time, an interrupt request signal (INTTMn0) is generated. The square-wave output is cleared to 0 by setting TCEn0 to 0.

Table 7-5 shows the square-wave output range, and Figure 7-15 shows the timing of square-wave output.

Caution Be sure to stop the timer operation before overwriting the count clock with different data.

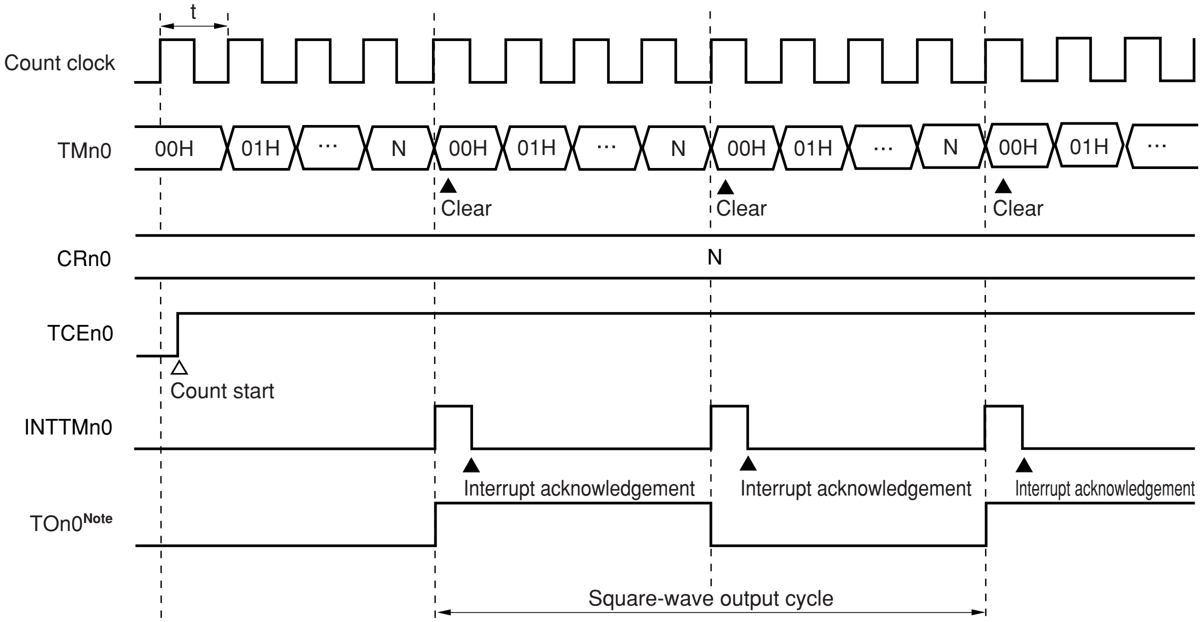
Remark n = A, B

Table 7-5. Square-Wave Output Range of Timer B0

TCLB02	TCLB01	TCLB00	Minimum Pulse Width	Maximum Pulse Width	Resolution
0	0	0	$2^2/f_X$ (0.4 μs)	$2^{10}/f_X$ (102 ms)	$2^2/f_X$ (0.4 μs)
0	0	1	$2^3/f_X$ (0.8 μs)	$2^{11}/f_X$ (205 ms)	$2^3/f_X$ (0.8 μs)
0	1	0	f_{TMI} input cycle	f_{TMI} input cycle × 2^8	f_{TMI} input cycle
0	1	1	$f_{TMI}/2$ input cycle	$f_{TMI}/2$ input cycle × 2^8	$f_{TMI}/2$ input cycle
1	0	0	$f_{TMI}/2^2$ input cycle	$f_{TMI}/2^2$ input cycle × 2^8	$f_{TMI}/2^2$ input cycle
1	0	1	$f_{TMI}/2^3$ input cycle	$f_{TMI}/2^3$ input cycle × 2^8	$f_{TMI}/2^3$ input cycle

Remarks 1. f_X: Main system clock oscillation frequency
2. f_{TMI}: Input frequency from TMIB0 pin
3. The parenthesized values apply to operation at f_X = 10.0 MHz

Figure 7-15. Timing of Square-Wave Output with 8-Bit Resolution



Note The initial value of TOn0 is low level when output is enabled (TOEn0 = 1).

Remarks 1. Square-wave output cycle = 2 (N + 1) × t: N = 00H to FFH

2. n = A, B

7.4.2 Operation as 16-bit timer counter

Timer A0 and timer B0 can be used as a 16-bit timer counter using cascade connection. In this case, 8-bit timer counter A0 (TMA0) is the higher 8 bits and 8-bit timer counter B0 (TMB0) is the lower 8 bits. 8-bit timer B0 controls reset and clear.

The following modes can be used for the 16-bit timer counter.

- Interval timer with 16-bit resolution
- External event counter with 16-bit resolution
- Square-wave output with 16-bit resolution

(1) Operation as interval timer with 16-bit resolution

The interval timer with 16-bit resolution repeatedly generates an interrupt at a time interval specified by the count value preset in 8-bit compare register A0 (CRA0) and 8-bit compare register B0 (CRB0).

To operate as an interval timer with 16-bit resolution, settings must be made in the following sequence.

<1> Disable operation of 8-bit timer counter A0 (TMA0) and 8-bit timer counter B0 (TMB0) (TCEA0 = 0, TCEB0 = 0).

<2> Disable timer output of TOA0, TOB0 (TOEA0 = 0, TOEB0 = 0).

<3> Set the count clock for timer B0 (see **Table 7-6**).

<4> Set the operation mode of timer A0 and 8-bit timer B0 to 16-bit timer counter mode (see **Figures 7-4** and **7-5**).

<5> Set a count value in CRA0 and CRB0.

<6> Enable the operation of TMA0 and TMB0 (TCEB0 = 1[Note]).

Note Start and clear of the timer in the 16-bit timer counter mode are controlled by TCEB0 (the value of TCEA0 is invalid).

When the count values of TMA0 and TMB0 match the values set in CRA0 and CRB0 respectively, both TMA0 and TMB0 are simultaneously cleared to 00H and counting continues. At the same time, an interrupt request signal (INTTMB0) is generated (INTTMA0 is not generated).

Table 7-6 shows interval time, and Figure 7-16 shows the timing of the interval timer operation.

Cautions 1. Be sure to stop the timer operation before overwriting the count clock with different data.

2. In the 16-bit timer counter mode, TOA0 cannot be used. Be sure to set TOEA0 = 0 to disable TOA0 output.

Table 7-6. Interval Time with 16-Bit Resolution

TCLB02	TCLB01	TCLB00	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	0	$2^2/f_X$ (0.4 μs)	$2^{18}/f_X$ (26.2 ms)	$2^2/f_X$ (0.4 μs)
0	0	1	$2^3/f_X$ (0.8 μs)	$2^{19}/f_X$ (52.4 ms)	$2^3/f_X$ (0.8 μs)
0	1	0	f_{TMI} input cycle	f_{TMI} input cycle × 2^{16}	f_{TMI} input cycle
0	1	1	$f_{TMI}/2$ input cycle	$f_{TMI}/2$ input cycle × 2^{16}	$f_{TMI}/2$ input cycle
1	0	0	$f_{TMI}/2^2$ input cycle	$f_{TMI}/2^2$ input cycle × 2^{16}	$f_{TMI}/2^2$ input cycle
1	0	1	$f_{TMI}/2^3$ input cycle	$f_{TMI}/2^3$ input cycle × 2^{16}	$f_{TMI}/2^3$ input cycle

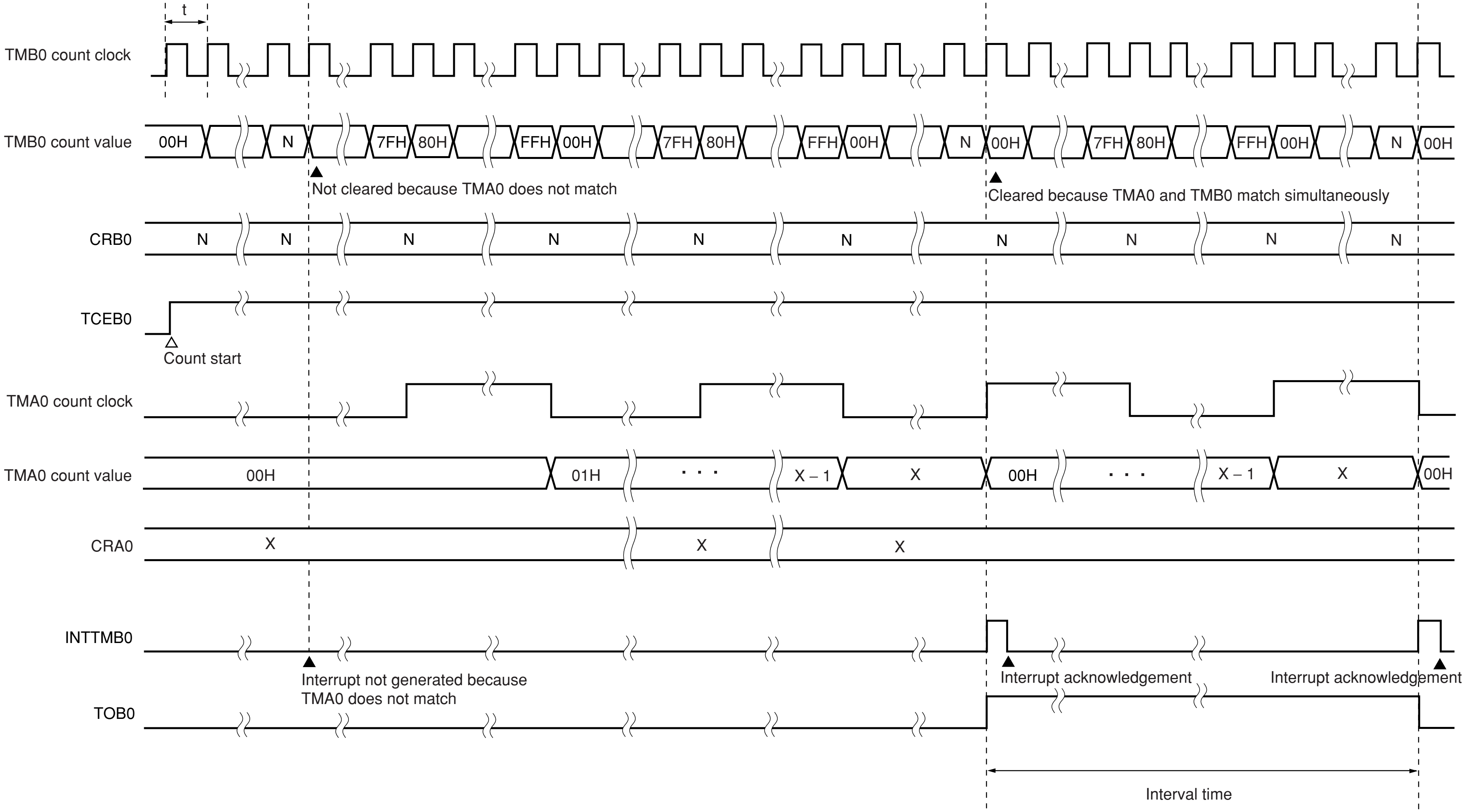
Remarks 1. f_X: Main system clock oscillation frequency

2. f_{TMI}: Input frequency from TMIB0 pin

3. The parenthesized values apply to operation at f_X = 10.0 MHz

★

Figure 7-16. Timing of Interval Timer Operation with 16-Bit Resolution



Remark Interval time = (256X + N + 1) × t: X = 00H to FFH, N = 00H to FFH

(2) Operation as external event counter with 16-bit resolution

The external event counter counts the number of external clock pulses input to the TMIB0/P07/TOA0 pin by TMA0 and TMB0.

To operate as an external event counter with 16-bit resolution, settings must be made in the following sequence.

★ <1> Disable operation of 8-bit timer counter A0 (TMA0) and 8-bit timer counter B0 (TMB0) (TCEA0 = 0, TCEB0 = 0).

★ <2> Disable timer output of TOA0 and TOB0 (TOEA0 = 0, TOEB0 = 0).

<3> Set P07 to input mode (PM07 = 1).

<4> Select the external input clock for timer B0 (see **Table 7-6**).

<5> Set the operation mode of timer A0 and timer B0 to 16-bit timer counter mode (see **Figures 7-4** and **7-5**).

<6> Set a count value in CRA0 and CRB0.

<7> Enable the operation of TMA0 and TMB0 (TCEB0 = 1[Note]).

Note Start and clear of the timer in the 16-bit timer counter mode are controlled by TCEB0 (the value of TCEA0 is invalid).

Each time the valid edge is input, the values of TMA0 and TMB0 are incremented.

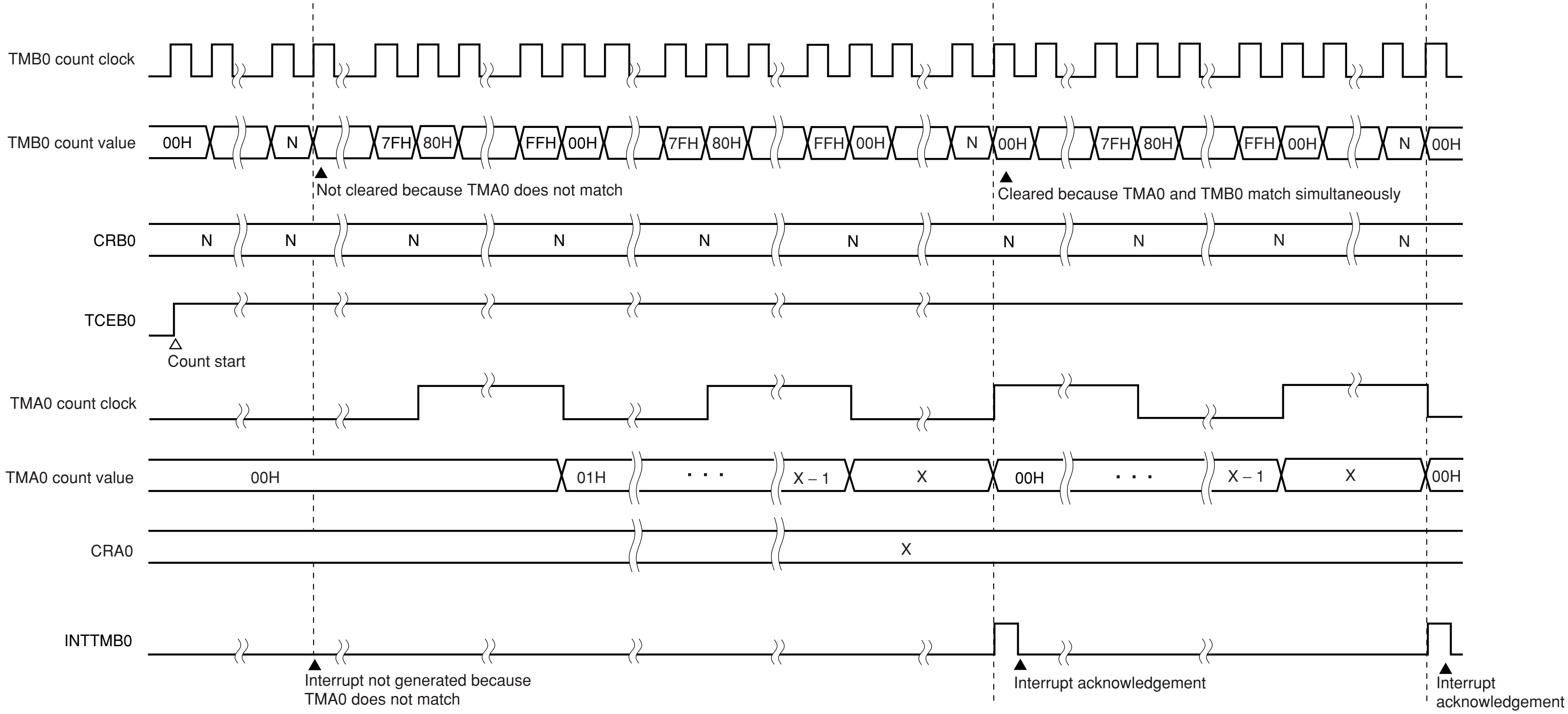
When the count values of TMA0 and TMB0 simultaneously match the values set in CRA0 and CRB0 respectively, both TMA0 and TMB0 are cleared to 00H and counting continues. At the same time, an interrupt request signal (INTTMB0) is generated (INTTMA0 is not generated).

Figure 7-17 shows the timing of the external event counter operation.

Caution Be sure to stop the timer operation before overwriting the count clock with different data.

★

Figure 7-17. Timing of External Event Counter Operation with 16-Bit Resolution


TMB0 count clock
TMB0 count value
00H
N
7FH
80H
FFH
00H
7FH
80H
FFH
00H
N
00H
7FH
80H
FFH
00H
N
00H
Not cleared because TMA0 does not match
Cleared because TMA0 and TMB0 match simultaneously
CRB0
N
TCEB0
Count start
TMA0 count clock
TMA0 count value
00H
01H
· · ·
X – 1
X
00H
· · ·
X – 1
X
00H
CRA0
X
INTTMB0
Interrupt not generated because TMA0 does not match
Interrupt acknowledgement
Interrupt acknowledgement

Remark X = 00H to FFH, N = 00H to FFH

(3) Operation as square-wave output with 16-bit resolution

Square waves of any frequency can be output at an interval specified by the count value preset in CRA0 and CRB0.

To operate as a square-wave output with 16-bit resolution, settings must be made in the following sequence.

★ <1> Disable operation of 8-bit timer counter A0 (TMA0) and 8-bit timer counter B0 (TMB0) (TCEA0 = 0, TCEB0 = 0).

★ <2> Disable timer output of TOA0 and TOB0 (TOEA0 = 0, TOEB0 = 0).

<3> Set a count clock for timer B0.

<4> Set the operation mode of timer A0 and timer B0 to 16-bit timer counter mode (see **Figures 7-4** and **7-5**).

<5> Set count values in CRA0 and CRB0.

<6> Set P06 to output mode (PM06 = 0) and P06 output latch to 0 and enable TOB0 output (TOEB0 = 1) (TOA0 cannot be used).

<7> Enable the operation of TMB0 (TCEB0 = 1**Note**).

Note Start and clear of the timer in the 16-bit timer counter mode are controlled by TCEB0 (the value of TCEA0 is invalid).

When the count values of TMA0 and TMB0 simultaneously match the values set in CRA0 and CRB0 respectively, the TOB0 pin output status will be inverted. Through application of this mechanism, square waves of any frequency can be output. As soon as a match occurs, TMA0 and TMB0 are cleared to 00H and counting continues. At the same time, an interrupt request signal (INTTMB0) is generated (INTTMA0 is not generated).

The square-wave output is cleared to 0 by setting TCEB0 to 0.

Table 7-7 shows the square-wave output range, and Figure 7-18 shows timing of square-wave output.

Cautions 1. Be sure to stop the timer operation before overwriting the count clock with different data.

2. In the 16-bit timer counter mode, TOA0 cannot be used. Be sure to set TOEA0 = 0 to disable TOA0 output.

★ **Table 7-7. Square-Wave Output Range with 16-Bit Resolution**

TCLB02	TCLB01	TCLB00	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	0	$2^2/f_X$ (0.4 μs)	$2^{18}/f_X$ (26.2 ms)	$2^2/f_X$ (0.4 μs)
0	0	1	$2^3/f_X$ (0.8 μs)	$2^{19}/f_X$ (52.4 ms)	$2^3/f_X$ (0.8 μs)
0	1	0	f_{TMI} input cycle	f_{TMI} input cycle × 2^{16}	f_{TMI} input cycle
0	1	1	$f_{TMI}/2$ input cycle	$f_{TMI}/2$ input cycle × 2^{16}	$f_{TMI}/2$ input cycle
1	0	0	$f_{TMI}/2^2$ input cycle	$f_{TMI}/2^2$ input cycle × 2^{16}	$f_{TMI}/2^2$ input cycle
1	0	1	$f_{TMI}/2^3$ input cycle	$f_{TMI}/2^3$ input cycle × 2^{16}	$f_{TMI}/2^3$ input cycle

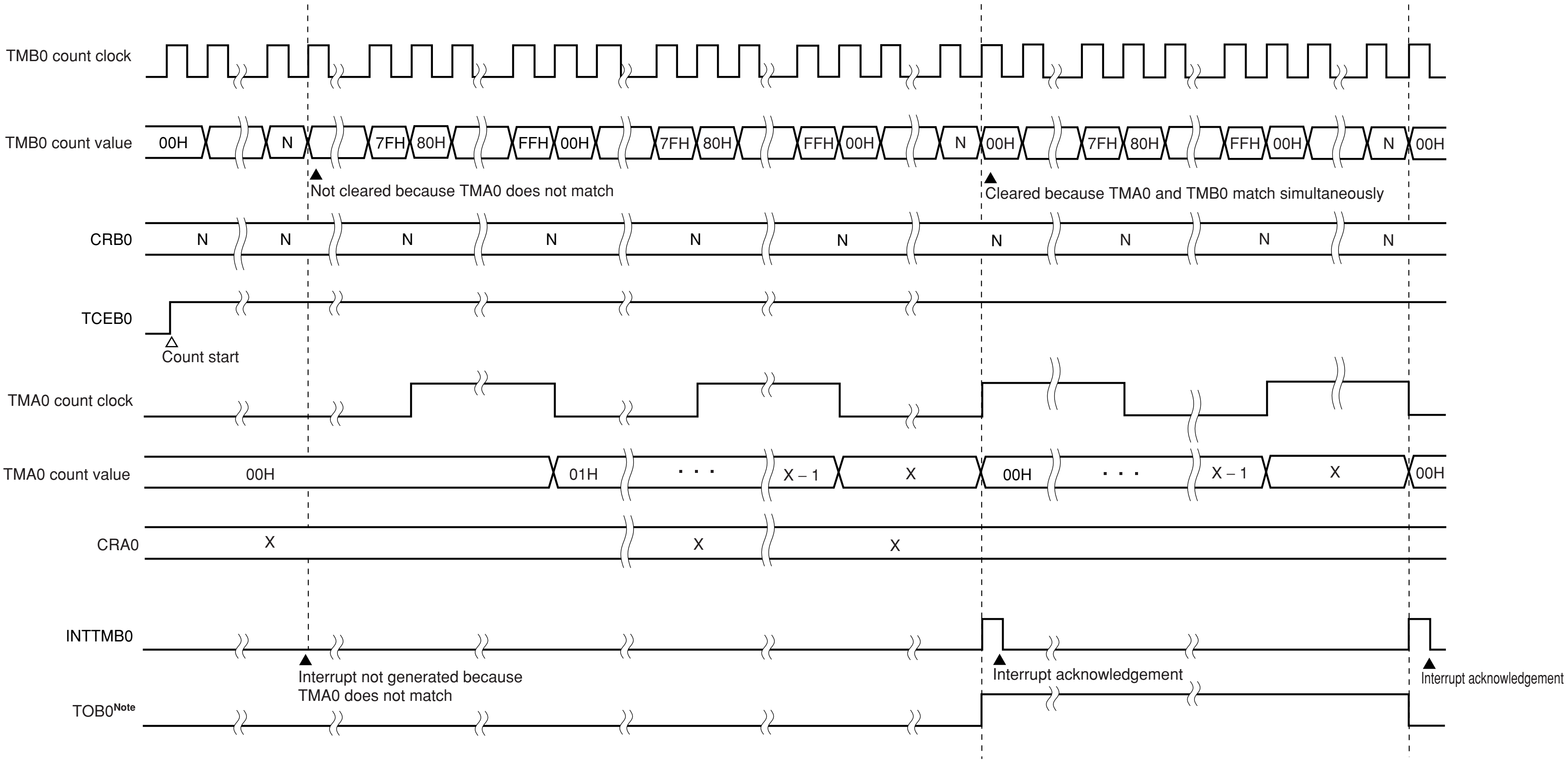
Remarks 1. f_X: Main system clock oscillation frequency

2. f_{TMI}: Input frequency from TMIB0 pin

3. The parenthesized values apply to operation at f_X = 10.0 MHz

★

Figure 7-18. Timing of Square-Wave Output with 16-Bit Resolution


TMB0 count clock
TMB0 count value
00H
N
7FH
80H
FFH
00H
7FH
80H
FFH
00H
N
00H
7FH
80H
FFH
00H
N
00H
Not cleared because TMA0 does not match
Cleared because TMA0 and TMB0 match simultaneously
CRB0
N
TCEB0
Count start
TMA0 count clock
TMA0 count value
00H
01H
· · ·
X – 1
X
00H
· · ·
X – 1
X
00H
CRA0
X
X
X
INTTMB0
Interrupt not generated because TMA0 does not match
Interrupt acknowledgement
Interrupt acknowledgement
TOB0Note

Note The initial value of TOB0 is low level when output is enabled (TOEB0 = 1).

Remark X = 00H to FFH, N = 00H to FFH

7.4.3 Operation as carrier generator

An arbitrary carrier clock generated by TMB0 can be output in the cycle set in TMA0.

To operate timer A0 and timer B0 as carrier generators, settings must be made in the following sequence.

★ <1> Disable operation of 8-bit timer counter A0 (TMA0) and 8-bit timer counter B0 (TMB0) (TCEA0 = 0, TCEB0 = 0).

★ <2> Disable timer output of TOA0 and TOB0 (TOEA0 = 0, TOEB0 = 0).

★ <3> Set the count clock for timer A0 and timer B0.

★ <4> Set the operation mode of timer A0 and timer B0 to carrier generator mode (see **Figures 7-4** and **7-5**).

★ <5> Set count values in CRA0, CRB0, and CRHB0.

<6> Set remote control output to carrier clock (RMCB0 (bit 2 of carrier generator output control register B0 (TCAB0)) = 0).
Input the required value to NRZBB0 (bit 1 of TCAB0) by program.
Input a value to NRZB0 (bit 0 of TCAB0) before it is reloaded from NRZBB0.

<7> Set P06 to output mode (PM06 = 0) and the P06 output latch to 0 and enable TOB0 output by setting TOEB0 to 1.

<8> Enable the operation of TMA0 and TMB0 (TCEA0 = 1, TCEB0 = 1).

The operation of the carrier generator is as follows.

<1> When the count value of TMB0 matches the value set in CRB0, an interrupt request signal (INTTMB0) is generated and output status of timer B0 is inverted, which makes the compare register switch from CRB0 to CRHB0.

<2> After that, when the count value of TMB0 matches the value set in CRHB0, an interrupt request signal (INTTMB0) is generated and output status of timer B0 is inverted again, which makes the compare register switch from CRHB0 to CRB0.

<3> The carrier clock is generated by repeating <1> and <2> above.

<4> When the count value of TMA0 matches the value set in CRA0, an interrupt request signal (INTTMA0) is generated. The rising edge of INTTMA0 is the data reload signal of NRZBB0 and is transferred to NRZB0.

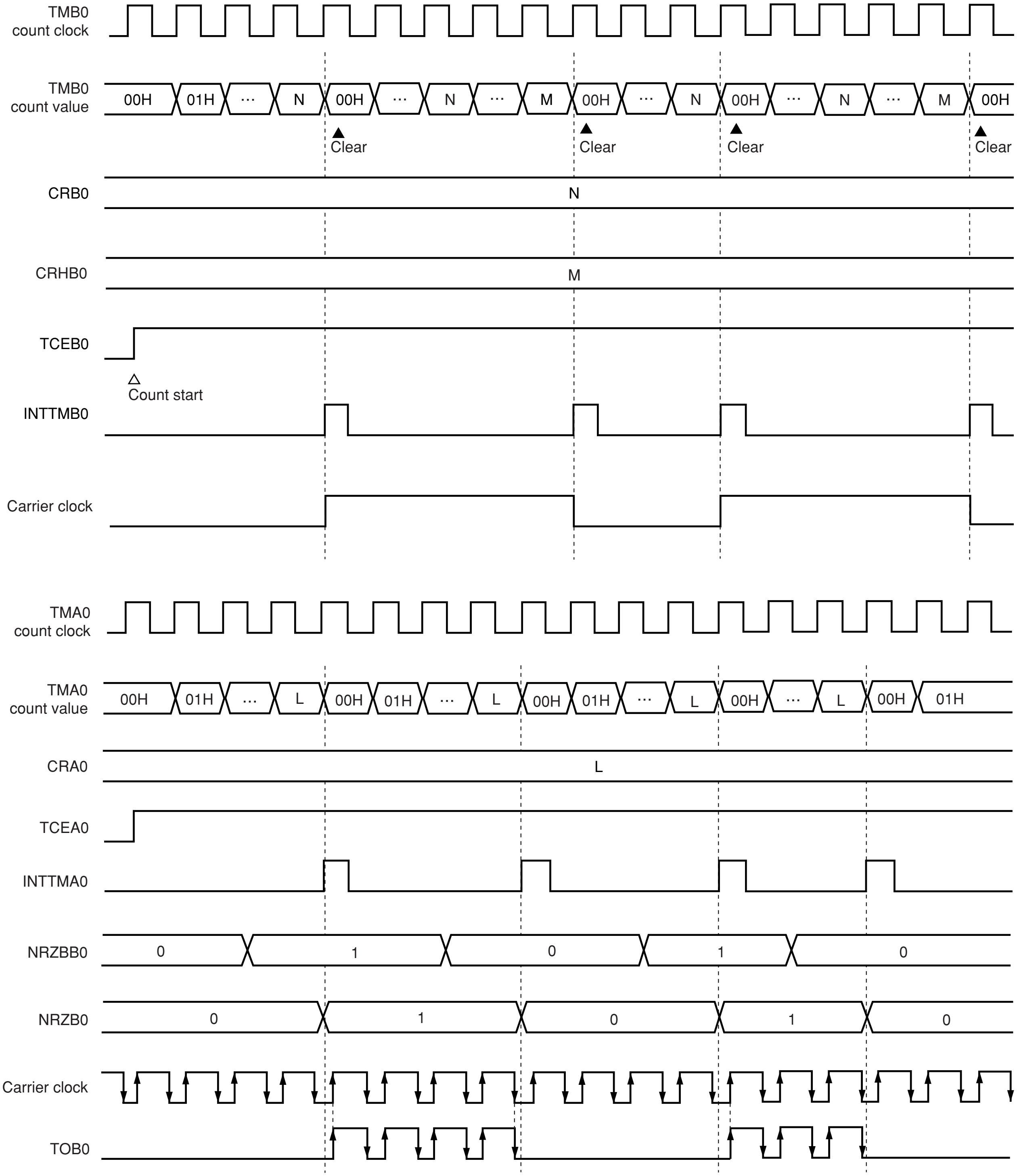
<5> When NRZB0 is 1, a carrier clock is output from TOB0 pin.

Cautions 1. TCAB0 cannot be set by a 1-bit memory manipulation instruction. Be sure to use an 8-bit memory manipulation instruction.

2. When setting the carrier generator operation again after stopping it once, reset NRZBB0 because the previous data is not retained. In this case also a 1-bit memory manipulation instruction cannot be used. Be sure to use an 8-bit memory manipulation instruction.

Figures 7-19 to 7-21 show the operation timing of the carrier generator.

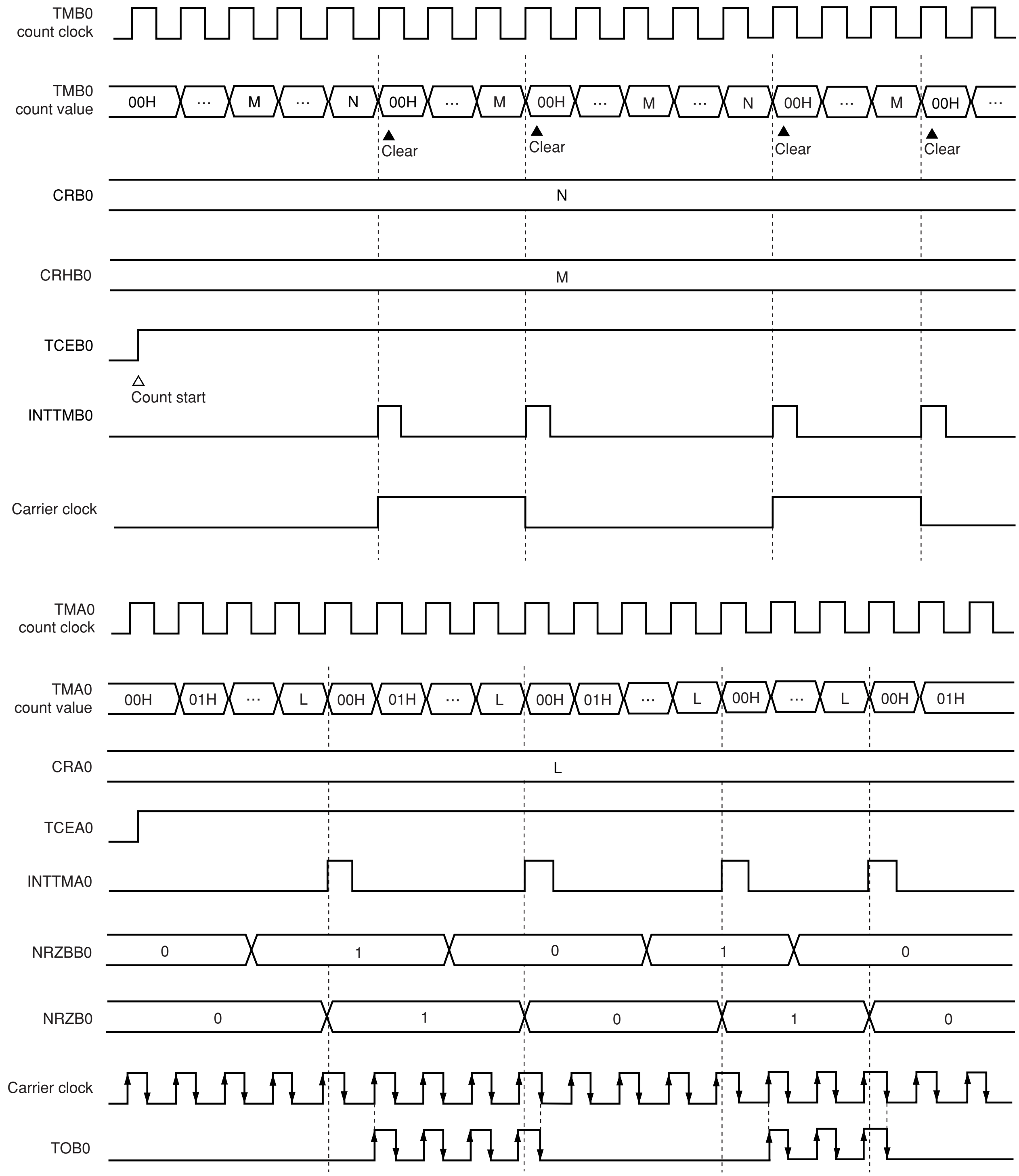
★ **Figure 7-19. Timing of Carrier Generator Operation (When CRB0 = N, CRHB0 = M (M > N))**



Remarks 1. 00H ≤ N < M ≤ FFH

2. L = 00H to FFH

★ **Figure 7-20. Timing of Carrier Generator Operation (When CRB0 = N, CRHB0 = M (M < N))**

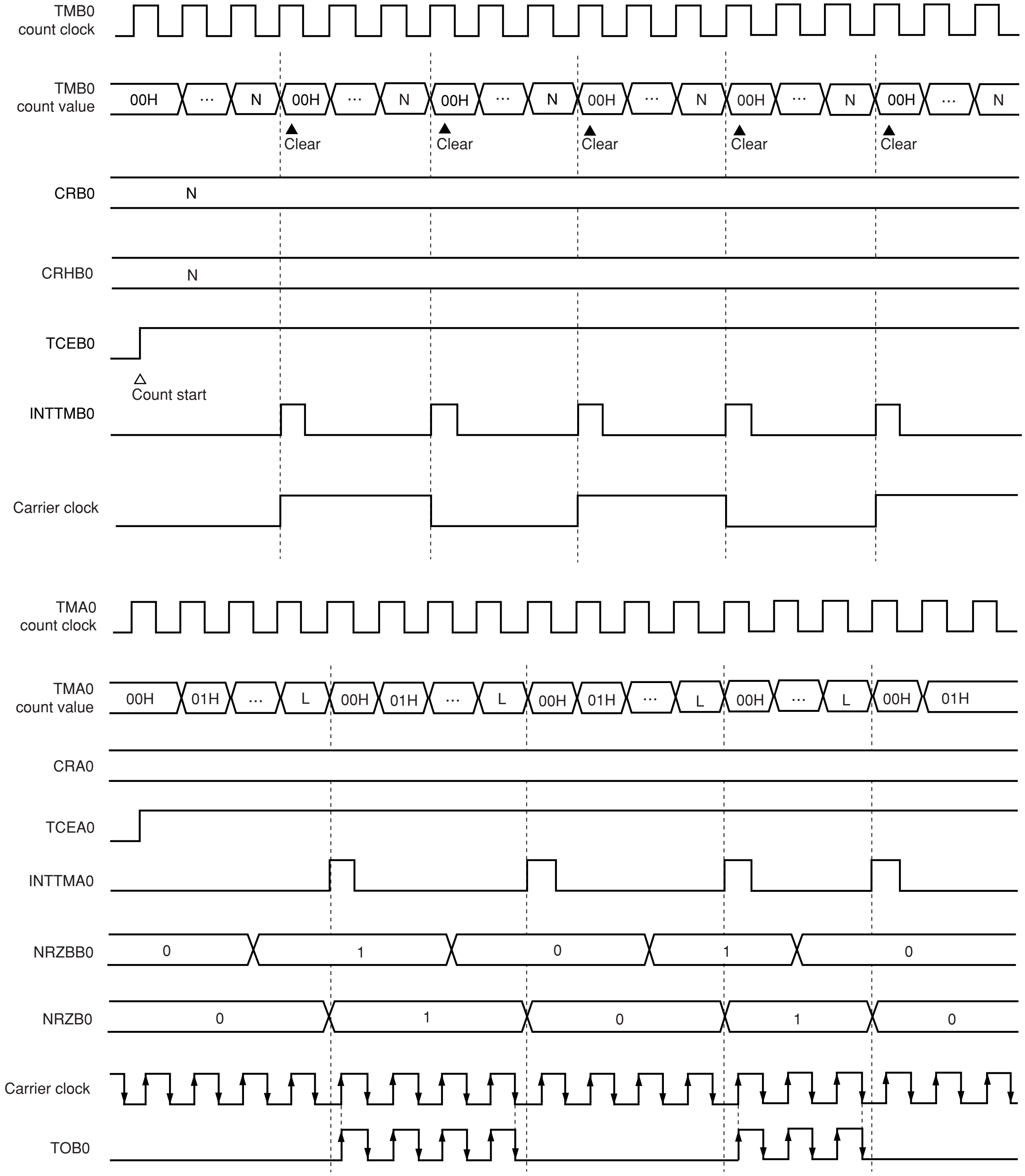


Remarks 1. 00H ≤ M < N ≤ FFH

2. L = 00H to FFH

★

Figure 7-21. Timing of Carrier Generator Operation (When CRB0 = CRHB0 = N)



Remark N, L = 00H to FFH

7.4.4 Operation as PWM output (timer B0 only)

In the PWM output mode, a pulse of any duty ratio can be output by setting a low-level width using CRB0 and a high-level width using CRHB0.

To operate timer B0 in PWM output mode, settings must be made in the following sequence.

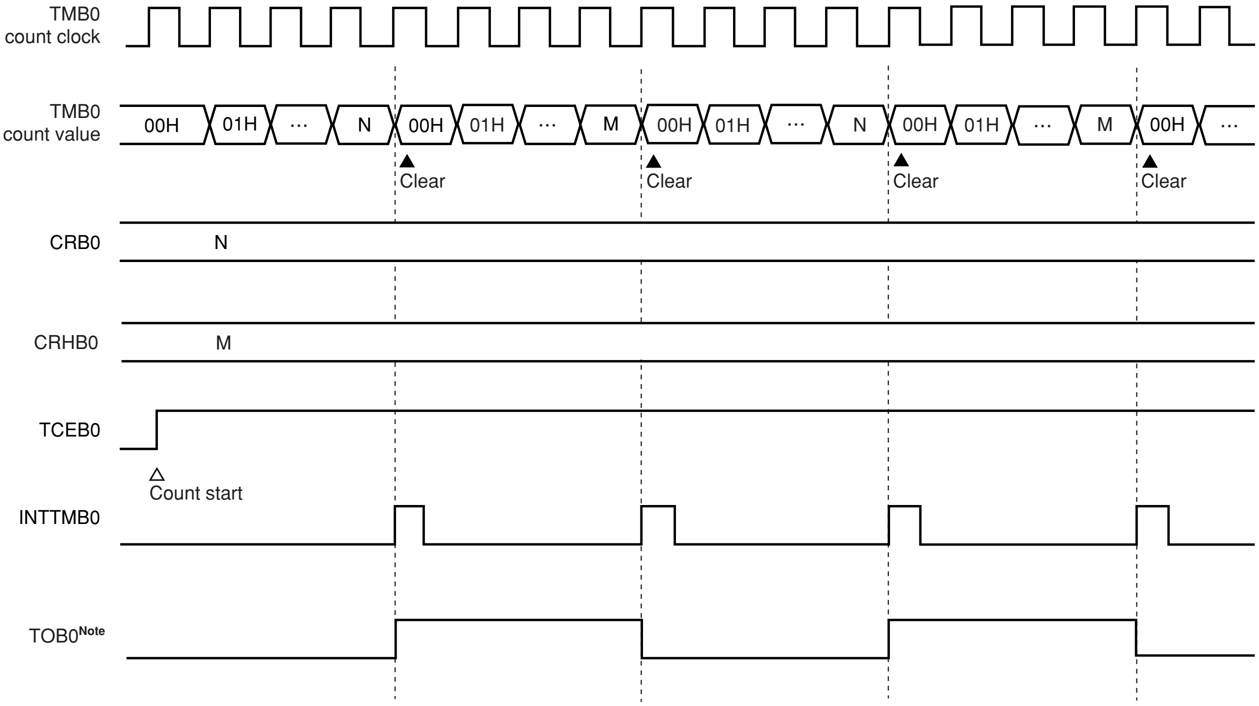
★ <1> Disable operation of 8-bit timer counter B0 (TMB0) (TCEB0 = 0).
<2> Disable timer output of TOB0 (TOEB0 = 0).
★ <3> Set the count clock for timer B0.
★ <4> Set the operation mode of timer B0 to PWM output mode (see **Figure 7-5**).
★ <5> Set count values in CRB0 and CRHB0.
<6> Set P06 to output mode (PM06 = 0) and the P06 output latch to 0 and enable timer output of TOB0 (TOEB0 = 1).
<7> Enable the operation of TMB0 (TCEB0 = 1).

The operation in the PWM output mode is as follows.

<1> When the count value of TMB0 matches the value set in CRB0, an interrupt request signal (INTTMB0) is generated and output status of timer B0 is inverted, which makes the compare register switch from CRB0 to CRHB0.
<2> A match between TMB0 and CRB0 clears the TMB0 value to 00H and then counting starts again.
<3> After that, when the count value of TMB0 matches the value set in CRHB0, an interrupt request signal (INTTMB0) is generated and output status of timer B0 is inverted again, which makes the compare register switch from CRHB0 to CRB0.
<4> A match between TMB0 and CRHB0 clears the TMB0 value to 00H and then counting starts again.

A pulse of any duty ratio is output by repeating <1> to <4> above. Figures 7-22 and 7-23 show the operation timing in the PWM output mode.

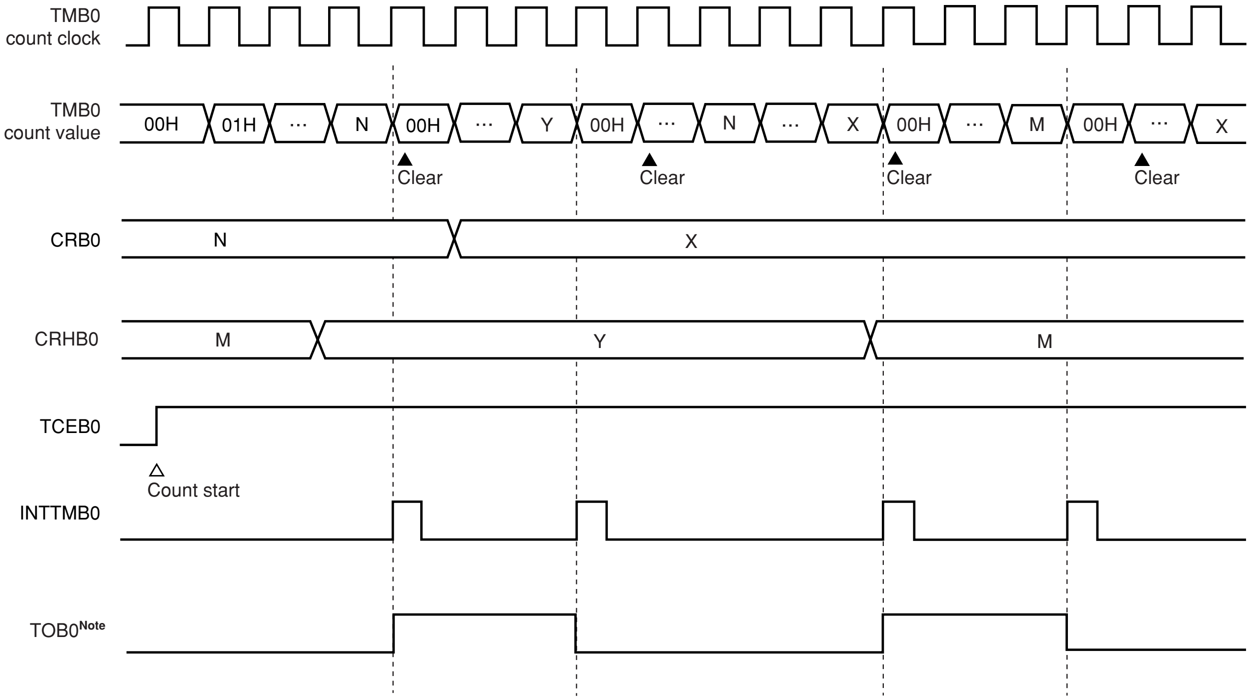
★ **Figure 7-22. PWM Output Mode Timing (Basic Operation)**



Note The initial value of TOB0 is low level when output is enabled (TOEB0 = 1).

Remark N, M = 00H to FFH

★ **Figure 7-23. PWM Output Mode Timing (When CRB0 and CRHB0 Are Overwritten)**



Note The initial value of TOB0 is low level when output is enabled (TOEB0 = 1).

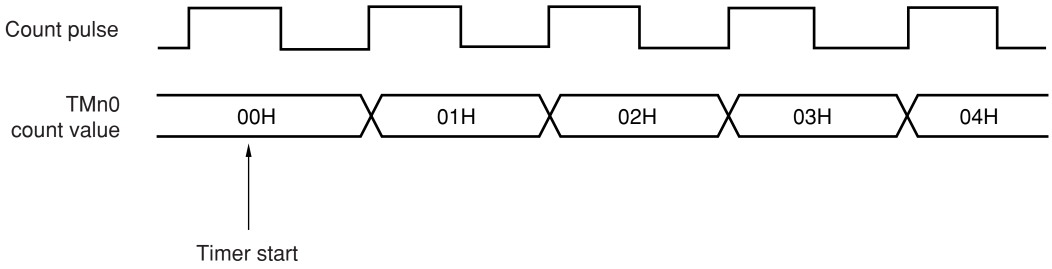
Remark N, M, X, Y = 00H to FFH

7.5 8-Bit Timer A0, B0 Cautions

(1) Error on starting timer

An error of up to 1 clock is included in the time between the timer being started and a match signal being generated. This is because 8-bit timer counter n0 (TMn0) is started asynchronously to the count pulse.

Figure 7-24. Start Timing of 8-Bit Timer Counter



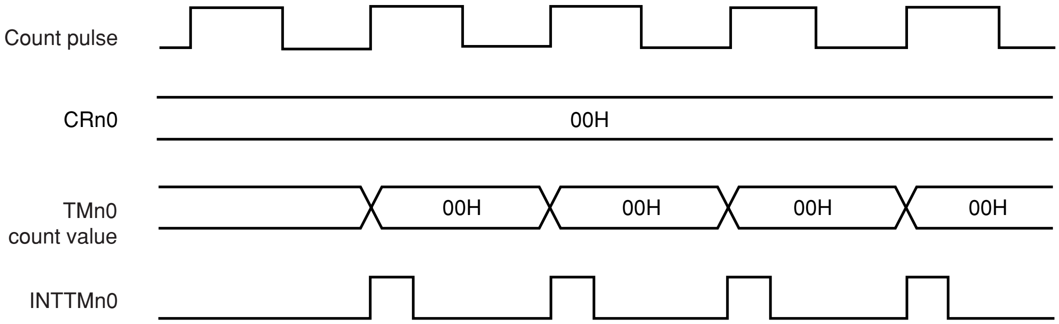
Remark n = A, B

(2) Setting of 8-bit compare register n0

8-bit compare register n0 (CRn0) can be set to 00H.

Therefore, one pulse can be counted.

Figure 7-25. Timing of 1-Pulse Count Operation (8-Bit Resolution)

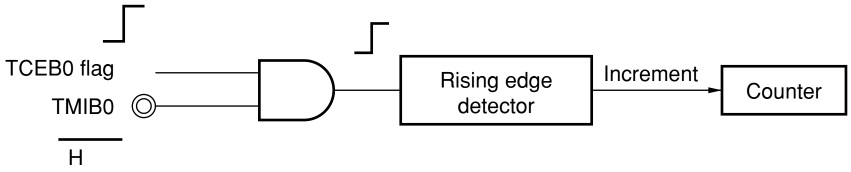


Remark n = A, B

(3) Count value if timer is started when TMIB0 pin is high

When an external clock input from the TMIB0 pin is selected as the count clock, the count value starts from 01H if the timer is enabled (TCEB0 = 0 → 1) while the TMIB0 pin is high. This is because the signal input from the TMIB0 pin is internally ANDed with the TCEB0 signal, and therefore the rising edge is input to the timer immediately after TCEB0 has been set and the counter is incremented. Either use the timer recognizing that the count is the count value + 1, or start the timer while the TMIB0 pin is low.

★



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

8.1 Outline of 8-Bit Timer/Event Counters 50 and 51

8-bit timer/event counters 50 and 51 can be used as an interval timer, an external event counter, to output square wave output with any selected frequency, and PWM output.

8.2 Functions of 8-Bit Timer/Event Counters 50 and 51

8-bit timer/event counters 50 and 51 have the following functions.

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

(1) Interval timer
These counters generate interrupt requests at the preset time interval.

(2) External event counter
These counters can measure the number of pulses of an externally input signal.

(3) Square wave output
These counters can output a square wave with any selected frequency.

(4) PWM output
These counters can output PWM.

Figures 8-1 and 8-2 show the 8-bit timer/event counter block diagrams.

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

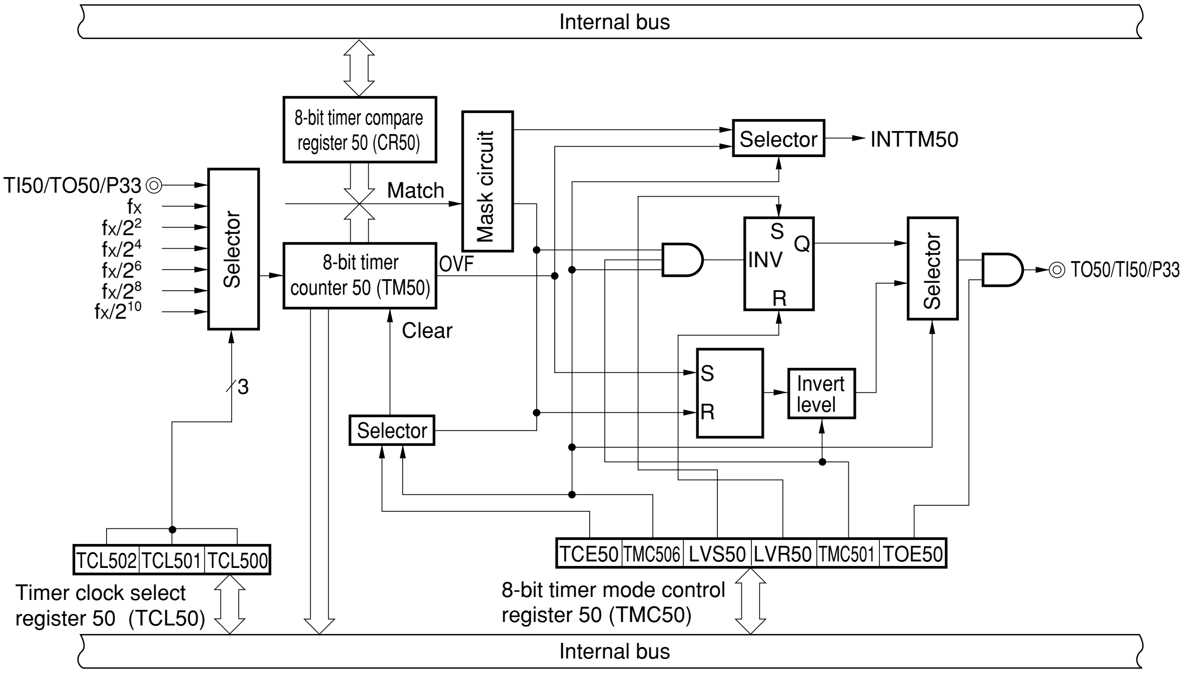
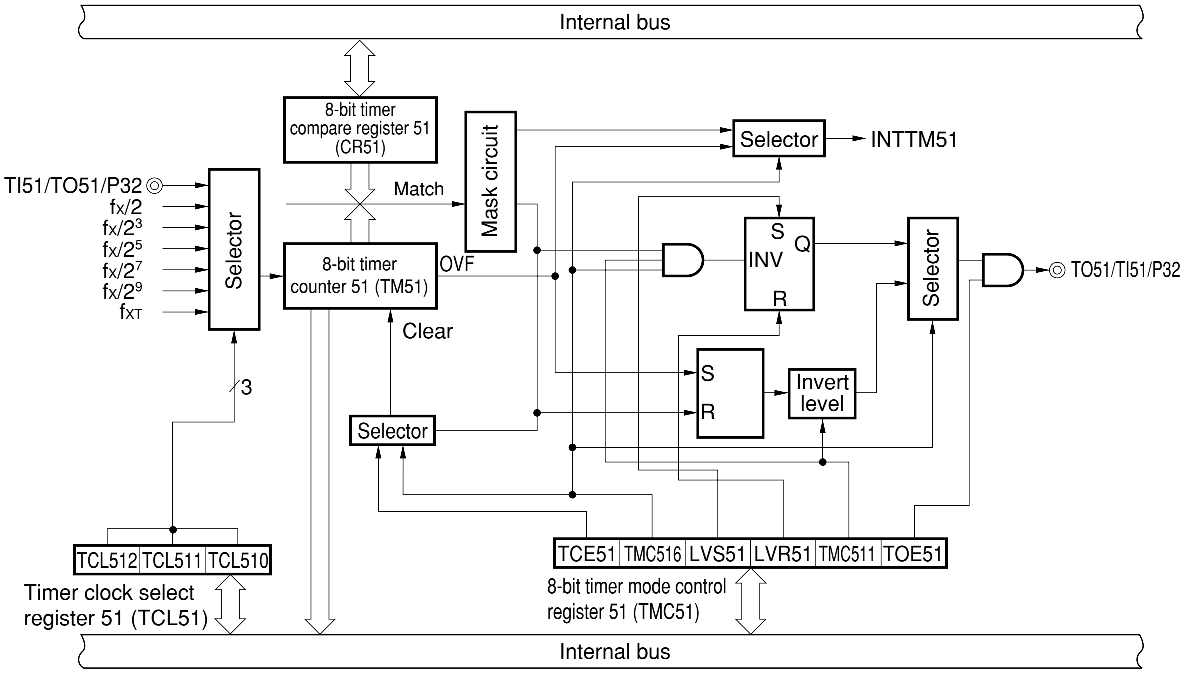


Figure 8-2. Block Diagram of 8-Bit Timer/Event Counter 51



8.3 Configuration of 8-Bit Timer/Event Counters 50 and 51

8-bit timer/event counters 50 and 51 consist of the following hardware.

Table 8-1. Configuration of 8-Bit Timer/Event Counters 50 and 51

Item	Configuration
Timer register	8-bit timer counter 5n (TM5n)
Register	8-bit timer compare register 5n (CR5n)
Timer output	2 (TO5n)
Control registers	Timer clock select register 5n (TCL5n) 8-bit timer mode control register 5n (TMC5n) Port mode register 3 (PM3)**Note**

Note See **Figure 4-9 Block Diagram of P32 to P34**.

Remark n = 0, 1

(1) 8-bit timer counter 5n (TM5n: n = 0, 1)

TM5n is an 8-bit read-only register which counts the count pulses.

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

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

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

<2> When TCE5n is cleared

<3> When TM5n and CR5n match in clear & start mode if this mode was entered upon match of the TM5n and CR5n values.

Remark n = 0, 1

★ **(2) 8-bit timer compare register 5n (CR5n: n = 0, 1)**

CR5n is a register that can be read/written by an 8-bit memory manipulation instruction.

The value set in CR5n is constantly compared with the 8-bit timer counter 5n (TM5n) count value, and an interrupt request (INTTM5n) is generated if they match, except in PWM mode.

In PWM mode, the TO5n pin becomes the active level due to TM5n overflow, and when the TM5n value and the CR5n value match, the TO5n pin becomes the inactive level.

It is possible to rewrite the value of CR5n within 00H to FFH during a count operation.

$\overline{\text{RESET}}$ input makes the value of this register undefined.

Remark n = 0, 1

8.4 Registers to Control 8-Bit Timer/Event Counters 50 and 51

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

- Timer clock select register 5n (TCL5n)
- 8-bit timer mode control register 5n (TMC5n)
- Port mode register 3 (PM3)

Remark n = 0, 1

(1) Timer clock select register 5n (TCL5n: n = 0, 1)

This register sets the count clock of 8-bit timer/event counter 5n and the valid edge of the TI5n input.
TCL5n is set by an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets the value of this register to 00H.

Figure 8-3. Format of Timer Clock Select Register 50 (TCL50)

Address: FF71H 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	$f_X/2^2$ (2.5 MHz)
1	0	0	$f_X/2^4$ (625 kHz)
1	0	1	$f_X/2^6$ (156 kHz)
1	1	0	$f_X/2^8$ (39.1 kHz)
1	1	1	$f_X/2^{10}$ (9.77 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: Main system clock oscillation frequency
2. Figures in parentheses are for operation with f_X = 10 MHz

Figure 8-4. Format of Timer Clock Select Register 51 (TCL51)

Address: FF74H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TCL51	0	0	0	0	0	TCL512	TCL511	TCL510

TCL512	TCL511	TCL510	Count clock selection
0	0	0	TI51 falling edge
0	0	1	TI51 rising edge
0	1	0	$f_X/2$ (5 MHz)
0	1	1	$f_X/2^3$ (1.25 MHz)
1	0	0	$f_X/2^5$ (313 kHz)
1	0	1	$f_X/2^7$ (78.1 kHz)
1	1	0	$f_X/2^9$ (19.5 kHz)
1	1	1	f_{XT} (32.768 kHz)

Cautions 1. When rewriting TCL51 to other data, stop the timer operation beforehand.

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

Remarks 1. f_X: Main system clock oscillation frequency

2. Figures in parentheses are for operation with f_X = 10 MHz, f_{XT} = 32.768 kHz

(2) 8-bit timer mode control register 5n (TMC5n: n = 0, 1)

TMC5n is a register which sets the following.

<1> 8-bit timer counter 5n (TM5n) count operation control
<2> 8-bit timer counter 5n (TM5n) operation 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

TMC5n is set by a 1-bit or 8-bit memory manipulation instruction.

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

Figure 8-5 shows the TMC5n format.

Figure 8-5. Format of 8-Bit Timer Mode Control Register 5n (TMC5n)

Address: FF70H (TMC50) FF73H (TMC51) After reset: 00H R/W

Symbol	$\boxed{7}$	6	5	4	$\boxed{3}$	$\boxed{2}$	1	$\boxed{0}$
TMC5n	TCE5n	TMC5n6	0	0	LVS5n	LVR5n	TMC5n1	TOE5n

TCE5n	TM5n count operation control
0	After clearing to 0, count operation disabled (prescaler disabled)
1	Count operation start

TMC5n6	TM5n operation mode selection
0	Clear and start mode by matching between TM5n and CR5n
1	PWM (free-running) mode

LVS5n	LVR5n	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

TMC5n1	In other modes (TMC5n6 = 0)	In PWM mode (TMC5n6 = 1)
	Timer F/F control	Active level selection
0	Inversion operation disabled	Active high
1	Inversion operation enabled	Active low

TOE5n	Timer output control
0	Output disabled (port mode)
1	Output enabled

Remarks 1. In PWM mode, PWM output will be inactive because TCE5n = 0.

2. If LVS5n and LVR5n are read after data is set, 0 is read.

3. n = 0, 1

(3) Port mode register 3 (PM3)

This register sets port 3 input/output in 1-bit units.

When using the P33/TO50/TI50 and P32/TO51/TI51 pins for timer output, set PM33, PM32 and the output latches of P33 and P32 to 0.

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

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

Figure 8-6. Format of Port Mode Register 3 (PM3)

Address: FF23H After reset: FFH R/W

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

PM3n	P3n pin I/O mode selection (n = 0 to 5)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

8.5 Operation of 8-Bit Timer/Event Counters 50 and 51

8.5.1 Interval timer operation

The 8-bit timer/event counters operate as interval timers which generate interrupt requests repeatedly at intervals of the count value preset to 8-bit timer compare register 5n (CR5n).

When the count values of 8-bit timer counter 5n (TM5n) match the values set to CR5n, counting continues with the TM5n values cleared to 0 and the interrupt request signals (INTTM5n) are generated.

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

See **8.6 8-Bit Timer/Event Counter 50 and 51 Cautions (2)** about the operation when the compare register value is changed during timer count operation.

[Setting]

<1> Set the registers.
- TCL5n: Select count clock.
- CR5n: Compare value
- ★ TMC5n: Count operation stops, clear and start mode by match of TM5n and CR5n.
  (TMC5n = 0000×××0B × = don't care)

<2> After TCE5n = 1 is set, count operation starts.

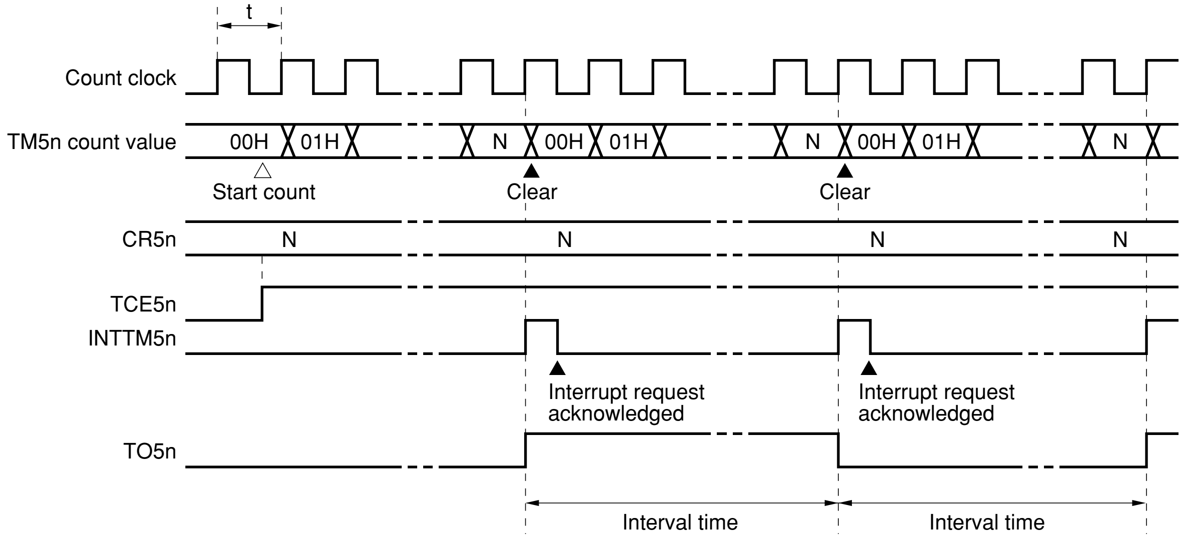
<3> If the values of TM5n and CR5n match, INTTM5n is generated (TM5n is cleared to 00H).

<4> INTTM5n generates repeatedly at the same interval. Set TCE5n to 0 to stop count operation.

Remark n = 0, 1

Figure 8-7. Interval Timer Operation Timings (1/3)

★ **(a) Basic operation**

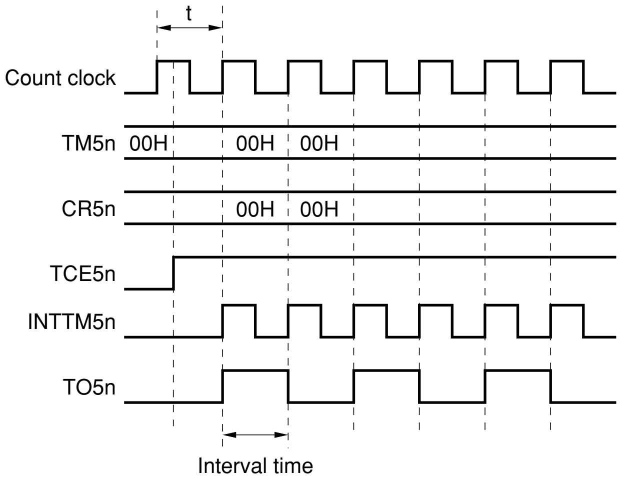


Remarks 1. Interval time = (N + 1) × t
N = 00H to FFH

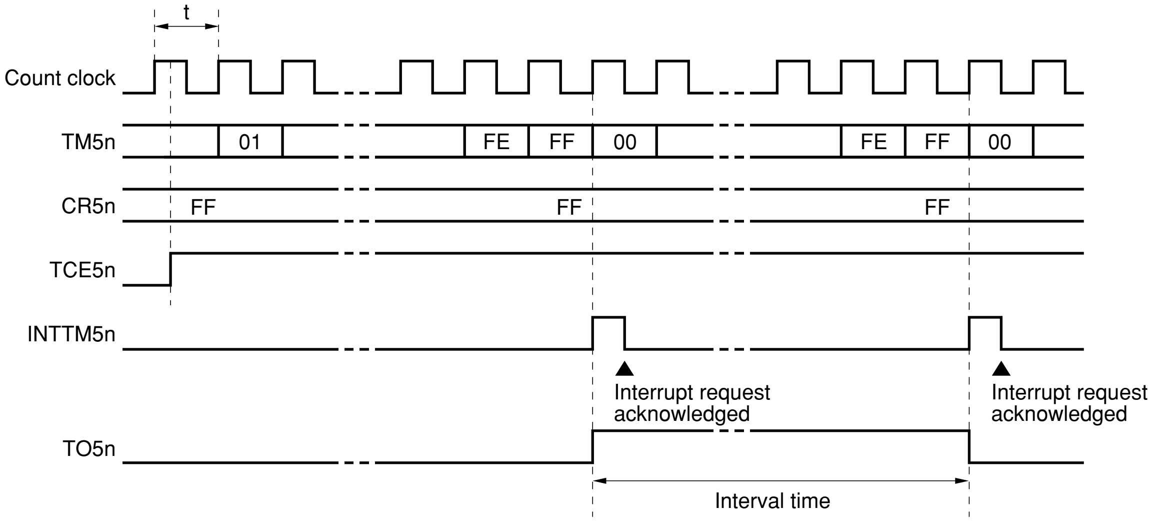
2. n = 0, 1

Figure 8-7. Interval Timer Operation Timings (2/3)

(b) When CR5n = 00H



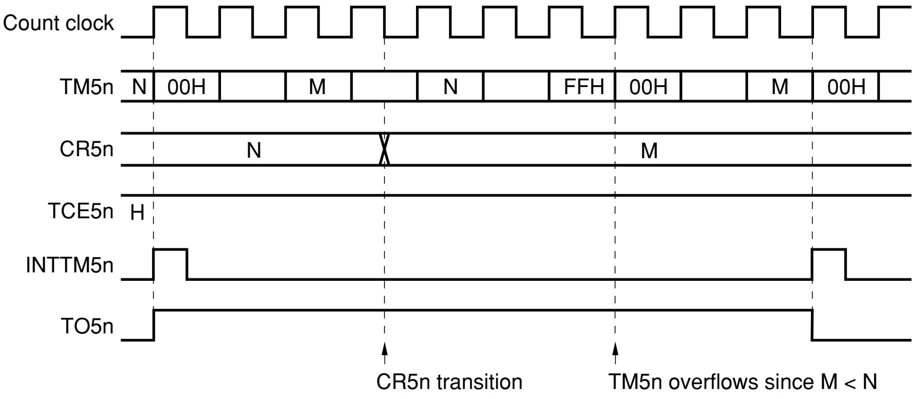
(c) When CR5n = FFH



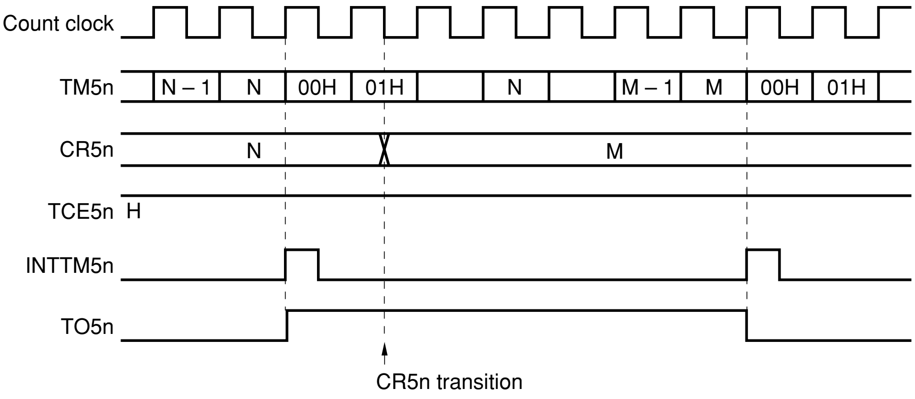
Remark n = 0, 1

Figure 8-7. Interval Timer Operation Timings (3/3)

(d) Operated by CR5n transition (M < N)



(e) Operated by CR5n transition (M > N)



Remark n = 0, 1

8.5.2 External event counter operation

The external event counter counts the number of external clock pulses to be input to TI5n by 8-bit timer counter 5n (TM5n).

TM5n is incremented each time the valid edge specified by timer clock select register 5n (TCL5n) is input. Either the rising or falling edge can be selected.

When the TM5n count value matches the value of 8-bit timer compare register 5n (CR5n), TM5n is cleared to 0 and the interrupt request signal (INTTM5n) is generated.

Subsequently, whenever the TM5n count value matches the value of CR5n, INTTM5n is generated.

★ **[Setting]**

<1> Set each register.

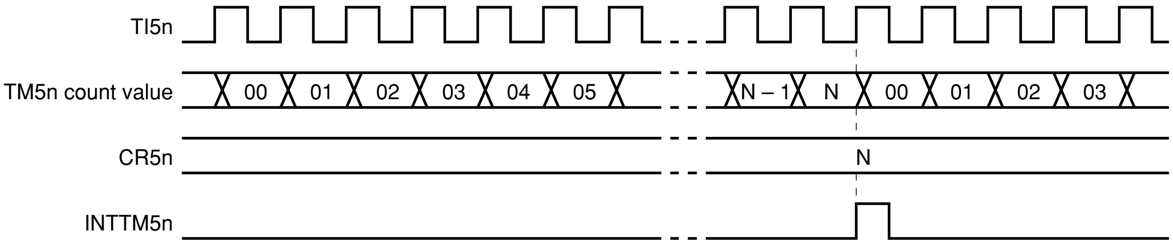
- TCL5n: Select TI5n input edge.
  TI5n falling edge → TCL5n = 00H
  TI5n rising edge → TCL5n = 01H
- CR5n: Compare value
- TMC5n: Stop the count operation, select the mode in which clear & start occurs on match of TM5n and CR5n, disable the timer F/F inversion operation, disable timer output.
  (TMC5n = 0000××00B (× = Don't care))

<2> When TCE5n = 1 is set, the number of pulses input from TI5n is counted.

<3> When the values of TM5n and CR5n match, INTTM5n is generated (TM5n is cleared to 00H).

<4> After these settings, INTTM5n is generated each time the values of TM5n and CR5n match.

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



★ **Remark** N = 00H to FFH
n = 0, 1

8.5.3 Square-wave output operation

A square wave with any selected frequency is output at intervals determined according to the value preset to 8-bit timer compare register 5n (CR5n).

The TO5n pin output status is reversed at intervals determined according to the count value preset to CR5n by setting bit 0 (TOE5n) of 8-bit timer mode control register 5n (TMC5n) to 1. This enables a square wave with any selected frequency to be output (duty = 50%).

★ **[Setting]**

<1> Set each register.

- Set port latches (P32, P33)Note and port mode registers (PM32, PM33)Note to 0.
- TCL5n: Select count clock
- CR5n: Compare value
- TMC5n: Counter operation stops, clear and start mode by match of TM5n and CR5n

LVS5n	LVR5n	Timer Output F/F Status Setting
1	0	High-level output
0	1	Low-level output

Timer output F/F reverse enable
Timer output enable → TOE5n = 1
(TMC5n = 00001011B or 00000111B)

<2> After TCE5n = 1 is set, count operation starts.

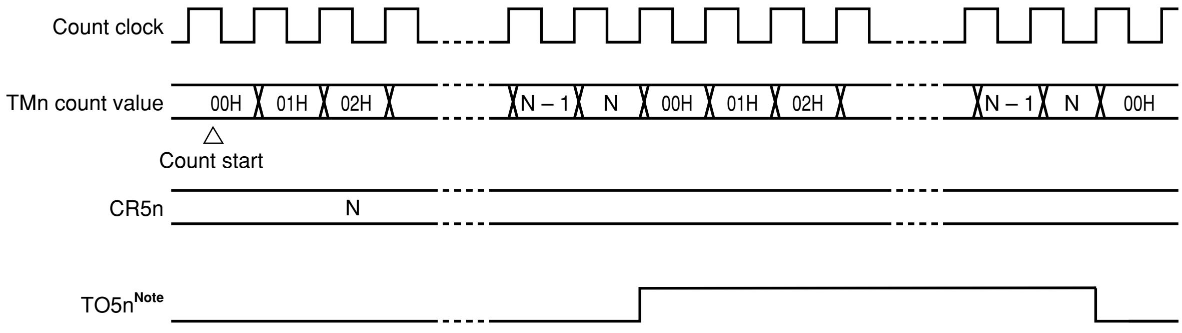
<3> Timer output F/F is reversed by match of TM5n and CR5n. After INTTM5n is generated, TM5n is cleared to 00H.

<4> Timer output F/F is reversed at the same interval and square wave is output from TO5n.
The frequency is as follows.

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

Note 8-bit timer/event counter 50: P33, PM33
8-bit timer/event counter 51: P32, PM32

Figure 8-9. Square-Wave Output Operation Timing



Note The TO5n output initial value can be set by bits 2 and 3 (LVR5n, LVS5n) of 8-bit timer mode control register 5n (TMC5n).

Remark n = 0, 1

8.5.4 PWM output operation

The 8-bit timer/event counter operates as PWM output when bit 6 (TMC5n6) of 8-bit timer mode control register 5n (TMC5n) is set to 1.

The duty rate pulse determined by the value set to 8-bit timer compare register 5n (CR5n) is output from TO5n.

Set the active level width of the PWM pulse to CR5n; the active level can be selected with bit 1 (TMC5n1) of TMC5n.

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

Enable/disable for PWM output can be selected with bit 0 (TOE5n) of TMC5n.

Caution CR5n can be rewritten in PWM mode only once in a cycle.

Remark n = 0, 1

(1) PWM output basic operation

★ **[Setting]**

<1> Set each register.

- Set the port latches (P32, P33)[Note] and port mode registers (PM32, PM33)[Note] to 0.
- TCL5n: Select the count clock.
- CR5n: Compare value
- TMC5n: Stop the count operation, select PWM mode.
  The timer output F/F is not changed.

TMC5n1	Active Level Selection
0	Active-high
1	Active-low

Timer output enabled
(TMC5n = 01000001B or 01000011B)

<2> The count operation starts when TCE5n = 1.
Set TCE5n to 0 to stop the count operation.

Note 8-bit timer/event counter 50: P33, PM33
8-bit timer/event counter 51: P32, PM32

[PWM output operation]

<1> PWM output (output from TO5n) outputs an inactive level after the count operation starts until an overflow is generated.

<2> When an overflow is generated, the active level set in <1> of setting is output.
The active level is output until CR5n matches the count value of 8-bit timer counter 5n (TM5n).

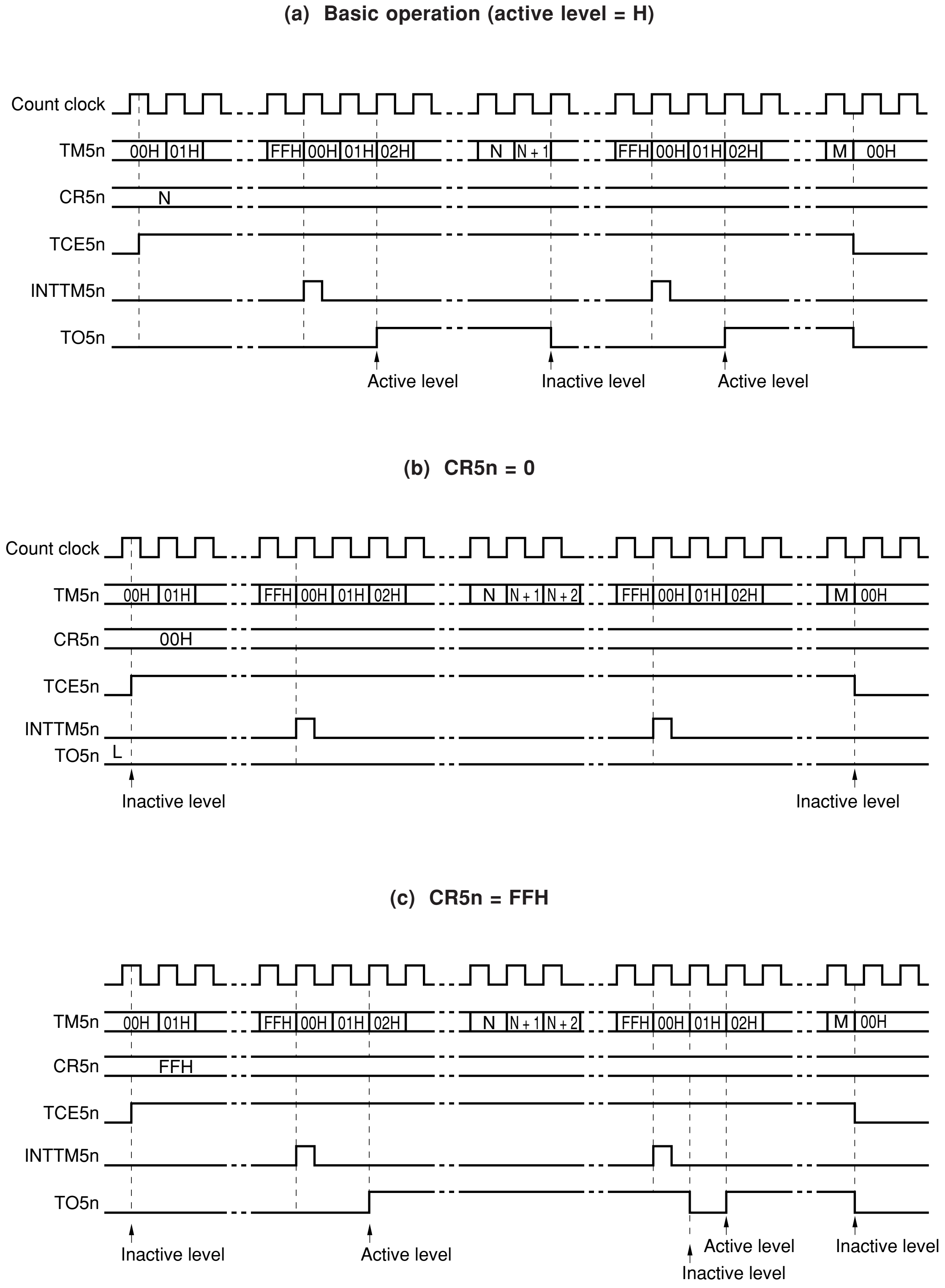
<3> After CR5n matches the count value, PWM output outputs the inactive level again until an overflow is generated.

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

<5> When the count operation is stopped with TCE5n = 0, PWM output becomes the inactive level.

Remark n = 0, 1

Figure 8-10. PWM Output Operation Timing



(a) Basic operation (active level = H)

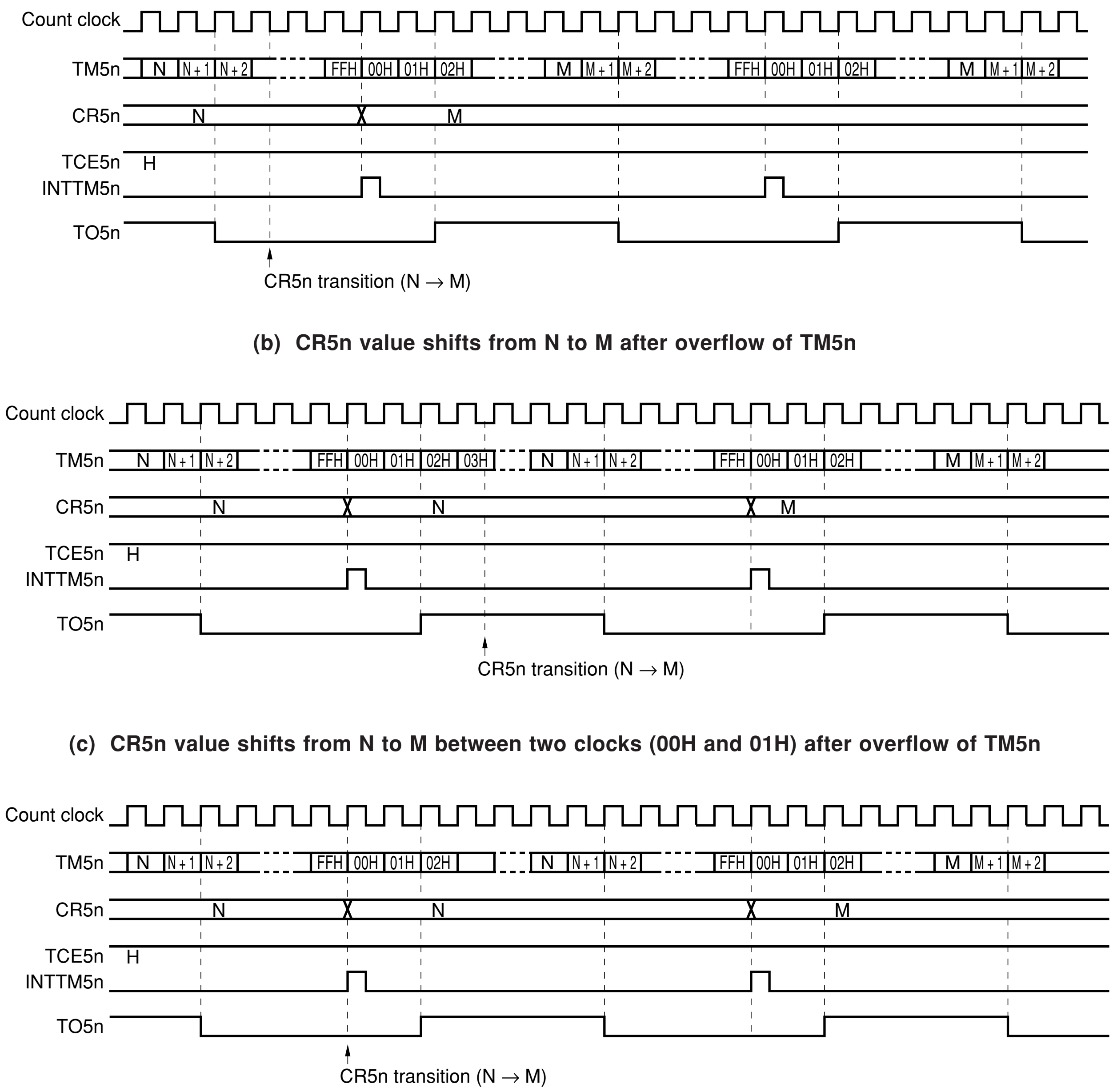
(b) CR5n = 0

(c) CR5n = FFH

Remark n = 0, 1

(2) Operated by CR5n transition

Figure 8-11. Timing of Operation by CR5n Transition



(a) CR5n value shifts from N to M before overflow of TM5n

(b) CR5n value shifts from N to M after overflow of TM5n

(c) CR5n value shifts from N to M between two clocks (00H and 01H) after overflow of TM5n

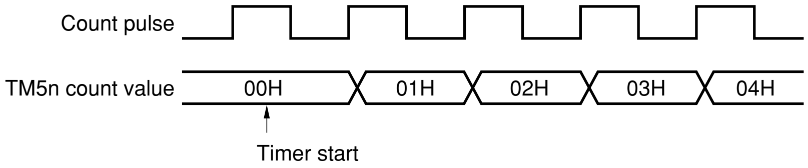
Remark n = 0, 1

8.6 8-Bit Timer/Event Counter 50 and 51 Cautions

(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 5n (TM5n) is started asynchronously to the count pulse.

Figure 8-12. 8-Bit Timer Counter Start Timing

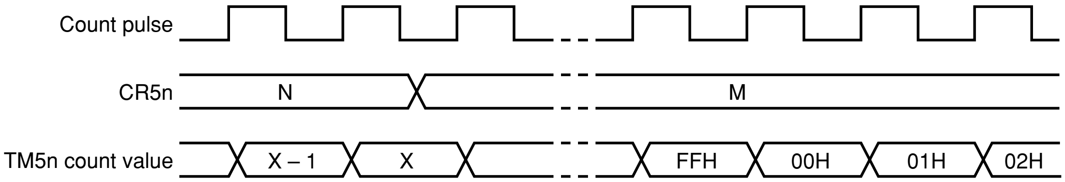


Remark n = 0, 1

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

If the value after 8-bit timer compare register 5n (CR5n) is changed is smaller than the value of 8-bit timer counter 5n (TM5n), TM5n continues counting, overflows and then restarts counting from 0. Thus, if the value (M) after CR5n change is smaller than value (N) before the change, it is necessary to restart the timer after changing CR5n.

Figure 8-13. Timing After Compare Register Change During Timer Count Operation



Caution Except when the TI5n input is selected, always set TCE5n = 0 before setting the stop state.

Remarks 1. N > X > M

2. n = 0, 1

(3) TM5n (n = 0, 1) reading during timer operation

When reading TM5n during operation, select a count clock with a high/low level waveform longer than two cycles of the CPU clock because the count clock stops temporarily. For example, in the case where the CPU clock (f_{CPU}) is f_X, when the selected count clock is $f_X/4$ or below, TM5n can be read.

Remark n = 0, 1

CHAPTER 9 WATCH TIMER

9.1 Outline of Watch Timer

The watch timer generates interrupt requests (INTWTN0 and INTWTNI0) at the preset time interval.

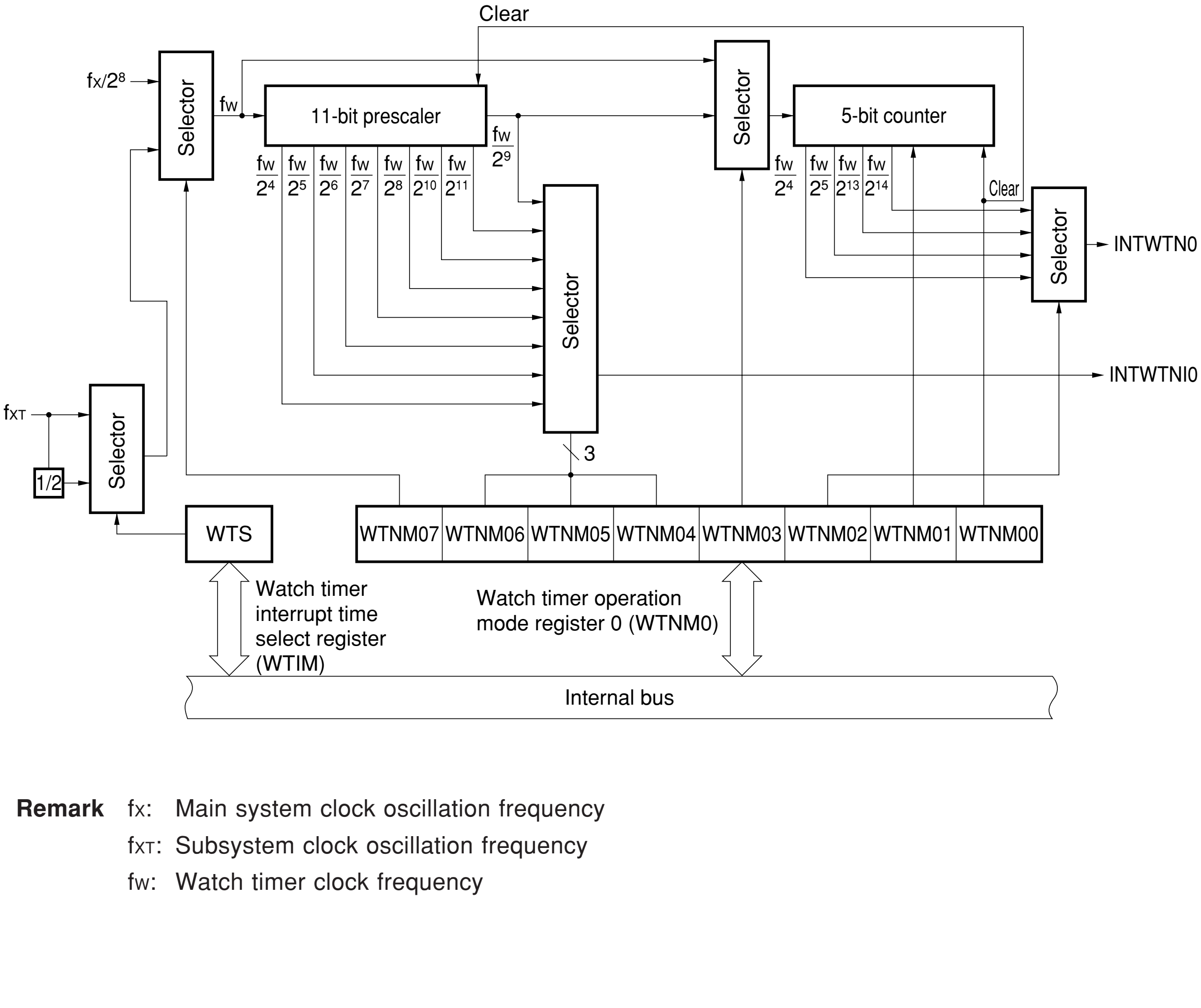
9.2 Watch Timer Functions

The watch timer has the following functions.

- Watch timer
- Interval timer

The watch timer and the interval timer can be used simultaneously.
Figure 9-1 shows the watch timer block diagram.

Figure 9-1. Watch Timer Block Diagram



Remark fX: Main system clock oscillation frequency
fXT: Subsystem clock oscillation frequency
fW: Watch timer clock frequency

(1) Watch timer

By using the main system clock or subsystem clock, interrupt requests (INTWTN0) are generated at preset intervals.

An interrupt request (INTWTN0) occurs at an interval of 0.5 second when using the 32.768 kHz subsystem clock. Also, an interrupt request (INTWTN0) can be generated at an interval of 1.0 seconds when using the 32.768 kHz subsystem clock via a setting in the watch timer interrupt time select register (WTIM).

Table 9-1. Watch Timer Interrupt Request Time

Interrupt Request Time	When Operated at f_X = 10 MHz	When Operated at f_{XT} = 32.768 kHz	When Operated at $f_{XT}/2$ = 16.384 kHz
$2^4/f_W$	409.6 μs	488 μs	977 μs
$2^5/f_W$	819.2 μs	977 μs	1.95 ms
$2^{13}/f_W$	0.2 s	0.25 s	0.5 s
$2^{14}/f_W$	0.41 s	0.5 s	1.0 s

Remark f_W: Watch timer clock frequency ($f_X/2^8$, f_{XT}, or $f_{XT}/2$)
f_X: Main system clock oscillation frequency
f_{XT}: Subsystem clock oscillation frequency

(2) Interval timer

By using the main system clock or subsystem clock, interrupt requests (INTWTNI0) are generated at preset intervals.

Table 9-2. Interval Timer Interval Time

Interval Time	When Operated at f_X = 10 MHz	When Operated at f_{XT} = 32.768 kHz	When Operated at $f_{XT}/2$ = 16.384 kHz
$2^4/f_W$	409.6 μs	488 μs	977 μs
$2^5/f_W$	819.2 μs	977 μs	1.95 ms
$2^6/f_W$	1.64 ms	1.95 ms	3.91 ms
$2^7/f_W$	3.28 ms	3.91 ms	7.81 ms
$2^8/f_W$	6.56 ms	7.81 ms	15.6 ms
$2^9/f_W$	13.1 ms	15.6 ms	31.2 ms
$2^{10}/f_W$	26.2 ms	31.2 ms	62.4 ms
$2^{11}/f_W$	52.4 ms	62.4 ms	125 ms

Remark f_W: Watch timer clock frequency ($f_X/2^8$, f_{XT}, or $f_{XT}/2$)
f_X: Main system clock oscillation frequency
f_{XT}: Subsystem clock oscillation frequency

9.3 Watch Timer Configuration

The watch timer consists of the following hardware.

Table 9-3. Watch Timer Configuration

Item	Configuration
Prescaler	11 bits × 1, 5 bits × 1
Control register	Watch timer operation mode register 0 (WTNM0) Watch timer interrupt time select register (WTIM)

9.4 Registers to Control Watch Timer

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

- Watch timer operation mode register 0 (WTNM0)
- Watch timer interrupt time select register (WTIM)

(1) Watch timer operation mode register 0 (WTNM0)

This register enables/disables the watch timer operation, sets the 11-bit prescaler interval time, and controls the 5-bit counter operation.

WTNM0 is set by a 1-bit or 8-bit memory manipulation instruction.

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

Figure 9-2. Format of Watch Timer Operation Mode Register 0 (WTNM0)

Address: FF41H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
WTNM0	WTNM07	WTNM06	WTNM05	WTNM04	WTNM03	WTNM02	WTNM01	WTNM00

WTNM07	Watch timer count clock selection
0	$f_X/2^8$ (39.1 kHz)
1	f_{XT} (32.768 kHz) or $f_{XT}/2$ (16.384 kHz)Note

WTNM06	WTNM05	WTNM04	11-bit prescaler interval time selection			
				WTNM07 = 0	WTNM07 = 1	
					WTS = 0	WTS = 1
0	0	0	$2^4/f_W$	$2^{12}/f_X$ (409.6 μs)	$2^4/f_{XT}$ (488 μs)	$2^5/f_{XT}$ (977 μs)
0	0	1	$2^5/f_W$	$2^{13}/f_X$ (819.2 μs)	$2^5/f_{XT}$ (977 μs)	$2^6/f_{XT}$ (1.95 ms)
0	1	0	$2^6/f_W$	$2^{14}/f_X$ (1.64 ms)	$2^6/f_{XT}$ (1.95 ms)	$2^7/f_{XT}$ (3.91 ms)
0	1	1	$2^7/f_W$	$2^{15}/f_X$ (3.28 ms)	$2^7/f_{XT}$ (3.91 ms)	$2^8/f_{XT}$ (7.81 ms)
1	0	0	$2^8/f_W$	$2^{16}/f_X$ (6.56 ms)	$2^8/f_{XT}$ (7.81 ms)	$2^9/f_{XT}$ (15.6 ms)
1	0	1	$2^9/f_W$	$2^{17}/f_X$ (13.1 ms)	$2^9/f_{XT}$ (15.6 ms)	$2^{10}/f_{XT}$ (31.2 ms)
1	1	0	$2^{10}/f_W$	$2^{18}/f_X$ (26.2 ms)	$2^{10}/f_{XT}$ (31.2 ms)	$2^{11}/f_{XT}$ (62.4 ms)
1	1	1	$2^{11}/f_W$	$2^{19}/f_X$ (52.4 ms)	$2^{11}/f_{XT}$ (62.4 ms)	$2^{12}/f_{XT}$ (125 ms)

WTNM03	WTNM02	Selection of interrupt request time of the watch timer			
			WTNM07 = 0	WTNM07 = 1	
				WTS = 0	WTS = 1
0	0	$2^{14}/f_W$	$2^{22}/f_X$ (0.41 s)	$2^{14}/f_{XT}$ (0.5 s)	$2^{15}/f_{XT}$ (1.0 s)
0	1	$2^{13}/f_W$	$2^{21}/f_X$ (0.2 s)	$2^{13}/f_{XT}$ (0.25 s)	$2^{14}/f_{XT}$ (0.5 s)
1	0	$2^5/f_W$	$2^{13}/f_X$ (819.2 μs)	$2^5/f_{XT}$ (977 μs)	$2^6/f_{XT}$ (1.95 ms)
★ 1	1	$2^4/f_W$	$2^{12}/f_X$ (409.6 μs)	$2^4/f_{XT}$ (488 μs)	$2^5/f_{XT}$ (977 μs)

WTNM01	5-bit counter operation control
0	Clear after operation stop
1	Start

WTNM00	Watch timer operation enable
0	Operation stop (clear both 11-bit prescaler and 5-bit counter)
1	Operation enable

Note This is the frequency (f_{XT} or $f_{XT}/2$) set with bit 0 (WTS) of the watch timer interrupt time select register (WTIM).

Caution Do not change the count clock, interval time, and interrupt request time (by using bits 2 to 7 (WTNM02 to WTNM07) of WTNM0) while the watch timer is operating.

Remarks 1. f_W: Watch timer clock frequency ($f_X/2^8$, f_{XT}, or $f_{XT}/2$)

f_X: Main system clock oscillation frequency

f_{XT}: Subsystem clock oscillation frequency

2. Figures in parentheses apply to operation with f_X = 10 MHz, f_{XT} = 32.768 kHz.

(2) Watch timer interrupt time select register (WTIM)

This register is used to set the interrupt time by selecting either the source clock or the clock divided by 2 for the subsystem clock to be input to the watch timer.

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

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

Figure 9-3. Format of Watch Timer Interrupt Time Select Register (WTIM)

Symbol	7	6	5	4	3	2	1	$\boxed{0}$	Address	After reset	R/W
WTIM	0	0	0	0	0	0	0	WTS	FF79H	00H	R/W

WTS	Selection of watch timer interrupt time[Note]
0	0.5 s (f_{XT})
1	1.0 s ($f_{XT}/2$)

Note The selection is only available when bit 7 (WTNM07) of watch timer operation mode register 0 (WTNM0) is 1.

Remark f_{XT}: Subsystem clock oscillation frequency

9.5 Watch Timer Operations

9.5.1 Watch timer operation

By using the main system clock or subsystem clock, the watch timer operates as a watch timer with preset timing intervals.

Bits 2, 3, and 7 (WTNM02, WTNM03, and WTNM07) of watch timer operation mode register 0 (WTNM0) enable the selection of the timing for the watch timer.

The watch timer generates an interrupt request (INTWTN0) at a fixed time interval.

An interrupt request (INTWTN0) occurs at an interval of 0.5 second when using the 32.768 kHz subsystem clock.

Also, an interrupt request (INTWTN0) can be generated at an interval of 1.0 seconds when using the 32.768 kHz subsystem clock via a setting in the watch timer interrupt time select register (WTIM).

If bit 0 (WTNM00) and bit 1 (WTNM01) of watch timer operation mode register 0 (WTNM0) are set to 1, the count operation starts. If set to 0, the 5-bit counter is cleared and the count operation stops.

For simultaneous operation of the interval timer, zero-second start can be achieved by setting WTNM01 to 0.

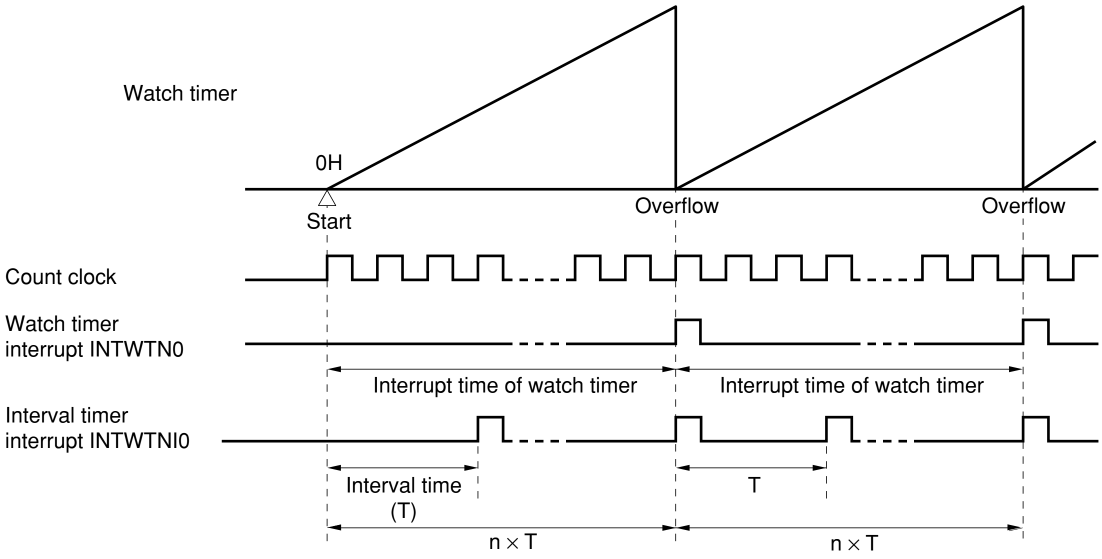
However, in this case, since the 11-bit prescaler is not cleared, at the first overflow (INTWTN0) after the watch timer's zero-second start, an error of up to $2^{11} \times 1/f_W$ seconds occurs.

9.5.2 Interval timer operation

The watch timer operates as an interval timer which generates interrupt requests (INTWTNI0) repeatedly at an interval of the preset count value.

The interval time can be selected with bits 4 to 6 and 7 (WTNM04 to WTNM06 and WTNM07) of watch timer operation mode register 0 (WTNM0).

Figure 9-4. Operation Timing of Watch Timer/Interval Timer


Watch timer
0H
Start
Overflow
Overflow
Count clock
Watch timer interrupt INTWTN0
Interrupt time of watch timer
Interrupt time of watch timer
Interval timer interrupt INTWTNI0
Interval time (T)
T
n × T
n × T

Caution If the watch timer and 5-bit counter are enabled by watch timer operation mode register 0 (WTNM0) (by setting bits 0 (WTNM00) and 1 (WTNM01) of WTNM0 to 1), the time from this setting to the occurrence of the first interrupt request (INTWTN0) is not exactly the value set by bits 2 and 3 (WTNM02 and WTNM03) of WTNM0. This is because the 5-bit counter is late by one output cycle of the 11-bit prescaler in starting to count. The second INTWTN0 signal and those that follow are generated exactly at the set time.

Remark n: The number of interval timer operations

10.1 Outline of Watchdog Timer

The watchdog timer can also be used to generate a non-maskable interrupt request, maskable interrupt request, or $\overline{\text{RESET}}$ signal at the preset time intervals.

10.2 Watchdog Timer Functions

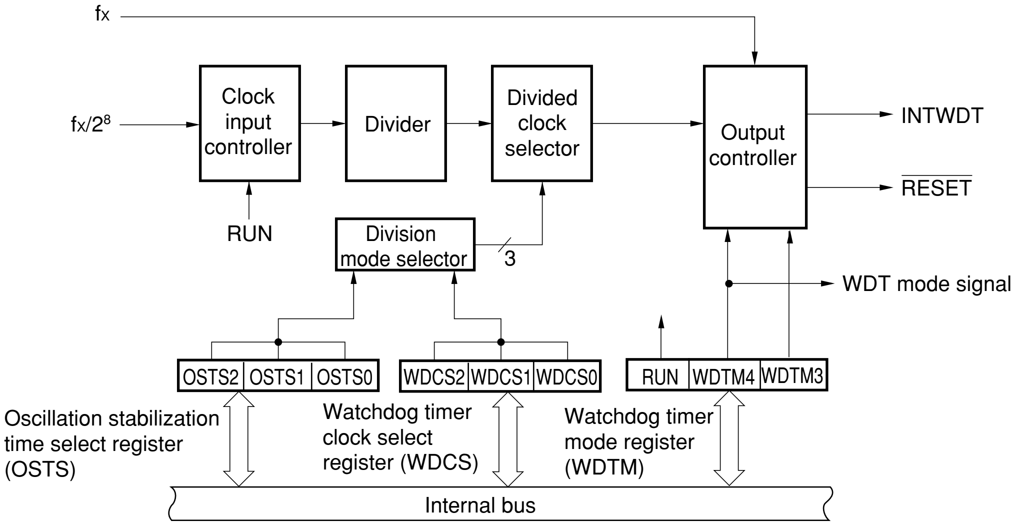
The watchdog timer has the following functions.

- Watchdog timer
- Interval timer
- Oscillation stabilization time selection

Caution Select the watchdog timer mode or the interval timer mode with the watchdog timer mode register (WDTM). (The watchdog timer and the interval timer cannot be used simultaneously.)

Figure 10-1 shows a block diagram of the watchdog timer.

Figure 10-1. Watchdog Timer Block Diagram



(1) Watchdog timer mode

A program loop is detected. Upon detection of a program loop, a non-maskable interrupt request or $\overline{\text{RESET}}$ can be generated.

Table 10-1. Watchdog Timer Program Loop Detection Time

Program Loop Detection Time
2^{12} × 1/fx (410 μs)
2^{13} × 1/fx (819 μs)
2^{14} × 1/fx (1.64 ms)
2^{15} × 1/fx (3.28 ms)
2^{16} × 1/fx (6.55 ms)
2^{17} × 1/fx (13.1 ms)
2^{18} × 1/fx (26.2 ms)
2^{20} × 1/fx (105 ms)

Remarks 1. fx: Main system clock oscillation frequency

2. Figures in parentheses are for operation with fx = 10 MHz

(2) Interval timer mode

Interrupt requests are generated at the preset time intervals.

Table 10-2. Interval Time

Interval Time
2^{12} × 1/fx (410 μs)
2^{13} × 1/fx (819 μs)
2^{14} × 1/fx (1.64 ms)
2^{15} × 1/fx (3.28 ms)
2^{16} × 1/fx (6.55 ms)
2^{17} × 1/fx (13.1 ms)
2^{18} × 1/fx (26.2 ms)
2^{20} × 1/fx (105 ms)

Remarks 1. fx: Main system clock oscillation frequency

2. Figures in parentheses are for operation with fx = 10 MHz

10.3 Watchdog Timer Configuration

The watchdog timer consists of the following hardware.

Table 10-3. Watchdog Timer Configuration

Item	Configuration
Control registers	Watchdog timer clock select register (WDCS) Watchdog timer mode register (WDTM) Oscillation stabilization time select register (OSTS)

10.4 Registers to Control Watchdog Timer

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

- Watchdog timer clock select register (WDCS)
- Watchdog timer mode register (WDTM)
- Oscillation stabilization time select register (OSTS)

(1) Watchdog timer clock select register (WDCS)

This register sets overflow time of the watchdog timer and the interval timer.

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

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

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

Address: FF42H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
WDCS	0	0	0	0	0	WDCS2	WDCS1	WDCS0

WDCS2	WDCS1	WDCS0	Overflow time of watchdog timer/interval timer
0	0	0	$2^{12}/f_X$ (410 μs)
0	0	1	$2^{13}/f_X$ (819 μs)
0	1	0	$2^{14}/f_X$ (1.64 ms)
0	1	1	$2^{15}/f_X$ (3.28 ms)
1	0	0	$2^{16}/f_X$ (6.55 ms)
1	0	1	$2^{17}/f_X$ (13.1 ms)
1	1	0	$2^{18}/f_X$ (26.2 ms)
1	1	1	$2^{20}/f_X$ (105 ms)

Remarks 1. f_X: Main system clock oscillation frequency

2. Figures in parentheses are for operation with f_X = 10 MHz

(2) Watchdog timer mode register (WDTM)

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

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

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

Figure 10-3. Format of Watchdog Timer Mode Register (WDTM)

Address: FFF9H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
WDTM	RUN	0	0	WDTM4	WDTM3	0	0	0

RUN	Watchdog timer operation mode selection[Note 1]
0	Count stop
1	Counter is cleared and counting starts

WDTM4	WDTM3	Watchdog timer operation mode selection[Note 2]
0	×	Interval timer mode[Note 3] (Maskable interrupt request occurs upon generation of an overflow)
1	0	Watchdog timer mode 1 (Non-maskable interrupt request occurs upon generation of an overflow)
1	1	Watchdog timer mode 2 (Reset operation is activated upon generation of an overflow)

Notes 1. Once set to 1, RUN cannot be cleared to 0 by software.
Thus, once counting starts, it can only be stopped by $\overline{\text{RESET}}$ input.

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

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

Caution When 1 is set to RUN so that the watchdog timer is cleared, the actual overflow time is up to 2^8/fx seconds shorter than the time set by watchdog timer clock select register (WDCS).

Remark ×: Don’t care

(3) Oscillation stabilization time select register (OSTS)

A register to select oscillation stabilization time from reset time or STOP mode released time to the time when oscillation is stabilized.

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

$\overline{\text{RESET}}$ input sets the value of this register to 04H. Thus, when releasing the STOP mode by $\overline{\text{RESET}}$ input, the time required to release is 2^{17}/fx.

Figure 10-4. Format of Oscillation Stabilization Time Select Register (OSTS)

Address: FFFAH After reset: 04H R/W

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

OSTS2	OSTS1	OSTS0	Selection of oscillation stabilization time
0	0	0	2^{12}/fx (410 μs)
0	0	1	2^{14}/fx (1.64 ms)
0	1	0	2^{15}/fx (3.28 ms)
0	1	1	2^{16}/fx (6.55 ms)
1	0	0	2^{17}/fx (13.1 ms)
Other than the above			Setting prohibited

Remarks 1. fx: Main system clock oscillation frequency

2. Figures in parentheses are for operation with fx = 10 MHz

10.5 Watchdog Timer Operations

10.5.1 Watchdog timer operation

When bit 4 (WDTM4) of the watchdog timer mode register (WDTM) is set to 1, the watchdog timer is operated to detect a program loop.

The program loop detection time interval is selected with bits 0 to 2 (WDCS0 to WDCS2) of the watchdog timer clock select register (WDCS).

Watchdog timer starts by setting bit 7 (RUN) of WDTM to 1. After the watchdog timer is started, set RUN to 1 within the set program loop time interval. The watchdog timer can be cleared and counting is started by setting RUN to 1. If RUN is not set to 1 and the program loop detection time is exceeded, system reset or a non-maskable interrupt request is generated according to WDTM bit 3 (WDTM3) value.

The watchdog timer continues operating in the HALT mode but it stops in the STOP mode. Thus, set RUN to 1 before the STOP mode is set, clear the watchdog timer and then execute the STOP instruction.

Cautions 1. The actual program loop detection time may be shorter than the set time by a maximum of 2^8/fx seconds.

2. When the subsystem clock is selected for CPU clock, watchdog timer count operation is stopped.

Table 10-4. Watchdog Timer Program Loop Detection Time

Program Loop Detection Time
2^{12} × 1/fx (410 μs)
2^{13} × 1/fx (819 μs)
2^{14} × 1/fx (1.64 ms)
2^{15} × 1/fx (3.28 ms)
2^{16} × 1/fx (6.55 ms)
2^{17} × 1/fx (13.1 ms)
2^{18} × 1/fx (26.2 ms)
2^{20} × 1/fx (105 ms)

Remarks 1. fx: Main system clock oscillation frequency

2. Figures in parentheses are for operation with fx = 10 MHz.

10.5.2 Interval timer operation

The watchdog timer operates as an interval timer which generates interrupt requests repeatedly at an interval of the preset count value when bit 4 (WDTM4) of the watchdog timer mode register (WDTM) is set to 0.

The interval time of interval timer is selected with bits 0 to 2 (WDCS0 to WDCS2) of the watchdog timer clock select register (WDCS). When 1 is set to bit 7 (RUN) of WDTM, the watchdog timer operates as an interval timer.

When the watchdog timer operated as the interval timer, the interrupt mask flag (WDTMK) and priority specify flag (WDTPR) are validated and the maskable interrupt request (INTWDT) can be generated. Among maskable interrupts, INTWDT has the highest priority at default.

The interval timer continues operating in the HALT mode but it stops in STOP mode. Thus, set RUN to 1 before the STOP mode is set, clear the interval timer and then execute the STOP instruction.

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

2. The interval time just after setting by WDTM may be shorter than the set time by a maximum of 2^8/fx seconds.

3. When the subsystem clock is selected for CPU clock, watchdog timer count operation is stopped.

Table 10-5. Interval Timer Interval Time

Interval Time
2^{12} × 1/fx (410 μs)
2^{13} × 1/fx (819 μs)
2^{14} × 1/fx (1.64 ms)
2^{15} × 1/fx (3.28 ms)
2^{16} × 1/fx (6.55 ms)
2^{17} × 1/fx (13.1 ms)
2^{18} × 1/fx (26.2 ms)
2^{20} × 1/fx (105 ms)

Remarks 1. fx: Main system clock oscillation frequency

2. Figures in parentheses are for operation with fx = 10 MHz.

CHAPTER 11 CLOCK OUTPUT CONTROLLER

11.1 Outline of Clock Output Controller

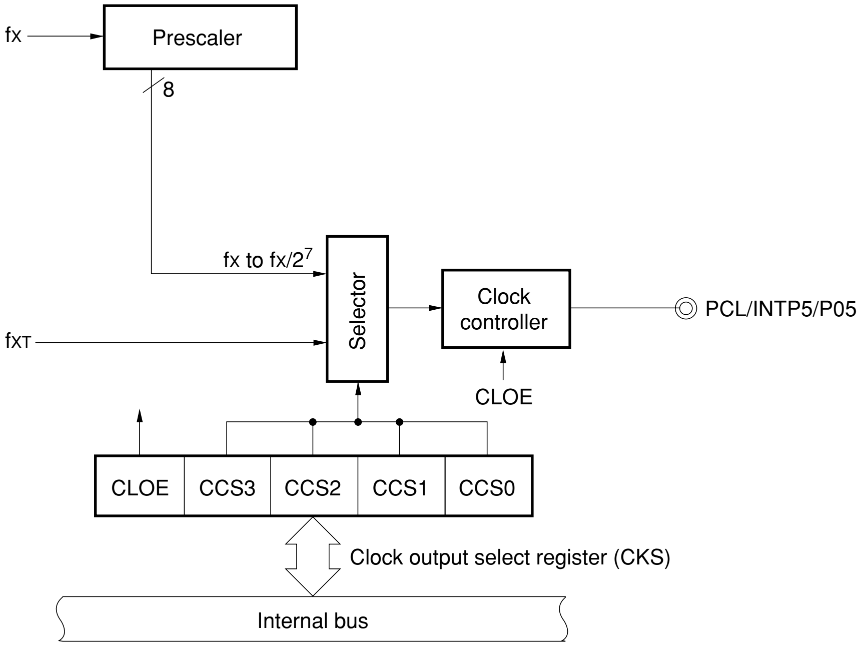
The clock output circuit supplies other devices with the divided main system clock and the subsystem clock.

11.2 Clock Output Controller Functions

The clock output controller is intended for carrier output during remote controlled transmission and clock output for supply to peripheral LSIs. The clock selected with the clock output select register (CKS) is output.

Figure 11-1 shows the block diagram of clock output controller.

Figure 11-1. Block Diagram of Clock Output Controller



11.3 Configuration of Clock Output Controller

The clock output controller consists of the following hardware.

Table 11-1. Configuration of Clock Output Controller

Item	Configuration
Control registers	Clock output select register (CKS) Port mode register 0 (PM0)[Note]

Note See **Figure 4-3 P05 to P07 Block Diagram**.

11.4 Registers to Control Clock Output Controller

The following two registers are used to control the clock output controller.

- Clock output select register (CKS)
- Port mode register 0 (PM0)

(1) Clock output select register (CKS)

This register enables/disables output of the clock output (PCL), and sets the output clock.

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

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

Figure 11-2. Format of Clock Output Select Register (CKS)

Address: FF40H After reset: 00H R/W

Symbol	7	6	5	$\boxed{4}$	3	2	1	0
CKS	0	0	0	CLOE	CCS3	CCS2	CCS1	CCS0

CLOE	PCL output enable/disable setting
0	Stop clock division circuit operation. PCL fixed to low level
1	Enable clock division circuit operation. PCL output enabled.

CCS3	CCS2	CCS1	CCS0	PCL output clock selection
0	0	0	0	f_X (10 MHz)
0	0	0	1	$f_X/2$ (5 MHz)
0	0	1	0	$f_X/2^2$ (2.5 MHz)
0	0	1	1	$f_X/2^3$ (1.25 MHz)
0	1	0	0	$f_X/2^4$ (625 kHz)
0	1	0	1	$f_X/2^5$ (312.5 kHz)
0	1	1	0	$f_X/2^6$ (156.3 kHz)
0	1	1	1	$f_X/2^7$ (78.1 kHz)
1	0	0	0	f_{XT} (32.768 kHz)
Other than above				Setting prohibited

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

3. Figures in parentheses are for operation with f_X = 10 MHz, f_{XT} = 32.768 kHz.

(2) Port mode register 0 (PM0)

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

When using the PCL/INTP5/P05 pin for clock output, set PM05 and the output latch of P05 to 0.

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

RESET input sets the value of this register to FFH.

Figure 11-3. Format of Port Mode Register 0 (PM0)

Address: FF20H After reset: FFH R/W

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

PM0n	P0n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

11.5 Clock Output Controller Operations

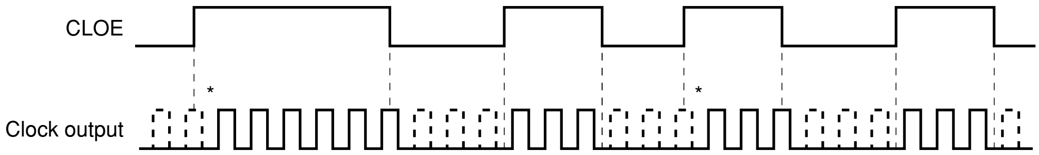
The clock pulse is output using the following procedure.

<1> Select the clock pulse output frequency with bits 0 to 3 (CCS0 to CCS3) of the clock output select register (CKS) (clock pulse output in disabled status).

<2> Set bit 4 (CLOE) of CKS to 1, and enable clock output.

Remark The clock output controller is designed not to output pulses with a small width during output enable/disable switching of the clock output. As shown in Figure 11-4, be sure to start output from the low period of the clock (marked with * in the figure). When stopping output, do so after securing the high level of the clock.

Figure 11-4. Remote Control Output Application Example



CHAPTER 12 8-BIT A/D CONVERTER (μPD780344, 780344Y SUBSERIES)

12.1 8-Bit A/D Converter Functions

The 8-bit A/D converter is an 8-bit resolution converter that converts analog inputs into digital values. It can control up to 8 analog input channels (ANI0 to ANI7).

(1) Hardware start

Conversion is started by trigger input (ADTRG: rising edge, falling edge, or both rising and falling edges can be specified).

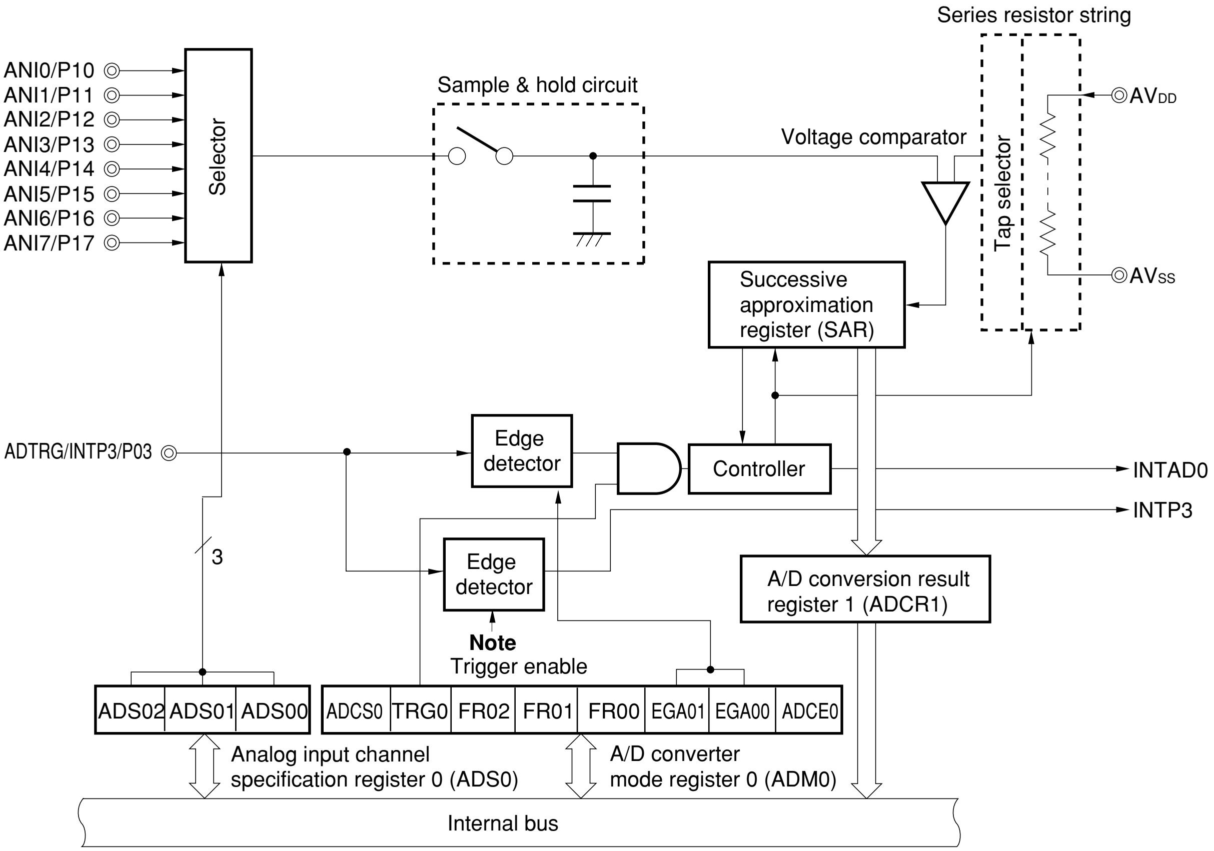
(2) Software start

Conversion is started by setting A/D converter mode register 0 (ADM0).

Select one channel for analog input from ANI0 to ANI7 to perform A/D conversion. In the case of hardware start, A/D conversion stops when an A/D conversion operation ends and an interrupt request (INTAD0) is generated. In the case of software start, A/D conversion is repeated. Each time an A/D conversion operation ends, an interrupt request (INTAD0) is generated.

Caution The μPD78F0354 and 78F0354Y have a 10-bit A/D converter, but this A/D converter can be used as an 8-bit A/D converter by using an object file assembled using a product in the μPD780344 and 780344Y Subseries.

★ **Figure 12-1. 8-Bit A/D Converter Block Diagram**



Note The valid edge of the external interrupt is specified by bit 3 of the EGP and EGN registers (see **Figure 19-5 Format of External Interrupt Rising Edge Enable Register (EGP), External Interrupt Falling Edge Enable Register (EGN)**).

12.2 8-Bit A/D Converter Configuration

The 8-bit A/D converter consists of the following hardware.

Table 12-1. 8-Bit A/D Converter Configuration

Item	Configuration
Analog input	8 channels (ANI0 to ANI7)
Registers	Successive approximation register (SAR) A/D conversion result register 1 (ADCR1)
Control registers	A/D converter mode register 0 (ADM0) Analog input channel specification register 0 (ADS0)

(1) Successive approximation register (SAR)

This register compares the analog input voltage value to the voltage tap (compare voltage) value applied from the series resistor string, and holds the result from the most significant bit (MSB).

When up to the least significant bit (LSB) is held (end of A/D conversion), the SAR contents are transferred to A/D conversion result register 1 (ADCR1).

(2) A/D conversion result register 1 (ADCR1)

ADCR1 is an 8-bit register that stores the A/D conversion result. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register.

ADCR1 is read by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ADCR1 to 00H.

Symbol									Address	After reset	R/W
ADCR1									FF15H	00H	R

Caution **When writing is performed to A/D converter mode register 0 (ADM0) and analog input channel specification register 0 (ADS0), the contents of ADCR1 may become undefined. Read the conversion result following conversion completion before writing to ADM0, ADS0. Using a timing other than the above may cause an incorrect conversion result to be read.**

(3) Sample & hold circuit

★ The sample & hold circuit samples the signal input to the analog input pin selected by the selector when A/D converter is started. This circuit holds the sampled analog input voltage value during A/D conversion.

(4) Voltage comparator

★ The voltage comparator compares the sampled analog input voltage to the series resistor string output voltage.

(5) Series resistor string

★ The series resistor string is connected between AV_{DD} and AV_{SS}, and generates a voltage to be compared to the analog input.

(6) ANI0 to ANI7 pins

These are eight analog input pins used to input analog signals to undergo A/D conversion to the A/D converter. ANI0 to ANI7 are alternate-function pins that can also be used for digital inputs.

★ **Cautions 1. Use the ANI0 to ANI7 input voltages within the specification range. If a voltage higher than or equal to AV_{DD} or lower than or equal to AV_{SS} is applied (even if within the absolute maximum rating range), the conversion value of that channel will be undefined and the conversion values of other channels may also be affected.**

2. Analog input (ANI0 to ANI7) pins are alternate function pins that can also be used as input port (P10 to P17) pins. When A/D conversion is performed by selecting any one of ANI0 to ANI7, do not access port 1 during conversion, as this may cause a lower conversion resolution.

3. When a digital pulse is applied to a pin adjacent to the pin being A/D converted, A/D conversion values may not be obtained as expected due to coupling noise. Thus, do not apply any pulse to a pin adjacent to the pin being A/D converted.

★ **(7) AV_{DD} pin**

This pin inputs the A/D converter analog power supply.

Use this pin at the same potential as the V_{DD0} or V_{DD1} pin even when the A/D converter is not used.

Signals input to ANI0 to ANI7 are converted into digital signals according to the voltage applied between AV_{DD} and AV_{SS}.

Caution A series resistor string of several 10 kΩ is connected between the AV_{DD} pin and AV_{SS} pin. Therefore, when the output impedance of the reference voltage is too high, it seems as if the AV_{DD} pin and the series resistor string are connected in series. This may cause a greater reference voltage error.

(8) AV_{SS} pin

This is the ground potential pin of the A/D converter. Always keep it at the same potential as the V_{SS0} or V_{SS1} pin even when not using the A/D converter.

12.3 Registers to Control 8-Bit A/D Converter

★ The following two registers are used to control the 8-bit A/D converter.

- A/D converter mode register 0 (ADM0)
- Analog input channel specification register 0 (ADS0)

(1) A/D converter mode register 0 (ADM0)

This register sets the conversion time for the analog input to be A/D converted, conversion start/stop, and an external trigger.

ADM0 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ADM0 to 00H.

Figure 12-2. Format of A/D Converter Mode Register 0 (ADM0)

Address: FF80H After reset: 00H R/W

Symbol	$\boxed{7}$	$\boxed{6}$	5	4	3	2	1	$\boxed{0}$
ADM0	ADCS0	TRG0	FR02	FR01	FR00	EGA01	EGA00	ADCE0

ADCS0	A/D conversion operation control
0	Stop conversion operation.
1	Enable conversion operation.

TRG0	Software start/hardware start selection
0	Software start
1	Hardware start

FR02	FR01	FR00	Conversion time selection^Note 1
0	0	0	144/fx (14.4 μs)
0	0	1	120/fx (Setting prohibited^Note 2)
0	1	0	96/fx (Setting prohibited^Note 2)
1	0	0	576/fx (57.6 μs)
1	0	1	480/fx (48.0 μs)
1	1	0	384/fx (38.4 μs)
Other than above			Setting prohibited

EGA01	EGA00	External trigger signal edge specification
0	0	No edge detection
0	1	Falling edge detection
1	0	Rising edge detection
1	1	Both falling and rising edge detection

ADCE0	Control of voltage booster for A/D converter circuit^Note 3
0	Stops operation.
1	Enables operation.

Notes 1. Set so that the A/D conversion time is 14 μs or more.

2. Setting prohibited because A/D conversion time is less than 14 μs at fx = 10 MHz.

★ **3.** 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 ADCE0, and it takes 14 μs from operation start to operation stabilization. Therefore, when ADCS0 is set to 1 after 14 μs or more has elapsed from the time ADCE0 is set to 1, the conversion result at that time has priority over the first conversion result.

Remarks 1. fx: Main system clock oscillation frequency

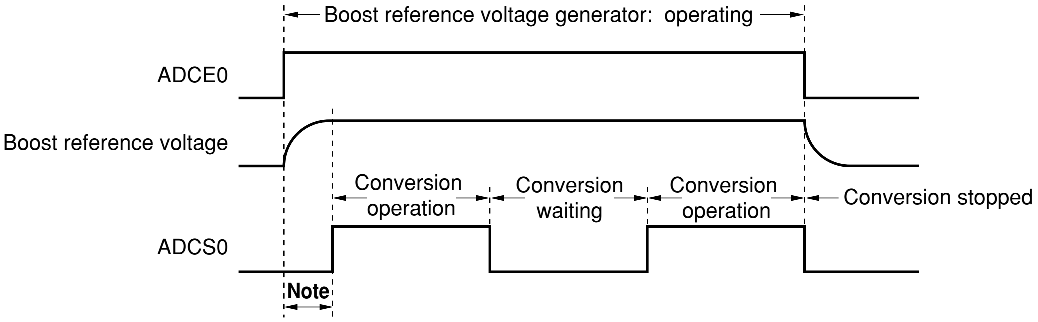
2. Figures in parentheses are for operation with fx = 10 MHz.

★ **Table 12-2. Settings of ADCS0 and ADCE0**

ADCS0	ADCE0	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[Note])
1	1	Conversion mode (reference voltage generator operates)

Note Data of the first conversion cannot be used immediately after A/D conversion is started.

★ **Figure 12-3. Timing Chart When Boost Reference Voltage Generator Is Used**



Note 14 μs or more is required for reference voltage stabilization.

Cautions 1. Stop A/D conversion operations once before rewriting FR00 to FR02 to other than the same data.

2. Before clearing ADCE0, clear ADCS0.

(2) Analog input channel specification register 0 (ADS0)

This register specifies the analog voltage input port for A/D conversion.

ADS0 is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ADS0 to 00H.

Figure 12-4. Format of Analog Input Channel Specification Register 0 (ADS0)

Address: FF81H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS0	0	0	0	0	0	ADS02	ADS01	ADS00

ADS02	ADS01	ADS00	Analog input channel specification
0	0	0	ANI0
0	0	1	ANI1
0	1	0	ANI2
0	1	1	ANI3
1	0	0	ANI4
1	0	1	ANI5
1	1	0	ANI6
1	1	1	ANI7

12.4 8-Bit A/D Converter Operations

12.4.1 Basic operations of 8-bit A/D converter

<1> Select one channel for A/D conversion using analog input channel specification register 0 (ADS0).

<2> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<3> When sampling has been done for a certain time, the sample & hold circuit is placed in the hold state and the input analog voltage is held until the A/D conversion operation is ended.

<4> Bit 7 of the successive approximation register (SAR) is set. The series resistor string voltage tap is set to (1/2) AV_{DD} by the tap selector.

<5> The voltage difference between the series resistor string voltage tap and analog input is compared by the voltage comparator. If the analog input is greater than (1/2) AV_{REF}, the MSB of SAR remains set. If the analog input is smaller than (1/2) AV_{DD}, the MSB is reset.

<6> Next, bit 6 of SAR is automatically set, and the operation proceeds to the next comparison. The series resistor string voltage tap is selected according to the preset value of bit 7, as described below.

- Bit 7 = 1: (3/4) AV_{DD}
- Bit 7 = 0: (1/4) AV_{DD}

The voltage tap and analog input voltage are compared and bit 6 of SAR is manipulated as follows.

- Analog input voltage ≥ Voltage tap: Bit 6 = 1
- Analog input voltage < Voltage tap: Bit 6 = 0

<7> Comparison is continued in this way up to bit 0 of SAR.

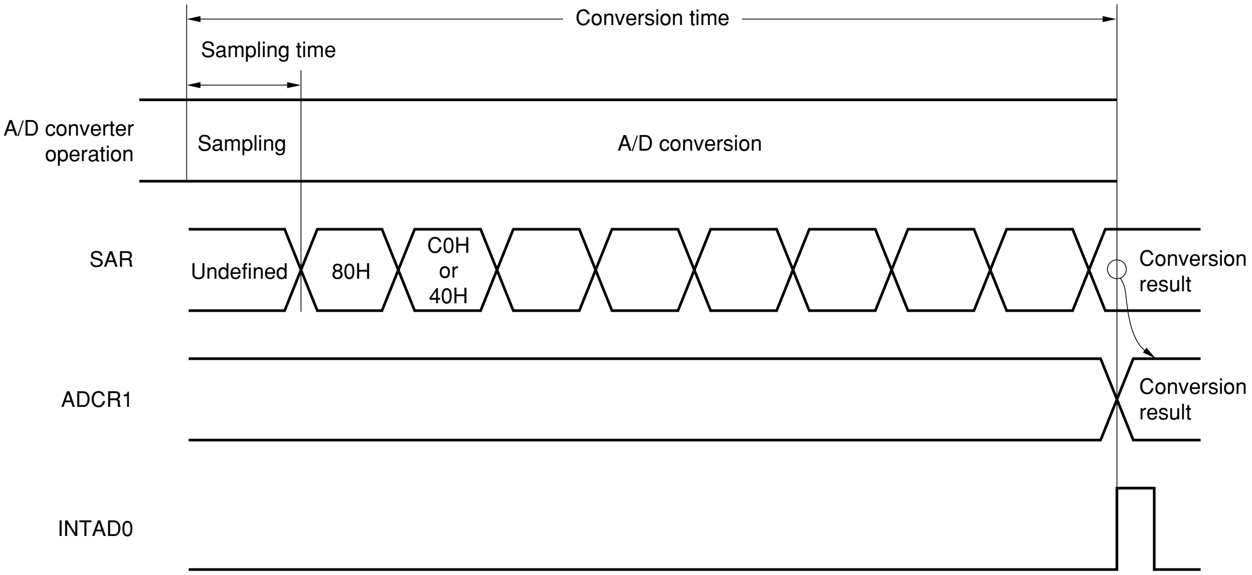
<8> Upon completion of the comparison of 8 bits, an effective digital result value remains in SAR, and the result value is transferred to and latched in A/D conversion result register 1 (ADCR1).

At the same time, the A/D conversion end interrupt request (INTAD0) can also be generated.

Cautions 1. Start conversion (ADCS0 = 1) after 14 μs have elapsed following the setting of ADCE0. If ADCE0 is not used, the first conversion result immediately after the setting of ADCS0 is undefined.

2. In standby mode, the A/D converter operation is stopped.

Figure 12-5. Basic Operation of 8-Bit A/D Converter



A/D conversion operations are performed continuously until bit 7 (ADCS0) of A/D converter mode register 0 (ADM0) is reset (0) by software.

If a write operation is performed to ADM0 or analog input channel specification register 0 (ADS0) during an A/D conversion operation, the conversion operation is initialized, and if the ADCS0 bit is set (1), conversion starts again from the beginning.

$\overline{\text{RESET}}$ input sets A/D conversion result register 1 (ADCR1) to 00H.

★ The completion of A/D conversion can be confirmed using the A/D conversion completion interrupt request flag (ADIF0).

12.4.2 Input voltage and conversion results

★ The relationship between the analog voltage input to the analog input pins (ANI0 to ANI7) and the logical A/D conversion result (stored in A/D conversion result register 1 (ADCR1)) is shown by the following expression.

★ $$\mathrm{ADCR1} = \mathrm{INT}\left(\frac{V_{IN}}{AV_{DD}} \times 256 + 0.5\right)$$

or

★ $$(\mathrm{ADCR1} - 0.5) \times \frac{AV_{DD}}{256} \leq V_{AIN} < (\mathrm{ADCR1} + 0.5) \times \frac{AV_{DD}}{256}$$

where, INT( ): Function which returns integer part of value in parentheses

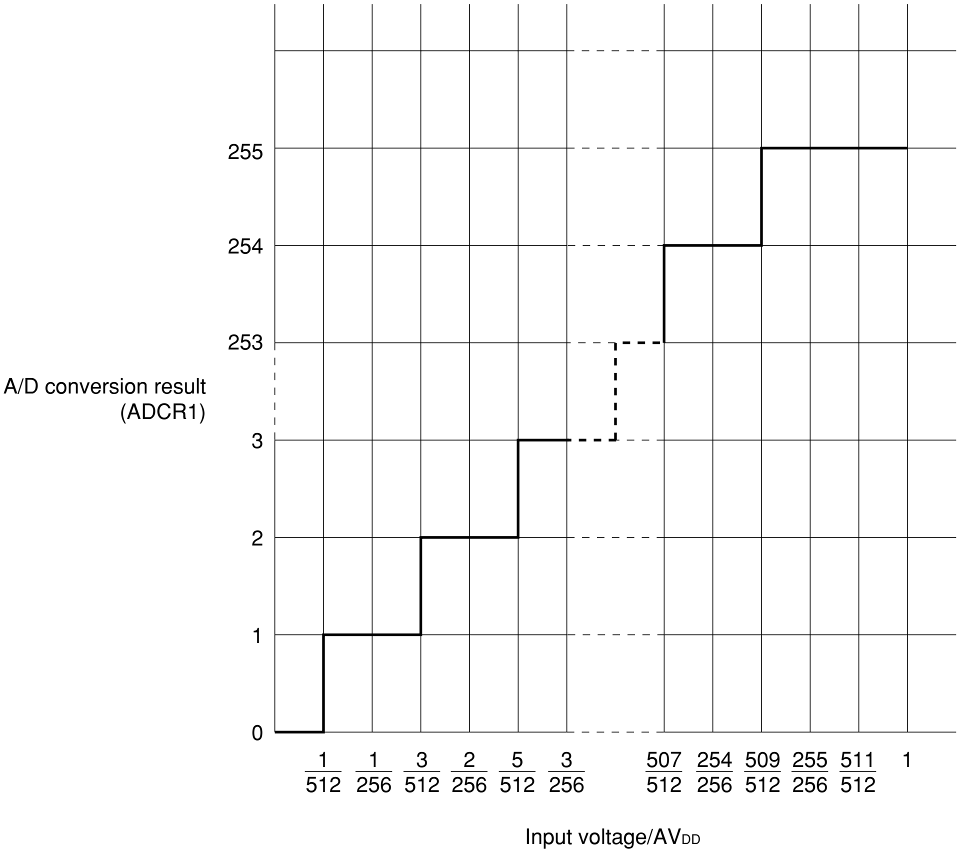
V_{AIN}: Analog input voltage

AV_{DD}: AV_{DD} pin voltage

ADCR1: A/D conversion result register 1 (ADCR1) value

Figure 12-6 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 12-6. Relationship Between Analog Input Voltage and A/D Conversion Result



12.4.3 8-bit A/D converter operation mode

Select one analog input channel from among ANI0 to ANI7 using analog input channel specification register 0 (ADS0) to start A/D conversion.

A/D conversion can be started in either of the following two ways.

- Hardware start: Conversion is started by trigger input (rising edge, falling edge, or both rising and falling edges specified).
- Software start: Conversion is started by specifying A/D converter mode register 0 (ADM0).

The A/D conversion result is stored in A/D conversion result register 1 (ADCR1), and the interrupt request signal (INTAD0) is simultaneously generated.

(1) A/D conversion by hardware start

When bit 6 (TRG0) and bit 7 (ADCS0) of A/D converter mode register 0 (ADM0) are set to 1 after bit 0 (ADCE0) is set to 1, the A/D conversion standby state is set. When the external trigger signal (ADTRG) is input, A/D conversion of the voltage applied to the analog input pins specified by analog input channel specification register 0 (ADS0) starts.

Upon the end of the A/D conversion, the conversion result is stored in A/D conversion result register 1 (ADCR1), and the interrupt request signal (INTAD0) is generated. After one A/D conversion operation is started and ended, the next conversion operation is not started until a new external trigger signal is input.

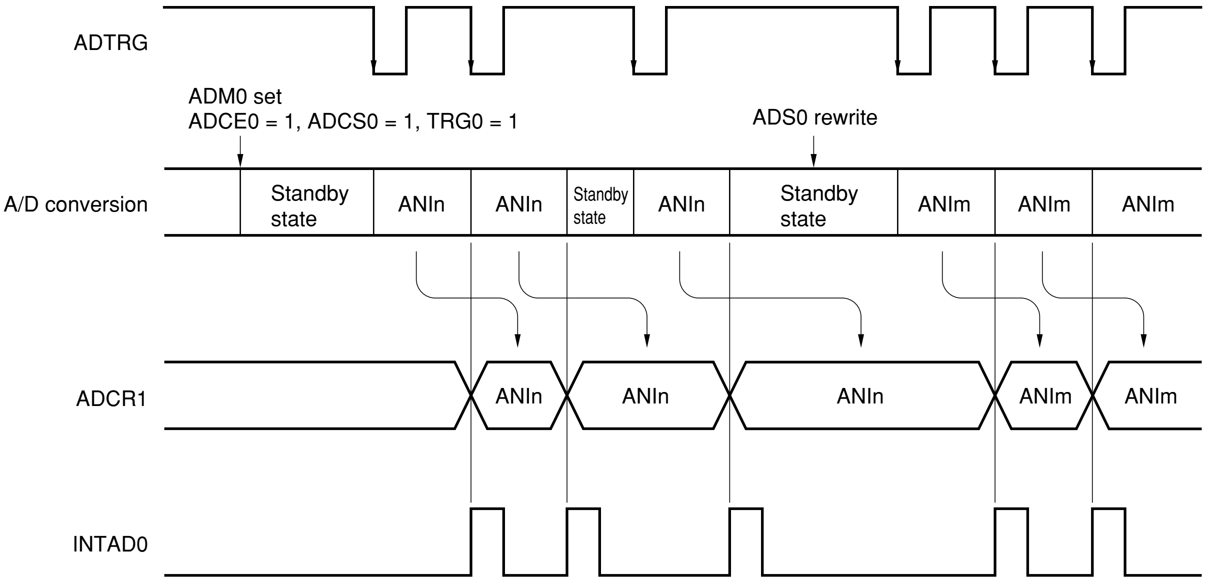
If ADS0 is rewritten during A/D conversion, the converter suspends A/D conversion and waits for a new external trigger signal to be input. When the external trigger input signal is reinput, A/D conversion is carried out from the beginning. If ADS0 is rewritten during A/D conversion wait, A/D conversion starts when the following external trigger input signal is input.

★ If ADCS0 is set to 0 during A/D conversion, the A/D conversion operation stops immediately.

Caution When P03/INTP3/ADTRG is used as the external trigger input (ADTRG), specify the valid edge by bits 1 and 2 (EGA00, EGA01) of A/D converter mode register 0 (ADM0) and set the interrupt mask flag (PMK3) to 1.

Figure 12-7. A/D Conversion by Hardware Start (When Falling Edge Is Specified)

★



Remarks 1. n = 0, 1, ..., 7

2. m = 0, 1, ..., 7

(2) A/D conversion by software start

When bit 6 (TRG0) and bit 7 (ADCS0) of A/D converter mode register 0 (ADM0) are set to 0 and 1 after bit 0 (ADCE0) is set to 1, A/D conversion of the voltage applied to the analog input pin specified by analog input channel specification register 0 (ADS0) starts.

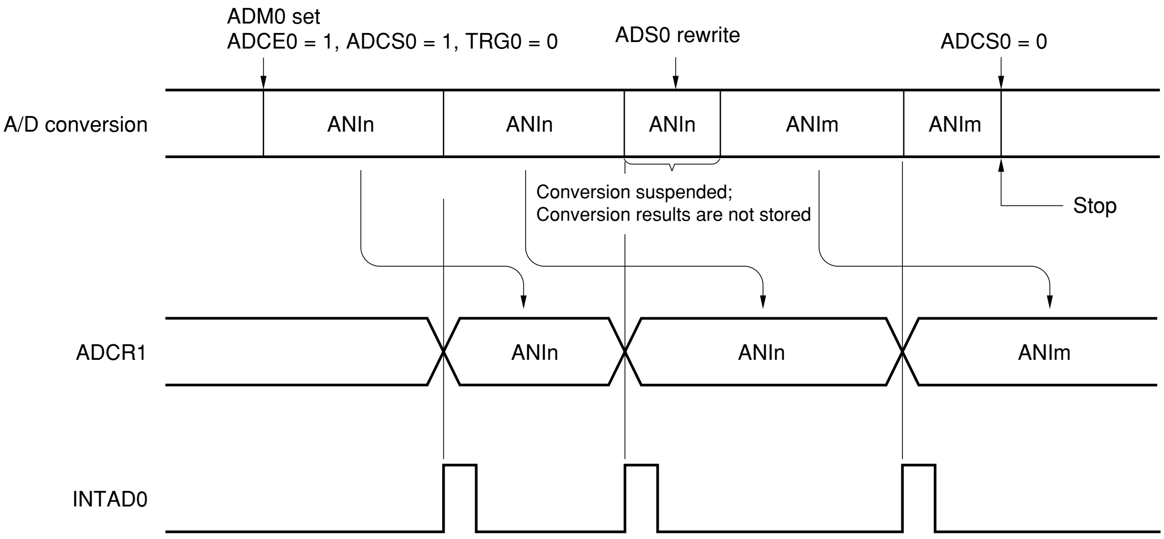
Upon the end of the A/D conversion, the conversion result is stored in A/D conversion result register 1 (ADCR1), and the interrupt request signal (INTAD0) is generated. After one A/D conversion operation is started and ended, the next conversion operation is immediately started. A/D conversion operations are repeated until new data is written to ADS0.

If ADS0 is rewritten during A/D conversion, the converter suspends A/D conversion and A/D conversion of the newly selected analog input channel is started.

★ If ADCS0 is set to 0 during A/D conversion, the A/D conversion operation stops immediately.

Figure 12-8. A/D Conversion by Software Start

★



Remarks 1. n = 0, 1, ..., 7

2. m = 0, 1, ..., 7

12.5 How to Read A/D Converter Characteristics Table

Here we will explain the special terms unique to A/D converters.

(1) Resolution

This is the minimum analog input voltage that can be identified. That is, the percentage of the analog input voltage per 1 bit of digital output is called 1LSB (Least Significant Bit). The percentage of 1LSB with respect to the full scale is expressed by %FSR (Full Scale Range).

When the resolution is 8 bits,

$$1\text{LSB} = 1/2^8 = 1/256$$
$$= 0.4\%\text{FSR}$$

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, differential linearity error and errors which are combinations of these express overall error.

Furthermore, quantization error is not included in overall error in the characteristics table.

(3) Quantization error

When analog values are converted to digital values, there naturally occurs a ±1/2LSB error. In an A/D converter, an analog input voltage in a range of ±1/2LSB are converted to the same digital code, so a quantization error cannot be avoided.

Furthermore, it is not included in the overall error, zero-scale error, full-scale error, integral linearity error, and differential linearity error in the characteristics table.

Figure 12-9. Overall Error

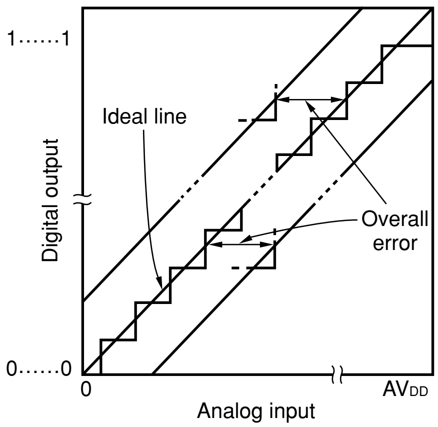
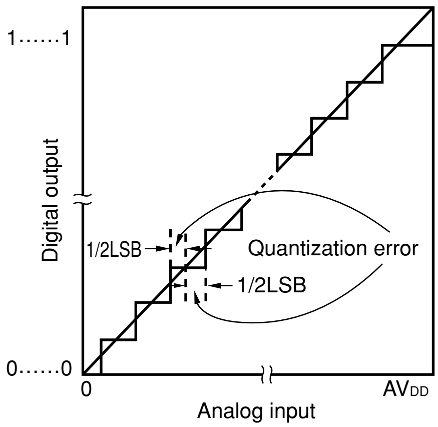


Figure 12-10. Quantization Error

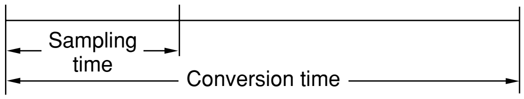


(4) Conversion time

★ This expresses the time from when sampling is started to the time when the digital output was obtained. Sampling time is included in the conversion time in the characteristics table.

(5) Sampling time

This is the time the analog switch is turned on for the analog voltage to be sampled by the sample & hold circuit.



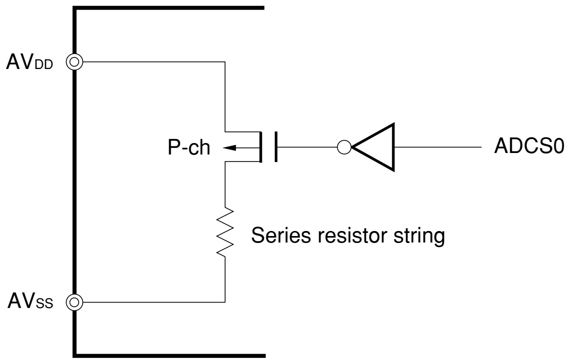
12.6 A/D Converter Cautions

(1) Power consumption in standby mode

The A/D converter stops operating in the standby mode. At this time, the power consumption can be reduced by setting bit 7 (ADCS0) of A/D converter mode register 0 (ADM0) to 0.

Figure 12-11 shows the circuit configuration of the series resistor string.

★ **Figure 12-11. Example of Series Resistor String Circuit Configuration**



(2) Input range of ANI0 to ANI7

The input voltages of ANI0 to ANI7 should be within the specification range. In particular, if a voltage higher ★ than or equal to AV_{DD} or lower than or equal to AV_{SS} is input (even if within the absolute maximum rating range), the conversion value of that channel will be undefined and the conversion values of other channels may also be affected.

(3) Conflicting operations

<1> Conflict between A/D conversion result register 1 (ADCR1) write and ADCR1 read by instruction upon the end of conversion

ADCR1 read is given priority. After the read operation, the new conversion result is written to ADCR1.

<2> Conflict between ADCR1 write and external trigger signal input upon the end of conversion

The external trigger signal is not accepted during A/D conversion. Therefore, the external trigger signal is not accepted during ADCR1 write.

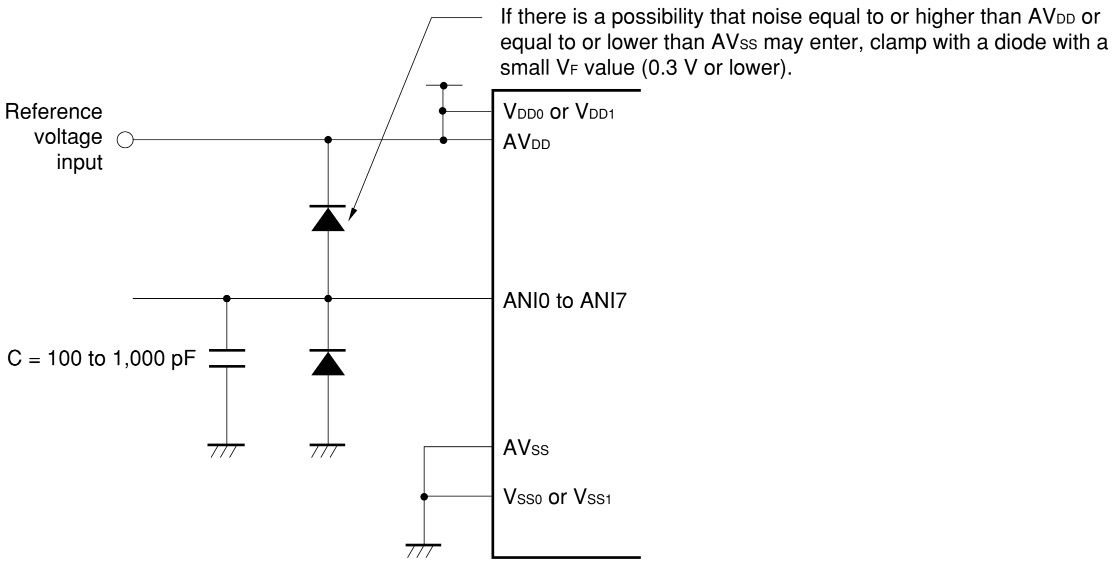
<3> Conflict between ADCR1 write and A/D converter mode register 0 (ADM0) write or analog input channel specification register 0 (ADS0) write upon the end of conversion

ADM0 or ADS0 write is given priority. ADCR1 write is not performed, nor is the conversion end interrupt request signal (INTAD0) generated.

(4) Noise countermeasures

★ To maintain the 8-bit resolution, attention must be paid to noise input to the AV_{DD} and ANI0 to ANI7 pins. Because the effect increases in proportion to the output impedance of the analog input source, it is recommended that a capacitor be connected externally as shown in Figure 12-12 to reduce noise.

★ **Figure 12-12. Analog Input Pin Connection**


If there is a possibility that noise equal to or higher than AV_{DD} or equal to or lower than AV_{SS} may enter, clamp with a diode with a small V_F value (0.3 V or lower).
Reference voltage input
V_{DD0} or V_{DD1}
AV_{DD}
ANI0 to ANI7
C = 100 to 1,000 pF
AV_{SS}
V_{SS0} or V_{SS1}

(5) ANI0/P10 to ANI7/P17

The analog input pins (ANI0 to ANI7) also function as input port pins (P10 to P17).

When A/D conversion is performed with any of the ANI0 to ANI7 pins selected, do not execute an input instruction to port 1 while conversion is in progress, as this may reduce the conversion resolution.

Also, if digital pulses are applied to a pin adjacent to the pin being A/D converted, the expected A/D conversion value may not be obtainable due to coupling noise. Therefore, avoid applying pulses to pins adjacent to the pin undergoing A/D conversion.

★ **(6) Input impedance of ANI0 to ANI7 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 kΩ or lower, or attach a capacitor of around 100 pF to the ANI0 to ANI7 pins (see **Figure 12-12**).

★ **(7) AV_{DD} pin input impedance**

A series resistor string of several 10 kΩ is connected between the AV_{DD} pin and the AV_{SS} pin.

Therefore, when the output impedance of the reference voltage is too high, it seems as if the AV_{DD} pin and the series resistor string are connected in series. This may cause a greater reference voltage error.

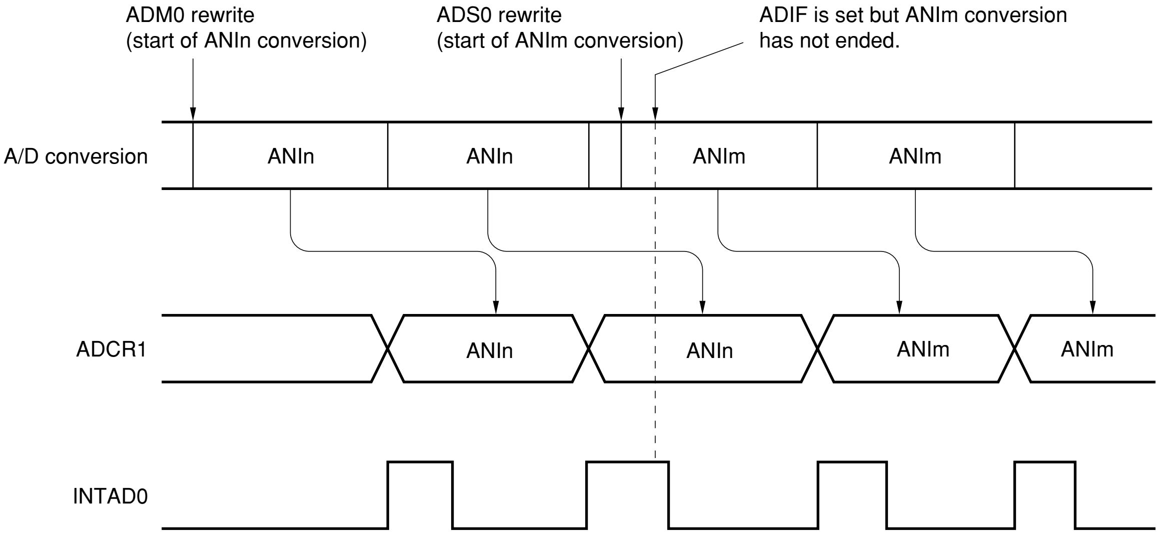
(8) Interrupt request flag (ADIF0)

The interrupt request flag (ADIF0) is not cleared even if analog input channel specification register 0 (ADS0) is changed.

Therefore, if an analog input pin is changed during A/D conversion, the A/D conversion result and conversion end interrupt request flag for the pre-change analog input may be set just before the ADS0 rewrite. Caution is therefore required since, at this time, when ADIF0 is read immediately just after the ADS0 rewrite, ADIF0 is set despite the fact that the A/D conversion for the post-change analog input has not ended.

When the A/D conversion is restarted after it is stopped, clear ADIF0 before restart.

Figure 12-13. A/D Conversion End Interrupt Request Generation Timing



Remarks 1. n = 0, 1, ..., 7

2. m = 0, 1, ..., 7

(9) Conversion results just after A/D conversion start

If bit 7 (ADCS0) of A/D converter mode register 0 (ADM0) is set to 1 without setting bit 0 (ADCE0) to 1, the first value converted immediately after A/D conversion has been started may not satisfy the rated value. Discard the first conversion result and use the second and subsequent results by polling the A/D conversion end interrupt request (INTAD0).

The same may apply if ADCS0 is set to 1 without the lapse of the wait time of 14 μs (MIN.) after ADCE0 has been set to 1. Make sure that the specified wait time elapses.

(10) A/D conversion result register 1 (ADCR1) read operation

When writing is performed to A/D converter mode register 0 (ADM0) and analog input channel specification register 0 (ADS0), the contents of ADCR1 may become undefined. Read the conversion result following conversion completion before writing to ADM0, ADS0. Using a timing other than the above may cause an incorrect conversion result to be read.

(11) Timing at which A/D conversion result is undefined

The A/D conversion value may be undefined if the timing of completion of A/D conversion and the timing of stopping the A/D conversion conflict with each other. Therefore, read the A/D conversion result during the A/D conversion operation. To read the conversion result after stopping the A/D conversion operation, be sure to stop the A/D conversion before the next conversion ends.

Figures 12-14 and 12-15 show the timing of reading the conversion result.

Figure 12-14. Timing of Reading Conversion Result (When Conversion Result Is Undefined)

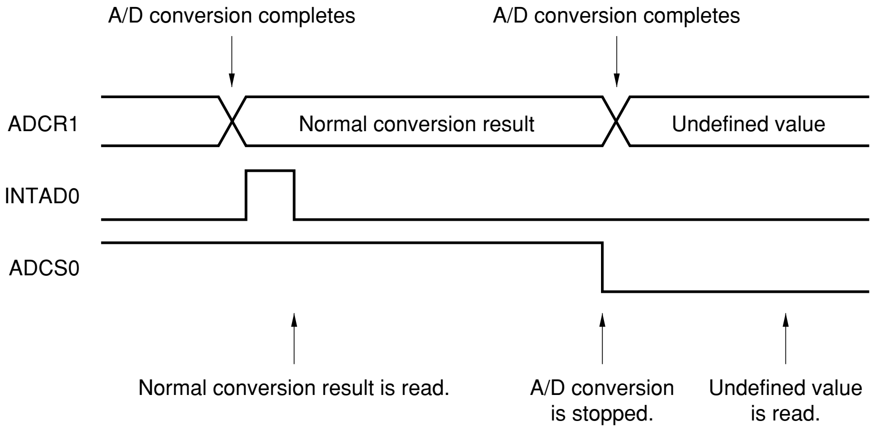
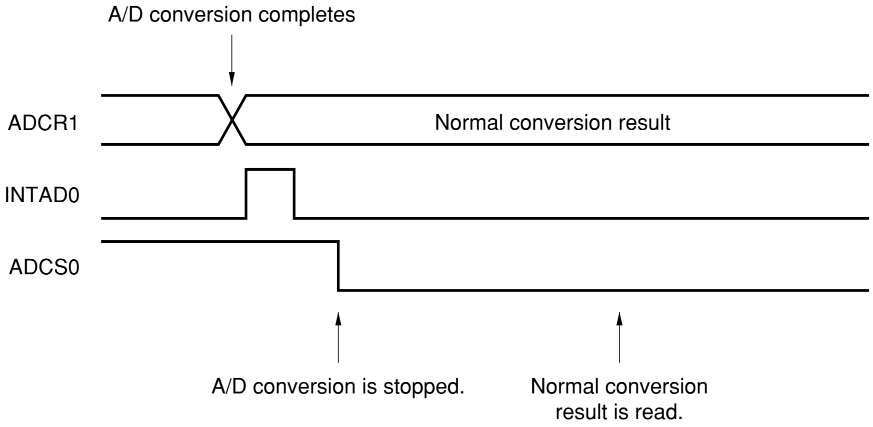


Figure 12-15. Timing of Reading Conversion Result (When Conversion Result Is Normal)



(12) Notes on board design

Locate analog circuits as far away from digital circuits as possible on the board because the analog circuits may be affected by the noise of the digital circuits. In particular, do not cross an analog signal line with a digital signal line, or wire an analog signal line in the vicinity of a digital signal line. Otherwise, the A/D conversion characteristics may be affected by the noise of the digital line.

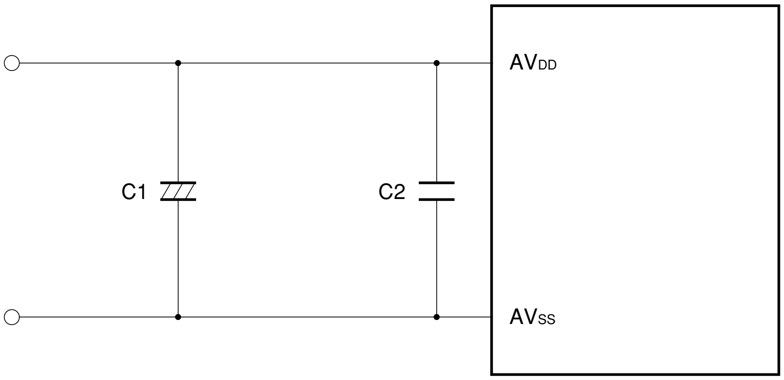
Connect AV_{SS} and V_{SS1} at one location on the board where the voltages are stable.

★ **(13) AV_{DD} and AV_{SS} pins**

Connect a capacitor between the AV_{DD} and AV_{SS} pins to minimize conversion errors due to noise. If an A/D conversion operation has been stopped and then is started, the voltage applied to the AV_{DD} and AV_{SS} pins becomes unstable, causing the accuracy of the A/D conversion to drop. To prevent this, also connect a capacitor between the AV_{DD} and AV_{SS} pins.

Figure 12-16 shows an example of connecting a capacitor.

★ **Figure 12-16. Example of Connecting Capacitor to AV_{DD} Pin**

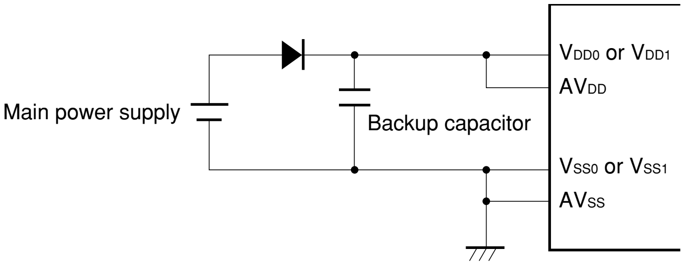


Remark C1: 4.7 μF to 10 μF (reference value)
C2: 0.01 μF to 0.1 μF (reference value)
Connect C2 as close to the pin as possible.

★ **(14) AV_{DD} pin**

The AV_{DD} pin supplies power to the analog circuit. It also supplies power to the input circuit of ANI0 to ANI7. Therefore, apply the same potential as that of the V_{DD0} or V_{DD1} pin to this pin, as shown in Figure 12-17, in an application where a backup power supply is used.

Figure 12-17. Processing of AV_{DD} Pin



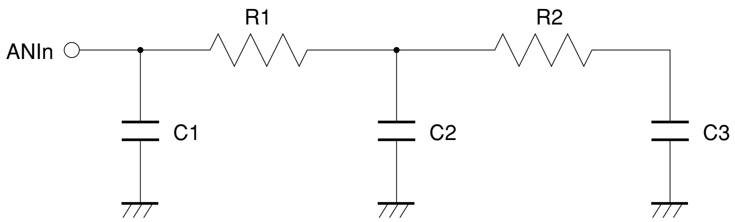
(15) Internal equivalent circuit of ANI0 to ANI7 pins and permissible signal source impedance

To complete sampling within the sampling time with sufficient A/D conversion accuracy, the impedance of the signal source such as a sensor must be sufficiently low. Figure 12-18 shows the internal equivalent circuit of the ANI0 to ANI7 pins.

If the impedance of the signal source is high, connect capacitors with a high capacitance to the ANI0 to ANI7 pins. An example of this is shown in Figure 12-19. In this case, however, the microcontroller cannot follow an analog signal with a high differential coefficient because a lowpass filter is created.

To convert a high-speed analog signal or to convert an analog signal in the scan mode, insert a low-impedance buffer.

Figure 12-18. Internal Equivalent Circuit of ANI0 to ANI7 Pins



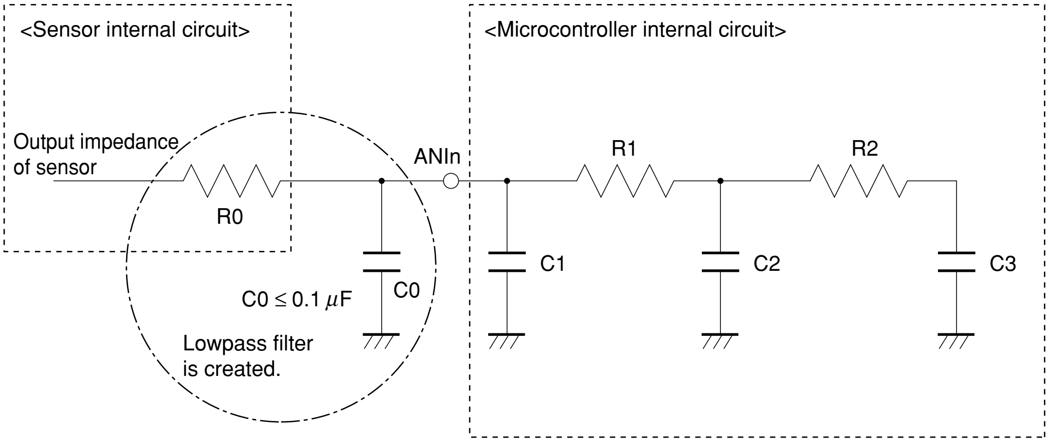
Remark n = 0 to 7

★ **Table 12-3. Resistances and Capacitances of Equivalent Circuit (Reference Values)**

AV_{DD}	R1	R2	C1	C2	C3
2.7 V	12 kΩ	8 kΩ	8 pF	3 pF	2 pF
4.5 V	4 kΩ	2.7 kΩ	8 pF	1.4 pF	2 pF

Caution The resistances and capacitances in Table 12-3 are not guaranteed values.

Figure 12-19. Example of Connection if Signal Source Impedance Is High


<Sensor internal circuit>
<Microcontroller internal circuit>
Output impedance of sensor
R0
ANIn
R1
R2
C0
C0 ≤ 0.1 μF
Lowpass filter is created.
C1
C2
C3

Remark n = 0 to 7

13.1 10-Bit A/D Converter Functions

The 10-bit A/D converter is a 10-bit resolution converter that converts analog inputs into digital signals. It can control up to 8 analog input channels (ANI0 to ANI7).

(1) Hardware start

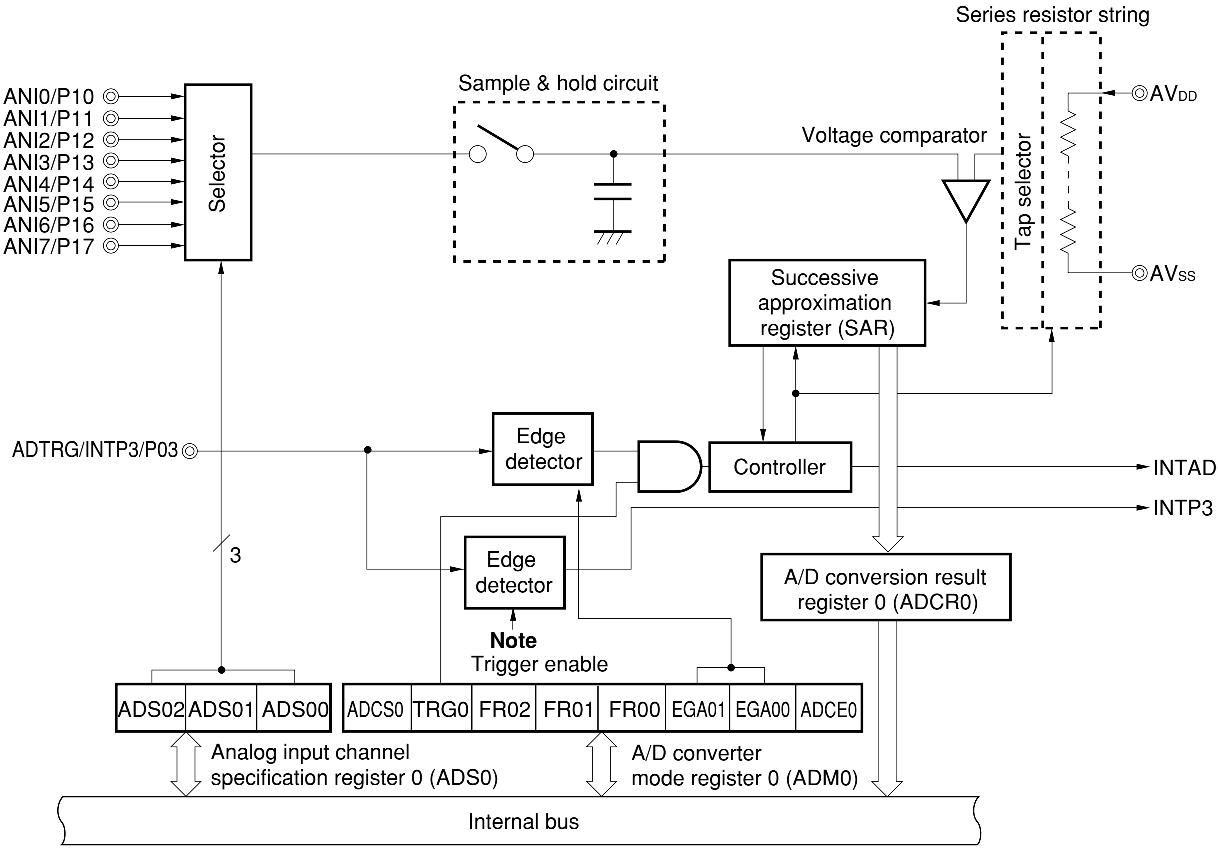
Conversion is started by trigger input (ADTRG: rising edge, falling edge, or both rising and falling edges can be specified).

(2) Software start

Conversion is started by setting A/D converter mode register 0 (ADM0).

Select one channel for analog input from ANI0 to ANI7 to start A/D conversion. In the case of hardware start, the A/D converter stops when A/D conversion is completed, and an interrupt request (INTAD0) is generated. In the case of software start, A/D conversion is repeated. Each time as A/D conversion operation ends, an interrupt request (INTAD0) is generated.

★ **Figure 13-1. 10-Bit A/D Converter Block Diagram**



Note The valid edge of the external interrupt is specified by bit 3 of the EGP and EGN registers (see **Figure 19-5 Format of External Interrupt Rising Edge Enable Register (EGP), External Interrupt Falling Edge Enable Register (EGN)**).

13.2 10-Bit A/D Converter Configuration

10-bit A/D converter consists of the following hardware.

Table 13-1. 10-Bit A/D Converter Configuration

Item	Configuration
Analog input	8 channels (ANI0 to ANI7)
Registers	Successive approximation register (SAR) A/D conversion result register 0 (ADCR0)
Control registers	A/D converter mode register 0 (ADM0) Analog input channel specification register 0 (ADS0)

(1) Successive approximation register (SAR)

This register compares the analog input voltage value to the voltage tap (compare voltage) value applied from the series resistor string, and holds the result from the most significant bit (MSB).

When up to the least significant bit (LSB) is hold (end of A/D conversion), the SAR contents are transferred to A/D conversion result register 0 (ADCR0).

(2) A/D conversion result register 0 (ADCR0)

This is a 16-bit register which stores the A/D conversion results. Lower 6 bits are fixed to 0. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register.

★ The result is stored in ADCR0 in order from the most significant bit (MSB). The higher 8 bits of the conversion result are input to FF0FH and the lower 2 bits of the conversion result are input to FF0EH.

ADCR0 is read by a 16-bit memory manipulation instruction.

★ $\overline{\text{RESET}}$ input sets the value of this register to 0000H.

Symbol	FF0FH								FF0EH								Address	After reset	R/W
ADCR0											0	0	0	0	0	0	FF0EH, FF0FH	0000H	R

Caution When writing is performed to A/D converter mode register 0 (ADM0) and analog input channel specification register 0 (ADS0), the contents of ADCR0 may become undefined. Read the conversion result following conversion completion before writing to ADM0, ADS0. Using a timing other than the above may cause an incorrect conversion result to be read.

(3) Sample & hold circuit

★ The sample & hold circuit samples the signal input to the analog input pin selected by the selector when A/D conversion is started. This circuit holds the sampled analog input voltage value during A/D conversion.

(4) Voltage comparator

★ The voltage comparator compares the sampled analog input voltage to the series resistor string output voltage.

(5) Series resistor string

★ The series resistor string is connected between AV_{DD} and AV_{SS}, and generates a voltage to be compared to the analog input.

(6) ANI0 to ANI7 pins

These are eight analog input pins used to input analog signals to undergo A/D conversion to the A/D converter. ANI0 to ANI7 can be used as input ports except for the pins specified as analog input by analog input channel specification register 0 (ADS0).

★ **Cautions 1. Use the ANI0 to ANI7 input voltages within the specification range. If a voltage higher than or equal to AV$_{DD}$ or lower than or equal to AV$_{SS}$ is applied (even if within the absolute maximum rating range), the conversion value of that channel will be undefined and the conversion values of other channels may also be affected.**

2. Analog input (ANI0 to ANI7) pins are alternate function pins that can also be used as input port (P10 to P17) pins. When A/D conversion is performed by selecting any one of ANI0 through ANI7, do not access port 1 during conversion, as this may cause a lower conversion resolution.

3. When a digital pulse is applied to a pin adjacent to the pin being A/D converted, A/D conversion values may not be obtained as expected due to coupling noise. Thus, do not apply any pulse to a pin adjacent to the pin being A/D converted.

★ **(7) AV$_{DD}$ pin**

This pin inputs the A/D converter analog power supply.

Use this pin at the same potential as the V$_{DD0}$ or V$_{DD1}$ pin even when the A/D converter is not used.

Signals input to ANI0 to ANI7 are converted into digital signals according to the voltage applied between AV$_{DD}$ and AV$_{SS}$.

Caution A series resistor string is connected between the AV$_{DD}$ and AV$_{SS}$ pins. Therefore, when the output impedance of the reference voltage is too high, it seems as if the AV$_{DD}$ pin and the series resistor string are connected in series. This may cause a greater reference voltage error.

(8) AV$_{SS}$ pin

This is the ground potential pin of the A/D converter. Always keep it at the same potential as the V$_{SS0}$ or V$_{SS1}$ pin when not using the A/D converter.

13.3 Registers to Control 10-Bit A/D Converter

★ The following two registers are used to control the 10-bit A/D converter.

- A/D converter mode register 0 (ADM0)
- Analog input channel specification register 0 (ADS0)

(1) A/D converter mode register 0 (ADM0)

This register sets the conversion time for the analog input to be A/D converted, conversion start/stop, and an external trigger.

ADM0 is set by a 1-bit or 8-bit memory manipulation instruction.

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

Figure 13-2. Format of A/D Converter Mode Register 0 (ADM0)

Address: FF80H After reset: 00H R/W

Symbol	$\boxed{7}$	$\boxed{6}$	5	4	3	2	1	$\boxed{0}$
ADM0	ADCS0	TRG0	FR02	FR01	FR00	EGA01	EGA00	ADCE0

ADCS0	A/D conversion operation control
0	Stop conversion operation.
1	Enable conversion operation.

TRG0	Software start/hardware start selection
0	Software start
1	Hardware start

FR02	FR01	FR00	Conversion time selection[Note 1]
0	0	0	144/f_X (14.4 μs)
0	0	1	120/f_X (Setting prohibited[Note 2])
0	1	0	96/f_X (Setting prohibited[Note 2])
1	0	0	576/f_X (57.6 μs)
1	0	1	480/f_X (48.0 μs)
1	1	0	384/f_X (38.4 μs)
Other than above			Setting prohibited

EGA01	EGA00	External trigger signal edge specification
0	0	No edge detection
0	1	Falling edge detection
1	0	Rising edge detection
1	1	Both falling and rising edge detection

ADCE0	Control of voltage booster for A/D converter circuit[Note 3]
0	Stops operation.
1	Enables operation.

Notes 1. Set so that the A/D conversion time is 14 μs or more.

2. Setting prohibited because A/D conversion time is less than 14 μs at f_X = 10 MHz.

★ **3.** 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 ADCE0, and it takes 14 μs from operation start to operation stabilization. Therefore, when ADCS0 is set to 1 after 14 μs or more has elapsed from the time ADCE0 is set to 1, the conversion result at that time has priority over the first conversion result.

Remarks 1. f_X: Main system clock oscillation frequency

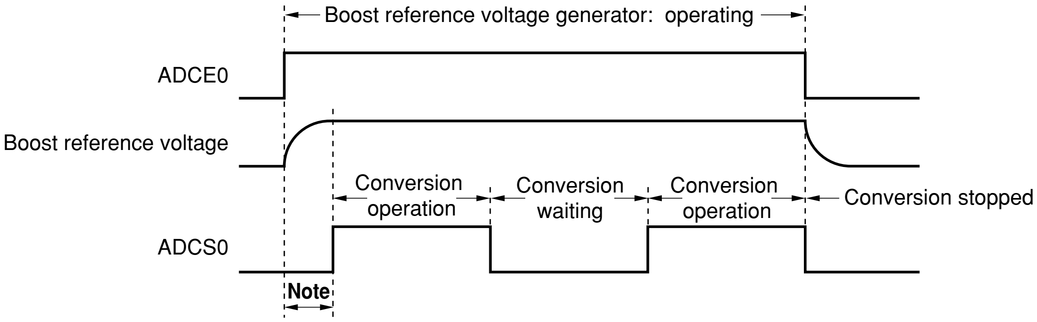
2. Figures in parentheses are for operation with f_X = 10 MHz.

★ **Table 13-2. Settings of ADCS0 and ADCE0**

ADCS0	ADCE0	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[Note])
1	1	Conversion mode (reference voltage generator operates)

Note Data of the first conversion cannot be used immediately after A/D conversion is started.

★ **Figure 13-3. Timing Chart When Boost Reference Voltage Generator Is Used**



Note 14 μs or more are required for reference voltage stabilization.

Cautions 1. Stop A/D conversion operations once before rewriting FR00 to FR02 to other than the same data.

2. Before clearing ADCE0, clear ADCS0.

(2) Analog input channel specification register 0 (ADS0)

This register specifies the analog voltage input port for A/D conversion.

ADS0 is set by an 8-bit memory manipulation instruction.

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

Figure 13-4. Format of Analog Input Channel Specification Register 0 (ADS0)

Address: FF81H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS0	0	0	0	0	0	ADS02	ADS01	ADS00

ADS02	ADS01	ADS00	Analog input channel specification
0	0	0	ANI0
0	0	1	ANI1
0	1	0	ANI2
0	1	1	ANI3
1	0	0	ANI4
1	0	1	ANI5
1	1	0	ANI6
1	1	1	ANI7

13.4 10-Bit A/D Converter Operation

13.4.1 Basic operations of 10-bit A/D converter

<1> Select one channel for A/D conversion using analog input channel specification register 0 (ADS0).

<2> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<3> When sampling has been done for a certain time, the sample & hold circuit is placed in the hold state and the input analog voltage is held until the A/D conversion operation is ended.

<4> Bit 9 of the successive approximation register (SAR) is set. The series resistor string voltage tap is set to (1/2) AV_{DD} 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_{DD}, the MSB of SAR remains set. If the analog input is smaller than (1/2) AV_{DD}, the MSB is reset. ★

<6> Next, bit 8 of SAR is automatically set, and the operation proceeds to the next comparison. The series resistor string voltage tap is selected according to the preset value of bit 9, as described below.

★ • Bit 9 = 1: (3/4) AV_{DD}

★ • Bit 9 = 0: (1/4) AV_{DD}

The voltage tap and analog input voltage are compared and bit 8 of SAR is manipulated as follows.

- Analog input voltage ≥ Voltage tap: Bit 8 = 1
- Analog input voltage < Voltage tap: Bit 8 = 0

<7> Comparison is continued in this way up to bit 0 of SAR.

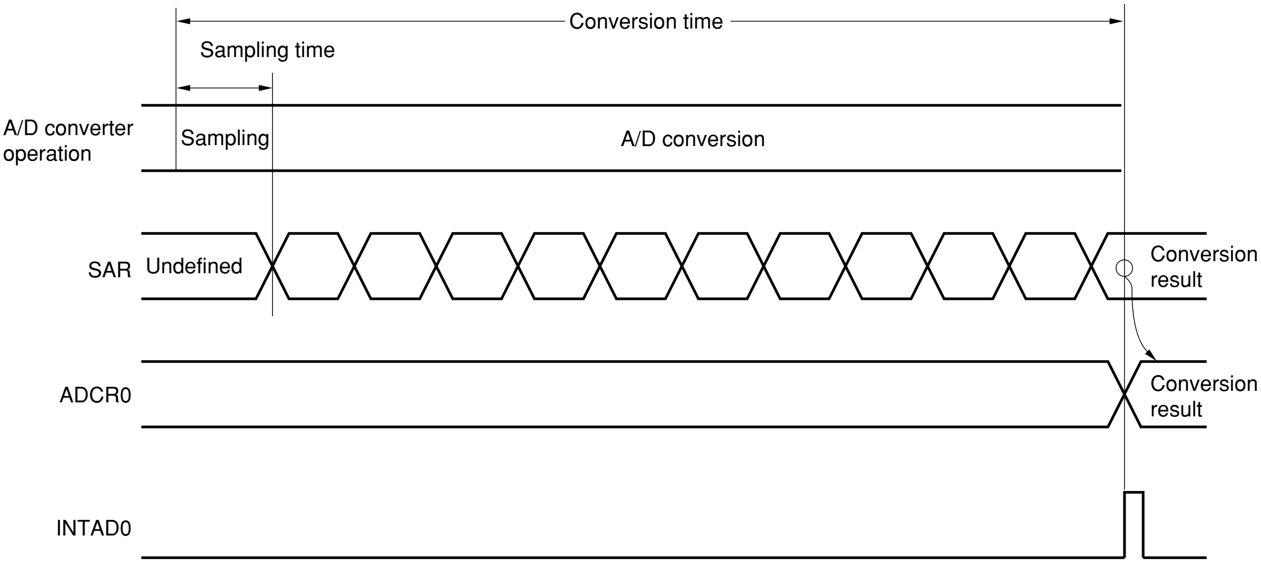
<8> Upon completion of the comparison of 10 bits, an effective digital result value remains in SAR, and the result value is transferred to and latched in A/D conversion result register 0 (ADCR0).

At the same time, the A/D conversion end interrupt request (INTAD0) can also be generated.

Cautions 1. Start conversion (ADCS0 = 1) after 14 μs have elapsed following the setting of ADCE0. If ADCE0 is not used, the first conversion result immediately after the setting of ADCS0 is undefined.

2. In standby mode, the A/D converter operation is stopped.

Figure 13-5. Basic Operation of 10-Bit A/D Converter



A/D conversion operations are performed continuously until bit 7 (ADCS0) of A/D converter mode register 0 (ADM0) is reset (0) by software.

If a write operation is performed to ADM0 or analog input channel specification register 0 (ADS0) during an A/D conversion operation, the conversion operation is initialized, and if the ADCS0 bit is set (1), conversion starts again from the beginning.

$\overline{\text{RESET}}$ input sets the A/D conversion result register 0 (ADCR0) to 00H.

★ The completion of A/D conversion can be confirmed using the A/D conversion completion interrupt request flag (ADIF0).

13.4.2 Input voltage and conversion results

★ The relationship between the analog voltage input to the analog input pins (ANI0 to ANI7) and the logical A/D conversion result (stored in A/D conversion result register 0 (ADCR0)) is shown by the following expression.

★ $$\mathrm{ADCR0} = \mathrm{INT}\left(\frac{V_{IN}}{AV_{DD}} \times 1{,}024 + 0.5\right)$$

or

★ $$(\mathrm{ADCR0} - 0.5) \times \frac{AV_{DD}}{1{,}024} \leq V_{AIN} < (\mathrm{ADCR0} + 0.5) \times \frac{AV_{DD}}{1{,}024}$$

where, INT( ): Function which returns integer part of value in parentheses

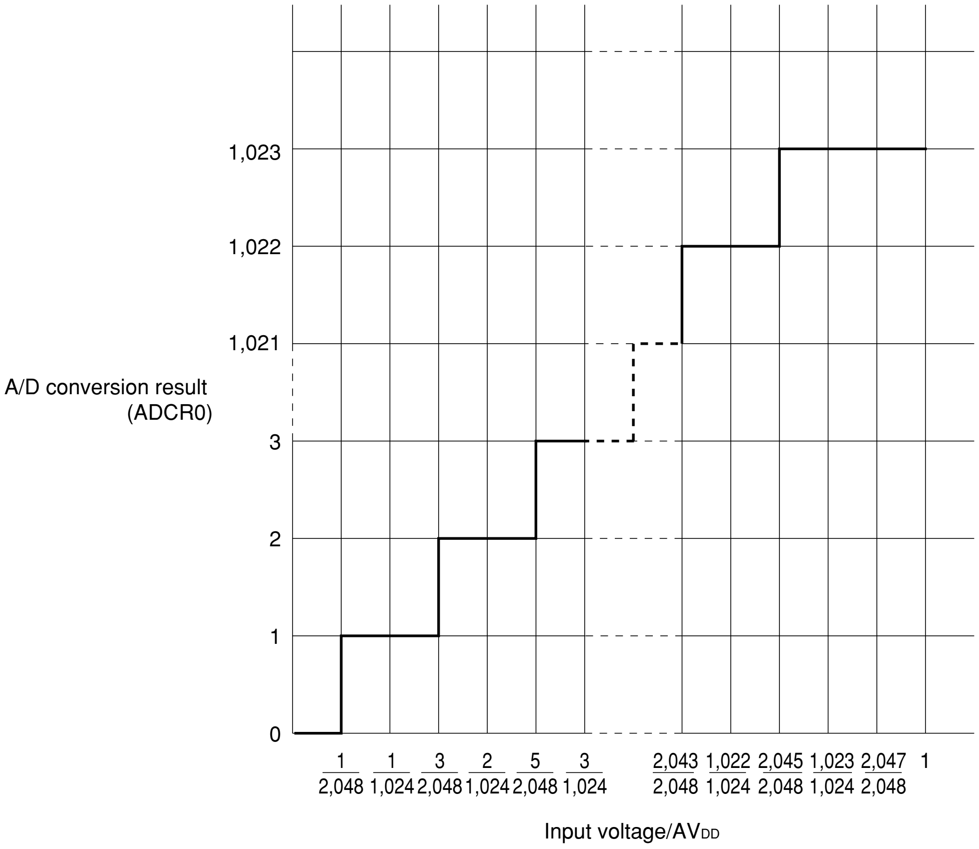
V_{AIN}: Analog input voltage

AV_{DD}: AV_{DD} pin voltage

ADCR0: A/D conversion result register 0 (ADCR0) value

Figure 13-6 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 13-6. Relationship Between Analog Input Voltage and A/D Conversion Result



13.4.3 10-bit A/D converter operation mode

Select one analog input channel from among ANI0 to ANI7 using analog input channel specification register 0 (ADS0) to start A/D conversion.

A/D conversion can be started in either of the following two ways.

- Hardware start: Conversion is started by trigger input (rising edge, falling edge, or both rising and falling edges specified).
- Software start: Conversion is started by specifying A/D converter mode register 0 (ADM0).

The A/D conversion result is stored in A/D conversion result register 0 (ADCR0), and the interrupt request signal (INTAD0) is simultaneously generated.

(1) A/D conversion by hardware start

When bit 6 (TRG0) and bit 7 (ADCS0) of A/D converter mode register 0 (ADM0) are set to 1 after bit 0 (ADCE0) is set to 1, the A/D conversion standby state is set. When the external trigger signal (ADTRG) is input, A/D conversion of the voltage applied to the analog input pin specified by analog input channel specification register 0 (ADS0) starts.

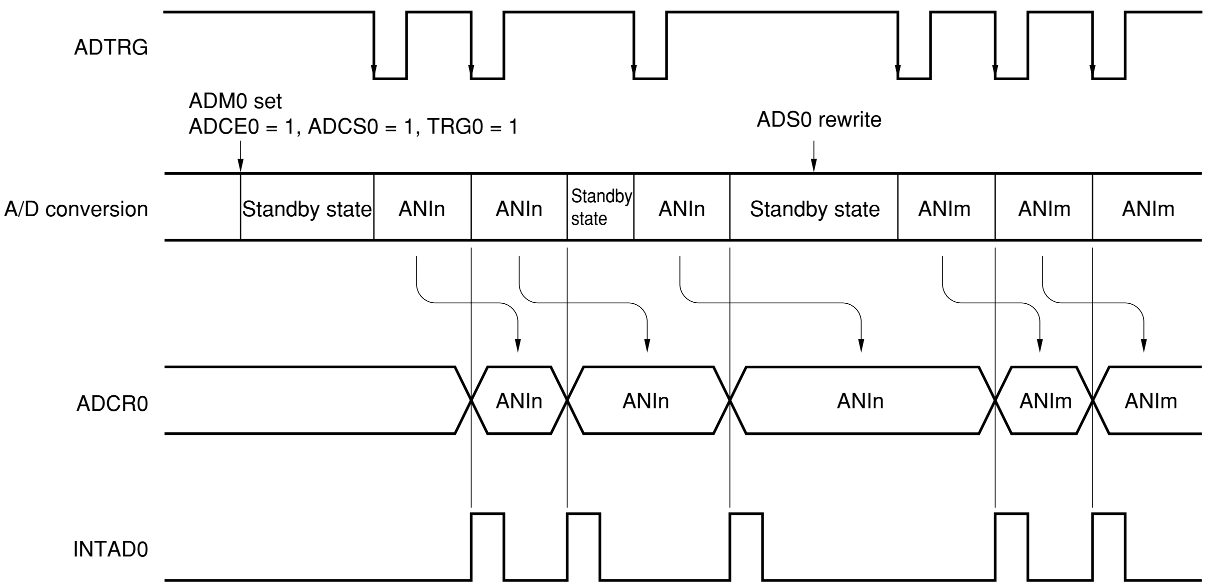
Upon the end of the A/D conversion, the conversion result is stored in A/D conversion result register 0 (ADCR0), and the interrupt request signal (INTAD0) is generated. After one A/D conversion operation is started and ended, the next conversion operation is not started until a new external trigger signal is input.

If ADS0 is rewritten during A/D conversion operation, the converter suspends A/D conversion and waits for a new external trigger signal to be input. When the external trigger input signal is reinput, A/D conversion is carried out from the beginning. If ADS0 is rewritten during A/D conversion wait, A/D conversion starts when the following external trigger input signal is input.

★ If ADCS0 is set to 0 during A/D conversion, the A/D conversion operation stops immediately.

Caution When P03/INTP3/ADTRG is used as the external trigger input (ADTRG), specify the valid edge by bits 1, 2 (EGA00, EGA01) of A/D converter mode register 0 (ADM0) and set the interrupt mask flag (PMK3) to 1.

Figure 13-7. A/D Conversion by Hardware Start (When Falling Edge Is Specified)



Remarks 1. n = 0, 1, ..., 7

2. m = 0, 1, ..., 7

(2) A/D conversion by software start

When bit 6 (TRG0) and bit 7 (ADCS0) of A/D converter mode register 0 (ADM0) are set to 0 and 1 after bit 0 (ADCE0) is set to 1, A/D conversion of the voltage applied to the analog input pin specified by analog input channel specification register 0 (ADS0) starts.

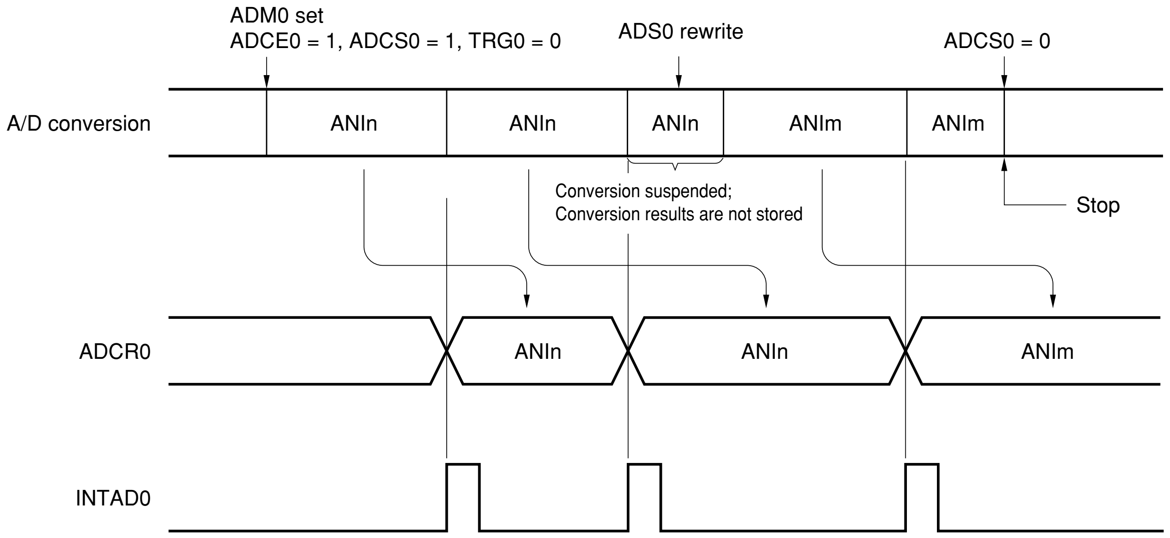
Upon the end of the A/D conversion, the conversion result is stored in A/D conversion result register 0 (ADCR0), and the interrupt request signal (INTAD0) is generated. After one A/D conversion operation is started and ended, the next conversion operation is immediately started. A/D conversion operations are repeated until new data is written to ADS0.

If ADS0 is rewritten during A/D conversion, the converter suspends A/D conversion operation and A/D conversion of the new selected analog input channel starts.

★ If ADCS0 is set to 0 during A/D conversion, the A/D conversion operation stops immediately.

Figure 13-8. A/D Conversion by Software Start

Remarks 1. n = 0, 1, ..., 7



2. m = 0, 1, ..., 7

13.5 How to Read the A/D Converter Characteristics Table

Here we will explain the special terms unique to A/D converters.

(1) Resolution

This is the minimum analog input voltage that can be identified. That is, the percentage of the analog input voltage per 1 bit of digital output is called 1LSB (Least Significant Bit). The percentage of 1LSB with respect to the full scale is expressed by %FSR (Full Scale Range).

When the resolution is 10 bits,

$$1\text{LSB} = 1/2^{10} = 1/1{,}024$$
$$= 0.098\%\text{FSR}$$

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, differential linearity error and errors which are combinations of these express overall error.

Furthermore, quantization error is not included in overall error in the characteristics table.

(3) Quantization error

When analog values are converted to digital values, there naturally occurs a ±1/2LSB error. In an A/D converter, an analog input voltage in a range of ±1/2LSB are converted to the same digital code, so a quantization error cannot be avoided.

Furthermore, it is not included in the overall error, zero-scale error, full-scale error, integral linearity error, and differential linearity error in the characteristics table.

Figure 13-9. Overall Error

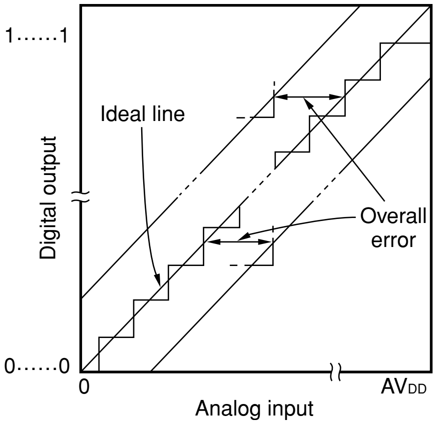
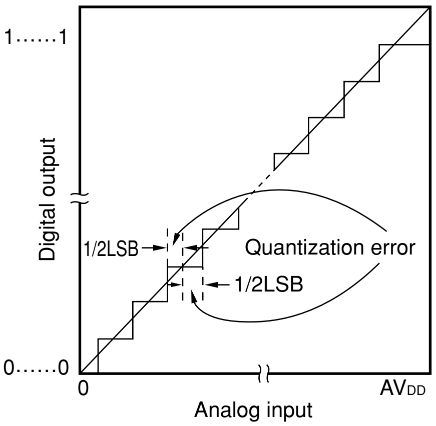


Figure 13-10. Quantization Error



(4) Zero-scale error

This shows the difference between the actual measured value of the analog input voltage and the theoretical value (1/2 LSB) when the digital output changes from 0……000 to 0……001. If the actual measured value is greater than the theoretical value, it shows the difference between the actual measured value of the analog input voltage and the theoretical value (3/2LSB) when the digital output changes from 0……001 to 0……010.

(5) Full-scale error

This shows the difference between the actual measured value of the analog input voltage and the theoretical value (full scale –3/2 LSB) when the digital output changes from 1……110 to 1……111.

(6) Integral linearity error

This shows the degree to which the conversion characteristics deviate from the ideal linear relationship. It expresses the maximum value of the difference between the actual measured value and the ideal straight line when the zero scale error and full scale error are 0.

(7) Differential linearity error

Ideal width to output a certain code is 1LSB. The following shows the difference between the actual measurement values and ideal values of the width when outputting a certain code.

Figure 13-11. Zero-Scale Error

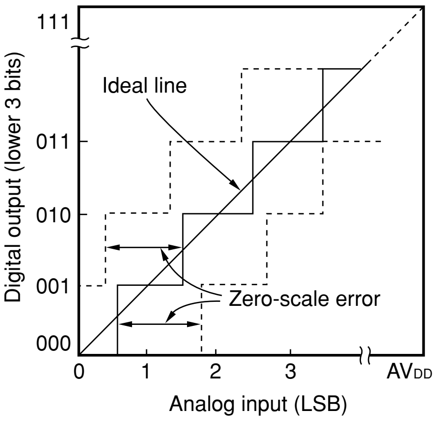


Figure 13-12. Full-Scale Error

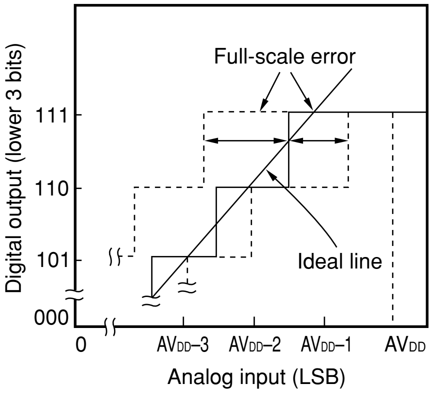


Figure 13-13. Integral Linearity Error

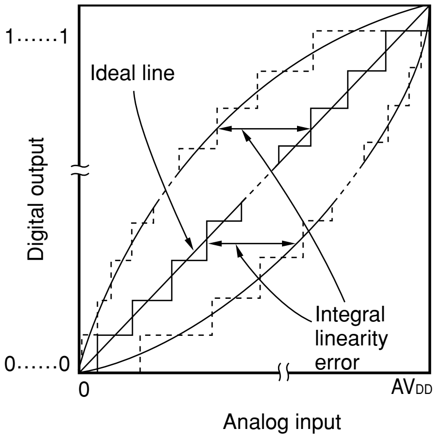
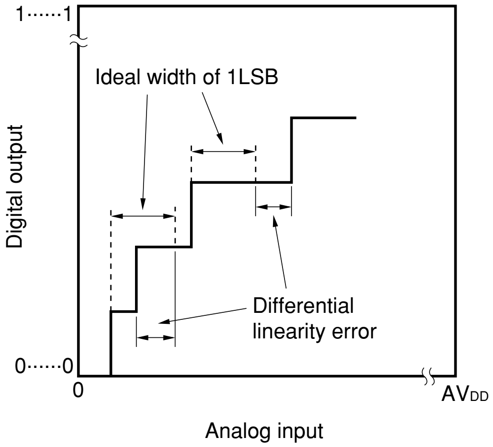


Figure 13-14. Differential Linearity Error

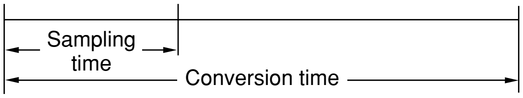


(8) Conversion time

★ This expresses the time from when the sampling is started to the time when the digital output was obtained. Sampling time is included in the conversion time in the characteristics table.

(9) Sampling time

This is the time the analog switch is turned on for the analog voltage to be sampled by the sample and hold circuit.



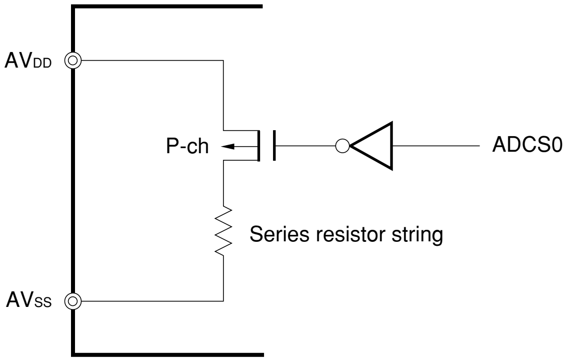
13.6 A/D Converter Cautions

(1) Power consumption in standby mode

The A/D converter stops operating in the standby mode. At this time, the power consumption can be reduced by setting bit 7 (ADCS0) of A/D converter mode register 0 (ADM0) to 0.

Figure 13-15 shows the circuit configuration of the series resistor string.

★ **Figure 13-15. Example of Series Resistor String Circuit Configuration**



(2) Input range of ANI0 to ANI7

The input voltages of ANI0 to ANI7 should be within the specification range. In particular, if a voltage higher ★ than or equal to AV_{DD} or lower than or equal to AV_{SS} is input (even if within the absolute maximum rating range), the conversion value of that channel will be undefined and the conversion values of other channels may also be affected.

(3) Conflicting operations

<1> Conflict between A/D conversion result register 0 (ADCR0) write and ADCR0 read by instruction upon the end of conversion

ADCR0 read is given priority. After the read operation, the new conversion result is written to ADCR0.

<2> Conflict between ADCR0 write and external trigger signal input upon the end of conversion

The external trigger signal is not accepted during A/D conversion. Therefore, the external trigger signal is not accepted during ADCR0 write.

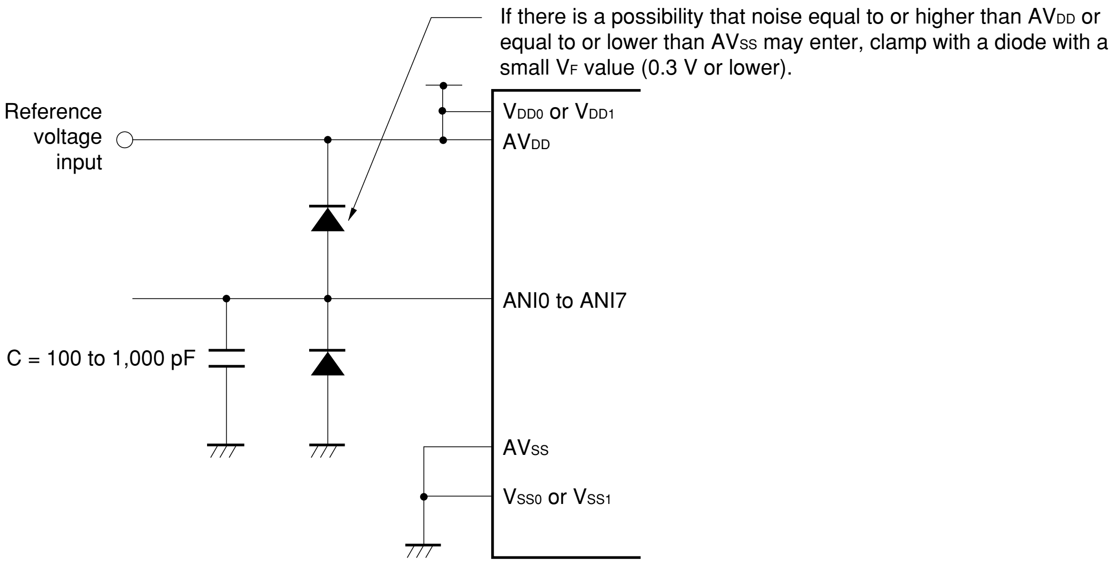
<3> Conflict between ADCR0 write and A/D converter mode register 0 (ADM0) write or analog input channel specification register 0 (ADS0) write upon the end of conversion

ADM0 or ADS0 write is given priority. ADCR0 write is not performed, nor is the conversion end interrupt request signal (INTAD0) generated.

(4) Noise countermeasures

★ To maintain the 10-bit resolution, attention must be paid to noise input to the AV$_{DD}$ and ANI0 to ANI7 pins. Because the effect increases in proportion to the output impedance of the analog input source, it is recommended that a capacitor be connected externally as shown in Figure 13-16 to reduce noise.

★ **Figure 13-16. Analog Input Pin Connection**



If there is a possibility that noise equal to or higher than AV$_{DD}$ or equal to or lower than AV$_{SS}$ may enter, clamp with a diode with a small V$_F$ value (0.3 V or lower).

(5) ANI0/P10 to ANI7/P17

The analog input pins (ANI0 to ANI7) also function as port pins.

When A/D conversion is performed with any of the ANI0 to ANI7 pins selected, do not execute an input instruction to port 1 while conversion is in progress, as this may reduce the conversion resolution.

Also, if digital pulses are applied to other analog input pins during A/D conversion, the expected A/D conversion value may not be obtainable due to coupling noise. Therefore, avoid applying pulses to other analog input pins during A/D conversion.

★ (6) Input impedance of ANI0 to ANI7 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 kΩ or lower, or attach a capacitor of around 100 pF to the ANI0 to ANI7 pins (see **Figure 13-16**).

★ (7) AV$_{DD}$ pin input impedance

A series resistor string is connected between the AV$_{DD}$ pin and the AV$_{SS}$ pin.

Therefore, when the output impedance of the reference voltage is too high, it seems as if the AV$_{DD}$ pin and the series resistor string are connected in series. This may cause a greater reference voltage error.

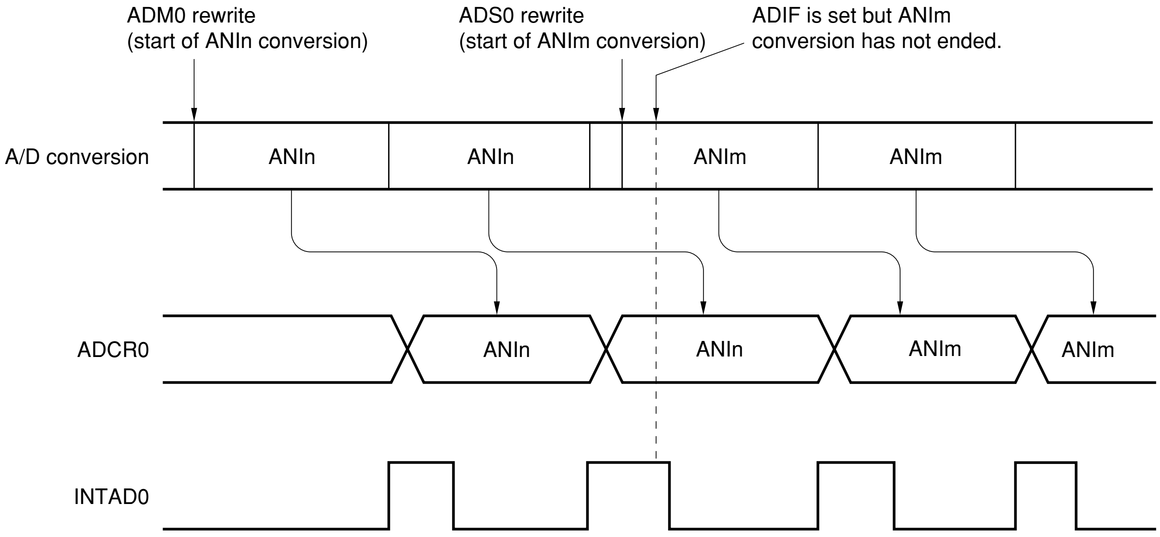
(8) Interrupt request flag (ADIF0)

The interrupt request flag (ADIF0) is not cleared even if analog input channel specification register 0 (ADS0) is changed.

Therefore, if an analog input pin is changed during A/D conversion, the A/D conversion result and conversion end interrupt request flag for the pre-change analog input may be set just before the ADS0 rewrite. Caution is therefore required since, at this time, when ADIF0 is read immediately just after the ADS0 rewrite, ADIF0 is set despite the fact that the A/D conversion for the post-change analog input has not ended.

When A/D conversion is restarted after it is stopped, clear ADIF0 before restart.

Figure 13-17. A/D Conversion End Interrupt Request Generation Timing



Remarks 1. n = 0, 1, ..., 7

2. m = 0, 1, ..., 7

(9) Conversion results just after A/D conversion start

If bit 7 (ADCS0) of A/D converter mode register 0 (ADM0) is set to 1 without setting bit 0 (ADCE0) to 1, the first value converted immediately after A/D conversion has been started may not satisfy the rated value. Discard the first conversion result and use the second and subsequent results by polling the A/D conversion end interrupt request (INTAD0).

The same may apply if ADCS0 is set to 1 without the lapse of the wait time of 14 μs (MIN.) after ADCE0 has been set to 1. Make sure that the specified wait time elapses.

(10) A/D conversion result register 0 (ADCR0) read operation

When writing is performed to A/D converter mode register 0 (ADM0) and analog input channel specification register 0 (ADS0), the contents of ADCR0 may become undefined. Read the conversion result following conversion completion before writing to ADM0, ADS0. Using a timing other than the above may cause an incorrect conversion result to be read.

(11) Timing at which A/D conversion result is undefined

The A/D conversion value may be undefined if the timing of completion of A/D conversion and the timing of stopping the A/D conversion conflict with each other. Therefore, read the A/D conversion result during the A/D conversion operation. To read the conversion result after stopping the A/D conversion operation, be sure to stop the A/D conversion before the next conversion ends.

Figures 13-18 and 13-19 show the timing of reading the conversion result.

Figure 13-18. Timing of Reading Conversion Result (When Conversion Result Is Undefined)

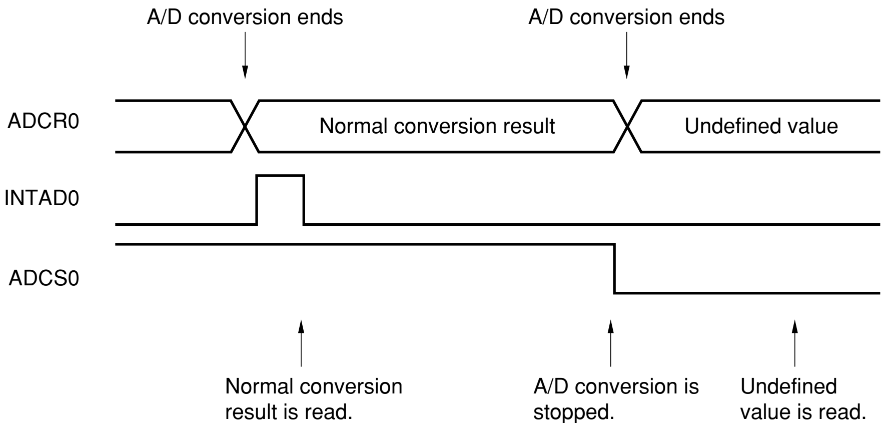
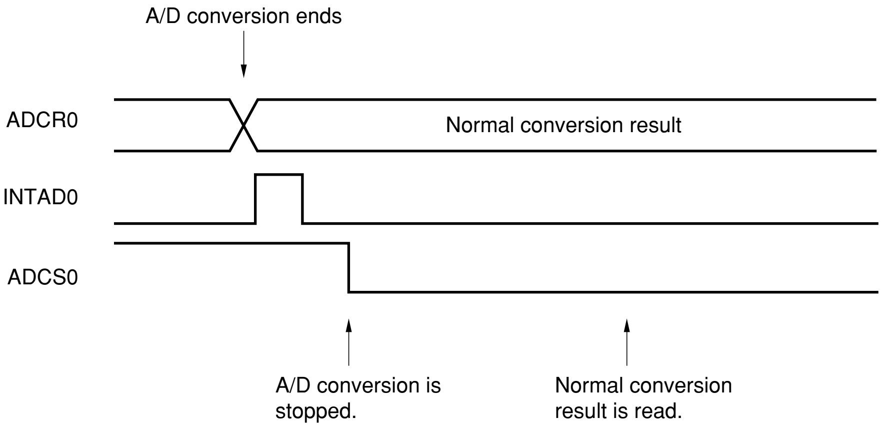


Figure 13-19. Timing of Reading Conversion Result (When Conversion Result Is Normal)



(12) Notes on board design

Locate analog circuits as far away from digital circuits as possible on the board because the analog circuits may be affected by the noise of the digital circuits. In particular, do not cross an analog signal line with a digital signal line, or wire an analog signal line in the vicinity of a digital signal line. Otherwise, the A/D conversion characteristics may be affected by the noise of the digital line.

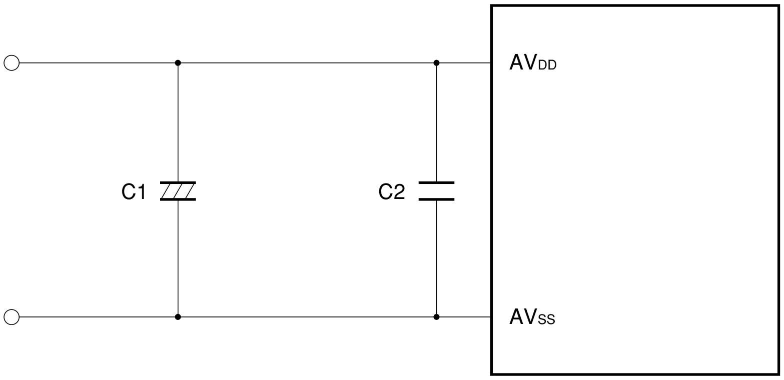
Connect AV_{SS} and V_{SS1} at one location on the board where the voltages are stable.

★ **(13) AV_{DD} pin and AV_{SS} pins**

Connect a capacitor to the AV_{DD} and AV_{SS} pins to minimize conversion errors due to noise. If an A/D conversion operation has been stopped and then started, the voltage applied to the AV_{DD} and AV_{SS} pins becomes unstable, causing the accuracy of the A/D conversion to drop. To prevent this, also connect a capacitor to the AV_{DD} and AV_{SS} pins.

Figure 13-20 shows an example of connecting capacitors.

★ **Figure 13-20. Example of Connecting Capacitor to AV_{DD} Pin**

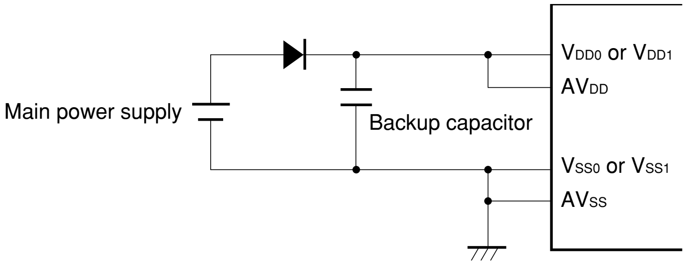


Remark C1: 4.7 μF to 10 μF (reference value)
C2: 0.01 μF to 0.1 μF (reference value)
Connect C2 as close to the pin as possible.

★ **(14) AV_{DD} pin**

The AV_{DD} pin supplies power to the analog circuit. It also supplies power to the input circuit of ANI0 to ANI7. Therefore, apply the same potential as that of the V_{DD0} or V_{DD1} pin to this pin, as shown in Figure 13-21, in an application where a backup power supply is used.

Figure 13-21. Processing of AV_{DD} Pin



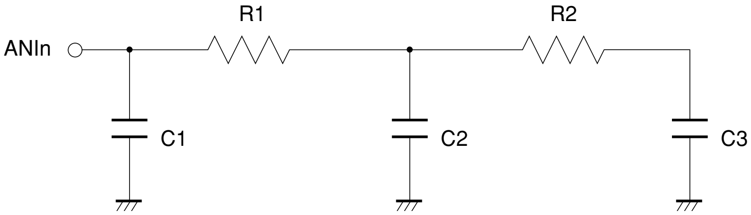
(15) Internal equivalent circuit of ANI0 to ANI7 pins and permissible signal source impedance

To complete sampling within the sampling time with sufficient A/D conversion accuracy, the impedance of the signal source such as a sensor must be sufficiently low. Figure 13-22 shows the internal equivalent circuit of the ANI0 to ANI7 pins.

If the impedance of the signal source is high, connect capacitors with a high capacitance to the ANI0 to ANI7 pins. An example of this is shown in Figure 13-23. In this case, however, the microcontroller cannot follow an analog signal with a high differential coefficient because a lowpass filter is created.

To convert a high-speed analog signal or to convert an analog signal in scan mode, insert a low-impedance buffer.

Figure 13-22. Internal Equivalent Circuit of ANI0 to ANI7 Pins



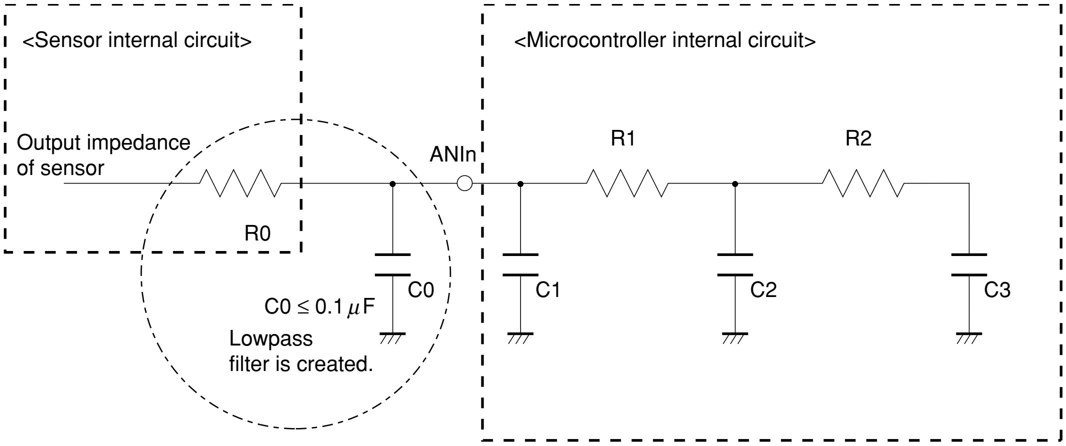
Remark n = 0 to 7

★ **Table 13-3. Resistances and Capacitances of Equivalent Circuit (Reference Values)**

AV_{DD}	R1	R2	C1	C2	C3
2.7 V	12 kΩ	8 kΩ	8 pF	3 pF	2 pF
4.5 V	4 kΩ	2.7 kΩ	8 pF	1.4 pF	2 pF

Caution The resistances and capacitances in Table 13-3 are not guaranteed values.

Figure 13-23. Example of Connection if Signal Source Impedance Is High


<Sensor internal circuit>
<Microcontroller internal circuit>
Output impedance of sensor
R0
ANIn
R1
R2
C0
C0 ≤ 0.1 μF
Lowpass filter is created.
C1
C2
C3

Remark n = 0 to 7

14.1 Functions of Serial Interface SIO3

Serial interface SIO3 has the following two modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed. For details, see **14.4.1 Operation stop mode**.

(2) 3-wire serial I/O mode (fixed as MSB first)

This is an 8-bit data transfer mode using three lines: a serial clock line ($\overline{\text{SCK3}}$), serial output line (SO3), and serial input line (SI3).

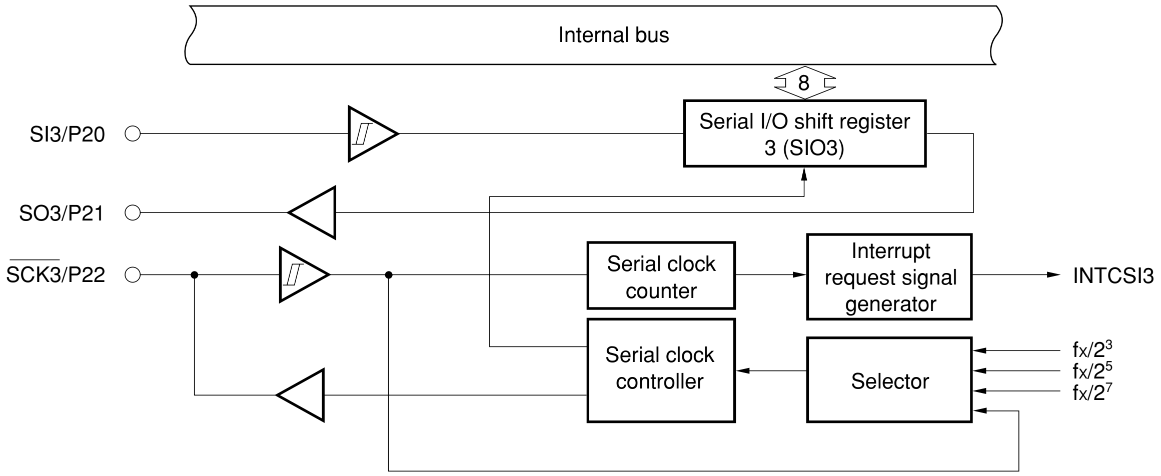
Since simultaneous transmit and receive operations are enabled in 3-wire serial I/O mode, the processing time for data transfers is reduced.

The first bit of the serially transferred 8-bit data is fixed as the MSB.

3-wire serial I/O mode is useful for connection to a peripheral IC incorporating a clocked serial interface, or a display controller, etc. For details, see **14.4.2 3-wire serial I/O mode**.

Figure 14-1 shows a block diagram of the serial interface SIO3.

Figure 14-1. Block Diagram of Serial Interface SIO3



14.2 Configuration of Serial Interface SIO3

Serial interface SIO3 consists of the following hardware.

Table 14-1. Configuration of Serial Interface SIO3

Item	Configuration
Register	Serial I/O shift register 3 (SIO3)
Control register	Serial operation mode register 3 (CSIM3)

(1) Serial I/O shift register 3 (SIO3)

This is an 8-bit register that performs parallel-serial conversion and serial transmit/receive (shift operations) synchronized with the serial clock.

When “1” is set to bit 7 (CSIE3) of serial operation mode register 3 (CSIM3), a serial operation can be started by writing data to or reading data from SIO3.

When transmitting, data written to SIO3 is output to the serial output (SO3).

When receiving, data is read from the serial input (SI3) and written to SIO3.

SIO3 is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes the value of this register undefined.

Caution Do not access SIO3 during a transmit operation unless the access is triggered by a transfer start. (Read operations are disabled when MODE = 0 and write operations are disabled when MODE = 1.)

14.3 Register to Control Serial Interface SIO3

The serial interface SIO3 is controlled by serial operation mode register 3 (CSIM3).

(1) Serial operation mode register 3 (CSIM3)

This register is used to enable or disable SIO3's serial clock, operation modes, and specific operations.
CSIM3 is set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets the value of this register to 00H.

Caution In 3-wire serial I/O mode, set the port mode register (PMXX) as follows. Set the output latch of a port set to output mode (PMXX = 0) to 0.

During serial clock output (master transmission or master reception)	PM22 = 0; Sets P22 ($\overline{\text{SCK3}}$) to output mode P22 = 0; Sets output latch of P22 to 0
During serial clock input (slave transmission or slave reception)	PM22 = 1; Sets P22 ($\overline{\text{SCK3}}$) to input mode

★ Transmit/receive mode	PM21 = 0; Sets P21 (SO3) to output mode
	P21 = 0; Sets output latch of P21 to 0
	PM20 = 1; Sets P20 (SI3) to input mode

Receive mode	PM20 = 1; Sets P20 (SI3) to input mode

Figure 14-2. Format of Serial Operation Mode Register 3 (CSIM3)

Address: FFAFH After reset: 00H R/W

Symbol	$\boxed{7}$	6	5	4	3	2	1	0
CSIM3	CSIE3	0	0	0	0	MODE	SCL31	SCL30

CSIE3	Enable/disable specification for SIO3		
	Shift register operation	Serial counter	Port
0	Operation stop	Clear	Port functionNote 1
1	Operation enable	Count operation enable	Serial function + port functionNote 2

★

MODE	Transfer operation modes and flags		
	Operation mode	Transfer start trigger	SO3/P21 pin function
0	Transmit/transmit and receive mode	Write to SIO3	SO3
1	Receive-only mode	Read from SIO3	P21

SCL31	SCL30	Clock selection
0	0	External clock input to $\overline{SCK3}$
0	1	$f_X/2^3$ (1.25 MHz)
1	0	$f_X/2^5$ (312.5 kHz)
1	1	$f_X/2^7$ (78.125 kHz)

Notes 1. When CSIE3 = 0 (SIO3 operation stop status), the SI3, SO3, and $\overline{SCK3}$ pins can be used for port functions.

2. When CSIE3 = 1 (SIO3 operation enabled status), the SI3 pin can be used as a port pin if only the transmit function is used, and the SO3 pin can be used as a port pin if only the receive-only mode is used.

Remarks 1. f_X: Main system clock oscillation frequency

2. Figures in parentheses are for operation with f_X = 10 MHz.

14.4 Operations of Serial Interface SIO3

This section explains the two modes of serial interface SIO3.

14.4.1 Operation stop mode

Because serial transfer is not performed in this mode, the power consumption can be reduced.

In addition, pins can be used as normal I/O ports.

(1) Register settings

Operation stop mode is set by serial operation mode register 3 (CSIM3).

CSIM3 is set by a 1-bit or 8-bit memory manipulation instruction.

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

Address: FFAFH After reset: 00H R/W

Symbol	$\boxed{7}$	6	5	4	3	2	1	0
CSIM3	CSIE3	0	0	0	0	MODE	SCL31	SCL30

CSIE3	SIO3 operation enable/disable specification		
	Shift register operation	Serial counter	Port
0	Operation stop	Clear	Port function[Note 1]
1	Operation enable	Count operation enable	Serial function + port function[Note 2]

Notes 1. When CSIE3 = 0 (SIO3 operation stop status), the SI3, SO3, and $\overline{\text{SCK3}}$ pins can be used for port functions.

2. When CSIE3 = 1 (SIO3 operation enabled status), the SI3 pin can be used as a port pin if only the transmit function is used, and the SO3 pin can be used as a port pin if only the receive-only mode is used.

14.4.2 3-wire serial I/O mode

The 3-wire serial I/O mode can be used for connection to a peripheral IC incorporating a clocked serial interface, a display controller, etc.

This mode executes data transfers via three lines: a serial clock line ($\overline{\text{SCK3}}$), serial output line (SO3), and serial input line (SI3).

(1) Register settings

3-wire serial I/O mode is set by serial operation mode register 3 (CSIM3).

CSIM3 is set by a 1-bit or 8-bit memory manipulation instruction.

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

Caution In 3-wire serial I/O mode, set the port mode register (PMXX) as follows. Set the output latch of a port set to output mode (PMXX = 0) to 0.

During serial clock output (master transmission or master reception)	PM22 = 0; Sets P22 ($\overline{\text{SCK3}}$) to output mode P22 = 0; Sets output latch of P22 to 0
During serial clock input (slave transmission or slave reception)	PM22 = 1; Sets P22 ($\overline{\text{SCK3}}$) to input mode
Transmit/receive mode	PM21 = 0; Sets P21 (SO3) to output mode P21 = 0; Sets output latch of P21 to 0 ★ PM20 = 1; Sets P20 (SI3) to input mode
Receive mode	PM20 = 1; Sets P20 (SI3) to input mode

Address: FFAFH After reset: 00H R/W

Symbol	$\boxed{7}$	6	5	4	3	2	1	0
CSIM3	CSIE3	0	0	0	0	MODE	SCL31	SCL30

CSIE3	Enable/disable specification for SIO3		
	Shift register operation	Serial counter	Port
0	Operation stop	Clear	Port function[Note 1]
1	Operation enable	Count operation enable	Serial function + port function[Note 2]

MODE	Transfer operation modes and flags		
	Operation mode	Transfer start trigger	SO3/P21 pin function
0	Transmit/transmit and receive mode	Write to SIO3	SO3
1	Receive-only mode	Read from SIO3	P21

SCL31	SCL30	Clock selection
0	0	External clock input to $\overline{\text{SCK3}}$
0	1	$f_X/2^3$ (1.25 MHz)
1	0	$f_X/2^5$ (312.5 kHz)
1	1	$f_X/2^7$ (78.125 kHz)

Notes 1. When CSIE3 = 0 (SIO3 operation stop status), the SI3, SO3, and $\overline{\text{SCK3}}$ pins can be used for port functions.

2. When CSIE3 = 1 (SIO3 operation enabled status), the SI3 pin can be used as a port pin if only the transmit function is used, and the SO3 pin can be used as a port pin if only the receive-only mode is used.

Remarks 1. f_X: Main system clock oscillation frequency

2. Figures in parentheses are for operation with f_X = 10 MHz.

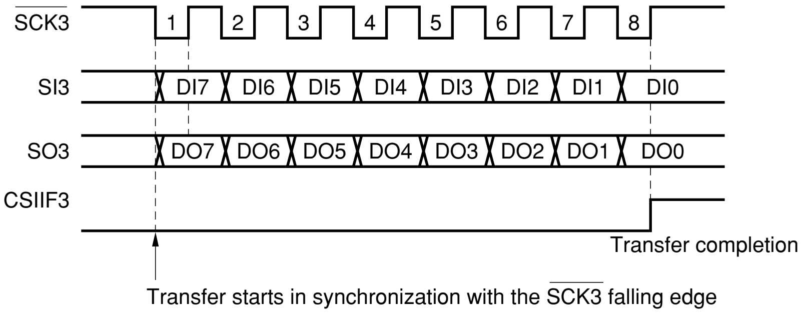
(2) Communication operations

In the 3-wire serial I/O mode, data is transmitted and received in 8-bit units. Each bit of data is transmitted or received in synchronization with the serial clock.

Serial I/O shift register 3 (SIO3) is shifted in synchronization with the falling edge of the serial clock. Transmission data is held in the SO3 latch and is output from the SO3 pin. Data that is received via the SI3 pin in synchronization with the rising edge of the serial clock is latched to SIO3.

Completion of an 8-bit transfer automatically stops operation of SIO3 and sets the interrupt request flag (CSIIF3).

Figure 14-3. Timing of 3-Wire Serial I/O Mode



(3) Transfer start

A serial transfer starts when the following two conditions have been satisfied and transfer data has been set (or read) to serial I/O shift register 3 (SIO3).

- SIO3 operation control bit (CSIE3) = 1
- After an 8-bit serial transfer, either the internal serial clock is stopped or $\overline{SCK3}$ is set to high level.
- Transmit/transmit and receive mode
  When CSIE3 = 1 and MODE = 0, transfer starts when writing to SIO3.
- Receive-only mode
  When CSIE3 = 1 and MODE = 1, transfer starts when reading from SIO3.

Caution After data has been written to SIO3, transfer will not start even if the CSIE3 bit value is set to “1”.

Completion of an 8-bit transfer automatically stops the serial transfer operation and interrupt request flag (CSIIF3) is set.

CHAPTER 15 SERIAL INTERFACE CSI1

15.1 Functions of Serial Interface CSI1

Serial interface CSI1 has the following two modes.

- Operation stop mode
- 3-wire serial I/O mode

(1) Operation stop mode

This mode is used when serial transfer is not performed. In this mode, the power consumption can be reduced.

(2) 3-wire serial I/O mode (MSB/LSB first selectable)

This mode is used to transfer 8-bit data by using three lines: a serial clock line (SCK1) and two serial data lines (SI1 and SO1).

The processing time of data transfer can be shortened in the 3-wire serial I/O mode because transmission and reception can be simultaneously executed in this mode. In addition, whether 8-bit data is transferred with the MSB or LSB first can be specified, so this interface can be connected to any device.

The 3-wire serial I/O mode can be used for connecting peripheral ICs and display controllers having a conventional clocked serial interface.

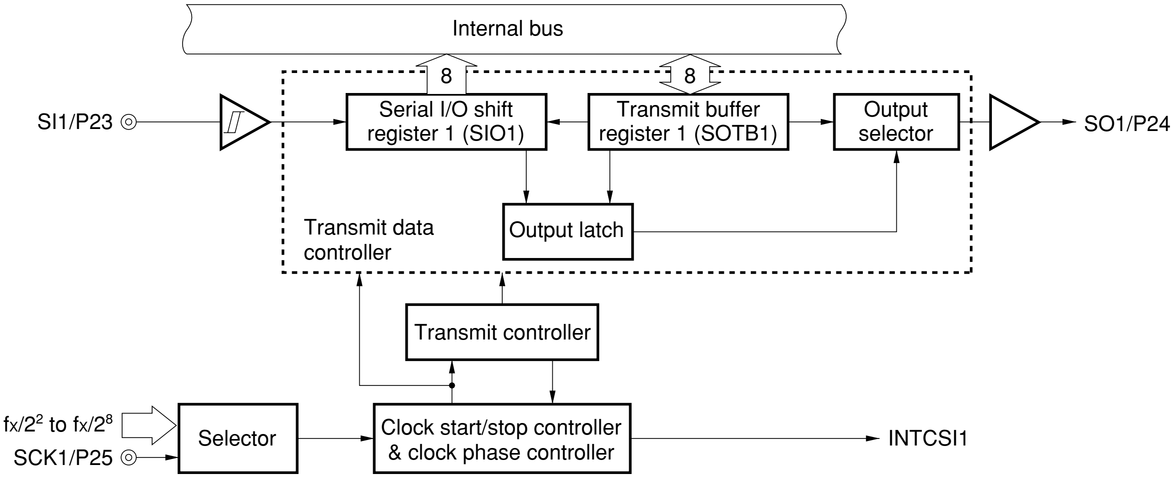
15.2 Configuration of Serial Interface CSI1

Serial interface CSI1 consists of the following hardware.

Table 15-1. Configuration of Serial Interface CSI1

Item	Configuration
Registers	Transmit buffer register 1 (SOTB1) Serial I/O shift register 1 (SIO1)
Control registers	Serial operation mode register 1 (CSIM1) Serial clock select register 1 (CSIC1)

Figure 15-1. Block Diagram of Serial Interface CSI1



(1) Transmit buffer register 1 (SOTB1)

This register sets transmit data.

Transmission/reception is started by writing data to SOTB1 when bit 6 (TRMD1) of serial operation mode register 1 (CSIM1) is 1.

The data written to SOTB1 is converted from parallel data into serial data by serial I/O shift register 1, and output to the serial output (SO1) pin.

SOTB1 can be written or read by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes the value of this register undefined.

Caution Do not access SOTB1 when CSOT1 = 1 (during serial communication).

(2) Serial I/O shift register 1 (SIO1)

This is an 8-bit register that converts data from parallel into serial or vice versa.

Reception is started by reading data from SIO1 if bit 6 (TRMD1) of serial operation mode register 1 (CSIM1) is 0.

During reception, the data is read from the serial input pin (SI1) to SIO1.

This register can be read by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes the value of this register undefined.

Caution Do not access SIO1 when CSOT1 = 1 (during serial communication).

15.3 Registers to Control Serial Interface CSI1

Serial interface CSI1 is controlled by the following two registers.

- Serial operation mode register 1 (CSIM1)
- Serial clock select register 1 (CSIC1)

(1) Serial operation mode register 1 (CSIM1)

This register is used to select an operation mode and enable or disable the operation.

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

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

Figure 15-2. Format of Serial Operation Mode Register 1 (CSIM1)

Address: FFB0H After reset: 00H R/W[Note 1]

Symbol	$\boxed{7}$	6	5	4	3	2	1	0
CSIM1	CSIE1	TRMD1	0	DIR1	0	0	0	CSOT1

CSIE1	Operation control in 3-wire serial I/O mode
0	Stops operation (SI1/P23, SO1/P24, and SCK1/P25 pins can be used as general-purpose port pins).
1	Enables operation (SI1/P23, SO1/P24, and SCK1/P25 pins can be used in 3-wire serial I/O mode).

TRMD1[Note 2]	Transmit/receive mode selection
0[Note 3]	Receive mode (transmission disabled).
1	Transmit/receive mode

DIR1[Note 4]	First bit specification
0	MSB
1	LSB

CSOT1[Note 5]	Operation mode flag
0	Communication is stopped.
1	Communication is in progress.

Notes **1.** Bit 0 is a read-only bit.

2. Do not rewrite TRMD1 when CSOT1 = 1 (during serial communication).

★ **3.** The SO1 output is fixed to the low level when TRMD1 is 0. Reception is started when data is read from SIO1.

4. Do not overwrite these bits when CSOT1 = 1 (during serial communication).

5. CSOT1 is cleared if CSIE1 is cleared to 0 (operation stops).

Caution Be sure to set bit 5 to 0.

(2) Serial clock select register 1 (CSIC1)

This register is used to select the phase of the data clock and the transfer clock.

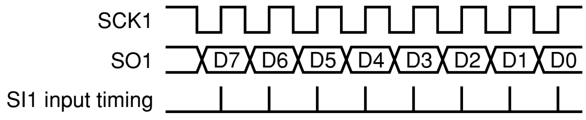
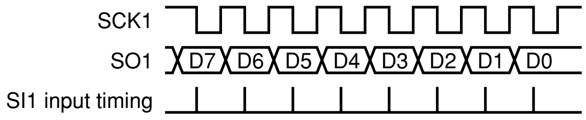
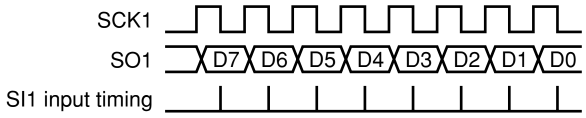
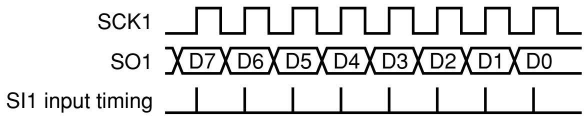
This register is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the value of this register to 10H.

Figure 15-3. Format of Serial Clock Select Register 1 (CSIC1)

Address: FFB1H After reset: 10H R/W

Symbol	7	6	5	4	3	2	1	0
CSIC1	0	0	0	CKP1	DAP1	CKS12	CKS11	CKS10

CKP1	DAP1	Data clock phase selection	Type
0	0		1
0	1		2
1	0		3
1	1		4

CKS12	CKS11	CKS10	Transfer clock CSI1 selection
0	0	0	fx/2^2 (2.5 MHz)
0	0	1	fx/2^3 (1.25 MHz)
0	1	0	fx/2^4 (625 kHz)
0	1	1	fx/2^5 (312.5 kHz)
1	0	0	fx/2^6 (156.25 kHz)
1	0	1	fx/2^7 (78.125 kHz)
1	1	0	fx/2^8 (39.0625 kHz)
1	1	1	External clock

★ **Cautions 1. When CSIE1 = 1 (operation enable) or when the P23/SI1, P24/SO1, and P25/SCK1 pins are used as general-purpose ports, do not write to CSIC1.**

2. The phase type of the data clock is type 3 after reset.

Remark Figures in parentheses are for operation with fx = 10 MHz

15.4 Operations of Serial Interface CSI1

Serial interface CSI1 can be used in the following two modes.

- Operation stop mode
- 3-wire serial I/O mode

15.4.1 Operation stop mode

Serial transfer is not executed in this mode. Therefore, the power consumption can be reduced. In addition, the P23/SI1, P24/SO1, and P25/SCK1 pins can be used as normal I/O port pins in this mode.

(1) Register setting

The operation stop mode is set by serial operation mode register 1 (CSIM1).

(a) Serial operation mode register 1 (CSIM1)

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

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

Address: FFB0H After reset: 00H R/W

Symbol	$\boxed{7}$	6	5	4	3	2	1	0
CSIM1	CSIE1	TRMD1	0	DIR1	0	0	0	CSOT1

CSIE1	Operation control in 3-wire serial I/O mode
0	Stops operation (SI1/P23, SO1/P24, and SCK1/P25 pins can be used as general-purpose port pins).
★ 1	Enables operation (SI1/P23, SO1/P24, and SCK1/P25 pins can be used in 3-wire serial I/O mode).

15.4.2 3-wire serial I/O mode

The 3-wire serial I/O mode can be used for connecting peripheral ICs and display controllers having a conventional clocked serial interface.

In this mode, communication is executed by using three lines: serial clock (SCK1), serial output (SO1), and serial input (SI1) lines.

(1) Register setting

The 3-wire serial I/O mode is set by using serial operation mode register 1 (CSIM1) and serial clock select register 1 (CSIC1).

(a) Serial operation mode register 1 (CSIM1)

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

RESET input sets the value of this register to 00H.

Address: FFB0H After reset: 00H R/W[Note 1]

Symbol	7	6	5	4	3	2	1	0
CSIM1	CSIE1	TRMD1	0	DIR1	0	0	0	CSOT1

CSIE1	Operation control in 3-wire serial I/O mode
0	Stops operation (SI1/P23, SO1/P24, and SCK1/P25 pins can be used as general-purpose port pins).
1	★ Enables operation (SI1/P23, SO1/P24, and SCK1/P25 pins can be used in 3-wire serial I/O mode).

TRMD1[Note 2]	Transmit/receive mode selection
0[Note 3]	Receive mode (transmission disabled).
1	Transmit/receive mode

DIR1[Note 4]	First bit specification
0	MSB
1	LSB

CSOT1[Note 5]	Operation mode flag
0	Communication is stopped.
1	Communication is in progress.

Notes

1. Bit 0 is a read-only bit.

2. Do not rewrite TRMD1 when CSOT1 = 1 (during serial communication).

★ **3.** The SO1 output is fixed to low level when TRMD1 is 0. Reception is started when data is read from SIO1.

4. Do not overwrite these bits when CSOT1 = 1 (during serial communication).

5. CSOT1 is cleared if CSIE1 is cleared to 0 (operation stops).

Caution Be sure to set bit 5 to 0.

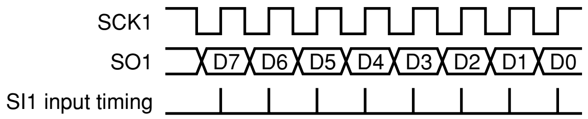
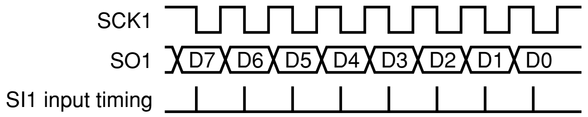
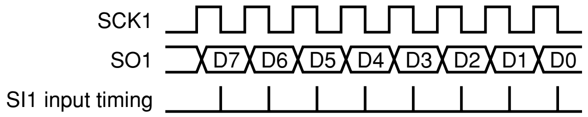
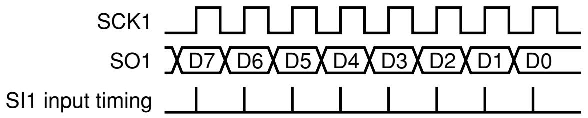
(b) Serial clock select register 1 (CSIC1)

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

$\overline{\text{RESET}}$ input sets the value of this register to 10H.

Address: FFB1H After reset: 10H R/W

Symbol	7	6	5	4	3	2	1	0
CSIC1	0	0	0	CKP1	DAP1	CKS12	CKS11	CKS10

CKP1	DAP1	Data clock phase selection	Type
0	0		1
0	1		2
1	0		3
1	1		4

CKS12	CKS11	CKS10	Transfer clock CSI1 selection
0	0	0	$f_X/2^2$ (2.5 MHz)
0	0	1	$f_X/2^3$ (1.25 MHz)
0	1	0	$f_X/2^4$ (625 kHz)
0	1	1	$f_X/2^5$ (312.5 kHz)
1	0	0	$f_X/2^6$ (156.25 kHz)
1	0	1	$f_X/2^7$ (78.125 kHz)
1	1	0	$f_X/2^8$ (39.0625 kHz)
1	1	1	External clock

★ **Cautions 1. When CSIE1 = 1 (operation enable) or when the P23/SI1, P24/SO1, and P25/SCK1 pins are used as general-purpose ports, do not write to CSIC1.**

2. The phase type of the data clock is type 3 after reset.

Remark Figures in parentheses are for operation with f_X = 10 MHz

(2) Setting of port

<1> Transmit/receive mode

(a) To use externally input clock as system clock (SCK1)

Bit 3 (PM23) of port mode register 2: Set to 1
Bit 4 (PM24) of port mode register 2: Cleared to 0
Bit 5 (PM25) of port mode register 2: Set to 1
Bit 4 (P24) of port 2: Cleared to 0

(b) To use internal clock as system clock (SCK1)

Bit 3 (PM23) of port mode register 2: Set to 1
Bit 4 (PM24) of port mode register 2: Cleared to 0
Bit 5 (PM25) of port mode register 2: Cleared to 0
Bit 4 (P24) of port 2: Cleared to 0
Bit 5 (P25) of port 2: Cleared to 0

<2> Receive mode (with transmission disabled)

(a) To use externally input clock as system clock (SCK1)

Bit 3 (PM23) of port mode register 2: Set to 1
Bit 5 (PM25) of port mode register 2: Set to 1

(b) To use internal clock as system clock (SCK1)

Bit 3 (PM23) of port mode register 2: Set to 1
Bit 5 (PM25) of port mode register 2: Cleared to 0
Bit 5 (P25) of port 2: Cleared to 0

Remark The transmit/receive mode or receive mode is selected by using bit 6 (TRMD1) of serial operation mode register 1 (CSIM1).

(3) Communication operation

In the 3-wire serial I/O mode, data is transmitted or received in 8-bit units. Each bit of the data is transmitted or received in synchronization with the serial clock.

Data can be transmitted or received if bit 6 (TRMD1) of serial operation mode register 1 (CSIM1) is 1. Transmission/reception is started when a value is written to transmit buffer register 1 (SOTB1). Data can be received if bit 6 (TRMD1) of serial operation mode register 1 (CSIM1) is 0. Reception is started when data is read from serial I/O shift register 1 (SIO1).

After communication has been started, bit 0 (CSOT1) of CSIM1 is set to 1. When communication of 8-bit data has been completed, a communication completion interrupt flag (CSIIF1) is set, and CSOT1 is cleared to 0. Then the next communication is enabled.

Caution Do not access the control register and data register when CSOT1 = 1 (during serial communication).

Figure 15-4. Timing in 3-Wire Serial I/O Mode (1/2)

(1) Transmission/reception timing (Type 1; TRMD1 = 1, DIR1 = 0, CKP1 = 0, DAP1 = 0)

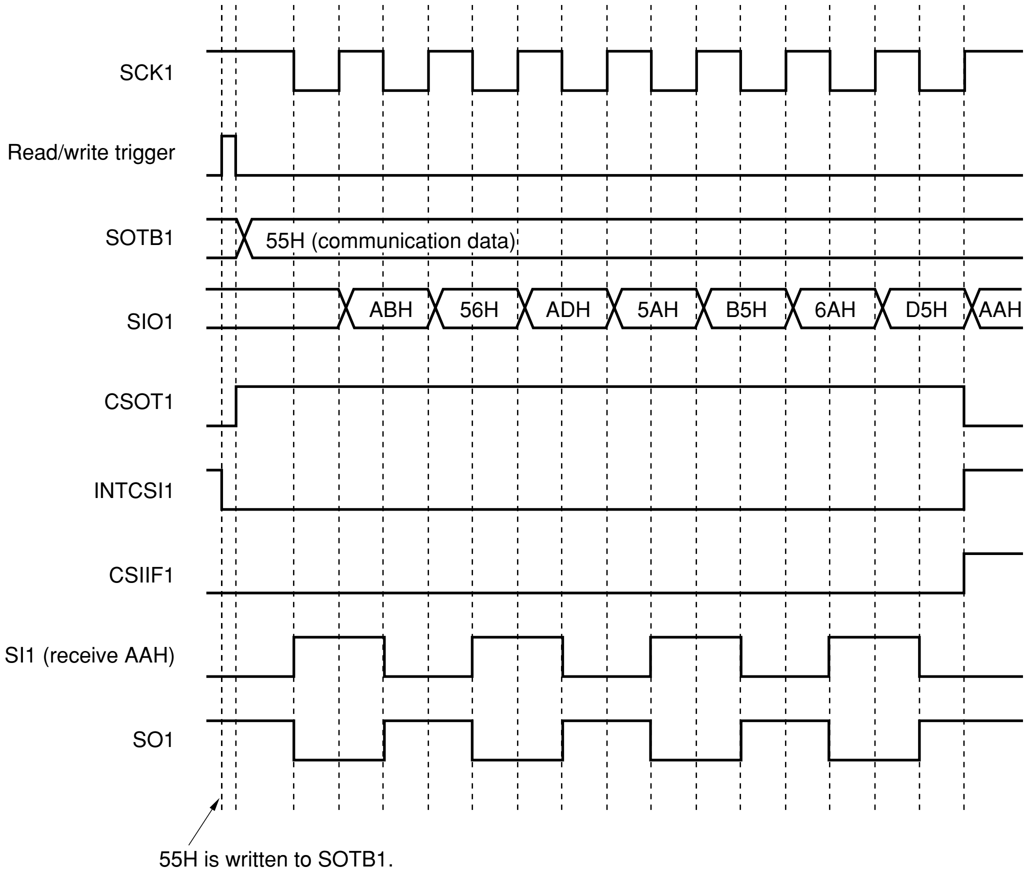


Figure 15-4. Timing in 3-Wire Serial I/O Mode (2/2)

(2) Transmission/reception timing (Type 2; TRMD1 = 1, DIR1 = 0, CKP1 = 0, DAP1 = 1)

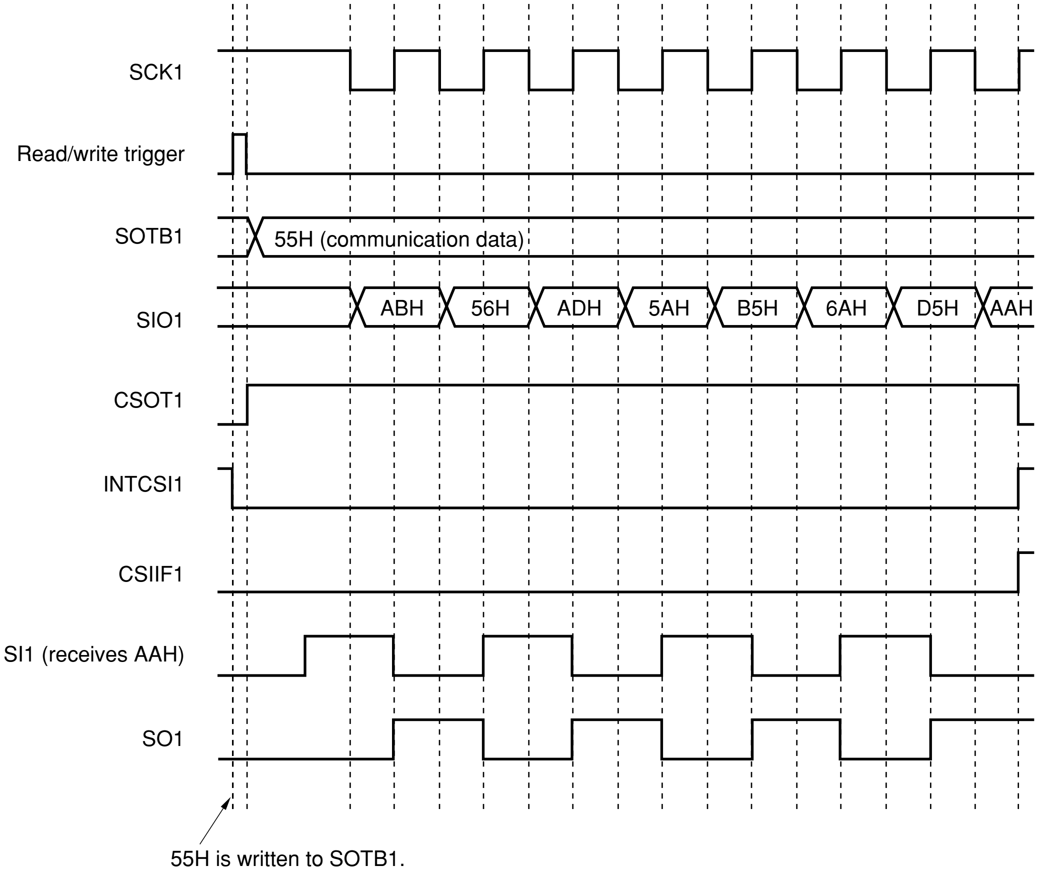
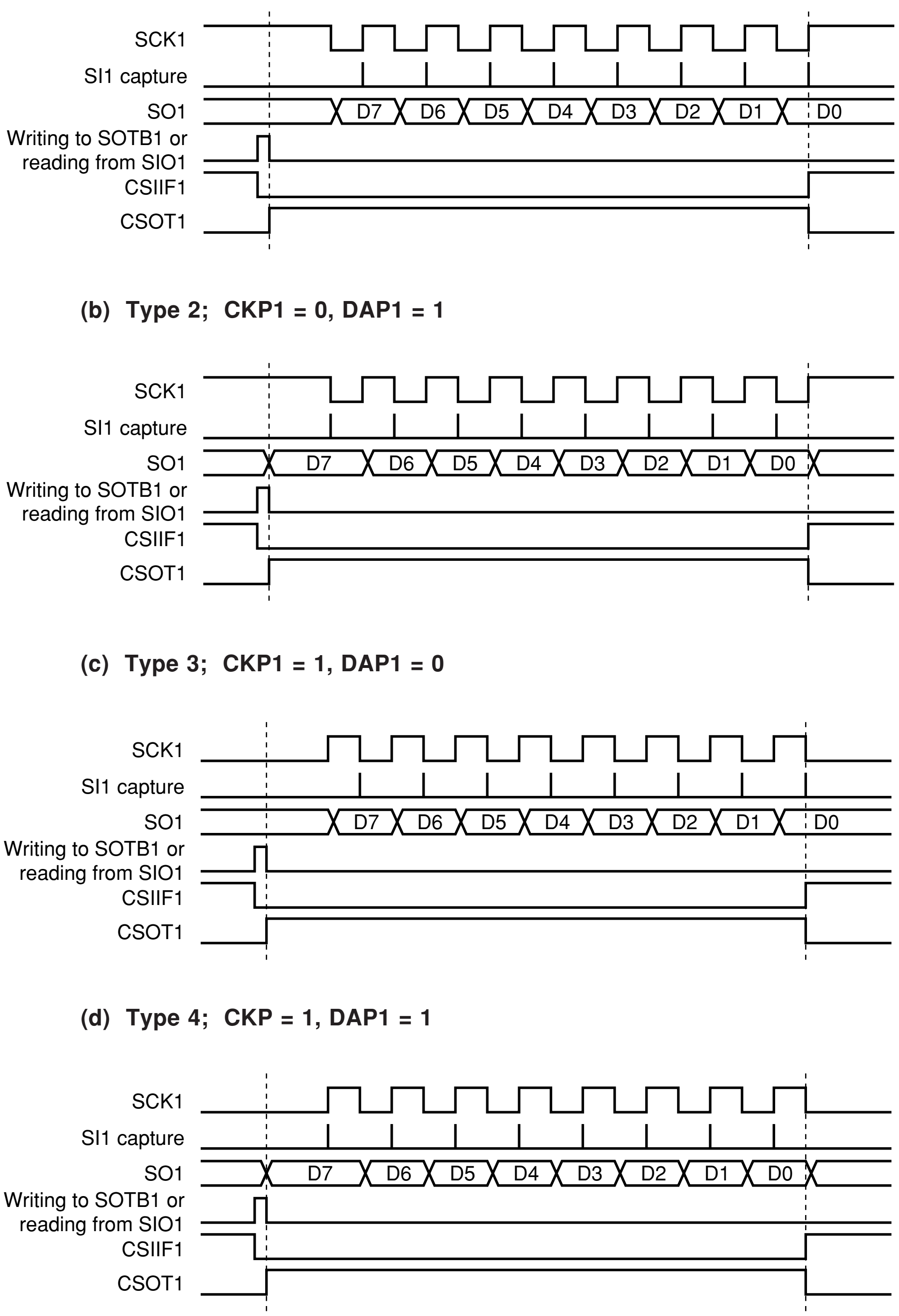


Figure 15-5. Timing of Clock/Data Phase



(a) Type 1; CKP1 = 0, DAP1 = 0

(b) Type 2; CKP1 = 0, DAP1 = 1

(c) Type 3; CKP1 = 1, DAP1 = 0

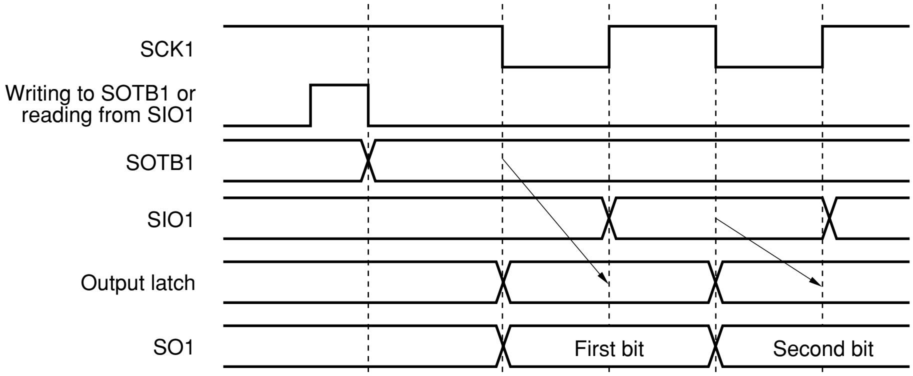
(d) Type 4; CKP = 1, DAP1 = 1

(4) Timing of output to SO1 pin (first bit)

When communication is started, the value of transmit buffer register 1 (SOTB1) is output from the SO1 pin. The output operation of the first bit at this time is explained below.

Figure 15-6. Output Operation of First Bit

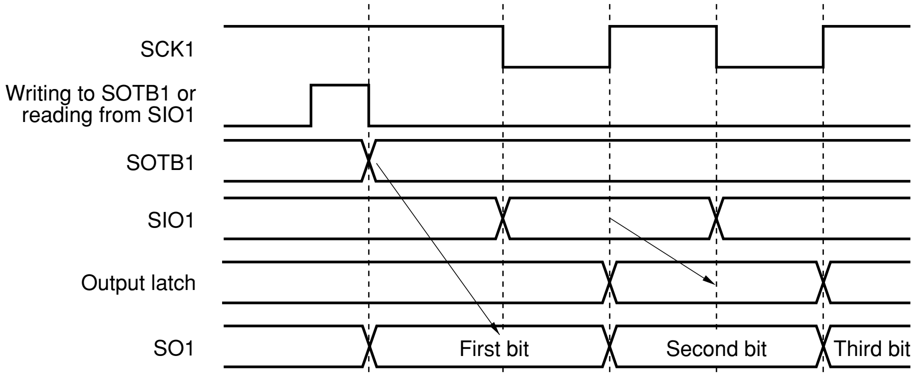
(1) CKP1 = 0, DAP1 = 0 (or CKP1 = 1, DAP1 = 0)



The first bit is directly latched to the output latch from the SOTB1 register at the falling (or rising) edge of SCK1, and is output from the SO1 pin via the output selector. At the next rising (or falling) edge of SCK1, the value of the SOTB1 register is transferred to the SIO1 register and shifted by 1 bit. At the same time, the first bit of the receive data is stored in the SIO1 register via the SI1 pin.

The second and subsequent bits are latched to the output latch from SIO1 at the next falling (or rising) edge of SCK1 and the data is output from the SO1 pin.

(2) CKP1 = 0, DAP1 = 1 (or CKP1 = 1, DAP1 = 1)



The first bit is directly output from the SOTB1 register to the SO1 pin via the output selector at the falling edge of the write signal of SOTB1 or the read signal of the SIO1 register. At the next falling (or rising) edge of SCK1, the value of the SOTB1 register is transferred to the SIO1 register and shifted by 1 bit. At the same time, the first bit of the received data is stored in the SIO1 register via the SI1 pin.

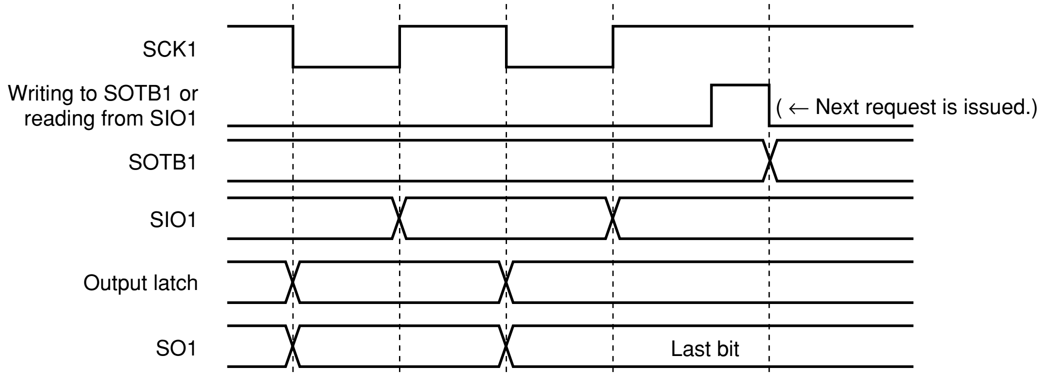
The second and subsequent bits are latched to the output latch from SIO1 at the next rising (or falling) edge of SCK1 and the data is output from the SO1 pin.

(5) Output value of SO1 pin (last bit)

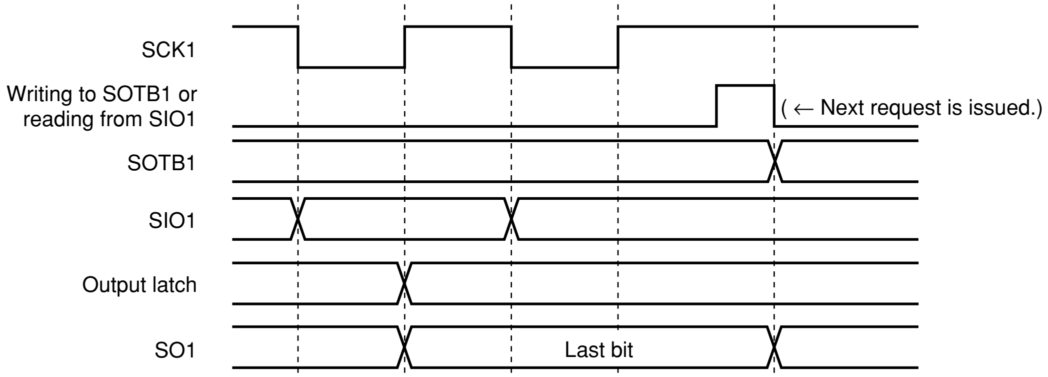
After communication has been completed, the SO1 pin holds the output value of the last bit.

Figure 15-7. Output Value of SO1 Pin (Last Bit)

(1) Type 1; CKP1 = 0 and DAP1 = 0 (or CKP1 = 1, DAP1 = 0)


SCK1
Writing to SOTB1 or reading from SIO1
(← Next request is issued.)
SOTB1
SIO1
Output latch
SO1
Last bit

(2) Type 2; CKP1 = 0 and DAP1 = 1 (or CKP1 = 1, DAP1 = 1)


SCK1
Writing to SOTB1 or reading from SIO1
(← Next request is issued.)
SOTB1
SIO1
Output latch
SO1
Last bit

CHAPTER 16 SERIAL INTERFACE UART0

16.1 Functions of Serial Interface UART0

Serial interface UART0 has the following two modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed to reduce power consumption.

For details, see **16.4.1 Operation stop mode**.

★ **(2) Asynchronous serial interface (UART) mode (fixed to LSB-first)**

This mode enables full-duplex operation wherein one byte of data after the start bit is transmitted and received.

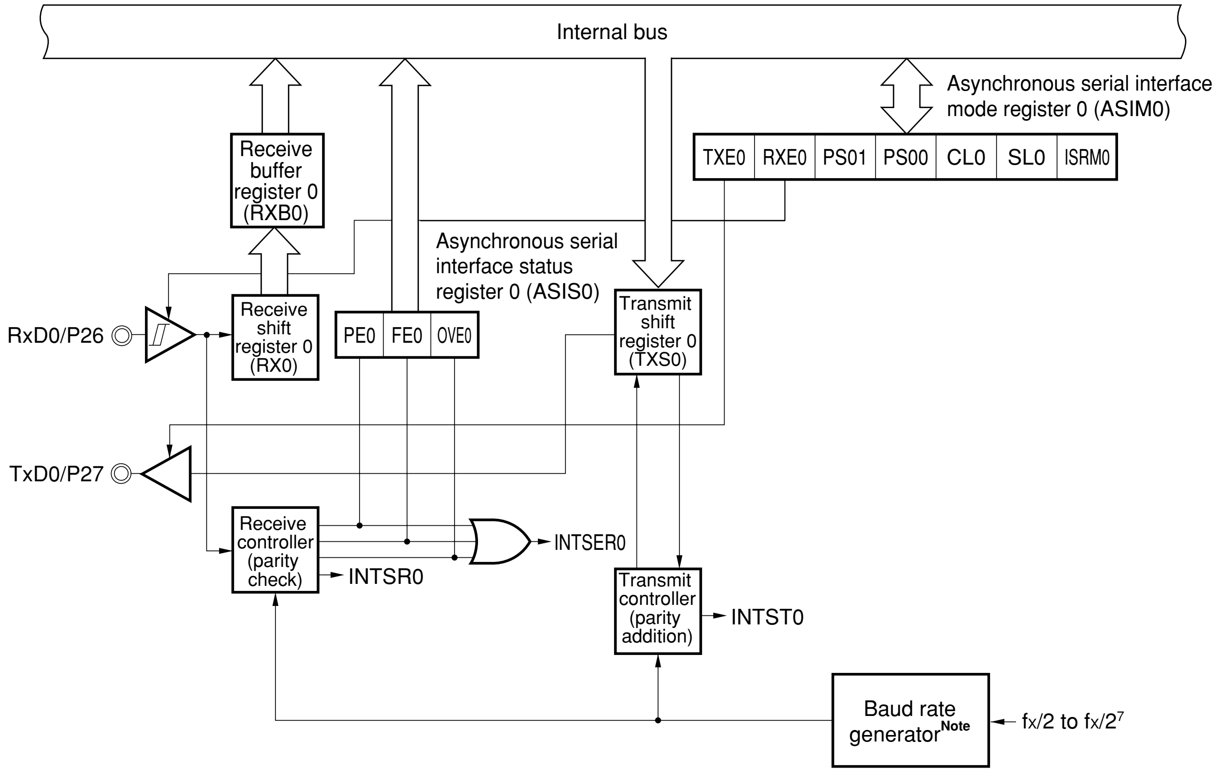
The on-chip baud rate generator dedicated to UART enables communications using a wide range of selectable baud rates.

The UART baud rate generator can also be used to generate a MIDI-standard baud rate (31.25 kbps).

For details, see **16.4.2 Asynchronous serial interface (UART) mode**.

Figure 16-1 shows a block diagram of serial interface UART0.

Figure 16-1. Block Diagram of Serial Interface UART0

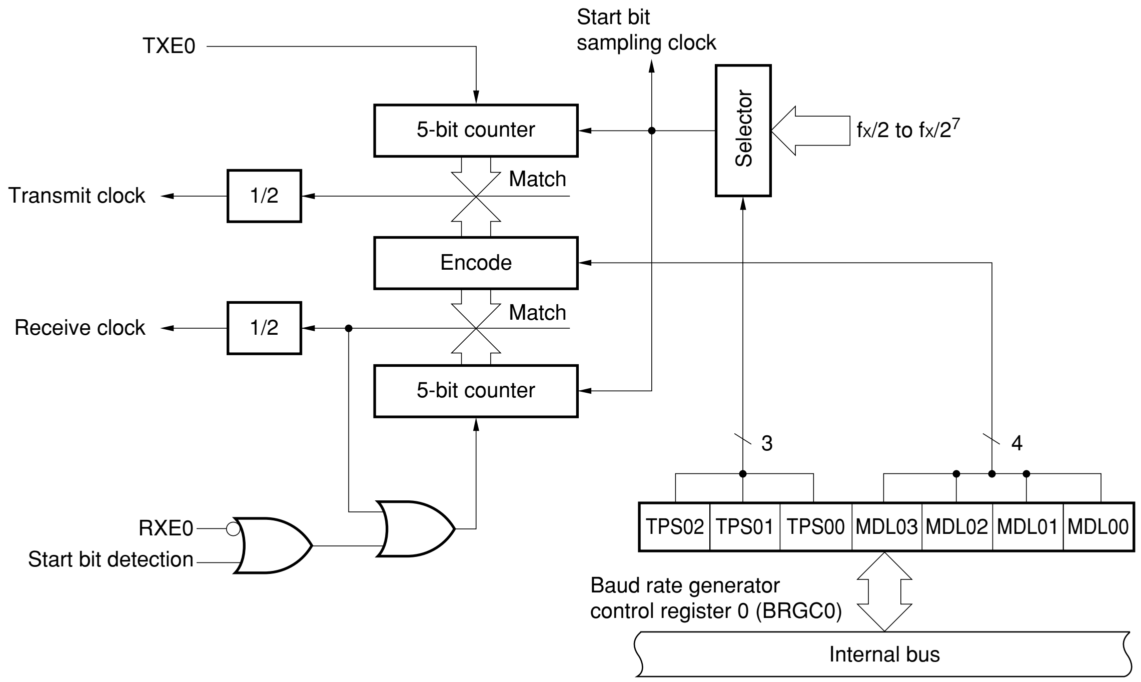


Internal bus
Asynchronous serial interface mode register 0 (ASIM0)
TXE0
RXE0
PS01
PS00
CL0
SL0
ISRM0
Receive buffer register 0 (RXB0)
Asynchronous serial interface status register 0 (ASIS0)
RxD0/P26
Receive shift register 0 (RX0)
PE0
FE0
OVE0
Transmit shift register 0 (TXS0)
TxD0/P27
Receive controller (parity check)
INTSER0
INTSR0
Transmit controller (parity addition)
INTST0
Baud rate generatorNote
fx/2 to fx/2^7

Note For the configuration of the baud rate generator, refer to **Figure 16-2**.

Figure 16-2. Baud Rate Generator Block Diagram

★



Remark TXE0: Bit 7 of asynchronous serial interface mode register 0 (ASIM0)
RXE0: Bit 6 of asynchronous serial interface mode register 0 (ASIM0)

16.2 Configuration of Serial Interface UART0

Serial interface UART0 consists of the following hardware.

Table 16-1. Configuration of Serial Interface (UART0)

Item	Configuration
Registers	Transmit shift register 0 (TXS0) Receive shift register 0 (RX0) Receive buffer register 0 (RXB0)
Control registers	Asynchronous serial interface mode register 0 (ASIM0) Asynchronous serial interface status register 0 (ASIS0) Baud rate generator control register 0 (BRGC0)

(1) Transmit shift register 0 (TXS0)

This is the register for setting transmit data. Data written to TXS0 is transmitted as serial data.

When the data length is set as 7 bits, bits 0 to 6 of the data written to TXS0 are transferred as transmit data.

Writing data to TXS0 starts the transmit operation.

TXS0 can be written by an 8-bit memory manipulation instruction. It cannot be read.

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

Caution Do not write to TXS0 during a transmit operation.

The same address is assigned to TXS0 and receive buffer register 0 (RXB0). A read operation reads values from RXB0.

(2) Receive shift register 0 (RX0)

This register converts serial data input via the RxD0 pin to parallel data. When one byte of data is received at this register, the receive data is transferred to receive buffer register 0 (RXB0).

RX0 cannot be manipulated directly by a program.

(3) Receive buffer register 0 (RXB0)

This register is used to hold receive data. When one byte of data is received, one byte of new receive data is transferred from the receive shift register (RX0).

When the data length is set as 7 bits, receive data is transferred to bits 0 to 6 of RXB0. In this case, the MSB of RXB0 always becomes 0.

RXB0 can be read by an 8-bit memory manipulation instruction. It cannot be written to.

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

Caution The same address is assigned to RXB0 and transmit shift register 0 (TXS0). During a write operation, values are written to TXS0.

(4) Transmit controller

The transmit controller controls transmit operations, such as adding a start bit, parity bit, and stop bit to data that is written to transmit shift register 0 (TXS0), based on the values set to asynchronous serial interface mode register 0 (ASIM0).

(5) Receive controller

The receive controller controls receive operations based on the values set to asynchronous serial interface mode register 0 (ASIM0). During a receive operation, it performs error checking, such as for parity errors, and sets various values to asynchronous serial interface status register 0 (ASIS0) according to the type of error that is detected.

16.3 Registers to Control Serial Interface UART0

Serial interface UART0 is controlled by the following three types of registers.

- Asynchronous serial interface mode register 0 (ASIM0)
- Asynchronous serial interface status register 0 (ASIS0)
- Baud rate generator control register 0 (BRGC0)

(1) Asynchronous serial interface mode register 0 (ASIM0)

This is an 8-bit register that controls serial interface UART0's serial transfer operations.
ASIM0 is set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets the value of this register to 00H.
Figure 16-3 shows the format of ASIM0.

Caution In UART mode, set the port mode register (PMXX) as follows. Set the output latch of a port set to output mode (PMXX = 0) to 0.

- **During receive operation**
  Set P26 (RxD0) to input mode (PM26 = 1)
- **During transmit operation**
  Set P27 (TxD0) to output mode (PM27 = 0)
- **During transmit/receive operation**
  Set P26 to input mode, and P27 to output mode

Figure 16-3. Format of Asynchronous Serial Interface Mode Register 0 (ASIM0)

Address: FFA0H After reset: 00H R/W

Symbol	$\boxed{7}$	$\boxed{6}$	5	4	3	2	1	0
ASIM0	TXE0	RXE0	PS01	PS00	CL0	SL0	ISRM0	0

TXE0	RXE0	Operation mode	RxD0/P26 pin function	TxD0/P27 pin function
0	0	Operation stop	Port function (P26)	Port function (P27)
0	1	UART mode (receive only)	Serial function (RxD0)	
1	0	UART mode (transmit only)	Port function (P26)	Serial function (TxD0)
1	1	UART mode (transmit and receive)	Serial function (RxD0)	

PS01	PS00	Parity bit specification
0	0	No parity
0	1	Zero parity always added during transmission No parity detection during reception (parity errors do not occur)
1	0	Odd parity
1	1	Even parity

CL0	Character length specification
0	7 bits
1	8 bits

SL0	Stop bit length specification for transmit data
0	1 bit
1	2 bits

ISRM0	Receive completion interrupt control when error occurs
0	Receive completion interrupt request is issued when an error occurs
1	Receive completion interrupt request is not issued when an error occurs

★ **Caution Before rewriting ASIM0 with a value other than the same value, temporarily stop the operation.**

(2) Asynchronous serial interface status register 0 (ASIS0)

When a receive error occurs in UART mode, this register indicates the type of error.

ASIS0 can be read by an 8-bit memory manipulation instruction.

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

Figure 16-4. Format of Asynchronous Serial Interface Status Register 0 (ASIS0)

Address: FFA1H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS0	0	0	0	0	0	PE0	FE0	OVE0

PE0	Parity error flag
0	No parity error
1	Parity error (Transmit data parity did not match)

FE0	Framing error flag
0	No framing error
1	Framing error[Note 1] (Stop bit not detected)

OVE0	Overrun error flag
0	No overrun error
1	Overrun error[Note 2] (Next receive operation was completed before data was read from receive buffer register 0 (RXB0))

Notes 1. Even if the stop bit length is set to 2 bits by setting bit 2 (SL0) in asynchronous serial interface mode register 0 (ASIM0), stop bit detection during a receive operation only applies to a stop bit length of 1 bit.

★ **2.** Once an overrun error has occurred, further overrun errors will occur until the contents of RXB0 are read.

(3) Baud rate generator control register 0 (BRGC0)

This register sets the serial clock for the serial interface.

BRGC0 is set by an 8-bit memory manipulation instruction.

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

Figure 16-5 shows the format of BRGC0.

Figure 16-5. Format of Baud Rate Generator Control Register 0 (BRGC0)

Address: FFA2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
BRGC0	0	TPS02	TPS01	TPS00	MDL03	MDL02	MDL01	MDL00

TPS02	TPS01	TPS00	Source clock selection for 5-bit counter	n
0	0	0	Setting prohibited	—
0	0	1	$f_X/2$	1
0	1	0	$f_X/2^2$	2
0	1	1	$f_X/2^3$	3
1	0	0	$f_X/2^4$	4
1	0	1	$f_X/2^5$	5
1	1	0	$f_X/2^6$	6
1	1	1	$f_X/2^7$	7

MDL03	MDL02	MDL01	MDL00	Output clock selection for baud rate generator	k
0	0	0	0	$f_{SCK}/16$	0
0	0	0	1	$f_{SCK}/17$	1
0	0	1	0	$f_{SCK}/18$	2
0	0	1	1	$f_{SCK}/19$	3
0	1	0	0	$f_{SCK}/20$	4
0	1	0	1	$f_{SCK}/21$	5
0	1	1	0	$f_{SCK}/22$	6
0	1	1	1	$f_{SCK}/23$	7
1	0	0	0	$f_{SCK}/24$	8
1	0	0	1	$f_{SCK}/25$	9
1	0	1	0	$f_{SCK}/26$	10
1	0	1	1	$f_{SCK}/27$	11
1	1	0	0	$f_{SCK}/28$	12
1	1	0	1	$f_{SCK}/29$	13
1	1	1	0	$f_{SCK}/30$	14
1	1	1	1	Setting prohibited	—

Caution Writing to BRGC0 during a communication operation may cause abnormal output from the baud rate generator and disable further communication operations. Therefore, do not write to BRGC0 during a communication operation.

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{SCK}: Source clock for 5-bit counter

3. n: Value set via TPS00 to TPS02 ($1 \leq n \leq 7$)

4. k: Value set via MDL00 to MDL03 ($0 \leq k \leq 14$)

★ **5.** Baud rate calculation

$$[\text{Baud rate}] = \frac{f_X}{2^{n+1}(k + 16)} \text{ [Hz]}$$

16.4 Serial Interface UART0 Operations

This section explains the two modes of serial interface UART0.

16.4.1 Operation stop mode

Because serial transfer is not performed during this mode, the power consumption can be reduced.
In addition, pins can be used as normal ports.

(1) Register settings

Operation stop mode is set by asynchronous serial interface mode register 0 (ASIM0).
ASIM0 is set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets the value of this register to 00H.

Address: FFA0H After reset: 00H R/W

Symbol	$\boxed{7}$	$\boxed{6}$	5	4	3	2	1	0
ASIM0	TXE0	RXE0	PS01	PS00	CL0	SL0	ISRM0	0

TXE0	RXE0	Operation mode	RxD0/P26 pin function	TxD0/P27 pin function
0	0	Operation stop	Port function (P26)	Port function (P27)
0	1	UART mode (receive only)	Serial function (RxD0)	
1	0	UART mode (transmit only)	Port function (P26)	Serial function (TxD0)
1	1	UART mode (transmit and receive)	Serial function (RxD0)	

★ **Caution Before rewriting ASIM0 with a value other than the same value, temporarily stop the operation.**

16.4.2 Asynchronous serial interface (UART) mode

This mode enables full-duplex operation wherein one byte of data after the start bit is transmitted or received.

The on-chip baud rate generator dedicated to UART enables communications using a wide range of selectable baud rates.

The UART baud rate generator can also be used to generate a MIDI-standard baud rate (31.25 kbps).

(1) Register settings

UART mode settings are performed by asynchronous serial interface mode register 0 (ASIM0), asynchronous serial interface status register 0 (ASIS0), and baud rate generator control register 0 (BRGC0).

(a) Asynchronous serial interface mode register 0 (ASIM0)

ASIM0 is set by a 1-bit or 8-bit memory manipulation instruction.

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

Caution In UART mode, set the port mode register (PMXX) as follows. Set the output latch of a port set to output mode (PMXX = 0) to 0.

- **During receive operation**
  Set P26 (RxD0) to input mode (PM26 = 1)
- **During transmit operation**
  Set P27 (TxD0) to output mode (PM27 = 0)
- **During transmit/receive operation**
  Set P26 to input mode, and P27 to output mode

Address: FFA0H After reset: 00H R/W

Symbol	$\boxed{7}$	$\boxed{6}$	5	4	3	2	1	0
ASIM0	TXE0	RXE0	PS01	PS00	CL0	SL0	ISRM0	0

TXE0	RXE0	Operation mode	RxD0/P26 pin function	TxD0/P27 pin function
0	0	Operation stop	Port function (P26)	Port function (P27)
0	1	UART mode (receive only)	Serial function (RxD0)	
1	0	UART mode (transmit only)	Port function (P26)	Serial function (TxD0)
1	1	UART mode (transmit and receive)	Serial function (RxD0)	

PS01	PS00	Parity bit specification
0	0	No parity
0	1	Zero parity always added during transmission No parity detection during reception (parity errors do not occur)
1	0	Odd parity
1	1	Even parity

CL0	Character length specification
0	7 bits
1	8 bits

SL0	Stop bit length specification for transmit data
0	1 bit
1	2 bits

ISRM0	Receive completion interrupt control when error occurs
0	Receive completion interrupt request is issued when an error occurs
1	Receive completion interrupt request is not issued when an error occurs

★ **Caution Before rewriting ASIM0 with a value other than the same value, temporarily stop the operation.**

(b) Asynchronous serial interface status register 0 (ASIS0)

ASIS0 can be read by an 8-bit memory manipulation instruction.

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

Address: FFA1H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS0	0	0	0	0	0	PE0	FE0	OVE0

PE0	Parity error flag
0	No parity error
1	Parity error (Transmit data parity not matched)

FE0	Framing error flag
0	No framing error
1	Framing error[Note 1] (Stop bit not detected)

OVE0	Overrun error flag
0	No overrun error
1	Overrun error[Note 2] (Next receive operation was completed before data was read from receive buffer register 0 (RXB0))

Notes 1. Even if a stop bit length is set to 2 bits by setting bit 2 (SL0) in asynchronous serial interface mode register 0 (ASIM0), stop bit detection during a receive operation only applies to a stop bit length of 1 bit.

★ **2.** Once an overrun error has occurred, further overrun errors will occur until the contents of RXB0 are read.

(c) Baud rate generator control register 0 (BRGC0)

BRGC0 is set by an 8-bit memory manipulation instruction.

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

Address: FFA2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
BRGC0	0	TPS02	TPS01	TPS00	MDL03	MDL02	MDL01	MDL00

TPS02	TPS01	TPS00	Source clock selection for 5-bit counter	n
0	0	0	Setting prohibited	—
0	0	1	$f_X/2$	1
0	1	0	$f_X/2^2$	2
0	1	1	$f_X/2^3$	3
1	0	0	$f_X/2^4$	4
1	0	1	$f_X/2^5$	5
1	1	0	$f_X/2^6$	6
1	1	1	$f_X/2^7$	7

MDL03	MDL02	MDL01	MDL00	Output clock selection for baud rate generator	k
0	0	0	0	$f_{SCK}/16$	0
0	0	0	1	$f_{SCK}/17$	1
0	0	1	0	$f_{SCK}/18$	2
0	0	1	1	$f_{SCK}/19$	3
0	1	0	0	$f_{SCK}/20$	4
0	1	0	1	$f_{SCK}/21$	5
0	1	1	0	$f_{SCK}/22$	6
0	1	1	1	$f_{SCK}/23$	7
1	0	0	0	$f_{SCK}/24$	8
1	0	0	1	$f_{SCK}/25$	9
1	0	1	0	$f_{SCK}/26$	10
1	0	1	1	$f_{SCK}/27$	11
1	1	0	0	$f_{SCK}/28$	12
1	1	0	1	$f_{SCK}/29$	13
1	1	1	0	$f_{SCK}/30$	14
1	1	1	1	Setting prohibited	—

Caution Writing to BRGC0 during a communication operation may cause abnormal output from the baud rate generator and disable further communication operations. Therefore, do not write to BRGC0 during a communication operation.

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{SCK}: Source clock for 5-bit counter

3. n: Value set via TPS00 to TPS02 ($1 \leq n \leq 7$)

4. k: Value set via MDL00 to MDL03 ($0 \leq k \leq 14$)

The transmit/receive clock that is used to generate the baud rate is obtained by dividing the main system clock.

- Transmit/receive clock generation for baud rate by using main system clock
  The main system clock is divided to generate the transmit/receive clock. The baud rate generated from the main system clock is determined according to the following formula.

  $$[\text{Baud rate}] = \frac{f_X}{2^{n+1}(k + 16)} \quad [\text{Hz}]$$

  f_X: Main system clock oscillation frequency
  n: Value set via TPS00 to TPS02 ($1 \le n \le 7$)
  For details, see **Table 16-2**.
  k: Value set via MDL00 to MDL03 ($0 \le k \le 14$)

Table 16-2 shows the relationship between the 5-bit counter's source clock assigned to bits 4 to 6 (TPS00 to TPS02) of BRGC0 and the "n" value in the above formula and Table 16-3 shows the relationship between the main system clock and the baud rate.

Table 16-2. Relationship Between 5-Bit Counter's Source Clock and "n" Value

TPS02	TPS01	TPS00	5-Bit Counter's Source Clock Selected	n
0	0	0	Setting prohibited	—
0	0	1	$f_X/2$	1
0	1	0	$f_X/2^2$	2
0	1	1	$f_X/2^3$	3
1	0	0	$f_X/2^4$	4
1	0	1	$f_X/2^5$	5
1	1	0	$f_X/2^6$	6
1	1	1	$f_X/2^7$	7

Remark f_X: Main system clock oscillation frequency

Table 16-3. Relationship Between Main System Clock and Baud Rate

★

Baud Rate (bps)	f_X = 10 MHz		f_X = 9.8304 MHz		f_X = 8.3886 MHz		f_X = 8 MHz	
	BRGC0	ERR (%)	BRGC0	ERR (%)	BRGC0	ERR (%)	BRGC0	ERR (%)
600	–	–	–	–	–	–	–	–
1,200	–	–	–	–	7BH	1.10	7AH	0.16
2,400	70H	1.73	70H	0.00	6BH	1.10	6AH	0.16
4,800	60H	1.73	60H	0.00	5BH	1.10	5AH	0.16
9,600	50H	1.73	50H	0.00	4BH	1.10	4AH	0.16
19,200	40H	1.73	40H	0.00	3BH	1.10	3AH	0.16
31,250	34H	0.00	34H	–1.70	31H	–3.14	30H	0.00
38,400	30H	1.73	30H	0.00	2BH	1.10	2AH	0.16
76,800	20H	1.73	20H	0.00	1BH	1.10	1AH	0.16
115,200	16H	–1.36	16H	–3.03	12H	1.10	11H	2.12
153,600	10H	1.73	10H	0.00	–	–	–	–

Baud Rate (bps)	f_X = 7.3728 MHz		f_X = 5 MHz		f_X = 4.194304 MHz	
	BRGC0	ERR (%)	BRGC0	ERR (%)	BRGC0	ERR (%)
600	–	–	–	–	7BH	1.14
1,200	78H	0.00	70H	1.73	6BH	1.14
2,400	68H	0.00	60H	1.73	5BH	1.14
4,800	58H	0.00	50H	1.73	4BH	1.14
9,600	48H	0.00	40H	1.73	3BH	1.14
19,200	38H	0.00	30H	1.73	2BH	1.14
31,250	2DH	1.69	24H	0.00	21H	–1.31
38,400	28H	0.00	20H	1.73	1BH	1.14
76,800	18H	0.00	10H	1.73	–	–
115,200	10H	0.00	–	–	–	–
153,600	–	–	–	–	–	–

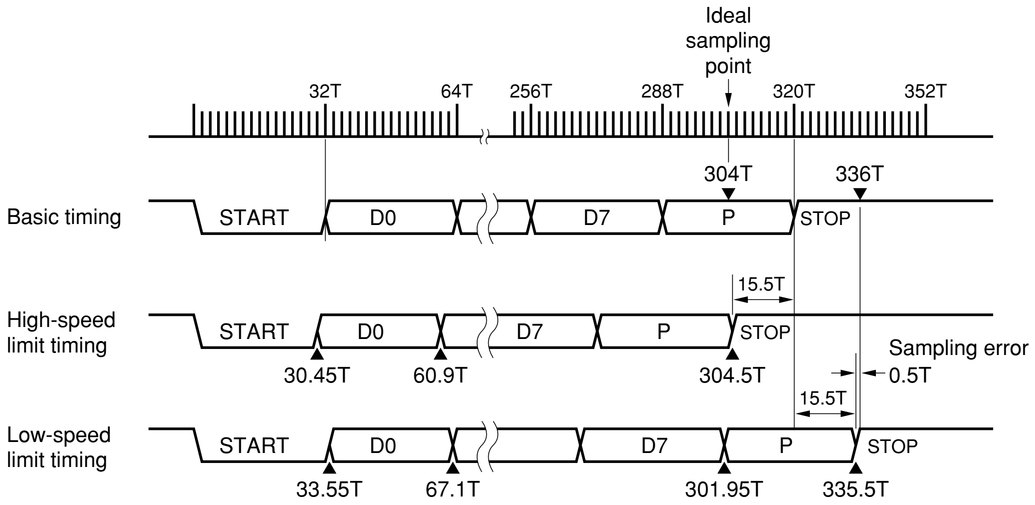
Remark f_X: Main system clock oscillation frequency
n: Value set via TPS00 to TPS02 ($1 \leq n \leq 7$)
k: Value set via MDL00 to MDL03 ($0 \leq k \leq 14$)

- **Error tolerance range for baud rate**

  The tolerance range for the baud rate depends on the number of bits per frame and the counter's division rate [1/(16 + k)].

  Figure 16-6 shows an example of the baud rate error tolerance range.

Figure 16-6. Baud Rate Error Tolerance (When k = 0), Including Sampling Errors



Remark T: 5-bit counter's source clock cycle

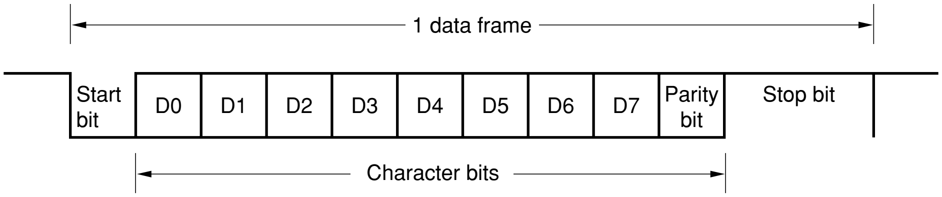
Baud rate error tolerance (when k = 0) = $\frac{\pm 15.5}{320} \times 100 = 4.8438$ (%)

(2) Communication operations

(a) Data format

Figure 16-7 shows the format of the transmit/receive data.

Figure 16-7. Format of Transmit/Receive Data in Asynchronous Serial Interface



1 data frame consists of the following bits.

- Start bit .... 1 bit
- ★ Character bits ... 7 bits or 8 bits (LSB first)
- Parity bit .... Even parity, odd parity, zero parity, or no parity
- Stop bit(s) .... 1 bit or 2 bits

Asynchronous serial interface mode register 0 (ASIM0) is used to set the character bit length, parity selection, and stop bit length within each data frame.

When “7 bits” is selected as the number of character bits, only the lower 7 bits (bits 0 to 6) are valid, so that during a transmission the highest bit (bit 7) is ignored and during reception the highest bit (bit 7) must be set to “0”.

★ Baud rate generator control register 0 (BRGC0) is used to set the serial transfer rate.

If a receive error occurs, information about the receive error can be recognized by reading asynchronous serial interface status register 0 (ASIS0).

(b) Parity types and operations

The parity bit is used to detect bit errors in communication data. Usually, the same type of parity bit is used by the transmitting and receiving sides. When odd parity or even parity is set, errors in the parity bit (the odd-number bit) can be detected. When zero parity or no parity is set, errors are not detected.

★ **(i) Even parity**

- During transmission
  The number of character bits in transmit data that includes a parity bit is controlled so that there are an even number of bits whose value is 1. The value of the parity bit is as follows.

  If the transmit data contains an odd number of character bits whose value is 1: the parity bit is “1”
  If the transmit data contains an even number of character bits whose value is 1: the parity bit is “0”

- During reception
  The number of character bits whose value is 1 is counted among the receive data that include a parity bit, and a parity error occurs when the counted result is an odd number.

★ **(ii) Odd parity**

- During transmission
  The number of character bits in transmit data that includes a parity bit is controlled so that there is an odd number of bits whose value is 1. The value of the parity bit is as follows.

  If the transmit data contains an odd number of character bits whose value is 1: the parity bit is “0”
  If the transmit data contains an even number of character bits whose value is 1: the parity bit is “1”

- During reception
  The number of character bits whose value is 1 is counted among the receive data that include a parity bit, and a parity error occurs when the counted result is an even number.

(iii) Zero parity

During transmission, the parity bit is set to “0” regardless of the transmit data.
During reception, the parity bit is not checked. Therefore, no parity errors will occur regardless of whether the parity bit is a “0” or a “1”.

(iv) No parity

No parity bit is added to the transmit data.
During reception, receive data is regarded as having no parity bit. Since there is no parity bit, no parity errors will occur.

(c) Transmission

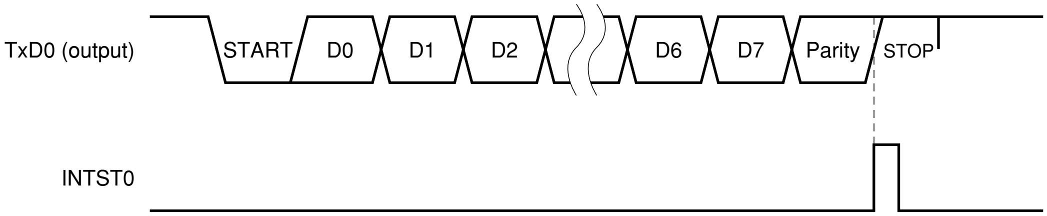
The transmit operation is enabled if bit 7 (TXE0) of asynchronous serial interface mode register 0 (ASIM0) is set to 1, the transmit operation is started when transmit data is written to transmit shift register 0 (TXS0). A start bit, parity bit, and stop bit(s) are automatically added to the data.

Starting the transmit operation shifts out the data in TXS0, thereby emptying TXS0, after which a transmit completion interrupt request (INTST0) is issued.

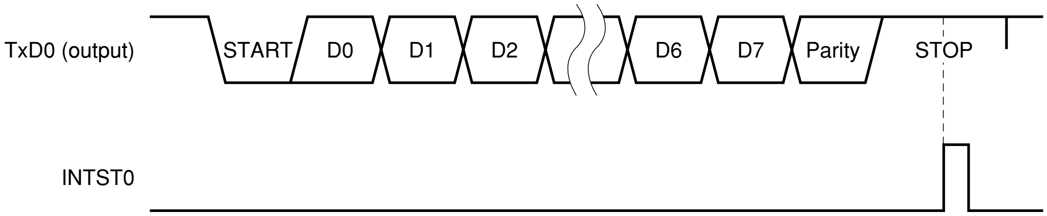
The timing of the transmit completion interrupt request is shown in Figure 16-8.

Figure 16-8. Timing of Asynchronous Serial Interface Transmit Completion Interrupt Request

(i) Stop bit length: 1 bit



(ii) Stop bit length: 2 bits



Caution Do not rewrite asynchronous serial interface mode register 0 (ASIM0) during a transmit operation. Rewriting ASIM0 register during a transmit operation may disable further transmit operations (in such cases, enter a $\overline{\text{RESET}}$ to restore normal operation).

(d) Reception

★ Receive operations are executed via level detection.

The receive operation is enabled when “1” is set to bit 6 (RXE0) of asynchronous serial interface mode register 0 (ASIM0), and input via the RxD0 pin is sampled.

The serial clock specified by BRGC0 is used to sample the RxD0 pin.

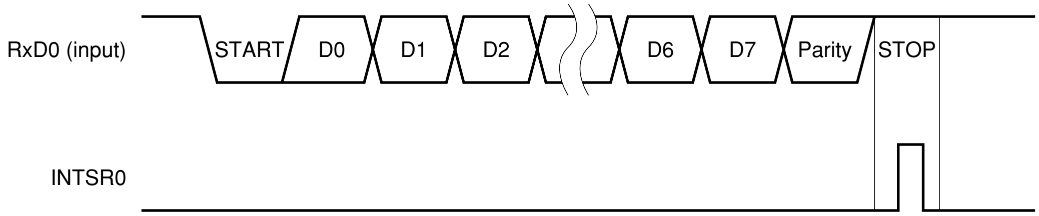
When the RxD0 pin goes low, the 5-bit counter of the baud rate generator begins counting and the start timing signal for data sampling is output when half of the specified baud rate time has elapsed. If sampling the RxD0 pin input with this start timing signal yields a low-level result, a start bit is recognized, after which the 5-bit counter is initialized and starts counting and data sampling begins. After the start bit is recognized, the character data, parity bit, and one-bit stop bit are detected, at which point reception of one data frame is completed.

Once reception of one data frame is completed, the receive data in the shift register is transferred to receive buffer register 0 (RXB0) and INTSR0 (receive completion interrupt request) occurs.

If the RXE0 bit is reset (to “0”) during a receive operation, the receive operation is stopped immediately. At this time, the contents of RXB0 and ASIS0 do not change, nor does INTSR0 or INTSER0 (receive error interrupt request) occur.

Figure 16-9 shows the timing of the asynchronous serial interface receive completion interrupt request.

Figure 16-9. Timing of Asynchronous Serial Interface Receive Completion Interrupt Request



★ **Caution Be sure to enable receive operations after setting the RxD0 pin input to high level; a receive operation is immediately started if receive operations are enabled when the RxD0 pin input is at low level.**

(e) Receive errors

Three types of errors can occur during a receive operation: parity error, framing error, or overrun error. If, as the result of data reception, an error flag is set to asynchronous serial interface status register 0 (ASIS0), a receive error interrupt request (INTSER0) will occur. Receive error interrupt requests are generated before receive completion interrupt requests (INTSR0). Table 16-4 lists the causes behind receive errors.

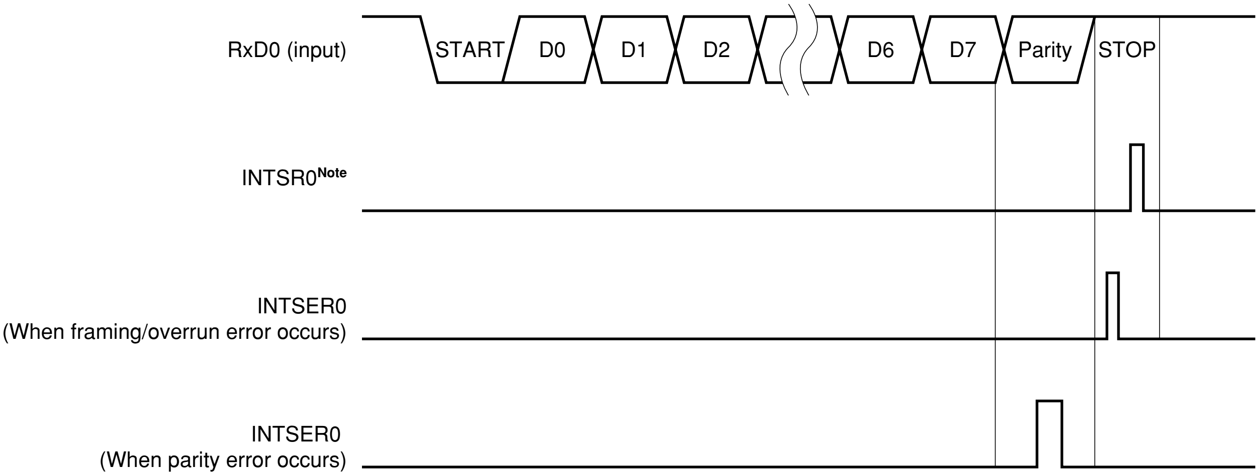
As part of receive error interrupt request (INTSER0) servicing, the contents of ASIS0 can be read to determine which type of error occurred during the receive operation (see **Table 16-4** and **Figure 16-10**).

The contents of ASIS0 are reset (to “0”) when receive buffer register 0 (RXB0) is read or when the next data is received (if the next data contains an error, its error flag will be set).

Table 16-4. Causes of Receive Errors

Receive Error	Cause	ASIS0 Value
★ Parity error	Specified parity does not match parity of receive data	04H
Framing error	Stop bit was not detected	02H
Overrun error	Reception of the next data was completed before data was read from receive buffer register 0 (RXB0)	01H

Figure 16-10. Receive Error Timing


RxD0 (input)
START
D0
D1
D2
D6
D7
Parity
STOP
INTSR0Note
INTSER0
(When framing/overrun error occurs)
INTSER0
(When parity error occurs)

★ **Note** Even if a receive error occurs when the ISRM0 bit has been set (1), INTSR0 does not occur.

Cautions 1. The contents of asynchronous serial interface status register 0 (ASIS0) are reset (to “0”) when receive buffer register 0 (RXB0) is read or when the next data is received. To obtain information about the error, be sure to read the contents of ASIS0 before reading RXB0.

★ **2. Be sure to read the contents of receive buffer register 0 (RXB0) after a receive completion interrupt request is generated even when a receive error has occurred. Overrun errors will occur during the next data receive operations and the receive error status will remain until the contents of RXB0 are read after a receive completion interrupt request is generated.**

CHAPTER 17 SERIAL INTERFACE IIC0 (μPD780344Y, 780354Y SUBSERIES ONLY)

17.1 Functions of Serial Interface IIC0

Serial interface IIC0 has the following two modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed. It can therefore be used to reduce power consumption.

(2) I^{2}C bus mode (multimaster supported)

This mode is used for 8-bit data transfers with several devices via two lines: a serial clock (SCL0) line and a serial data bus (SDA0) line.

This mode complies with the I^{2}C bus format and can output “start condition”, “data”, and “stop condition” data segments when transmitting via the serial data bus. These data segments are automatically detected by hardware during reception.

Since SCL0 and SDA0 are open-drain outputs, IIC0 requires pull-up resistors for the serial clock line (SCL0) and the serial data bus line (SDA0).

Figure 17-1 shows a block diagram of serial interface IIC0.

Figure 17-1. Block Diagram of Serial Interface IIC0

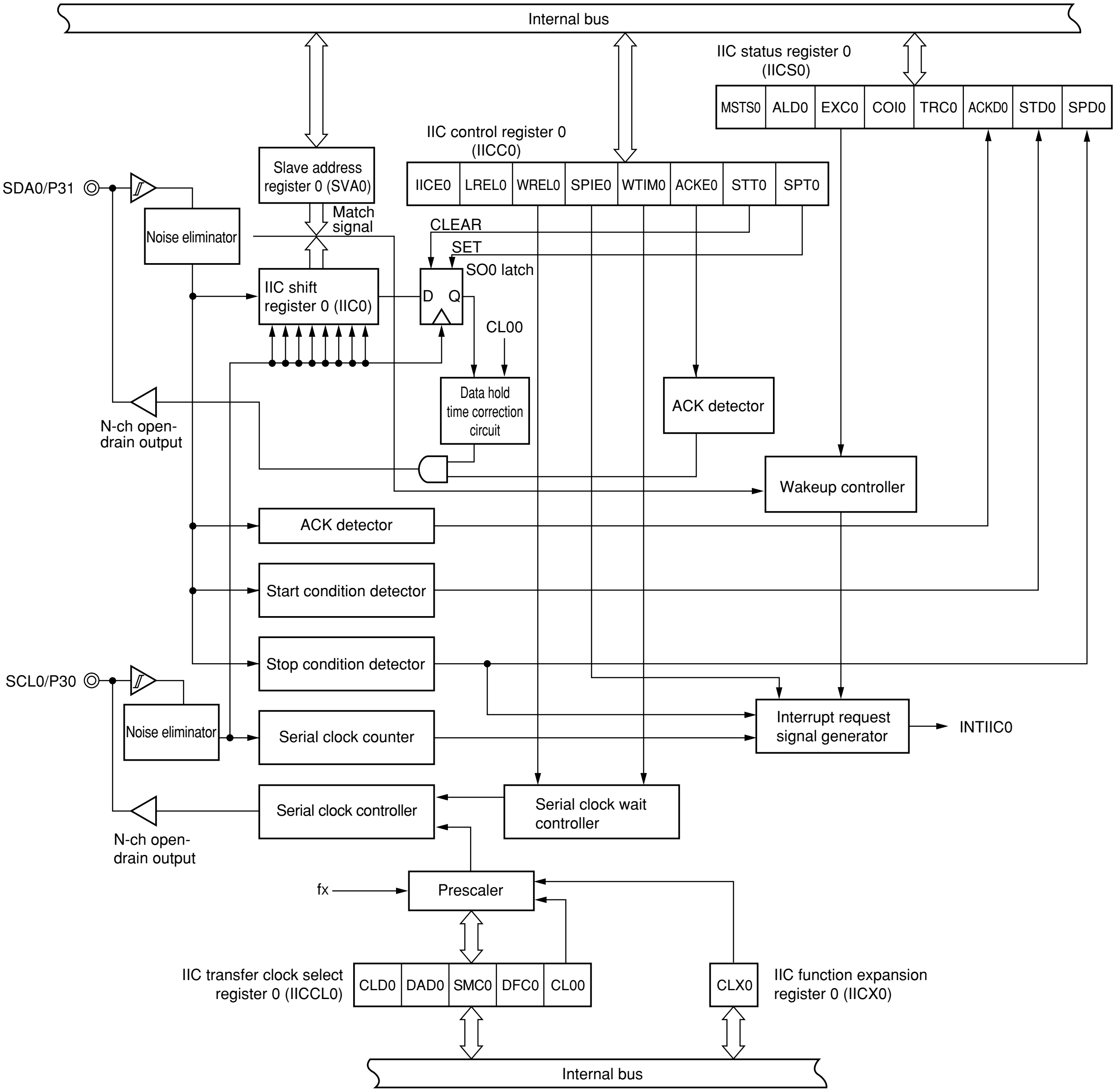
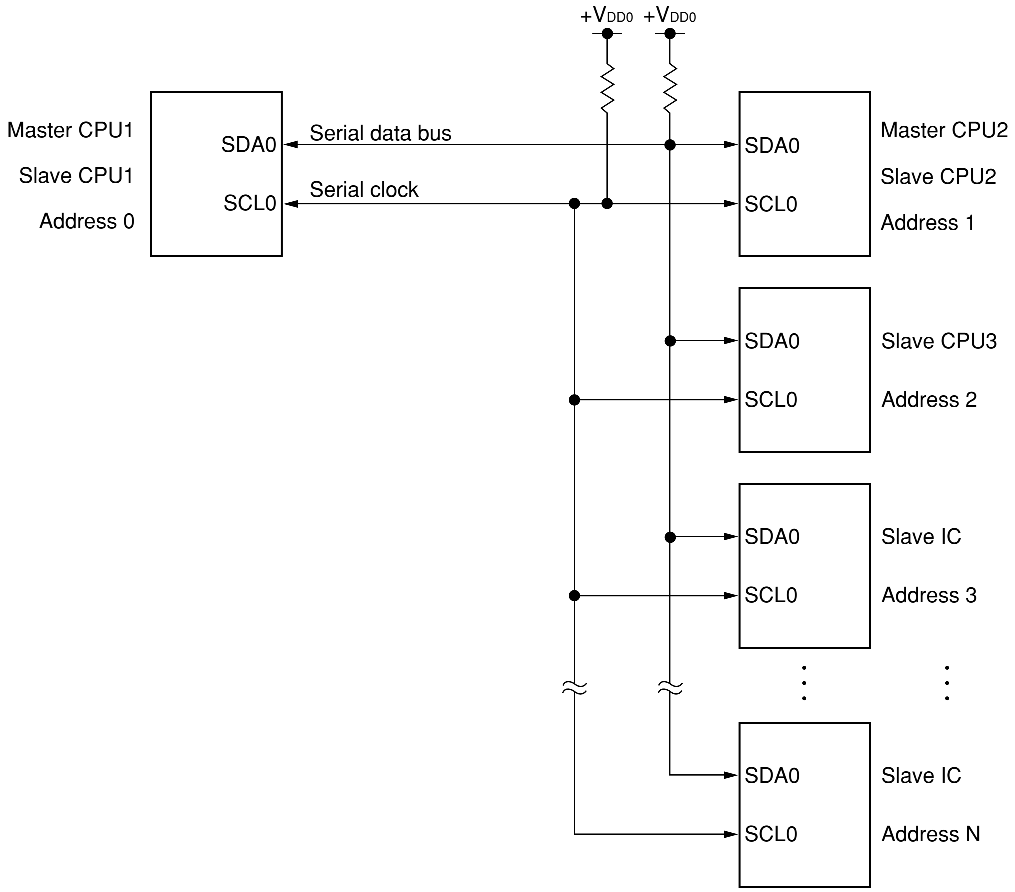


Figure 17-2 shows a serial bus configuration example.

Figure 17-2. Serial Bus Configuration Example Using I²C Bus



17.2 Serial Interface IIC0 Configuration

Serial interface IIC0 consists of the following hardware.

Table 17-1. Configuration of Serial Interface IIC0

Item	Configuration
Registers	IIC shift register 0 (IIC0) Slave address register 0 (SVA0)
Control registers	IIC control register 0 (IICC0) IIC status register 0 (IICS0) IIC transfer clock select register 0 (IICCL0) IIC function expansion register 0 (IICX0)

★ **(1) IIC shift register 0 (IIC0)**

IIC0 is used to convert 8-bit serial data to 8-bit parallel data and to convert 8-bit parallel data to 8-bit serial data in synchronization with the serial clock. IIC0 can be used for both transmission and reception.

Write and read operations to IIC0 are used to control the actual transmit and receive operations.

IIC0 is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets IIC0 to 00H.

Address: FF1FH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IIC0								

Caution Do not write to the IIC0 register during a data transfer.

★ **(2) Slave address register 0 (SVA0)**

This register sets local addresses when in slave mode.

SVA0 is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets SVA0 to 00H.

Address: FFABH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SVA0								0[Note]

Note Bit 0 is fixed to 0.

(3) SO0 latch

The SO0 latch is used to retain the SDA0 pin's output level.

(4) Wakeup controller

This circuit generates an interrupt request when the receive address matches the address value set to slave address register 0 (SVA0) or when an extension code is received.

(5) Prescaler

This selects the sampling clock to be used.

(6) Serial clock counter

This counter counts the serial clocks that are output or input during transmit/receive operations and is used to verify that 8-bit data was transmitted or received.

(7) Interrupt request signal generator

This circuit controls the generation of interrupt request signals (INTIIC0).
An I^{2}C interrupt request is generated following either of two triggers.

★
- Falling edge of the eighth or ninth serial clock (set by WTIM0 bit**Note**)
- Interrupt request generated when a stop condition is detected (set by SPIE0 bit**Note**)

Note WTIM0 bit: Bit 3 of the IIC control register 0 (IICC0)
SPIE0 bit: Bit 4 of the IIC control register 0 (IICC0)

(8) Serial clock controller

In master mode, this circuit generates the clock output via the SCL0 pin from the sampling clock.

(9) Serial clock wait controller

This circuit controls the wait timing.

(10) ACK output circuit, stop condition detector, start condition detector, and ACK detector

These circuits are used to output and detect various control signals.

(11) Data hold time correction circuit

This circuit generates the hold time for data corresponding to the falling edge of the serial clock.

17.3 Registers to Control Serial Interface IIC0

Serial interface IIC0 is controlled by the following four registers.

- IIC control register 0 (IICC0)
- IIC status register 0 (IICS0)
- IIC transfer clock select register 0 (IICCL0)
- IIC function expansion register 0 (IICX0)

(1) IIC control register 0 (IICC0)

This register is used to enable/disable I^{2}C operations, set wait timing, and set other I^{2}C operations.
IICC0 is set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets IICC0 to 00H.

Caution In I^{2}C bus mode, set the port mode register (PMXX) as follows. Set the output latch of a port set to output mode (PMXX) to 0.
- **Set P31 (SDA0) to output mode (PM31 = 0)**
- **Set P30 (SCL0) to output mode (PM30 = 0)**

Figure 17-3. Format of IIC Control Register 0 (IICC0) (1/3)

★ Address: FFA4H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IICC0	IICE0	LREL0	WREL0	SPIE0	WTIM0	ACKE0	STT0	SPT0

IICE0	I^2C operation enable
0	★ Stops operation. Resets IIC status register 0 (IICS0). Stops internal operation.
1	Enables operation.
Condition for clearing (IICE0 = 0)	Condition for setting (IICE0 = 1)
• Cleared by instruction • When $\overline{\text{RESET}}$ is input	• Set by instruction

LREL0	Exit from communications
0	Normal operation
1	This exits from the current communications operation and sets standby mode. This setting is automatically cleared after being executed. Its uses include cases in which a locally irrelevant extension code has been received. The SCL0 and SDA0 lines go into the high impedance state. The following flags are cleared. • STD0 • ACKD0 • TRC0 • COI0 • EXC0 • MSTS0 • STT0 • SPT0
The standby mode following exit from communications remains in effect until the following communications entry conditions are met. • After a stop condition is detected, restart is in master mode. • An address match or extension code reception occurs after the start condition.	
Condition for clearing (LREL0 = 0)$^{\text{Note}}$	Condition for setting (LREL0 = 1)
• Automatically cleared after execution • When $\overline{\text{RESET}}$ is input	• Set by instruction

WREL0	Cancel wait
0	Does not cancel wait
1	Cancels wait. This setting is automatically cleared after wait is canceled.
When WREL0 is set (wait canceled) during the wait period at the ninth clock pulse in transmission status (TRC0 = 1), the SDA0 line goes into the high impedance state (TRC0 = 0).	
Condition for clearing (WREL0 = 0)$^{\text{Note}}$	Condition for setting (WREL0 = 1)
• Automatically cleared after execution • When $\overline{\text{RESET}}$ is input	• Set by instruction

SPIE0	Enable/disable generation of interrupt request when stop condition is detected
0	Disable
1	Enable
Condition for clearing (SPIE0 = 0)$^{\text{Note}}$	Condition for setting (SPIE0 = 1)
• Cleared by instruction • When $\overline{\text{RESET}}$ is input	• Set by instruction

Note This flag's signal is invalid when IICE0 = 0.

Figure 17-3. Format of IIC Control Register 0 (IICC0) (2/3)

<table>
<tr><th>WTIM0</th><th>Control of wait and interrupt request generation</th></tr>
<tr><td>0</td><td>Interrupt request is generated at the eighth clock’s falling edge.
Master mode: After output of eight clocks, clock output is set to low level and wait is set.
Slave mode: After input of eight clocks, the clock is set to low level and wait is set for master device.</td></tr>
<tr><td>1</td><td>Interrupt request is generated at the ninth clock’s falling edge.
Master mode: After output of nine clocks, clock output is set to low level and wait is set.
Slave mode: After input of nine clocks, the clock is set to low level and wait is set for master device.</td></tr>
<tr><td colspan="2">This bit’s setting is invalid during an address transfer and is valid after the transfer is completed. When in master mode, a wait is inserted at the falling edge of the ninth clock during address transfers. For a slave device that has received a local address, a wait is inserted at the falling edge of the ninth clock after an ACK signal is issued. When the slave device has received an extension code, a wait is inserted at the falling edge of the eighth clock.</td></tr>
<tr><td>Condition for clearing (WTIM0 = 0)^{Note}</td><td>Condition for setting (WTIM0 = 1)</td></tr>
<tr><td>• Cleared by instruction
• When $\overline{\text{RESET}}$ is input</td><td>• Set by instruction</td></tr>
</table>

<table>
<tr><th>ACKE0</th><th>Acknowledge control</th></tr>
<tr><td>0</td><td>Disable acknowledge.</td></tr>
<tr><td>1</td><td>Enable acknowledge. During the ninth clock period, the SDA0 line is set to low level. However, the ACK is invalid during address transfers and is valid when EXC0 = 1.</td></tr>
<tr><td>Condition for clearing (ACKE0 = 0)^{Note}</td><td>Condition for setting (ACKE0 = 1)</td></tr>
<tr><td>• Cleared by instruction
• When $\overline{\text{RESET}}$ is input</td><td>• Set by instruction</td></tr>
</table>

<table>
<tr><th>STT0</th><th>Start condition trigger</th></tr>
<tr><td>0</td><td>Does not generate a start condition.</td></tr>
<tr><td>1</td><td>When bus is released (during STOP mode):
Generates a start condition (for starting as master). The SDA0 line is changed from high level to low level and then the start condition is generated. Next, after the rated amount of time has elapsed, SCL0 is changed to low level.
When bus is not used:
This trigger functions as a start condition reserve flag. When set, it releases the bus and then automatically generates a start condition.
Wait status (during master mode):
Generates a restart condition after wait is canceled.</td></tr>
<tr><td colspan="2">Cautions concerning set timing
• For master reception: Cannot be set during transfer. Can be set only at the waiting period when ACKE0 has been set to 0 and slave has been notified of final reception.
• For master transmission: A start condition may not be generated normally during the ACK period. Therefore, set it during the waiting period.
• Cannot be set at the same time as SPT0</td></tr>
<tr><td>Condition for clearing (STT0 = 0)</td><td>Condition for setting (STT0 = 1)</td></tr>
<tr><td>• Cleared by loss in arbitration
• Cleared after start condition is generated by master device
• Cleared by LREL0 = 1
• When IICE0 = 0
• When $\overline{\text{RESET}}$ is input</td><td>• Set by instruction</td></tr>
</table>

Note This flag’s signal is invalid when IICE0 = 0.

Figure 17-3. Format of IIC Control Register 0 (IICC0) (3/3)

<table>
<tr><th>SPT0</th><th colspan="2">Stop condition trigger</th></tr>
<tr><td>0</td><td colspan="2">Stop condition is not generated.</td></tr>
<tr><td>1</td><td colspan="2">Stop condition is generated (termination of master device’s transfer).
After the SDA0 line goes to low level, either set the SCL0 line to high level or wait until it goes to high level. Next, after the rated amount of time has elapsed, the SDA0 line changes from low level to high level and a stop condition is generated.</td></tr>
<tr><td colspan="3">Cautions concerning set timing
• For master reception: Cannot be set during transfer. Can be set only at the waiting period when ACKE0 has been set to 0 and slave has been notified of final reception.
• For master transmission: A stop condition cannot be generated normally during the ACK0 period. Therefore, set it during the waiting period.
• Cannot be set at the same time as STT0.
• SPT0 can be set only when in master mode.Note
• When WTIM0 has been set to 0, if SPT0 is set during the wait period that follows output of eight clocks, note that a stop condition will be generated during the high level period of the ninth clock. When a ninth clock must be output, WTIM0 should be changed from 0 to 1 during the wait period following output of eight clocks, and SPT0 should be set during the wait period that follows output of the ninth clock.</td></tr>
<tr><td colspan="2">Condition for clearing (SPT0 = 0)</td><td>Condition for setting (SPT0 = 1)</td></tr>
<tr><td colspan="2">• Cleared by loss in arbitration
• Automatically cleared after stop condition is detected
• Cleared by LREL0 = 1
• When IICE0 = 0
• When $\overline{\text{RESET}}$ is input</td><td>• Set by instruction</td></tr>
</table>

Note Set SPT0 only during master mode. However, you must set SPT0 and generate a stop condition before the first stop condition is detected following the switch to operation enable status. For details, see **17.5.15 Other cautions**.

Caution When bit 3 (TRC0) of IIC status register 0 (IICS0) is set to 1, WREL0 is set during the ninth clock and wait is canceled, after which TRC0 is cleared and the SDA0 line is set to high impedance.

Remarks 1. STD0: Bit 1 of IIC status register 0 (IICS0)
ACKD0: Bit 2 of IIC status register 0 (IICS0)
TRC0: Bit 3 of IIC status register 0 (IICS0)
COI0: Bit 4 of IIC status register 0 (IICS0)
EXC0: Bit 5 of IIC status register 0 (IICS0)
MSTS0: Bit 7 of IIC status register 0 (IICS0)

2. Bits 0 and 1 (SPT0, STT0) become 0 when they are read after data setting.

(2) IIC status register 0 (IICS0)

This register indicates the status of I^{2}C.

IICS0 is read by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets IICS0 to 00H.

Figure 17-4. Format of IIC Status Register 0 (IICS0) (1/3)

★ Address: FFA5H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
IICS0	MSTS0	ALD0	EXC0	COI0	TRC0	ACKD0	STD0	SPD0

MSTS0	Master device status
0	Slave device status or communication standby status
1	Master device communication status

Condition for clearing (MSTS0 = 0)	Condition for setting (MSTS0 = 1)
• When a stop condition is detected • When ALD0 = 1 • Cleared by LREL0 = 1 • When IICE0 changes from 1 to 0 • When $\overline{\text{RESET}}$ is input	• When a start condition is generated

ALD0	Detection of arbitration loss
0	This status means either that there was no arbitration or that the arbitration result was a “win”.
1	This status indicates the arbitration result was a “loss”. MSTS0 is cleared.

Condition for clearing (ALD0 = 0)	Condition for setting (ALD0 = 1)
• Automatically cleared after IICS0 is read[Note] • When IICE0 changes from 1 to 0 • When $\overline{\text{RESET}}$ is input	• When the arbitration result is a “loss”.

EXC0	Detection of extension code reception
0	Extension code was not received.
1	Extension code was received.

Condition for clearing (EXC0 = 0)	Condition for setting (EXC0 = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL0 = 1 • When IICE0 changes from 1 to 0 • When $\overline{\text{RESET}}$ is input	• When the higher 4 bits of the received address data is either “0000” or “1111” (set at the rising edge of the eighth clock).

Note This register is also cleared when a bit manipulation instruction is executed for bits other than IICS0.

Figure 17-4. Format of IIC Status Register 0 (IICS0) (2/3)

COI0	Detection of matching addresses
0	Addresses do not match.
1	Addresses match.

Condition for clearing (COI0 = 0)	Condition for setting (COI0 = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL0 = 1 • When IICE0 changes from 1 to 0 • When $\overline{\text{RESET}}$ is input	• When the received address matches the local address (SVA0) (set at the rising edge of the eighth clock).

TRC0	Detection of transmit/receive status
0	Receive status (other than transmit status). The SDA0 line is set for high impedance.
1	Transmit status. The value in the SO0 latch is enabled for output to the SDA0 line (valid starting at the falling edge of the first byte’s ninth clock).

Condition for clearing (TRC0 = 0)	Condition for setting (TRC0 = 1)
<Common to master and slave> • When a stop condition is detected • Cleared by LREL0 = 1 • When IICE0 changes from 1 to 0 • Cleared by WREL0 = 1[Note] • When ALD0 changes from 0 to 1 • When $\overline{\text{RESET}}$ is input <Master> • When “1” is output to the first byte’s LSB (transfer direction specification bit) <Slave> • When a start condition is detected • When “0” is input by the first byte’s LSB (transfer direction specification bit) • When not used for communication	<Master> • When a start condition is generated <Slave> • When “1” is input by the first byte’s LSB (transfer direction specification bit)

ACKD0	Detection of ACK
0	ACK was not detected.
1	ACK was detected.

Condition for clearing (ACKD0 = 0)	Condition for setting (ACKD0 = 1)
• When a stop condition is detected • At the rising edge of the next byte’s first clock • Cleared by LREL0 = 1 • When IICE0 changes from 1 to 0 • When $\overline{\text{RESET}}$ is input	• After the SDA0 line is set to low level at the rising edge of the SCL0’s ninth clock

Note When bit 3 (TRC0) of IIC status register 0 (IICS0) is set to 1, bit 5 (WREL0) of IIC control register 0 (IICC0) is set during the ninth clock and wait is canceled, after which TRC0 is cleared and the SDA0 line is set to high impedance.

Figure 17-4. Format of IIC Status Register 0 (IICS0) (3/3)

<table>
<tr><th>STD0</th><th colspan="2">Detection of start condition</th></tr>
<tr><td>0</td><td colspan="2">Start condition was not detected.</td></tr>
<tr><td>1</td><td colspan="2">Start condition was detected. This indicates that the address transfer period is in effect.</td></tr>
<tr><td colspan="2">Condition for clearing (STD0 = 0)</td><td>Condition for setting (STD0 = 1)</td></tr>
<tr><td colspan="2">• When a stop condition is detected
• At the rising edge of the next byte's first clock following address transfer
• Cleared by LREL0 = 1
• When IICE0 changes from 1 to 0
• When $\overline{\text{RESET}}$ is input</td><td>• When a start condition is detected</td></tr>
</table>

<table>
<tr><th>SPD0</th><th colspan="2">Detection of stop condition</th></tr>
<tr><td>0</td><td colspan="2">Stop condition was not detected.</td></tr>
<tr><td>1</td><td colspan="2">Stop condition was detected. The master device's communication was terminated and the bus was released.</td></tr>
<tr><td colspan="2">Condition for clearing (SPD0 = 0)</td><td>Condition for setting (SPD0 = 1)</td></tr>
<tr><td colspan="2">• At the rising edge of the address transfer byte's first clock following setting of this bit and detection of a start condition
• When IICE0 changes from 1 to 0
• When $\overline{\text{RESET}}$ is input</td><td>• When a stop condition is detected</td></tr>
</table>

Remark LREL0: Bit 6 of IIC control register 0 (IICC0)
IICE0: Bit 7 of IIC control register 0 (IICC0)

(3) IIC transfer clock select register 0 (IICCL0)

This register is used to set the transfer clock for the I^{2}C bus.

IICCL0 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets IICCL0 to 00H.

Figure 17-5. Format of IIC Transfer Clock Select Register 0 (IICCL0) (1/2)

★ Address: FFA6H After reset: 00H R/W[Note 1]

Symbol	7	6	$\boxed{5}$	$\boxed{4}$	3	2	1	0
IICCL0	0	0	CLD0	DAD0	SMC0	DFC0	0	CL00

<table>
<tr><th>CLD0</th><th colspan="2">Detection of SCL0 line level (valid only when IICE0 = 1)</th></tr>
<tr><td>0</td><td colspan="2">SCL0 line was detected at low level.</td></tr>
<tr><td>1</td><td colspan="2">SCL0 line was detected at high level.</td></tr>
<tr><td colspan="2">Condition for clearing (CLD0 = 0)</td><td>Condition for setting (CLD0 = 1)</td></tr>
<tr><td colspan="2">• When the SCL0 line is at low level
• When IICE0 = 0
• When $\overline{\text{RESET}}$ is input</td><td>• When the SCL0 line is at high level</td></tr>
</table>

<table>
<tr><th>DAD0</th><th colspan="2">Detection of SDA0 line level (valid only when IICE0 = 1)</th></tr>
<tr><td>0</td><td colspan="2">SDA0 line was detected at low level.</td></tr>
<tr><td>1</td><td colspan="2">SDA0 line was detected at high level.</td></tr>
<tr><td colspan="2">Condition for clearing (DAD0 = 0)</td><td>Condition for setting (DAD0 = 1)</td></tr>
<tr><td colspan="2">• When the SDA0 line is at low level
• When IICE0 = 0
• When $\overline{\text{RESET}}$ is input</td><td>• When the SDA0 line is at high level</td></tr>
</table>

<table>
<tr><th>SMC0</th><th colspan="2">Operation mode switching</th></tr>
<tr><td>0</td><td colspan="2">Operation in standard mode</td></tr>
<tr><td>1</td><td colspan="2">Operation in high-speed mode</td></tr>
<tr><td colspan="2">Condition for clearing (SMC0 = 0)</td><td>Condition for setting (SMC0 = 1)</td></tr>
<tr><td colspan="2">• Cleared by instruction
• When $\overline{\text{RESET}}$ is input</td><td>• Set by instruction</td></tr>
</table>

DFC0	Control of digital filter operation[Note 2]
0	Digital filter OFF
1	Digital filter ON

Notes 1. Bits 4 and 5 are read-only bits.

2. The digital filter can be used when in high-speed mode. The response time is slower when the digital filter is used.

Remark IICE0: Bit 7 of IIC control register 0 (IICC0)

Figure 17-5. Format of IIC Transfer Clock Select Register 0 (IICCL0) (2/2)

CL00	Selection of transfer rate		
	Standard mode	High-speed mode	
		CLX0 = 0	CLX0 = 1
0	f_X/88 (95.2 kHz)[Note 1]	f_X/48 (175 kHz)[Note 3]	f_X/24 (350 kHz)[Note 4]
1	f_X/172 (48.7 kHz)[Note 2]		

Notes 1. Transfer rate can only be used when f_X = 4.0 to 8.38 MHz.

2. Transfer rate can only be used when f_X = 8.38 to 10 MHz.

3. Transfer rate can only be used when f_X = 8.0 to 10 MHz.

4. Transfer rate can only be used when f_X = 8.0 to 8.38 MHz.

Caution Stop serial transfer once before rewriting CL00 to other than the same value.

Remarks 1. CLX0: Bit 0 of IIC function expansion register 0 (IICX0)

2. f_X: Main system clock oscillation frequency

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

(4) IIC function expansion register 0 (IICX0)

This register is used to set the function expansion for the I^2C bus.

IICX0 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets IICX0 to 00H.

Figure 17-6. Format of IIC Function Expansion Register 0 (IICX0)

Address: FFA8H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IICX0	0	0	0	0	0	0	0	CLX0

CLX0	Selection of transfer rate in high-speed mode
0	f_X/48 (175 kHz)[Note 1]
1	f_X/24 (350 kHz)[Note 2]

Notes 1. Transfer rate can only be used when f_X = 8.0 to 10 MHz.

2. Transfer rate can only be used when f_X = 8.0 to 8.38 MHz.

Cautions 1. This register is valid only in the high-speed mode. In the standard mode, this register is invalid.

2. Stop serial transfer once before rewriting IICX0 to other than the same value.

Remarks 1. f_X: Main system clock oscillation frequency

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

17.4 I²C Bus Mode Functions

17.4.1 Pin configuration

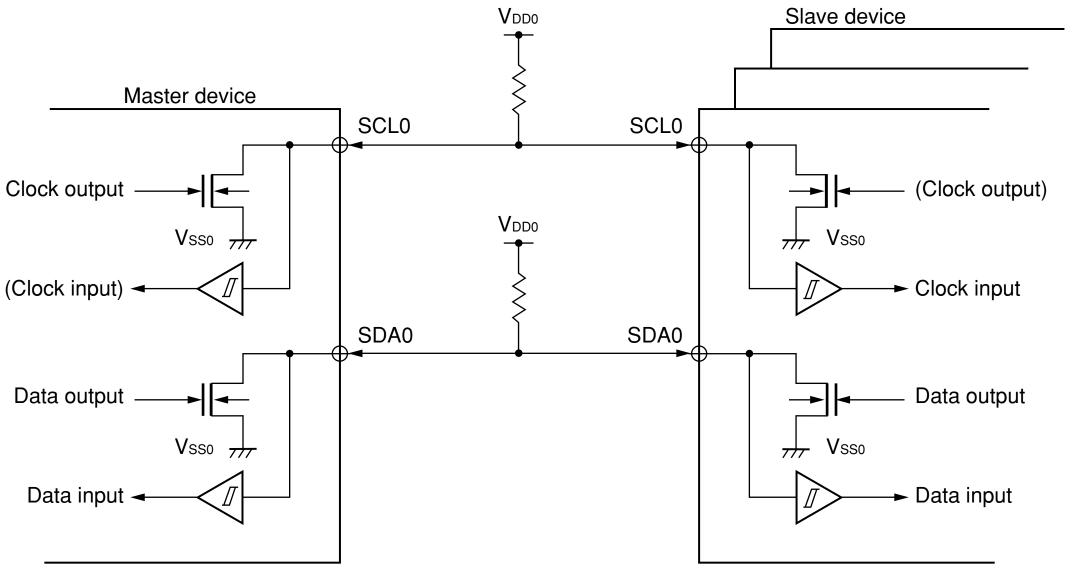
The serial clock pin (SCL0) and serial data bus pin (SDA0) are configured as follows.

(1) SCL0 ········· This pin is used for serial clock input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

(2) SDA0 ········· This pin is used for serial data input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

Since outputs from the serial clock line and the serial data bus line are N-ch open-drain outputs, an external pull-up resistor is required.

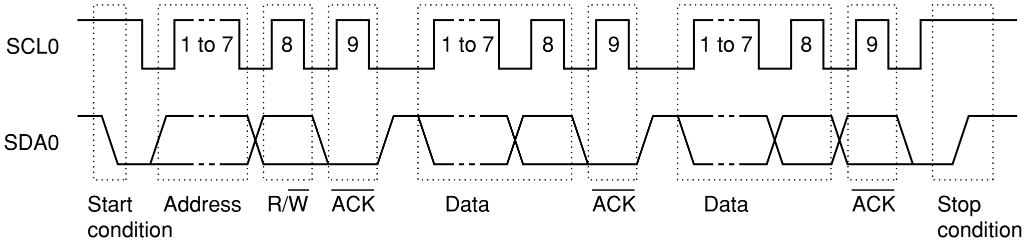
Figure 17-7. Pin Configuration Diagram



17.5 I^{2}C Bus Definitions and Control Methods

The following section describes the I^{2}C bus's serial data communication format and the signals used by the I^{2}C bus. Figure 17-8 shows the transfer timing for the "start condition", "data", and "stop condition" output via the I^{2}C bus's serial data bus.

Figure 17-8. I^{2}C Bus Serial Data Transfer Timing



The master device outputs the start condition, slave address, and stop condition.

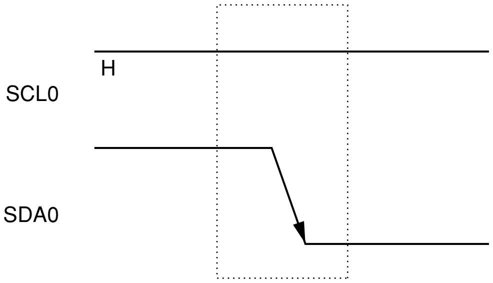
The acknowledge signal ($\overline{ACK}$) can be output by either the master or slave device (normally, it is output by the device that receives 8-bit data).

The serial clock (SCL0) is continuously output by the master device. However, in the slave device, the SCL0's low level period can be extended and a wait can be inserted.

17.5.1 Start conditions

A start condition is met when the SCL0 pin is at high level and the SDA0 pin changes from high level to low level. The start conditions for the SCL0 pin and SDA0 pin are signals that the master device outputs to the slave device when starting a serial transfer. The slave device includes hardware for detecting start conditions.

Figure 17-9. Start Conditions



A start condition is output when bit 1 (STT0) of IIC control register 0 (IICC0) is set (1) after a stop condition has been detected (SPD0: bit 0 = 1 in IIC status register 0 (IICS0)). When a start condition is detected, bit 1 of IICS0 (STD0) is set (1).

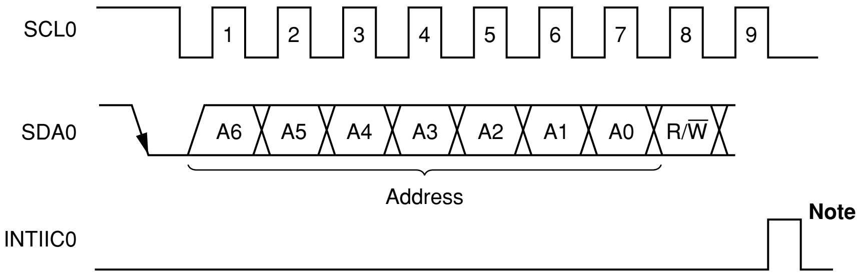
17.5.2 Addresses

The address is defined by the 7 bits of data that follow the start condition.

An address is a 7-bit data segment that is output in order to select one of the slave devices that are connected to the master device via bus lines. Therefore, each slave device connected via the bus lines must have a unique address.

The slave devices include hardware that detects the start condition and checks whether or not the 7-bit address data matches the data values stored in slave address register 0 (SVA0). If the address data matches the SVA0 values, the slave device is selected and communicates with the master device until the master device transmits a start condition or stop condition.

Figure 17-10. Address



Note INTIIC0 is not issued if data other than a local address or extension code is received during slave device operation.

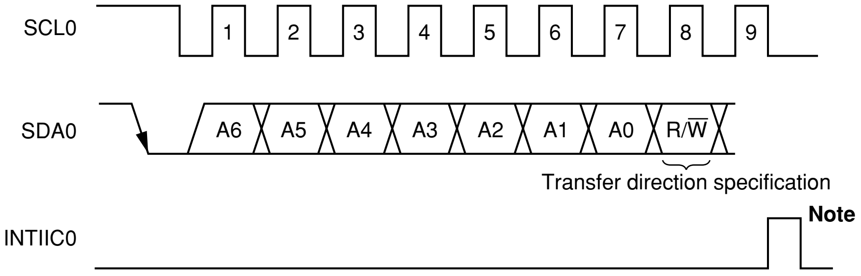
The slave address and the eighth bit, which specifies the transfer direction as described in **17.5.3 Transfer direction specification** below, are together written to IIC shift register 0 (IIC0) and are then output. Received addresses are written to IIC0.

The slave address is assigned to the higher 7 bits of IIC0.

17.5.3 Transfer direction specification

In addition to the 7-bit address data, the master device sends 1 bit that specifies the transfer direction. When this transfer direction specification bit has a value of “0”, it indicates that the master device is transmitting data to a slave device. When the transfer direction specification bit has a value of “1”, it indicates that the master device is receiving data from a slave device.

Figure 17-11. Transfer Direction Specification



Note INTIIC0 is not issued if data other than a local address or extension code is received during slave device operation.

17.5.4 Acknowledge ($\overline{\text{ACK}}$) signal

The acknowledge ($\overline{\text{ACK}}$) signal is used by the transmitting and receiving devices to confirm serial data reception.

The receiving device returns one $\overline{\text{ACK}}$ signal for each 8 bits of data it receives. The transmitting device normally receives an $\overline{\text{ACK}}$ signal after transmitting 8 bits of data. However, when the master device is the receiving device, it does not output an $\overline{\text{ACK}}$ signal after receiving the final data to be transmitted. The transmitting device detects whether or not an $\overline{\text{ACK}}$ signal is returned after it transmits 8 bits of data. When an $\overline{\text{ACK}}$ signal is returned, the reception is judged as normal and processing continues. If the slave device does not return an $\overline{\text{ACK}}$ signal, the master device outputs either a stop condition or a restart condition and then stops the current transmission. Failure to return an $\overline{\text{ACK}}$ signal may be caused by the following two factors.

(a) Reception was not performed normally.
(b) The final data was received.

When the receiving device sets the SDA0 line to low level during the ninth clock, the $\overline{\text{ACK}}$ signal becomes active (normal receive response).

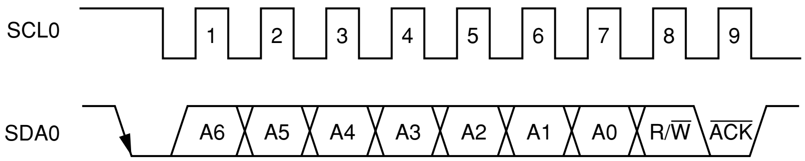
When bit 2 (ACKE0) of IIC control register 0 (IICC0) is set to 1, automatic $\overline{\text{ACK}}$ signal generation is enabled.

Transmission of the eighth bit following the 7 address data bits causes bit 3 (TRC0) of the IIC status register 0 (IICS0) to be set. When this TRC0 bit's value is "0", it indicates receive mode. Therefore, ACKE0 should be set to 1.

When the slave device is receiving (when TRC0 = 0), if the slave device does not need to receive any more data after receiving several bytes, setting ACKE0 to 0 will prevent the master device from starting transmission of the subsequent data.

Similarly, when the master device is receiving (when TRC0 = 0) and the subsequent data is not needed and when either a restart condition or a stop condition should therefore be output, setting ACKE0 to 0 will prevent the $\overline{\text{ACK}}$ signal from being returned. This prevents the MSB data from being output via the SDA0 line (i.e., stops transmission) during transmission from the slave device.

Figure 17-12. $\overline{\text{ACK}}$ Signal



When the local address is received, an $\overline{\text{ACK}}$ signal is automatically output in synchronization with the falling edge of the SCL0's eighth clock regardless of the ACKE0 value. No $\overline{\text{ACK}}$ signal is output if the received address is not a local address.

The $\overline{\text{ACK}}$ signal output method during data reception is based on the wait timing setting, as described below.

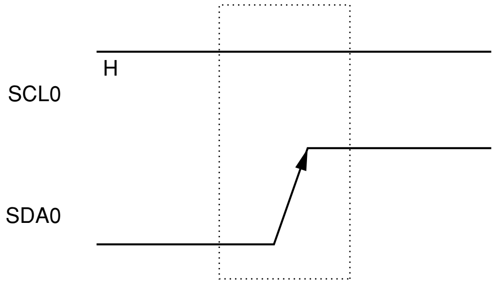
- When 8-clock wait is selected: $\overline{\text{ACK}}$ signal is output when ACKE0 is set to "1" before wait cancellation.
- When 9-clock wait is selected: $\overline{\text{ACK}}$ signal is automatically output at the falling edge of the SCL0's eighth clock if ACKE0 has already been set to "1".

17.5.5 Stop condition

When the SCL0 pin is at high level, changing the SDA0 pin from low level to high level generates a stop condition.

A stop condition is a signal that the master device outputs to the slave device when serial transfer has been completed. The slave device includes hardware that detects stop conditions.

Figure 17-13. Stop Condition



A stop condition is generated when bit 0 (SPT0) of IIC control register 0 (IICC0) is set (1). When the stop condition is detected, bit 0 (SPD0) of IIC status register 0 (IICS0) is set (1) and INTIIC0 is generated when bit 4 (SPIE0) of IICC0 is set (1).

17.5.6 Wait signal ($\overline{\text{WAIT}}$)

The wait signal ($\overline{\text{WAIT}}$) is used to notify the communication partner that a device (master or slave) is preparing to transmit or receive data (i.e., is in a wait state).

Setting the SCL0 pin to low level notifies the communication partner of the wait status. When the wait status has been canceled for both the master and slave devices, the next data transfer can begin.

Figure 17-14. Wait Signal (1/2)

(1) When master device has a nine-clock wait and slave device has an eight-clock wait (master transmits, slave receives, and ACKE0 = 1)

★

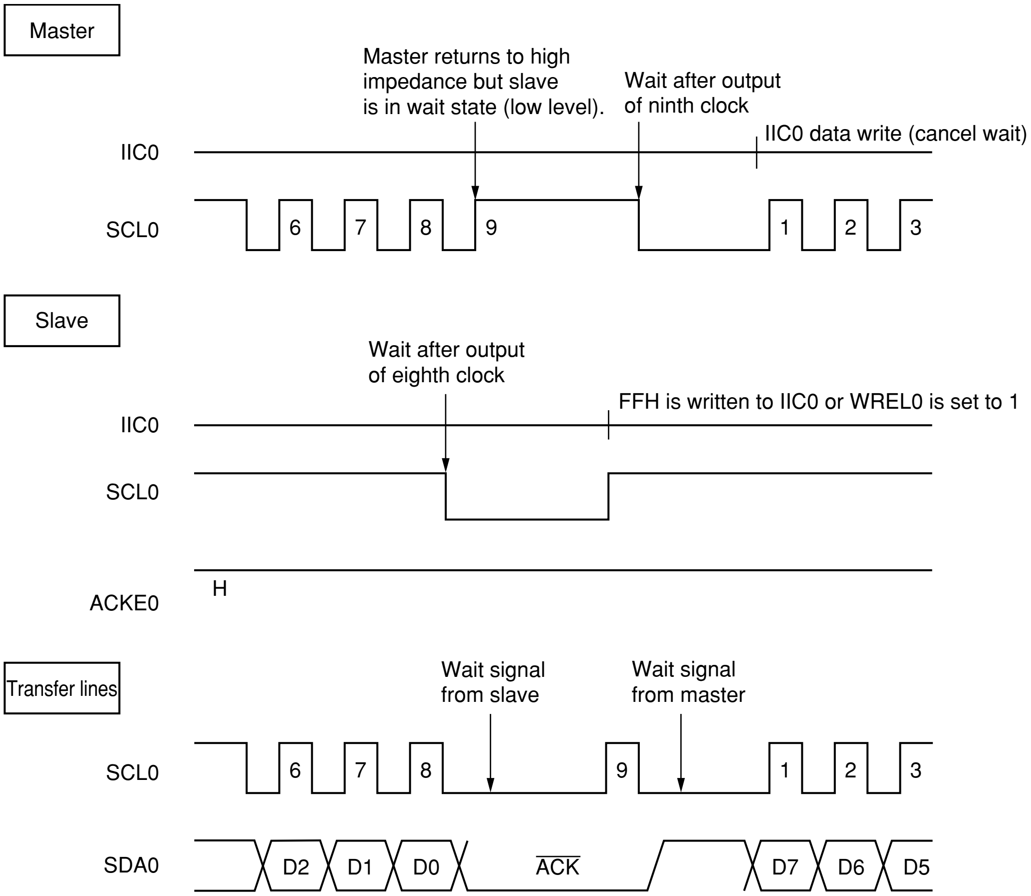
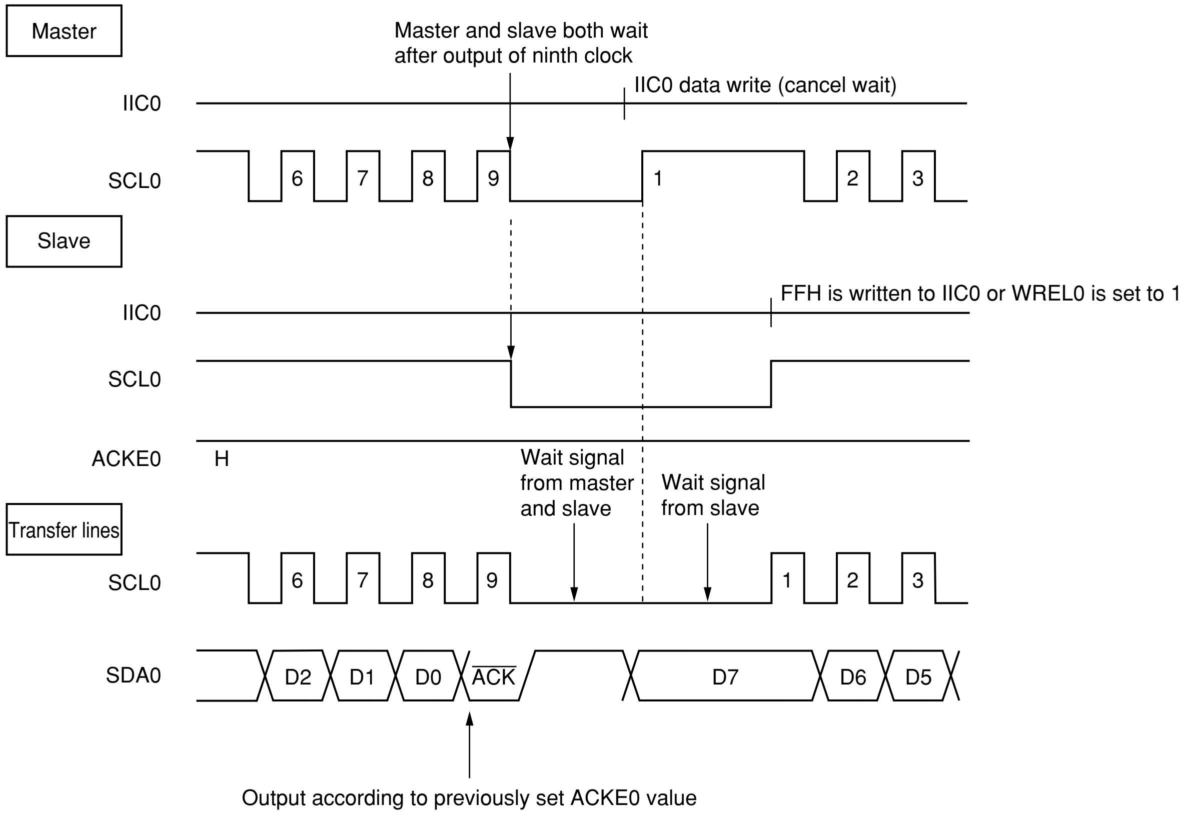

Master
Master returns to high impedance but slave is in wait state (low level).
Wait after output of ninth clock
IIC0 data write (cancel wait)
IIC0
SCL0
6
7
8
9
1
2
3
Slave
Wait after output of eighth clock
FFH is written to IIC0 or WREL0 is set to 1
IIC0
SCL0
ACKE0
H
Transfer lines
Wait signal from slave
Wait signal from master
SCL0
6
7
8
9
1
2
3
SDA0
D2
D1
D0
$\overline{\text{ACK}}$
D7
D6
D5

Figure 17-14. Wait Signal (2/2)

(2) When master and slave devices both have a nine-clock wait (master transmits, slave receives, and ACKE0 = 1)

★


Master
Master and slave both wait after output of ninth clock
IIC0 data write (cancel wait)
IIC0
SCL0
6
7
8
9
1
2
3
Slave
FFH is written to IIC0 or WREL0 is set to 1
IIC0
SCL0
ACKE0
H
Wait signal from master and slave
Wait signal from slave
Transfer lines
SCL0
6
7
8
9
1
2
3
SDA0
D2
D1
D0
$\overline{\text{ACK}}$
D7
D6
D5
Output according to previously set ACKE0 value

Remark ACKE0: Bit 2 of IIC control register 0 (IICC0)
WREL0: Bit 5 of IIC control register 0 (IICC0)

A wait may be automatically generated depending on the setting of bit 3 (WTIM0) of IIC control register 0 (IICC0).

Normally, when bit 5 (WREL0) of IICC0 is set to 1 or when FFH is written to IIC shift register 0 (IIC0), the wait status is canceled and the transmitting side writes data to IIC0 to cancel the wait status.

The master device can also cancel the wait status via either of the following methods.

- By setting bit 1 (STT0) of IICC0 to 1
- By setting bit 0 (SPT0) of IICC0 to 1

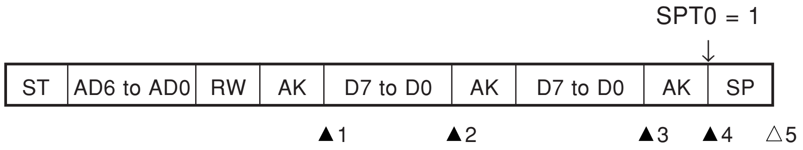
17.5.7 I^{2}C interrupt requests (INTIIC0)

The INTIIC0 interrupt request timing and IIC status register 0 (IICS0) settings corresponding to that timing are described below.

(1) Master device operation

(a) Start ~ Address ~ Data ~ Data ~ Stop (normal transmission/reception)

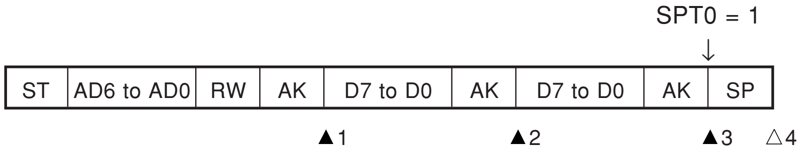
(i) When WTIM0 = 0



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B
▲3: IICS0 = 1000×000B (Sets WTIM0)
▲4: IICS0 = 1000××00B (Sets SPT0)
△5: IICS0 = 00000001B

Remark ▲ : Always generated
△ : Generated only when SPIE0 = 1
× : Don’t care

(ii) When WTIM0 = 1

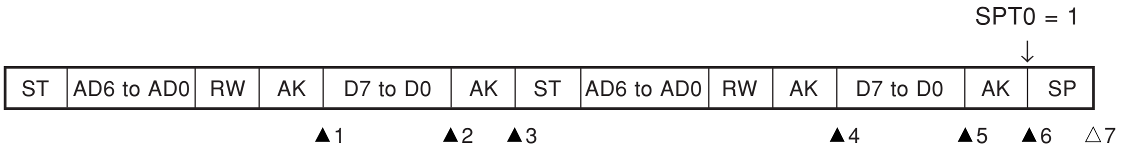


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×100B
▲3: IICS0 = 1000××00B (Sets SPT0)
△4: IICS0 = 00000001B

Remark ▲ : Always generated
△ : Generated only when SPIE0 = 1
× : Don’t care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

(i) When WTIM0 = 0



▲1: IICS0 = 1000×110B

▲2: IICS0 = 1000×000B (Sets WTIM0)

▲3: IICS0 = 1000××00B (Clears WTIM0, sets STT0)

▲4: IICS0 = 1000×110B

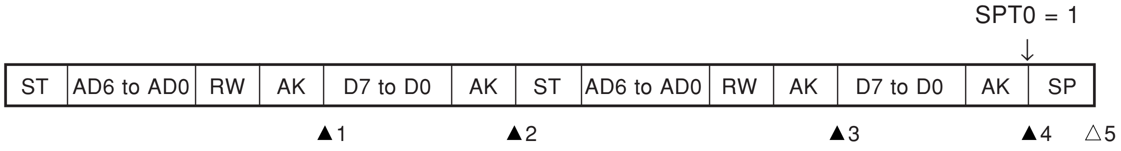
▲5: IICS0 = 1000×000B (Sets WTIM0)

▲6: IICS0 = 1000××00B (Sets SPT0)

△7: IICS0 = 00000001B

Remark ▲ : Always generated
△ : Generated only when SPIE0 = 1
× : Don’t care

(ii) When WTIM0 = 1



▲1: IICS0 = 1000×110B

▲2: IICS0 = 1000××00B (Sets STT0)

▲3: IICS0 = 1000×110B

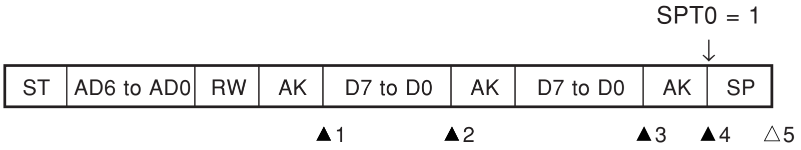
▲4: IICS0 = 1000××00B (Sets SPT0)

△5: IICS0 = 00000001B

Remark ▲ : Always generated
△ : Generated only when SPIE0 = 1
× : Don’t care

(c) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

(i) When WTIM0 = 0



▲1: IICS0 = 1010×110B

▲2: IICS0 = 1010×000B

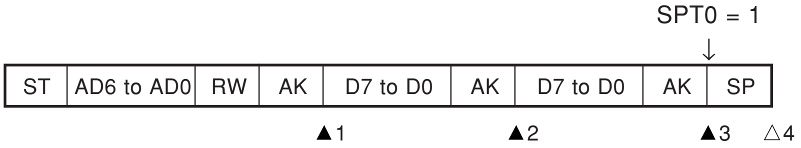
▲3: IICS0 = 1010×000B (Sets WTIM0)

★ ▲4: IICS0 = 1010××00B (Sets SPT0)

△5: IICS0 = 00000001B

Remark ▲ : Always generated
△ : Generated only when SPIE0 = 1
× : Don’t care

(ii) When WTIM0 = 1



▲1: IICS0 = 1010×110B

▲2: IICS0 = 1010×100B

▲3: IICS0 = 1010××00B (Sets SPT0)

△4: IICS0 = 00001001B

Remark ▲ : Always generated
△ : Generated only when SPIE0 = 1
× : Don’t care

(2) Slave device operation (when receiving slave address data (matches with SVA0))

(a) Start ~ Address ~ Data ~ Data ~ Stop

(i) When WTIM0 = 0

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
			▲1	▲2		▲3		△4

▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0001×000B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

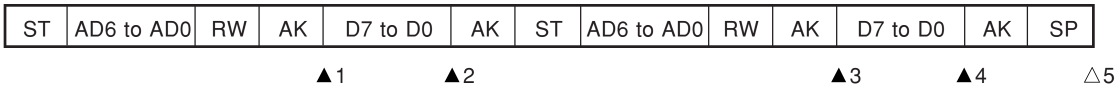
ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
			▲1		▲2		▲3	△4

▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×100B
▲3: IICS0 = 0001××00B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIM0 = 0 (after restart, matches with SVA0)



▲1: IICS0 = 0001×110B

▲2: IICS0 = 0001×000B

▲3: IICS0 = 0001×110B

▲4: IICS0 = 0001×000B

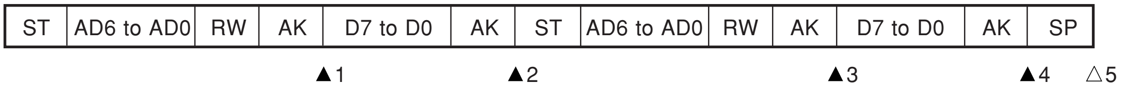
△5: IICS0 = 00000001B

Remark ▲ : Always generated

△ : Generated only when SPIE0 = 1

× : Don’t care

(ii) When WTIM0 = 1 (after restart, matches with SVA0)



▲1: IICS0 = 0001×110B

▲2: IICS0 = 0001××00B

▲3: IICS0 = 0001×110B

▲4: IICS0 = 0001××00B

△5: IICS0 = 00000001B

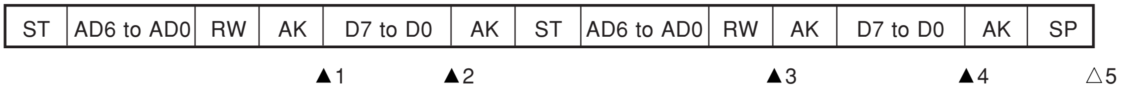
Remark ▲ : Always generated

△ : Generated only when SPIE0 = 1

× : Don’t care

(c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

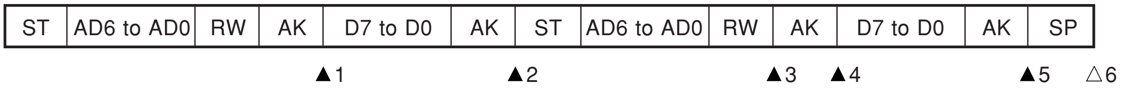
(i) When WTIM0 = 0 (after restart, extension code reception)



▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0010×010B
▲4: IICS0 = 0010×000B
△5: IICS0 = 00000001B

Remark ▲ : Always generated
△ : Generated only when SPIE0 = 1
× : Don't care

(ii) When WTIM0 = 1 (after restart, extension code reception)

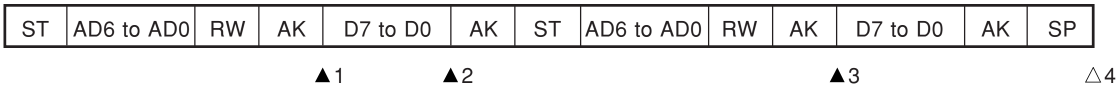


▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001××00B
▲3: IICS0 = 0010×010B
▲4: IICS0 = 0010×110B
▲5: IICS0 = 0010××00B
△6: IICS0 = 00000001B

Remark ▲ : Always generated
△ : Generated only when SPIE0 = 1
× : Don't care

(d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

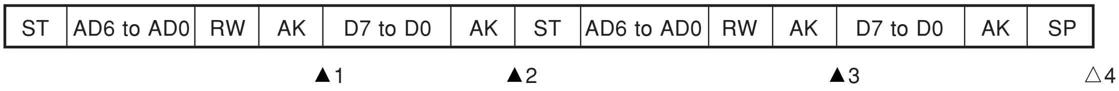
(i) When WTIM0 = 0 (after restart, does not match with address (= not extension code))



▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 00000×10B
△4: IICS0 = 00000001B

Remark ▲ : Always generated
△ : Generated only when SPIE0 = 1
× : Don't care

(ii) When WTIM0 = 1 (after restart, does not match with address (= not extension code))



▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001××00B
▲3: IICS0 = 00000×10B
△4: IICS0 = 00000001B

Remark ▲ : Always generated
△ : Generated only when SPIE0 = 1
× : Don't care

(3) Slave device operation (when receiving extension code)

(a) Start ~ Code ~ Data ~ Data ~ Stop

(i) When WTIM0 = 0

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
			▲1		▲2		▲3	△4

▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 0010×000B
△4: IICS0 = 00000001B

Remark ▲ : Always generated
△ : Generated only when SPIE0 = 1
× : Don't care

(ii) When WTIM0 = 1

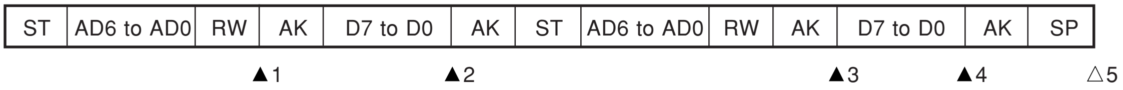
ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
			▲1	▲2		▲3		▲4 △5

▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010×100B
▲4: IICS0 = 0010××00B
△5: IICS0 = 00000001B

Remark ▲ : Always generated
△ : Generated only when SPIE0 = 1
× : Don't care

(b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

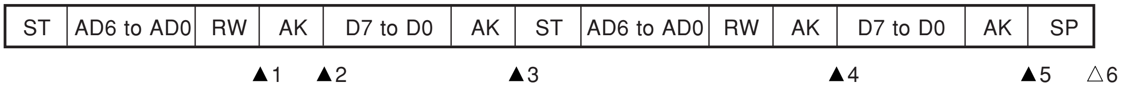
(i) When WTIM0 = 0 (after restart, matches with SVA0)



▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 0001×110B
▲4: IICS0 = 0001×000B
△5: IICS0 = 00000001B

Remark ▲ : Always generated
△ : Generated only when SPIE0 = 1
× : Don't care

(ii) When WTIM0 = 1 (after restart, matches with SVA0)

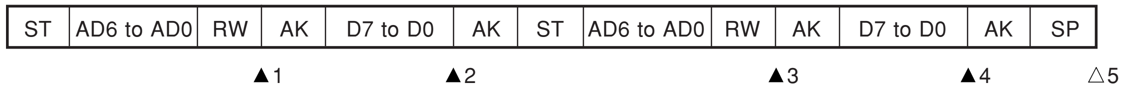


▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010××00B
▲4: IICS0 = 0001×110B
▲5: IICS0 = 0001××00B
△6: IICS0 = 00000001B

Remark ▲ : Always generated
△ : Generated only when SPIE0 = 1
× : Don't care

(c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

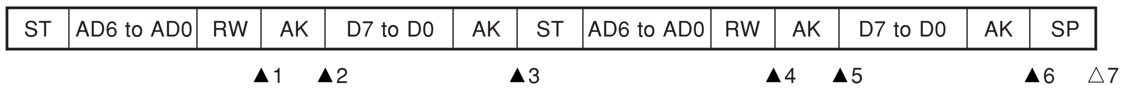
(i) When WTIM0 = 0 (after restart, extension code reception)



▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 0010×010B
▲4: IICS0 = 0010×000B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, extension code reception)



▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010××00B
▲4: IICS0 = 0010×010B
▲5: IICS0 = 0010×110B
▲6: IICS0 = 0010××00B
△7: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIM0 = 0 (after restart, does not match with address (= not extension code))

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
			▲1		▲2				▲3			△4

▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 00000×10B
△4: IICS0 = 00000001B

Remark ▲ : Always generated
△ : Generated only when SPIE0 = 1
× : Don't care

★ #### (ii) When WTIM0 = 1 (after restart, does not match with address (= not extension code))

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
			▲1	▲2		▲3						△4

▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010××00B
△4: IICS0 = 00000001B

Remark ▲ : Always generated
△ : Generated only when SPIE0 = 1
× : Don't care

(4) Operation without communication

(a) Start ~ Code ~ Data ~ Data ~ Stop

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
								△1

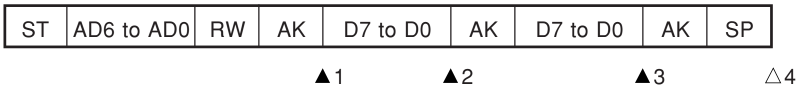
△1: IICS0 = 00000001B

Remark △ : Generated only when SPIE0 = 1

(5) Arbitration loss operation (operation as slave after arbitration loss)

(a) When arbitration loss occurs during transmission of slave address data

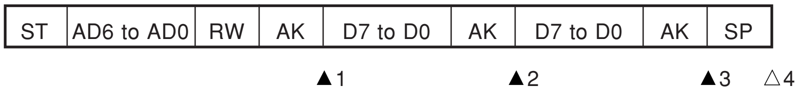
(i) When WTIM0 = 0



▲1: IICS0 = 0101×110B (**Example** When ALD0 is read during interrupt servicing)
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0001×000B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



▲1: IICS0 = 0101×110B (**Example** When ALD0 is read during interrupt servicing)
▲2: IICS0 = 0001×100B
▲3: IICS0 = 0001××00B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(b) When arbitration loss occurs during transmission of extension code

(i) When WTIM0 = 0

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP	
			▲1		▲2		▲3		△4

▲1: IICS0 = 0110×010B (**Example** When ALD0 is read during interrupt servicing)
▲2: IICS0 = 0010×000B
▲3: IICS0 = 0010×000B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(ii) When WTIM0 = 1

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP	
			▲1	▲2		▲3		▲4	△5

▲1: IICS0 = 0110×010B (**Example** When ALD0 is read during interrupt servicing)
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010×100B
▲4: IICS0 = 0010××00B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(6) Operation when arbitration loss occurs (no communication after arbitration loss)

(a) When arbitration loss occurs during transmission of slave address data (when WTIM0 = 1)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP	
				▲1					△2

▲1: IICS0 = 01000110B (**Example** When ALD0 is read during interrupt servicing)

△2: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1

(b) When arbitration loss occurs during transmission of extension code

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP	
			▲1						△2

▲1: IICS0 = 0110×010B (**Example** When ALD0 is read during interrupt servicing)
LREL0 is set to 1 by software

△2: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(c) When arbitration loss occurs during data transfer

(i) When WTIM0 = 0

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
			▲1	▲2				△3

▲1: IICS0 = 10001110B

▲2: IICS0 = 01000000B (**Example** When ALD0 is read during interrupt servicing)

△3: IICS0 = 00000001B

Remark ▲: Always generated

△: Generated only when SPIE0 = 1

(ii) When WTIM0 = 1

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
			▲1		▲2			△3

▲1: IICS0 = 10001110B

▲2: IICS0 = 01000100B (**Example** When ALD0 is read during interrupt servicing)

△3: IICS0 = 00000001B

Remark ▲: Always generated

△: Generated only when SPIE0 = 1

(d) When loss occurs due to restart condition during data transfer

(i) Not extension code (Example: unmatches with SVA0, WTIM0 = 1)

ST	AD6 to AD0	RW	AK	D7 to Dn	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
			▲1					▲2			△3

▲1: IICS0 = 1000×110B

▲2: IICS0 = 01000110B (**Example** When ALD0 is read during interrupt servicing)

△3: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care
n = 6 to 0

(ii) Extension code

ST	AD6 to AD0	RW	AK	D7 to Dn	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
			▲1					▲2			△3

▲1: IICS0 = 1000×110B

▲2: IICS0 = 0110×010B (**Example** When ALD0 is read during interrupt servicing)

LREL0 is set to 1 by software

△3: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care
n = 6 to 0

(e) When loss occurs due to stop condition during data transfer

ST	AD6 to AD0	RW	AK	D7 to Dn	SP	
				▲1		△2

▲1: IICS0 = 1000×110B
△2: IICS0 = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care
n = 6 to 0

(f) When arbitration loss occurs due to low-level data when attempting to generate a restart condition

(i) When WTIM0 = 1

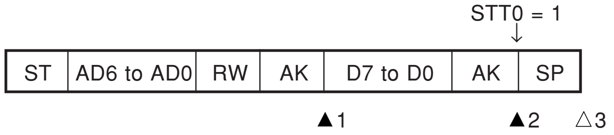
						STT0 = 1 ↓						
ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	D7 to D0	AK	SP		
				▲1		▲2		▲3			△4	

▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×100B (Sets STT0)
▲3: IICS0 = 01000100B (**Example** When ALD0 is read during interrupt servicing)
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(g) **When arbitration loss occurs due to a stop condition when attempting to generate a restart condition**

(i) **When WTIM0 = 1**



▲1: IICS0 = 1000×110B

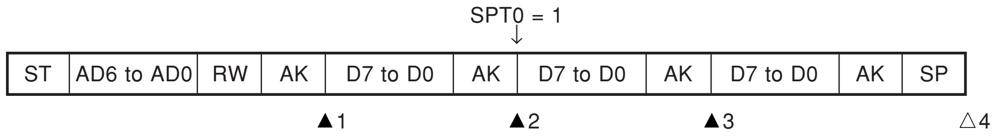
▲2: IICS0 = 1000××00B (Sets STT0)

△3: IICS0 = 01000001B

Remark ▲ : Always generated
△ : Generated only when SPIE0 = 1
× : Don’t care

(h) **When arbitration loss occurs due to low-level data when attempting to generate a stop condition**

(i) **When WTIM0 = 1**



▲1: IICS0 = 1000×110B

▲2: IICS0 = 1000××00B (Sets SPT0)

▲3: IICS0 = 01000000B (**Example** When ALD0 is read during interrupt servicing)

△4: IICS0 = 00000001B

Remark ▲ : Always generated
△ : Generated only when SPIE0 = 1
× : Don’t care

17.5.8 Interrupt request (INTIIC0) generation timing and wait control

The setting of bit 3 (WTIM0) of IIC control register 0 (IICC0) determines the timing by which INTIIC0 is generated and the corresponding wait control, as shown in Table 17-2.

Table 17-2. INTIIC0 Timing and Wait Control

WTIM	During Slave Device Operation			During Master Device Operation		
	Address	Data Reception	Data Transmission	Address	Data Reception	Data Transmission
0	9[Notes 1, 2]	8[Note 2]	8[Note 2]	9	8	8
1	9[Notes 1, 2]	9[Note 2]	9[Note 2]	9	9	9

Notes 1. The slave device's INTIIC0 signal and wait period occurs at the falling edge of the ninth clock only when there is a match with the address set to slave address register 0 (SVA0).
At this point, $\overline{\text{ACK}}$ is output regardless of the value set to IICC0's bit 2 (ACKE0). For a slave device that has received an extension code, INTIIC0 occurs at the falling edge of the eighth clock.
★ However, if an address mismatch is detected after a restart, INTIIC0 occurs at the falling edge of the ninth clock, but a wait period is not generated.

★ **2.** If the received address does not match the contents of slave address register 0 (SVA0) and an extension code has not received, neither INTIIC0 nor a wait occurs.

Remark The numbers in the table indicate the number of the serial clock's clock signals. Interrupt requests and wait control are both synchronized with the falling edge of these clock signals.

(1) During address transmission/reception

- ★ Slave device operation: According to the above **Notes 1** and **2**, the interrupt and wait timing are determined regardless of the WTIM0 bit.
- Master device operation: The interrupt and wait timing occur at the falling edge of the ninth clock regardless of the WTIM0 bit.

(2) During data reception

- Master/slave device operation: The interrupt and wait timing are determined according to the WTIM0 bit.

(3) During data transmission

- Master/slave device operation: The interrupt and wait timing are determined according to the WTIM0 bit.

(4) Wait cancellation method

The four wait cancellation methods are as follows.

- By setting bit 5 (WREL0) of IIC control register 0 (IICC0) to 1
- By writing to IIC shift register 0 (IIC0)
- By setting a start condition (setting bit 1 (STT0) of IICC0 to 1)[Note]
- By setting a stop condition (setting bit 0 (SPT0) of IICC0 to 1)[Note]

★ **Note** Master only

When 8-clock wait has been selected (WTIM0 = 0), the output level of $\overline{\text{ACK}}$ must be determined prior to wait cancellation.

(5) Stop condition detection

INTIIC0 is generated when a stop condition is detected.

17.5.9 Address match detection method

When in I²C bus mode, the master device can select a particular slave device by transmitting the corresponding slave address.

An address match can be detected automatically by hardware. An interrupt request (INTIIC0) occurs when a local address has been set to slave address register 0 (SVA0) and when the address set to SVA0 matches the slave address transmitted by the master device, or when an extension code has been received.

17.5.10 Error detection

In I²C bus mode, the status of the serial data bus (SDA0) during data transmission is captured by IIC shift register 0 (IIC0) of the transmitting device, so the IIC0 data prior to transmission can be compared with the transmitted IIC0 data to enable detection of transmission errors. A transmission error is judged as having occurred when the compared data values do not match.

17.5.11 Extension code

(1) When the higher 4 bits of the receive address are either “0000” or “1111”, the extension code reception flag (EXC0) is set for extension code reception and an interrupt request (INTIIC0) is issued at the falling edge of the eighth clock. The local address stored in slave address register 0 (SVA0) is not affected.

(2) If “111110××” is set to SVA0 by a 10-bit address transfer and “111110××” is transferred from the master device, the results are as follows. Note that INTIIC0 occurs at the falling edge of the eighth clock.

- Higher four bits of data match: EXC0 = 1**Note**
- Seven bits of data match: COI0 = 1**Note**

Note EXC0: Bit 5 of IIC status register 0 (IICS0)
COI0: Bit 4 of IIC status register 0 (IICS0)

(3) Since the processing after the interrupt request occurs differs according to the data that follows the extension code, such processing is performed by software.
For example, after the extension code is received, if you do not wish to operate the target device as a slave device, you can set bit 6 (LREL0) of IIC control register 0 (IICC0) to 1 to set the standby mode for the next communication operation.

Table 17-3. Extension Code Bit Definitions

Slave Address	R/W Bit	Description
0000 000	0	General call address
0000 000	1	Start byte
0000 001	×	CBUS address
0000 010	×	Address that is reserved for different bus format
1111 0××	×	10-bit slave address specification

17.5.12 Arbitration

When several master devices simultaneously output a start condition (when STT0 is set to 1 before STD0 is set to 1[Note]), communication among the master devices is performed as the number of clocks are adjusted until the data differs. This kind of operation is called arbitration.

When one of the master devices loses in arbitration, an arbitration loss flag (ALD0) in IIC status register 0 (IICS0) is set (1) at the timing at which the arbitration loss occurred, and the SCL0 and SDA0 lines are both set to high impedance, which releases the bus.

The arbitration loss is detected based on the timing of the next interrupt request (the eighth or ninth clock, when a stop condition is detected, etc.) and the ALD0 = 1 setting that has been made by software.

For details of interrupt request timing, see **17.5.7 I^{2}C interrupt requests (INTIIC0)**.

Note STD0: Bit 1 of IIC status register 0 (IICS0)
STT0: Bit 1 of IIC control register 0 (IICC0)

Figure 17-15. Arbitration Timing Example

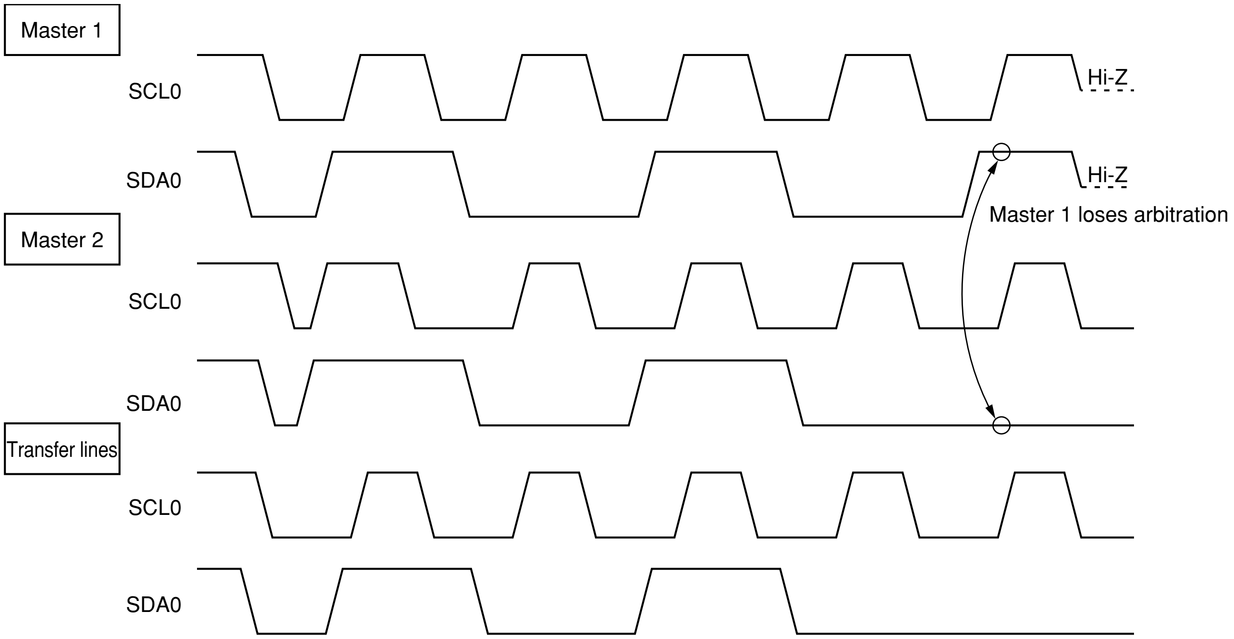


Table 17-4. Status During Arbitration and Interrupt Request Generation Timing

Status During Arbitration	Interrupt Request Generation Timing
During address transmission	At falling edge of eighth or ninth clock following byte transfer[Note 1]
Read/write data after address transmission	
During extension code transmission	
Read/write data after extension code transmission	
During data transmission	
During $\overline{\text{ACK}}$ signal transfer period after data transmission	
When restart condition is detected during data transfer	
When stop condition is detected during data transfer	When stop condition is output (when SPIE0 = 1)[Note 2]
When data is at low level while attempting to output a restart condition	At falling edge of eighth or ninth clock following byte transfer[Note 1]
When stop condition is detected while attempting to output a restart condition	When stop condition is output (when SPIE0 = 1)[Note 2]
When data is at low level while attempting to output a stop condition	At falling edge of eighth or ninth clock following byte transfer[Note 1]
When SCL0 is at low level while attempting to output a restart condition	

Notes 1. When WTIM0 (bit 3 of IIC control register 0 (IICC0)) = 1, an interrupt request occurs at the falling edge of the ninth clock. When WTIM0 = 0 and the extension code's slave address is received, an interrupt request occurs at the falling edge of the eighth clock.

2. When there is a chance that arbitration will occur, set SPIE0 = 1 for master device operation.

★ **Remark** SPIE0: Bit 4 of IIC control register 0 (IICC0)

17.5.13 Wakeup function

The I^{2}C bus slave function is a function that generates an interrupt request (INTIIC0) when a local address and extension code have been received.

This function makes processing more efficient by preventing unnecessary interrupt requests from occurring when addresses do not match.

When a start condition is detected, wakeup standby mode is set. This wakeup standby mode is in effect while addresses are transmitted due to the possibility that an arbitration loss may change the master device (which has output a start condition) to a slave device.

★ However, when a stop condition is detected, bit 4 (SPIE0) of IIC control register 0 (IICC0) is set regardless of the wakeup function, and this determines whether interrupt requests are enabled or disabled.

17.5.14 Communication reservation

To start master device communications when not currently using a bus, a communication reservation can be made to enable transmission of a start condition when the bus is released. There are two modes under which the bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled ($\overline{\text{ACK}}$ is not returned and the bus was released when bit 6 (LREL0) of IIC control register 0 (IICC0) was set to 1).

If bit 1 (STT0) of IICC0 is set (1) while the bus is not being used, a start condition is automatically generated and the wait status is set after the bus is released (when a stop condition is detected).

When the bus release is detected (when a stop condition is detected), writing to IIC shift register 0 (IIC0) causes the master's address transfer to start. At this point, IICC0's bit 4 (SPIE0) should be set (1).

When STT0 has been set (1), the operation mode (as start condition or as communication reservation) is determined according to the bus status.

- If the bus has been released .... A start condition is generated
- If the bus has not been released (standby mode) .... Communication reservation

Check whether the communication reservation operates or not using MSTS0 (bit 7 of IIC status register 0 (IICS0)) after STT0 is set and the wait time elapses.

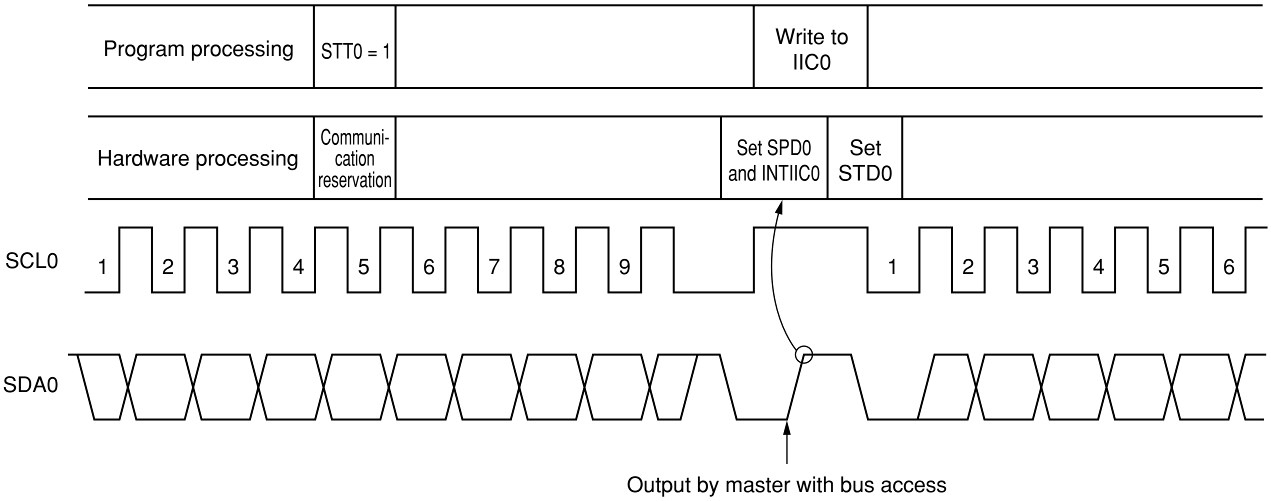
Wait periods, which should be set via software, are listed in Table 17-5. These wait periods can be set via the settings of bits 3 and 0 (SMC0 and CL00) of IIC transfer clock select register 0 (IICCL0).

Table 17-5. Wait Periods

SMC0	CL00	Wait Period
0	0	26 clocks
0	1	46 clocks
1	0	16 clocks
1	1	

Figure 17-16 shows communication reservation timing.

Figure 17-16. Communication Reservation Timing



Remark IIC0: IIC shift register 0
STT0: Bit 1 of IIC control register 0 (IICC0)
STD0: Bit 1 of IIC status register 0 (IICS0)
SPD0: Bit 0 of IIC status register 0 (IICS0)

Communication reservations are accepted at the following timing. After bit 1 (STD0) of IIC status register 0 (IICS0) is set to 1, a communication reservation can be made by setting bit 1 (STT0) of IIC control register 0 (IICC0) to 1 before a stop condition is detected.

Figure 17-17. Timing for Accepting Communication Reservations

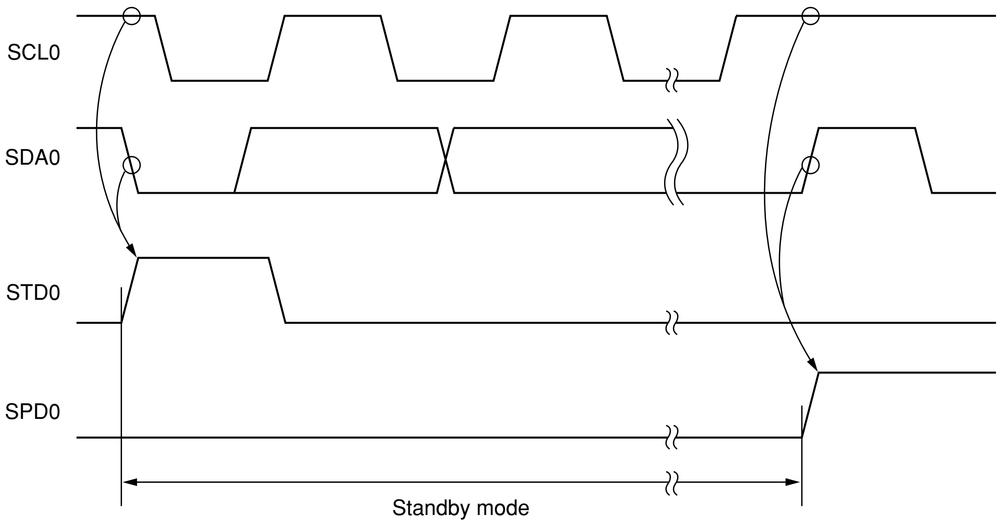
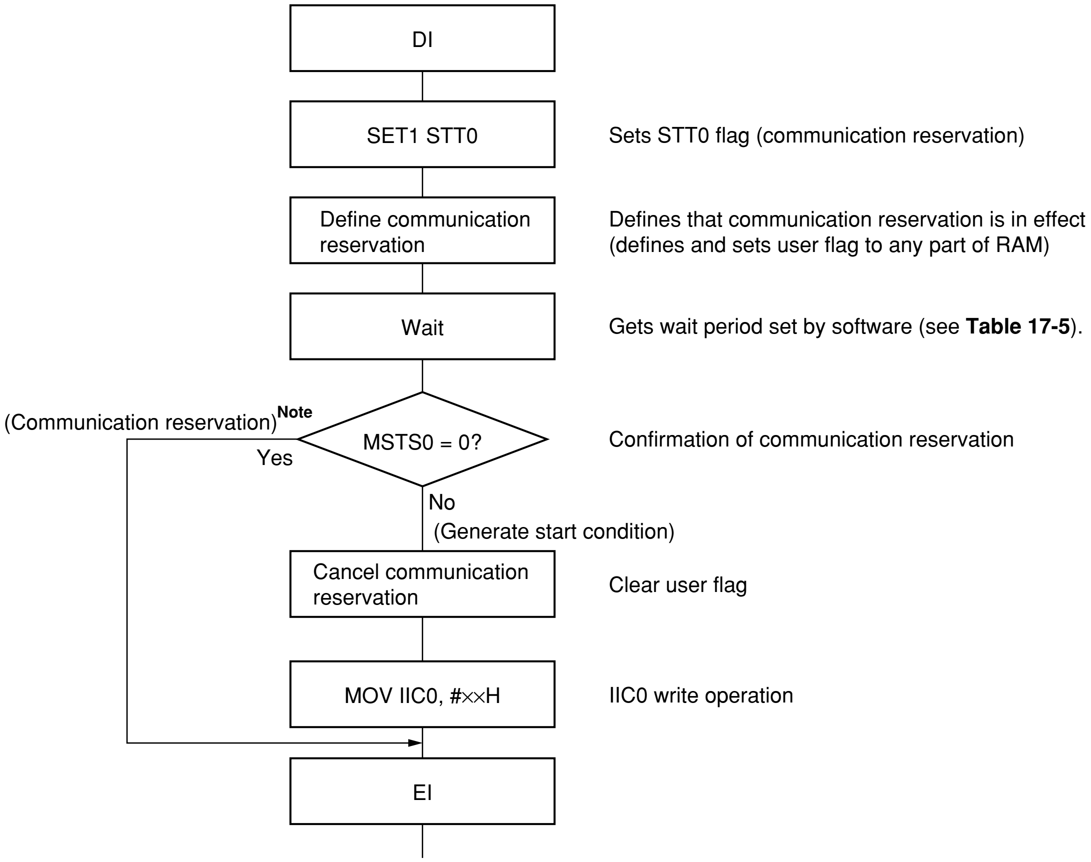


Figure 17-18 shows the communication reservation protocol.

Figure 17-18. Communication Reservation Protocol


DI
SET1 STT0
Sets STT0 flag (communication reservation)
Define communication reservation
Defines that communication reservation is in effect (defines and sets user flag to any part of RAM)
Wait
Gets wait period set by software (see **Table 17-5**).
(Communication reservation)Note
MSTS0 = 0?
Yes
Confirmation of communication reservation
No
(Generate start condition)
Cancel communication reservation
Clear user flag
MOV IIC0, #××H
IIC0 write operation
EI

Note The communication reservation operation executes a write to IIC shift register 0 (IIC0) when a stop condition interrupt request occurs.

Remark STT0: Bit 1 of IIC control register 0 (IICC0)
MSTS0: Bit 7 of IIC status register 0 (IICS0)
IIC0: IIC shift register 0

17.5.15 Other cautions

After a reset, when changing from a mode in which no stop condition has been detected (the bus has not been released) to a master device communication mode, first generate a stop condition to release the bus, then perform master device communication.

When using multiple masters, it is not possible to perform master device communication when the bus has not been released (when a stop condition has not been detected).

Use the following sequence for generating a stop condition.

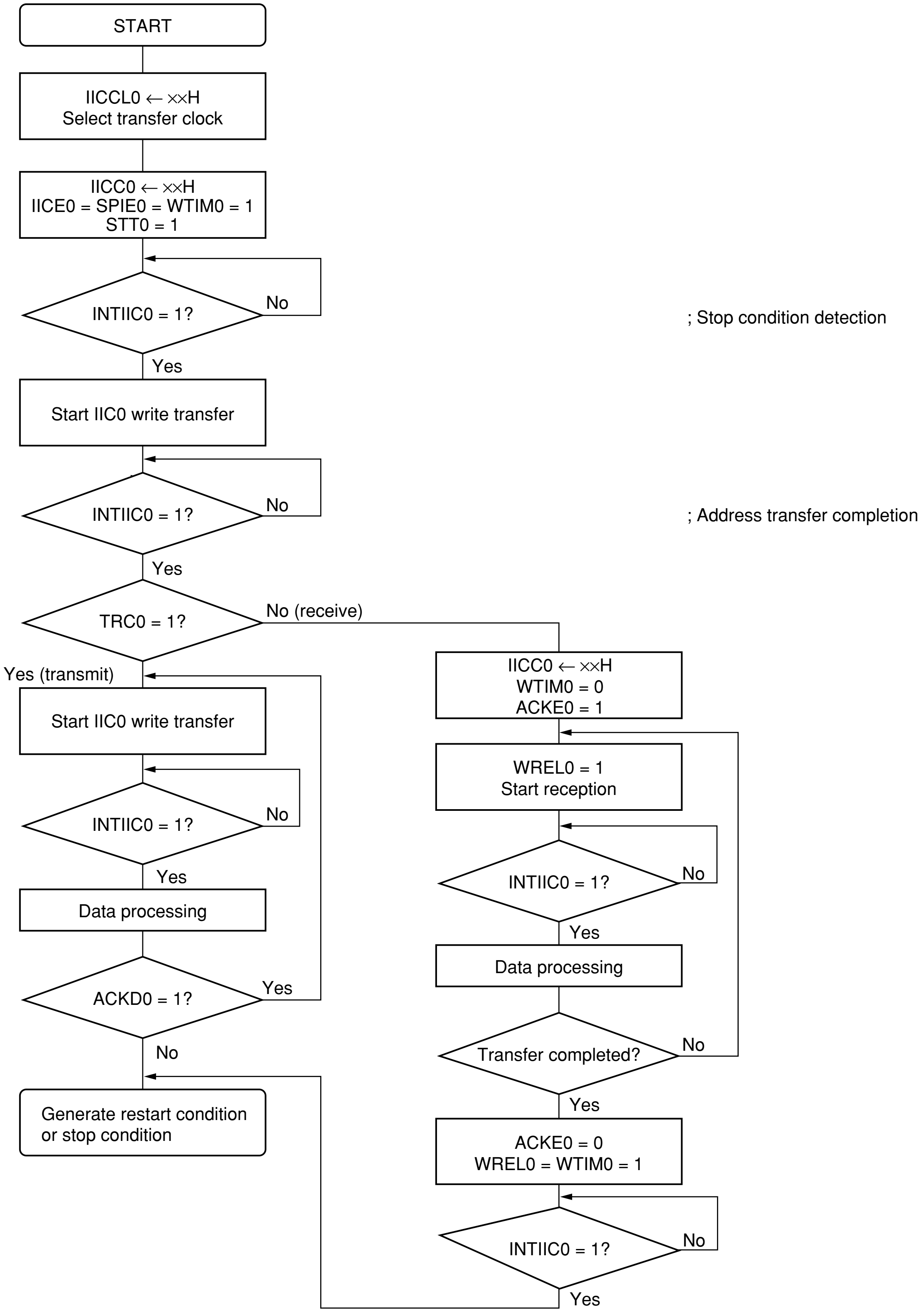
(a) Set IIC transfer clock select register 0 (IICCL0).
(b) Set bit 7 (IICE0) of IIC control register 0 (IICC0).
(c) Set bit 0 (SPT0) of IICC0.

17.5.16 Communication operations

(1) Master operation

An example of master operation is shown below.

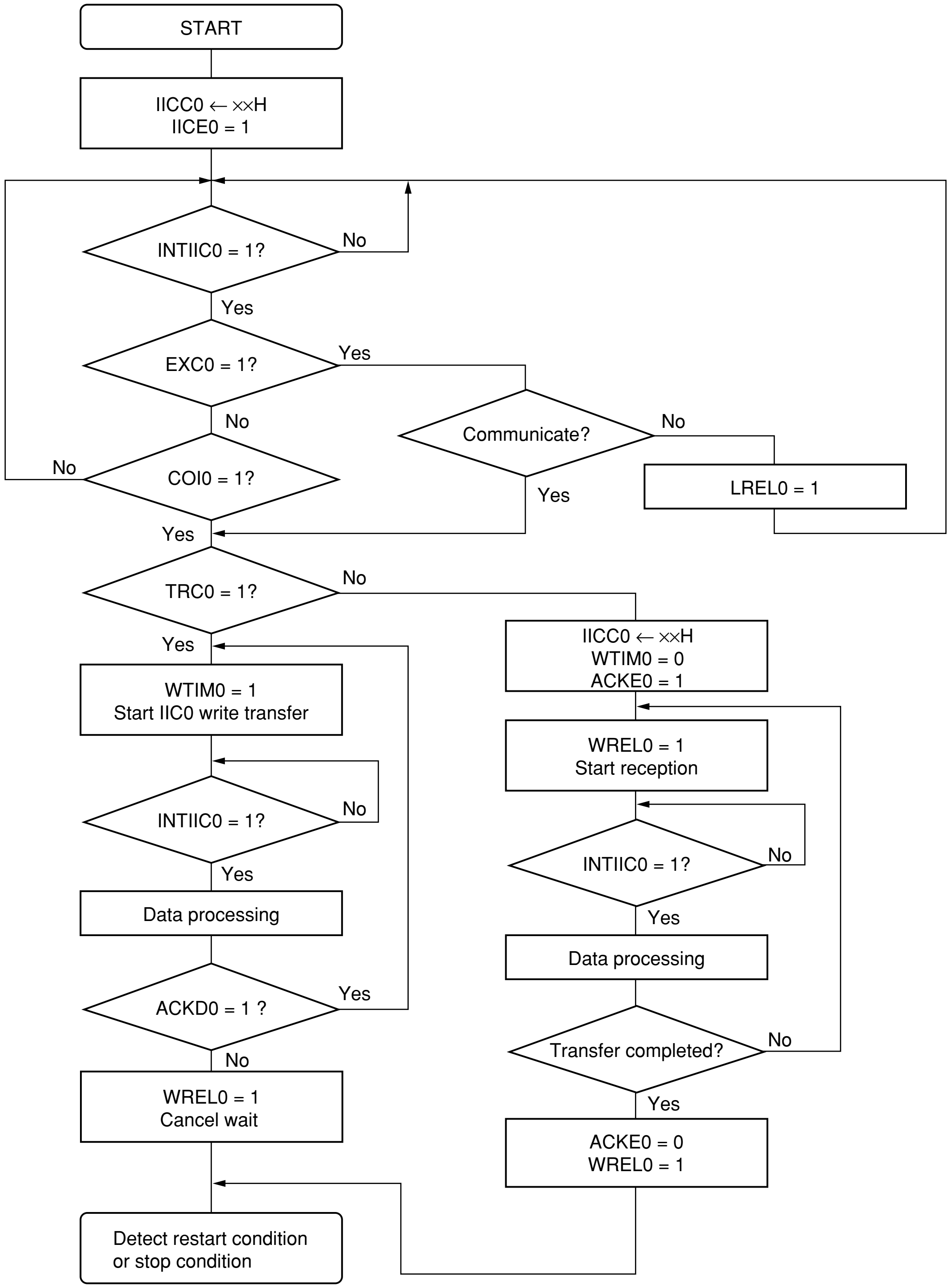
Figure 17-19. Master Operation Flowchart



(2) Slave operation

An example of slave operation is shown below.

Figure 17-20. Slave Operation Flowchart



17.6 Timing Charts

When using the I^2C bus mode, the master device outputs an address via the serial bus to select one of several slave devices as its communication partner.

After outputting the slave address, the master device transmits the TRC0 bit (bit 3 of IIC status register 0 (IICS0)) that specifies the data transfer direction and then starts serial communication with the slave device.

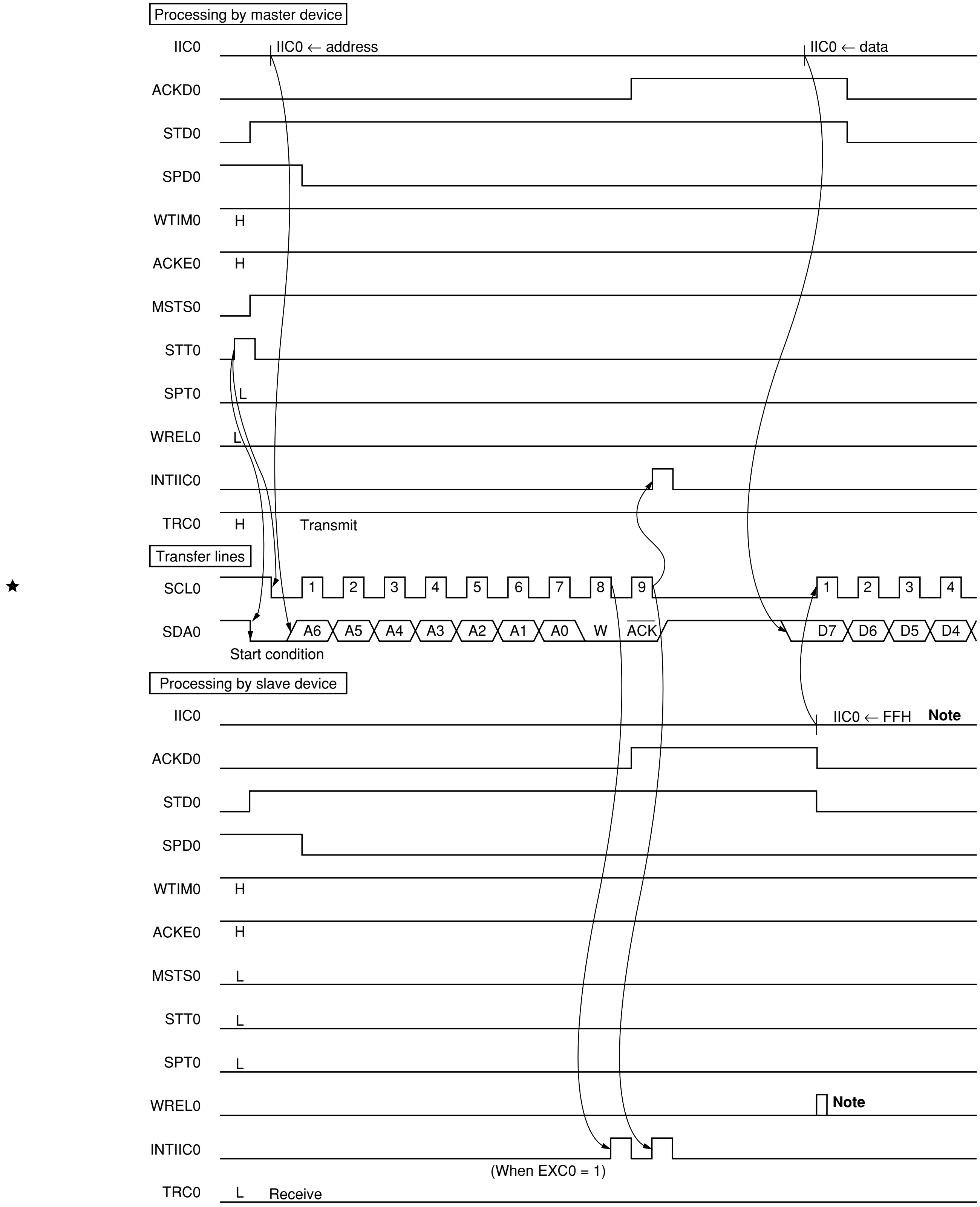
Figures 17-21 and 17-22 show timing charts of the data communication.

IIC shift register 0 (IIC0)'s shift operation is synchronized with the falling edge of the serial clock (SCL0). The transmit data is transferred to the SO0 latch and is output (MSB first) via the SDA0 pin.

Data input via the SDA0 pin is captured into IIC0 at the rising edge of SCL0.

Figure 17-21. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (1/3)

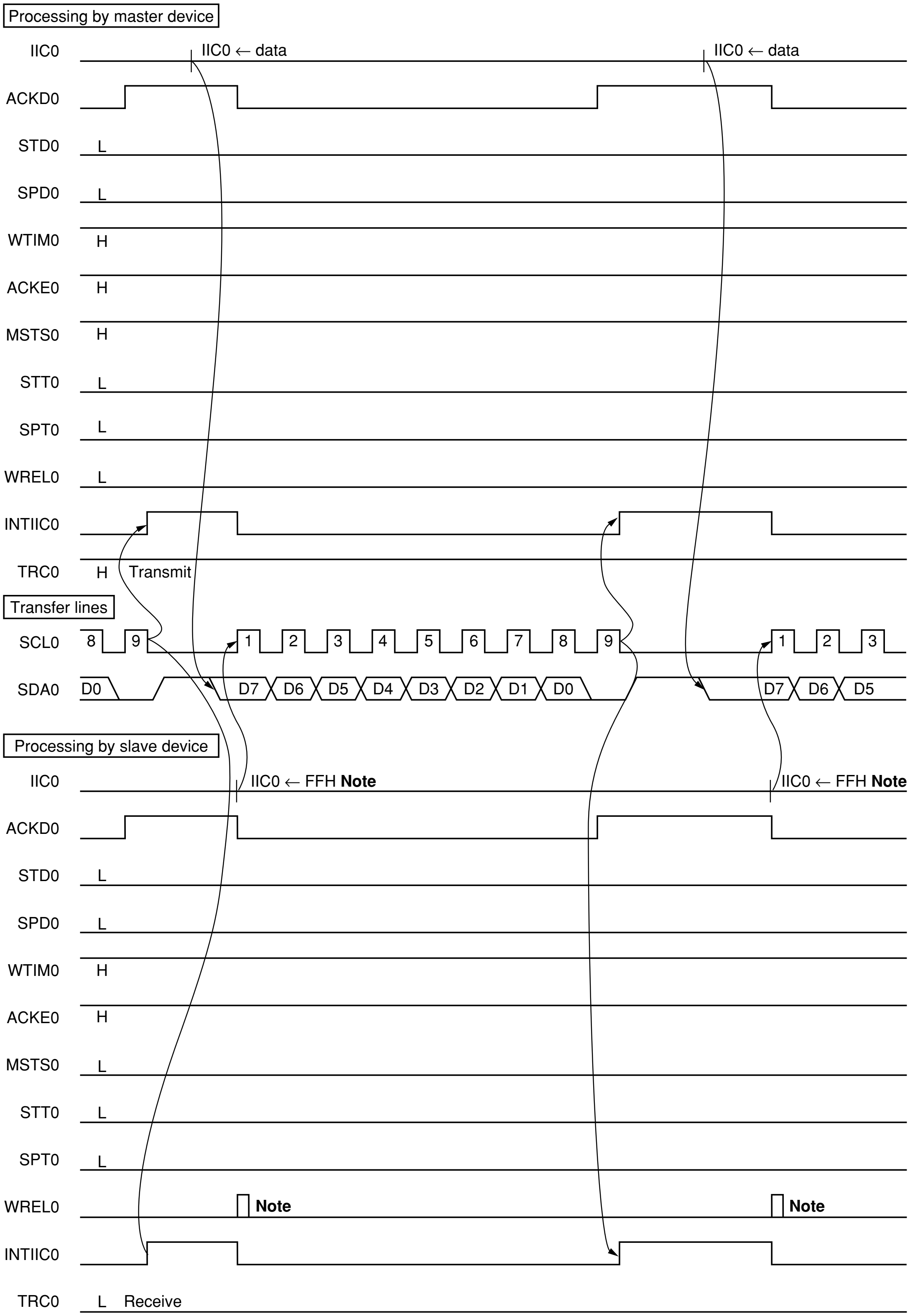
(1) Start condition ~ address



Note To cancel slave wait, write FFH to IIC0 or set WREL0.

Figure 17-21. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (2/3)

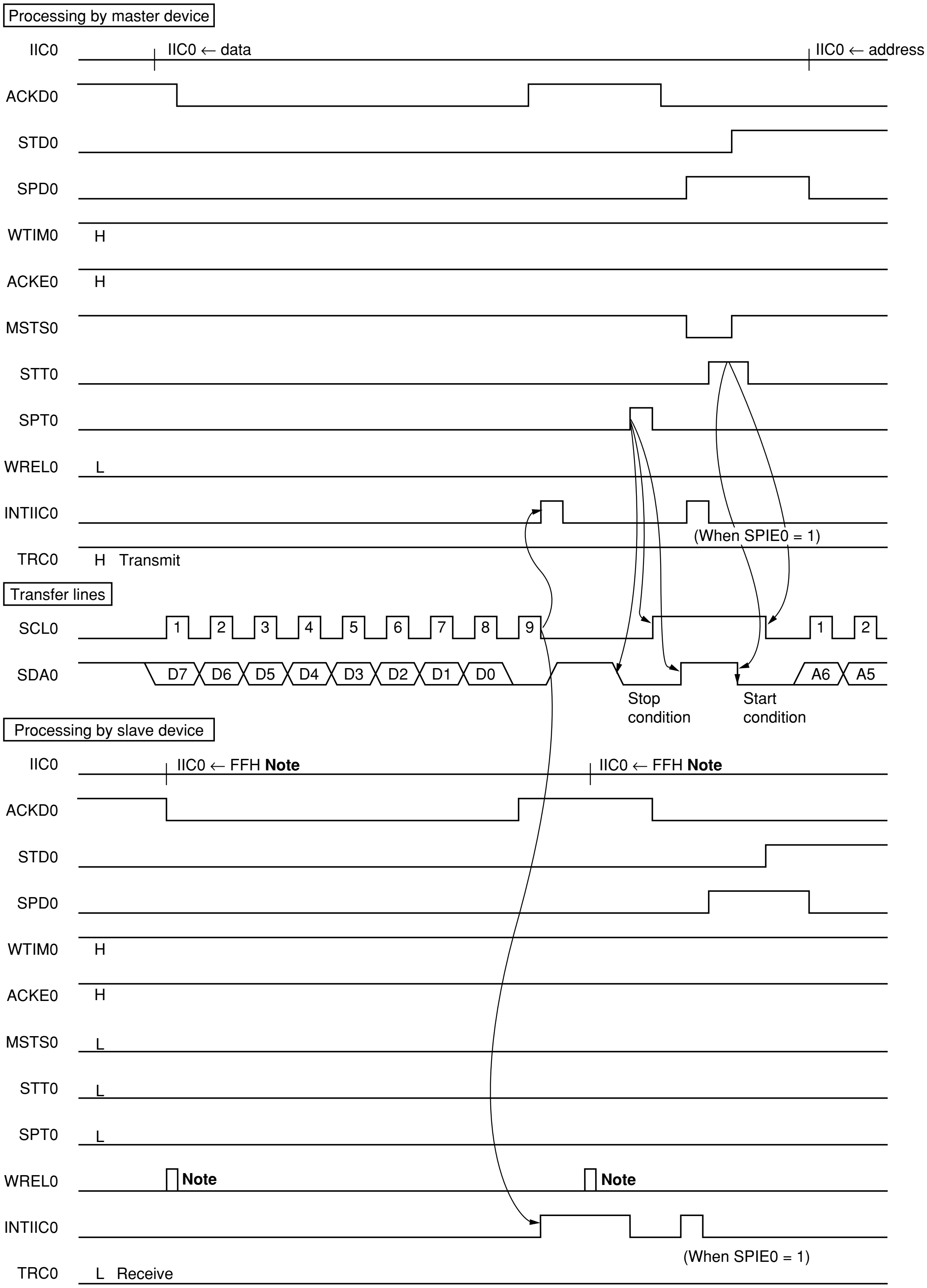
(2) Data



Note To cancel slave wait, write FFH to IIC0 or set WREL0.

Figure 17-21. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (3/3)

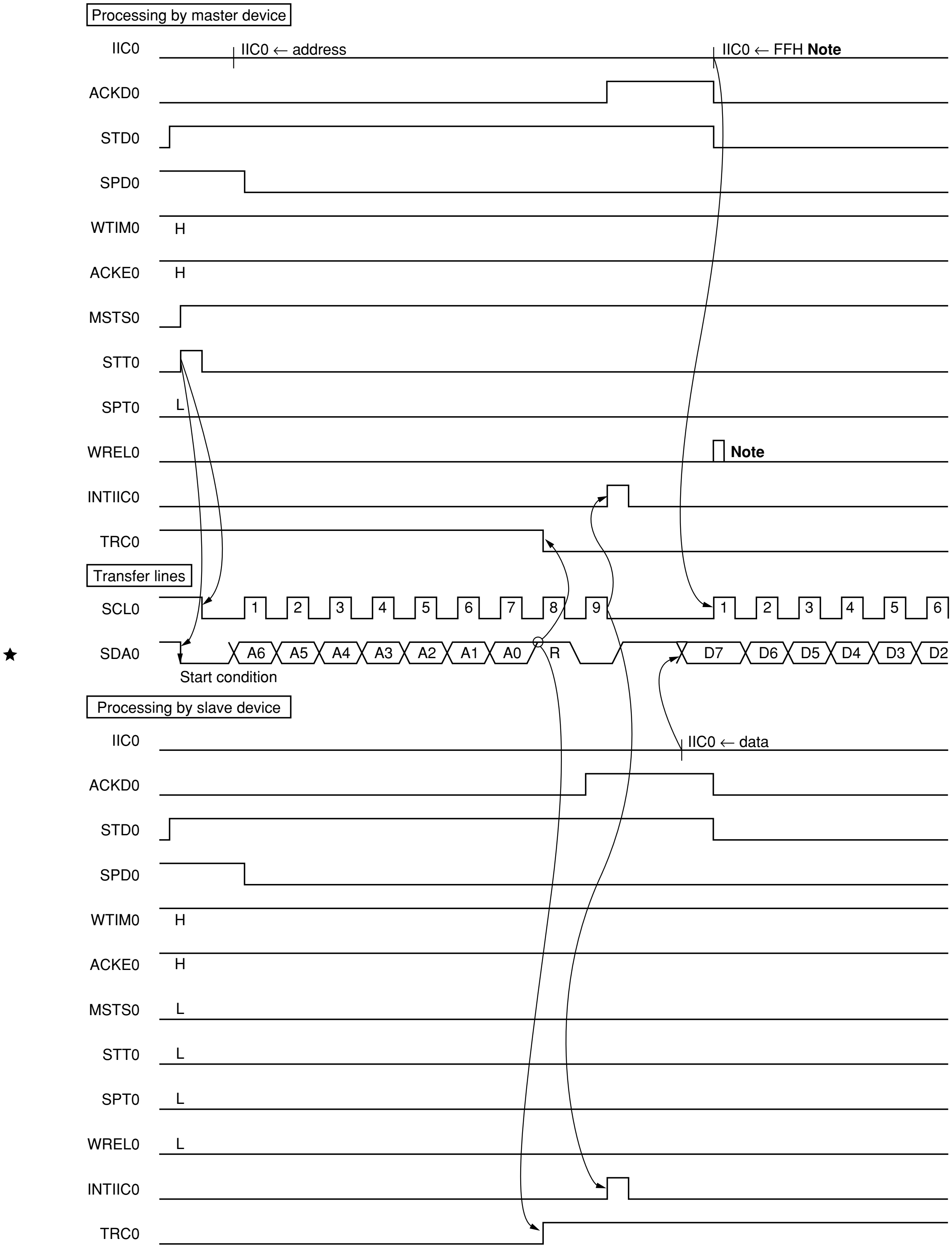
(3) Stop condition



Note To cancel slave wait, write FFH to IIC0 or set WREL0.

Figure 17-22. Example of Slave to Master Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (1/3)

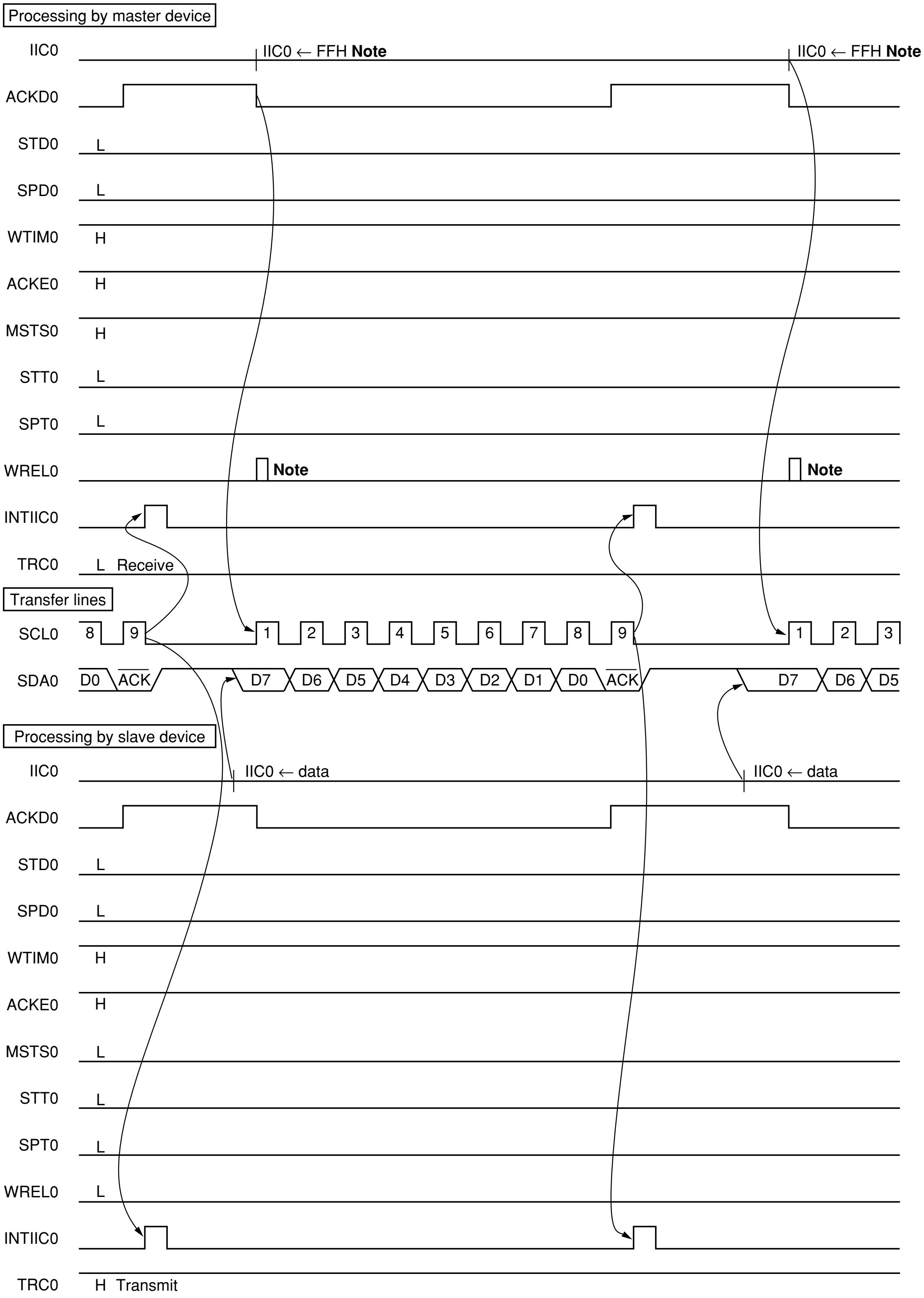
(1) Start condition ~ address



★ **Note** To cancel master wait, write FFH to IIC0 or set WREL0.

Figure 17-22. Example of Slave to Master Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (2/3)

(2) Data

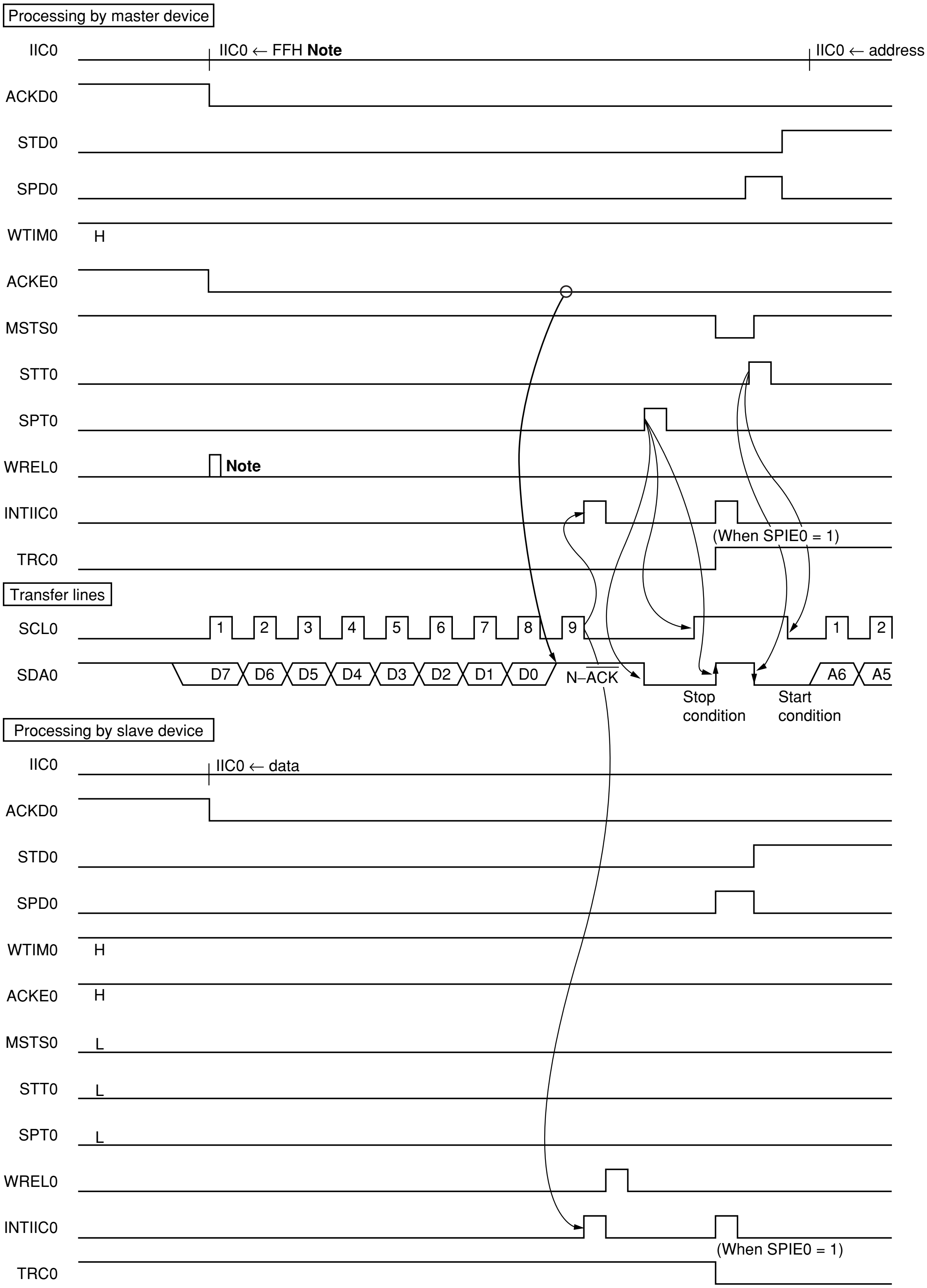


Note To cancel master wait, write FFH to IIC0 or set WREL0.

Figure 17-22. Example of Slave to Master Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (3/3)

(3) Stop condition

★



★ **Note** To cancel master wait, write FFH to IIC0 or set WREL0.

CHAPTER 18 LCD CONTROLLER/DRIVER

18.1 LCD Controller/Driver Functions

The internal LCD controller/driver of the μPD780344, 780354, 780344Y, and 780354Y Subseries has the following functions.

(1) Automatic output of segment signals and common signals by automatically reading display data memory

(2) Internal booster circuit employed for LCD driver reference voltage generator (×3 only).
Therefore, the LCD can be stably displayed even if the supply voltage drops due to a drop in the battery voltage.
In addition, two types of LCD driver reference voltages can be selected by using LCD gain adjust register 0.

(3) Three display modes selectable
- Static (up to 12 lines)
- 1/3 duty (1/3 bias)
- 1/4 duty (1/3 bias)

(4) Four types of frame frequencies selectable in each display mode

(5) Simultaneous driving of static display (up to 12 segments) and dynamic display. The operation mode of the alternate-function pins (S0 to S11) can be switched between the static display mode and dynamic display mode in 4-bit units.

(6) Blinking of LCD (only when subsystem clock is used).
Whether each segment blinks or not can be selected.
The blinking cycle can be selected from 0.5 s or 1.0 s.

(7) Operation with subsystem clock

(8) Operating voltage range: 1.8 to 5.5 V

Table 18-1. Segment Signals and Common Signals

Maximum Number of Segment Signals	Common Signals
40 lines (S0 to S39), of which 12 (S0 to S11) are selectable for static display, and 28 (S12 to S39) are also used with I/O port lines (P80 to P87, P90 to P97, P100 to P107, and P110 to P113)[Note].	Dynamic display: COM0 to COM3 Static display: SCOM0

Note The operation mode of the alternate-function pins can be switched between the port mode and segment signal mode in 1-bit units by using the pin function switching registers (PF8 to PF11).

Table 18-2 shows the maximum number of pixels that can be displayed in each display mode.

Table 18-2. Maximum Number of Pixels Displayed

Bias Mode	Time Division	Common Signals	Maximum Number of Pixels
—	Static	SCOM0	12 (12 segments × 1 common)
1/3	3	COM0 to COM2	120 (40 segments × 3 commons)
	4	COM0 to COM3	160 (40 segments × 4 commons)

18.2 LCD Controller/Driver Configuration

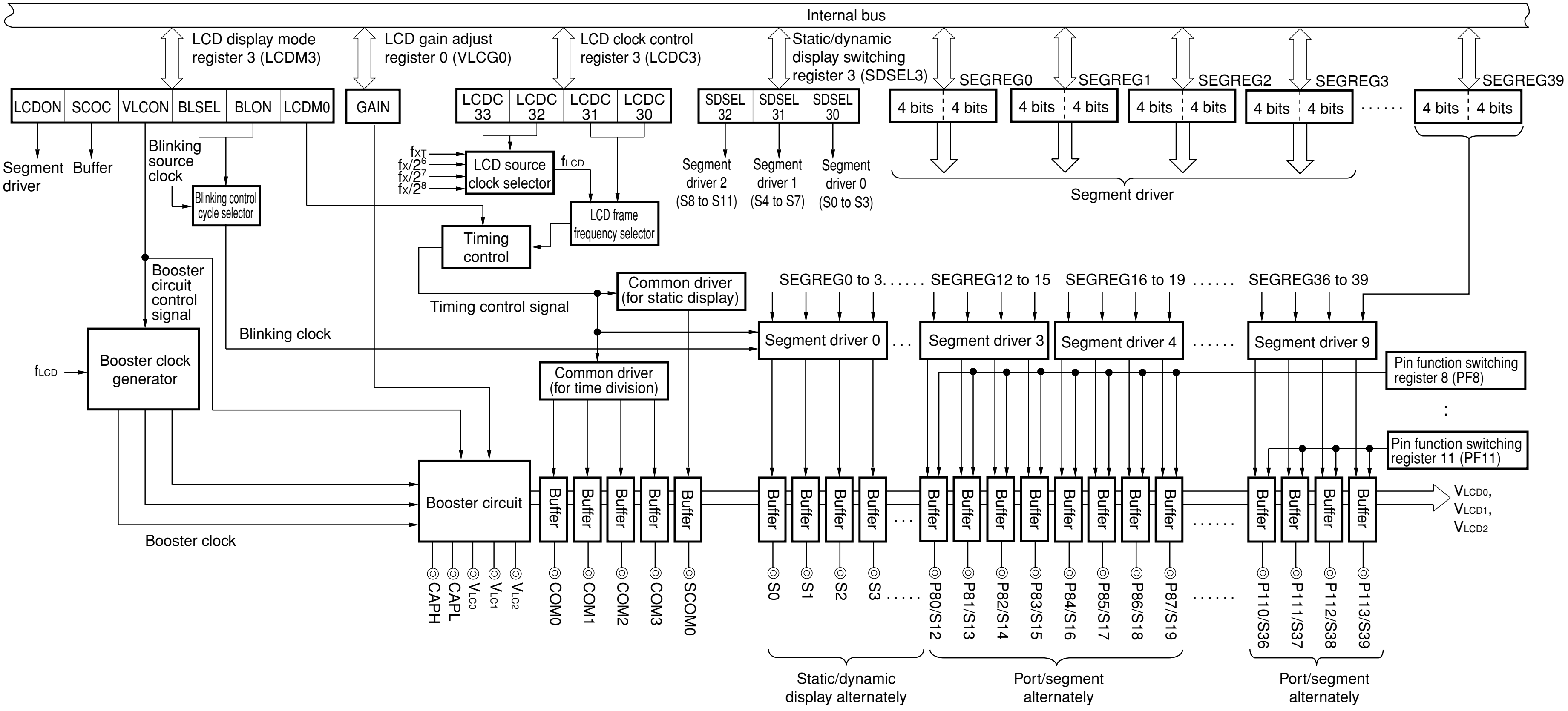
The LCD controller/driver consists of the following hardware.

Table 18-3. LCD Controller/Driver Configuration

Item	Configuration
Display output	Segment signal: 40 lines Dynamic/static alternated: 12 lines Segment/output port: 28 lines Common signal: 4 lines (for dynamic display) 1 line (for static display)
Control registers	LCD display mode register 3 (LCDM3) LCD clock control register 3 (LCDC3) LCD gain adjust register 0 (VLCG0) Static/dynamic display switching register 3 (SDSEL3) Pin function switching registers (PF8 to PF11)

★

Figure 18-1. LCD Controller/Driver Block Diagram



18.3 Registers to Control LCD Controller/Driver

The LCD controller/driver can be controlled by using the following five types of registers.

- LCD display mode register 3 (LCDM3)
- LCD clock control register 3 (LCDC3)
- LCD gain adjust register 0 (VLCG0)
- Static/dynamic display switching register 3 (SDSEL3)
- Pin function switching registers (PF8 to PF11)

(1) LCD display mode register 3 (LCDM3)

This register enables or disables display, controls the booster circuit and blinking display, and selects a display mode.

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

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

Figure 18-2. Format of LCD Display Mode Register 3 (LCDM3)

Address: FF90H After reset: 00H R/W

Symbol	$\boxed{7}$	$\boxed{6}$	$\boxed{5}$	$\boxed{4}$	$\boxed{3}$	2	1	0
LCDM3	LCDON	SCOC	VLCON	BLSEL	BLON	0	0	LCDM0

LCDON	Display control (enables output of display data)
0	Display OFF (All segment output pins output unselect signals.)
1	Display ON

SCOC	Output control of segment/common pins
0	Output GND level to segment/common pins.
1	Output select signal to segment/common pins.

VLCON	Booster circuit control
0	Stop booster circuit.
1	Operate booster circuit

BLSEL$^{\text{Note 1}}$	Blinking clock cycle selection
0	Blinking cycle of 0.5 s
1	Blinking cycle of 1.0 s

BLON$^{\text{Note 2}}$	Blinking display control
0	Blinking display OFF
1$^{\text{Note 3}}$	Blinking display ON

LCDM0$^{\text{Note 4}}$	Dynamic/static display alternate pins$^{\text{Notes 5, 6}}$		Dynamic pin	
	Time division	Bias mode	Time division	Bias mode
0	4	1/3	4	1/3
1	3	1/3	3	1/3

Notes 1. The BLSEL bit is valid only when the subsystem clock is used.

2. The corresponding segment pin can be blinked only if the blinking data memory (higher 4 bits of FA00 to FA27H) is set to 1.

3. Do not change the contents of the blinking data memory while BLON = 1.

4. Do not change LCDM0 while the LCD is in operation. Be sure to set this bit while LCDON = 0, SCOC = 0, and VLCON = 0.

5. The dynamic/static alternate pins are in the static display mode when this mode is selected by static/dynamic display switching register 3 (SDSEL3).

6. When static display is not used, the static display common output pin (SCOM0) outputs the GND potential.

Cautions 1. Set the LCDON, SCOC, and VLCON bits in the following sequence:

- **To display LCD while LCD booster circuit is stopped**
  - **(1) Set VLCON to 1. All the segment and common pins are in the GND output mode (SCOC = 0).**

    ↓
  - **(2) Set VLCON to 1 and wait 500 ms or longer with software.**

    ↓
  - **(3) Set SCOC to 1. All the segment and common pins output an unselect waveform and are in unselect display mode.**

    ↓
  - **(4) Set LCDON to 1. The value of the display RAM is reflected on the segment output waveform, and all segment and common pins are in select display mode.**

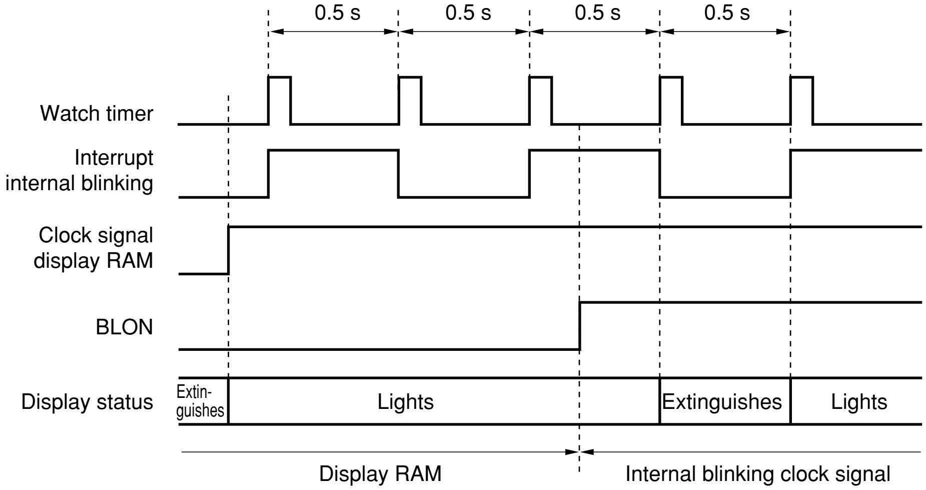
- **To stop LCD booster circuit while LCD displays**
  - **(1) Clear LCDON to 0. All the segment and common pins are in unselect display mode.**

    ↓
  - **(2) Clear SCOC to 0. All the segment and common pins are in GND output mode.**

    ↓
  - **(3) Clear VLCON to 0. The LCD booster circuit stops.**

2. The blinking cycle is generated using the interval time of the watch timer (0.5 s at 32.768 kHz). When the blinking function is not used (BLON = 0), the LCD lights or extinguishes depending on the setting of the display RAM, as shown in Figure 18-3. When the blinking function is used (BLON = 1), the LCD lights or extinguishes depending on the status of the internal blinking clock signal (set value of BLSEL), i.e., it lights if the internal blinking clock signal is "1" and extinguishes if the signal is "0".

Figure 18-3. Blinking Function



3. When using the blinking function, the LCD does not blink even if the data is rewritten while the LCD is in the extinguishing cycle (0.5 s or 1.0 s), unless the LCD moves to the lighting cycle.

(2) LCD clock control register 3 (LCDC3)

This register is used to select the LCD source clock and frame frequency.
It is set by an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets the value of this register to 00H.

Figure 18-4. Format of LCD Clock Control Register 3 (LCDC3)

Address: FF91H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LCDC3	0	0	0	0	LCDC33	LCDC32	LCDC31	LCDC30

LCDC33	LCDC32	Source clock selection (f_{LCD})
0	0	f_{XT} (32.768 kHz)
0	1	$f_X/2^6$ (156.25 kHz)
1	0	$f_X/2^7$ (78.125 kHz)
1	1	$f_X/2^8$ (39.0625 kHz)

LCDC31	LCDC30	Selection of reference clock generating frame frequency
0	0	$f_{LCD}/2^6$
0	1	$f_{LCD}/2^7$
1	0	$f_{LCD}/2^8$
1	1	$f_{LCD}/2^9$

Caution Do not rewrite LCDC3 while the LCD is operating. Be sure to set this bit while LCDON = 0, SCOC = 0, and VLCON = 0.

Remark Figures in parentheses are for operation with f_X = 10 MHz or f_{XT} = 32.768 kHz

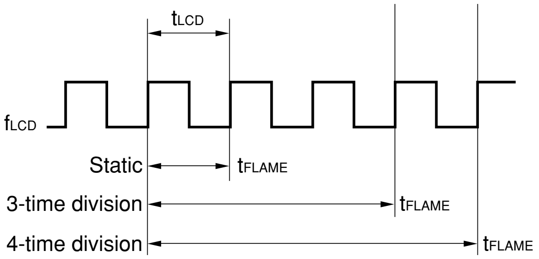
Table 18-4 shows the frame frequency if f_{XT} (32.768 kHz) is used as the source clock (f_{LCD}), and Figure 18-5 shows the relationship between the reference clock that generates the frame frequency, and the frame frequency.

Table 18-4. Frame Frequency

Frame Frequency \ Reference Clock Generating Frame Frequency		$f_{XT}/2^9$	$f_{XT}/2^8$	$f_{XT}/2^7$	$f_{XT}/2^6$
Display duty	Static	64 Hz	128 Hz	256 Hz[Note]	512 Hz[Note]
	1/3 duty	21 Hz	43 Hz	85 Hz	171 Hz[Note]
	1/4 duty	16 Hz	32 Hz	64 Hz	128 Hz

★ **Note** Set so that the frame frequency is 128 Hz or lower.

Figure 18-5. Relationship Between Reference Clock Generating Frame Frequency, and Frame Frequency



Remark f_{LCD}: Reference clock that generates frame frequency
t_{LCD}: LCD clock period
t_{FLAME}: Frame period

(3) LCD gain adjust register 0 (VLCG0)

This register controls the voltage boost level during the voltage boost operation.
VLCG0 is set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets the value of this register to 00H.

Figure 18-6. Format of LCD Gain Adjust Register 0 (VLCG0)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
VLCG0	0	0	0	0	0	0	0	GAIN	FF94H	00H	R/W

	GAIN	Reference voltage (V_{LCD2}) level selection[Note]
★	0	1.0 V (specification of the LCD panel used is 3 V)
★	1	1.5 V (specification of the LCD panel used is 4.5 V)

Note Select the settings according to the specifications of the LCD panel that is used.

★ **Caution Before changing the VLCG0 setting, be sure to stop voltage boosting (VLCON = 0).**

Remark The TYP. value is indicated as the reference voltage (V_{LCD2}) value.

(4) Static/dynamic display switching register 3 (SDSEL3)

This register is used to select the static or dynamic display mode of the segment pins (S0 to S11).

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

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

★ **Figure 18-7. Format of Static/Dynamic Display Switching Register 3 (SDSEL3)**

Address: FF92H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SDSEL3	0	0	0	0	0	SDSEL32	SDSEL31	SDSEL30

SDSEL32	SDSEL31	SDSEL30	Number of segments (for static display mode)	Number of segments (for dynamic display mode)
0	0	0	—	S0 to S39
0	0	1	S0 to S3	S4 to S39
0	1	1	S0 to S7	S8 to S39
1	1	1	S0 to S11	S12 to S39
Setting other than above is prohibited.			—	

Caution Do not rewrite SDSEL3 while the LCD is operating. Be sure to set this bit while LCDON = 0, SCOC = 0, and VLCON = 0. Note that SDSEL3 can be set only once after reset.

(5) Pin function switching registers (PF8 to PF11)

These registers are used to select whether the pins of ports 8 to 11 are used as port pins or segment pins in 1-bit units.

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

$\overline{\text{RESET}}$ input sets the values of these registers to 00H.

Figure 18-8. Format of Pin Function Switching Registers (PF8 to PF11)

Address: FF58H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PF8	PF87	PF86	PF85	PF84	PF83	PF82	PF81	PF80

Address: FF59H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PF9	PF97	PF96	PF95	PF94	PF93	PF92	PF91	PF90

Address: FF5AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PF10	PF107	PF106	PF105	PF104	PF103	PF102	PF101	PF100

Address: FF5BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PF11	0	0	0	0	PF113	PF112	PF111	PF110

	PFmn	Pin settings
★	0	I/O port
★	1	Segment output

Cautions 1. The pins specified as segment output pins by PF8 to PF11 can output their signals regardless of the value of the corresponding port mode register (PM8 to PM11).

★ **2. PF8 to PF11 can be set only once after a reset. To change the settings, a reset must be performed beforehand.**

Remark m = 8 to 11, n = 0 to 7

18.4 LCD Display RAM

The LCD display data and the LCD blinking select bits corresponding to the LCD display data are mapped to addresses FA00H to FA27H. The lower 4 bits of each of these addresses are an LCD display data area, and the higher 4 bits are an LCD blinking select bit area. The LCD blinking select bits correspond to the LCD display data (i.e., LCD blinking select bit 0 corresponds to bit 4 of the LCD display data, bit 1 to bit 5, bit 2 to bit 6, and bit 3 to bit 7).

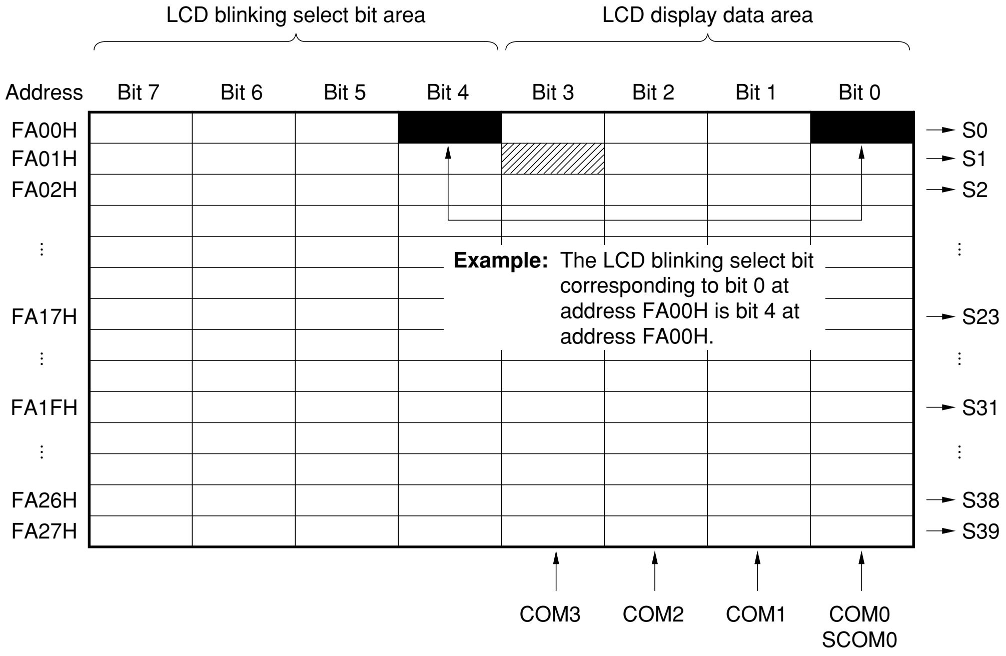
★ The data stored to the LCD display data area can be displayed on the LCD panel. For example, bit 3 at address FA01H (the shaded portion in Figure 18-9) is output to S1 pin at the COM3 timing.

The LCD blinking select bit is used to blink the corresponding segment by setting 1 to the bit to blink, and 1 to bit 3 (BLON) of LCD display mode register 3 (LCDM3). In this case, however, the display data of the corresponding segment must be 1.

Figure 18-9 shows the relationship between the LCD display data, contents of the blinking select bits, and segment/common output signals.

The area not used for display can be used as a normal RAM area.

Figure 18-9. Relationship Between LCD Display Data, Contents of Blinking Select Bits, and Segment/Common Output Signals (4-Time Division)

★


Example: The LCD blinking select bit corresponding to bit 0 at address FA00H is bit 4 at address FA00H.

Caution The higher 4 bits (LCD blinking select bit area) of each address, FA00H to FA27H, correspond to the lower 4 bits (LCD display data area). When the LCD does not blink, therefore, be sure to clear the corresponding bit in the blinking select bit area to 0.

18.5 LCD Controller/Driver Settings

Set the LCD controller/driver as follows:

(1) Specify whether P80/S12 to P87/S19, P90/S20 to P97/S27, P100/S28 toP107/S35, and P110/S36 to P113/S39 are used as segment output pins or port output pins, by using the pin function switching registers (PF8 to PF11).

(2) Specify the display mode of the segment output pins (S0 to S11) by using static/dynamic display switching register 3 (SDSEL3).

★ (3) Set the displayed default value to the LCD display data area (bits 0 to 3) of the LCD display RAM. The addresses and capacity of the LCD display RAM are FA00H to FA27H (40 bytes).
To use the blinking function, set the corresponding bit of the blinking select bit area (bits 4 to 7) in the LCD display RAM to 1.

(4) Specify the display mode using bit 0 (LCDM0) of LCD display mode register 3 (LCDM3).

(5) Select the source clock and frame frequency of the LCD using LCD clock control register 3 (LCDC3).

(6) Select the LCD reference voltage by using LCD gain adjust register 0 (VLCG0).

(7) Set bit 5 (VLCON) of LCD display mode register 3 (LCDM3) to 1 to start the operation of the booster circuit.

(8) Make sure that a wait time of 500 ms or longer elapses with software.

(9) Set bit 6 (SCOC) of LCD display mode register 3 (LCDM3) to 1 so that the unselect waveform is output to the segment pins and common pins.

(10) To use the blinking function, select a blinking cycle of 0.5 s or 1.0 s by using bit 4 (BLSEL) of LCD display mode register 3 (LCDM3).

(11) Set bit 7 (LCDON) of LCD display mode register 3 (LCDM3) to 1 to set the display to ON. To blink the LCD, set bit 3 (BLON) of LCD display mode register 3 (LCDM3) to 1 to set the display to ON.

Then, set the data of the display data memory and the timing of the blinking display according to the data to be displayed.

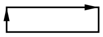
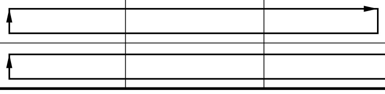
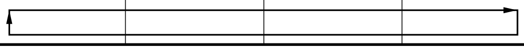
18.6 Common Signals and Segment Signals

★ An individual pixel on an LCD panel lights when the potential difference of the corresponding common signal and segment signal reaches or exceeds a given voltage (depending on the panel), and extinguishes when the potential difference drops lower than V_{LCD}.

(1) Common signals

For common signals, the selection timing order is as shown in Table 18-5 according to the number of time divisions set, and operations are repeated with these as the cycle. In the static mode, the same signal is output to SCOM0. With a 3-time-division operation, the COM3 pin is left open.

Table 18-5. COM Signals

COM Signal Time Division	COM0	COM1	COM2	COM3	SCOM0
Static	–	–	–	–	
3-time division				Open	–
4-time division					–

(2) Segment signals

Segment signals correspond to a 40-byte LCD display RAM (FA00H to FA27H[Note]). Bit 0, bit 1, bit 2, and bit 3 of each display data memory are read in synchronization with the SCOM0/COM0, COM1, COM2 and COM3 timing respectively, and if the value of the bit is 1, it is converted to the selection voltage. If the value of the bit is 0, it is converted to the non-selection voltage and output to a segment pin (S0 to S39[Note]).

Consequently, it is necessary to check what combination of front surface electrodes (corresponding to the segment signals) and rear surface electrodes (corresponding to the common signals) of the LCD display to be used form the display pattern, and then write bit data corresponding on a one-to-one basis with the pattern to be displayed.

In addition, because LCD display RAM bits 1 to 3 are not used with the static method, these can be used for other than display purposes.

LCD display RAM bits 4 to 7 are bits for LCD blinking selection. To use the LCD blinking function, set the relevant bit to 1.

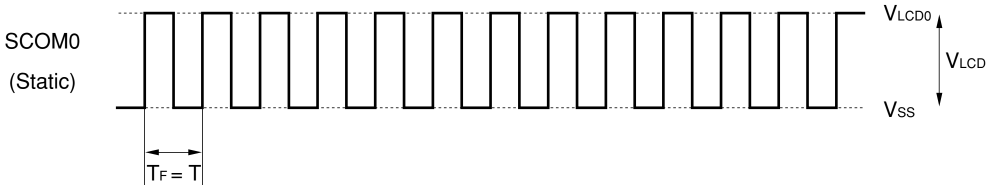
Note When ports 8 to 11 are used as the segment signal outputs.

(3) Common signal and segment signal output waveforms

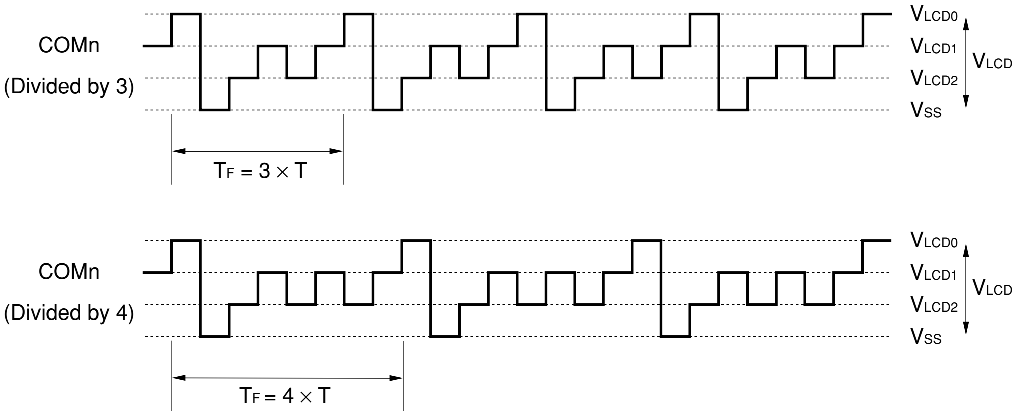
The voltages shown in Figures 18-10 and 18-11 are output for the common signals and segment signals. The ±V_{LCD} ON voltage is only produced when the common signal and segment signal are both at the selection voltage; other combinations produce the OFF voltage.

Figure 18-10. Common Signal Waveform

(a) Static display mode



(b) Dynamic display mode (1/3 bias method)

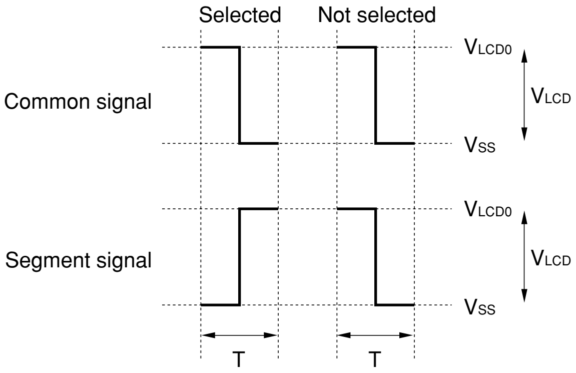


T: One LCDCL cycle

T_F: Frame frequency

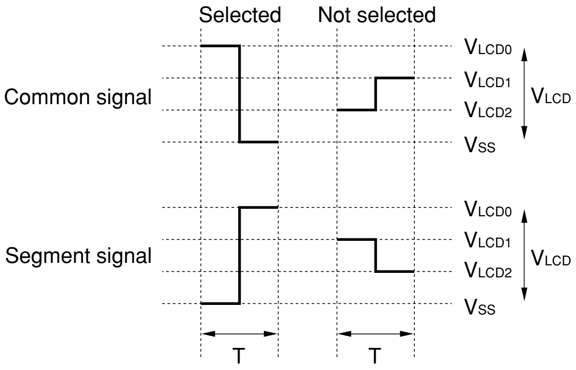
Figure 18-11. Common Signal and Segment Signal Voltages and Phases

(a) Static display mode



Remark T: One LCDCL cycle

(b) Dynamic display mode (1/3 bias method)



Remark T: One LCDCL cycle

18.7 Supplying LCD Drive Voltages V_{LCD0}, V_{LCD1}, and V_{LCD2}

The μPD780344, 780354, 780344Y, 780354Y Subseries contains a booster circuit (×3 only) to generate a supply voltage to drive the LCD. The internal LCD reference voltage (V_{LCD2}) is output from the V_{LC2} pin. A voltage two times higher than that on V_{LCD2} (V_{LCD1}) is output from the V_{LC1} pin and a voltage three times higher than that on V_{LCD2} (V_{LCD0}) is output from the V_{LC0} pin.

The LCD reference voltage (V_{LCD2}) can be varied by setting LCD gain adjust register 0 (VLCG0) as listed in Table 18-6.

In addition, an external capacitor (recommended value: 0.47 μF) is required because a capacitance division method is employed to generate the supply voltage to drive the LCD.

★ **Table 18-6. Output Voltages of V_{LC0} to V_{LC2} Pins**

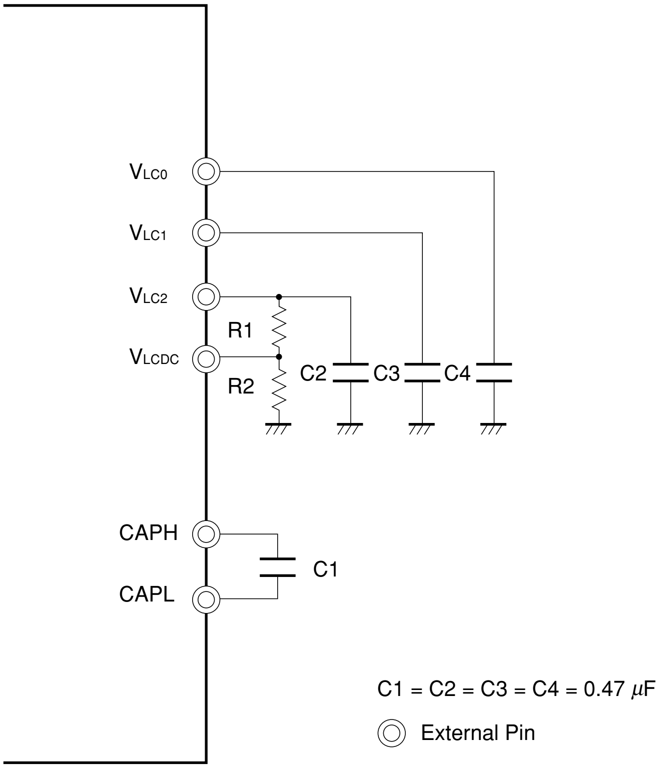
VLCG0 / LCD Driving Power Supply Pin	GAIN = 0	GAIN = 1
V_{LC0} pin	3.0 V	4.5 V
V_{LC1} pin	2.0 V	3.0 V
V_{LC2} pin (LCD reference voltage)	1.0 V	1.5 V

Cautions 1. When using the LCD function, do not leave the V_{LC0}, V_{LC1}, and V_{LC2} pins open. Refer to Figure 18-12 for connection.

★ **2. A constant LCD drive voltage can be supplied regardless of changes in V_{DD}.**

★ **Remark** For the LCD reference voltage (V_{LCD2}), refer to the LCD controller driver characteristics in **CHAPTER 25 ELECTRICAL SPECIFICATIONS**.

Figure 18-12. Example of LCD Driver Pin Connection



Remark Use a capacitor with as little leakage as possible. Use a nonpolar capacitor as C1.

18.8 Display Modes

18.8.1 Static display example

Figure 18-14 shows the connection of a static type 1-digit LCD panel with the display pattern shown in Figure 18-13 with the segment (S0 to S11) and common (SCOM0) signals. The display example is “5,” and the display data memory contents (addresses FA00H to FA07H) correspond to this.

★ In accordance with the display pattern in Figure 18-13, selection and non-selection voltages must be output to pins S0 to S7 as shown in Table 18-7 at the SCOM0 common signal timing. At this time, set the SDSEL3 register to 03H to set the S0 to S7 pins to the static display mode.

Table 18-7. Selection and Non-Selection Voltages (SCOM0)

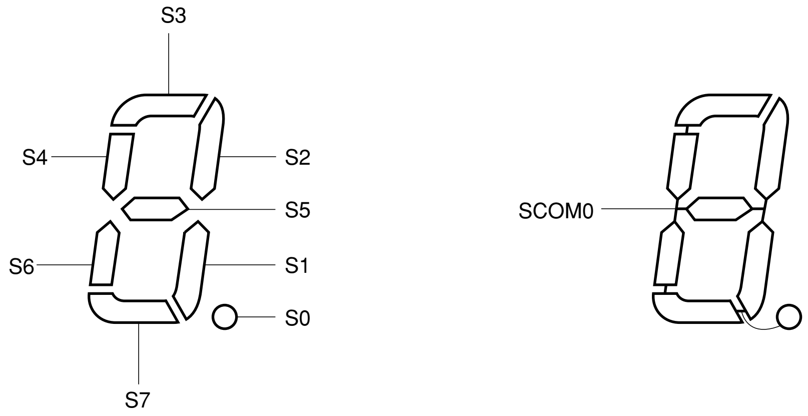
Segment Common	S0	S1	S2	S3	S4	S5	S6	S7
SCOM0	NS	S	NS	S	S	S	NS	S

S: Selection, NS: Non-selection

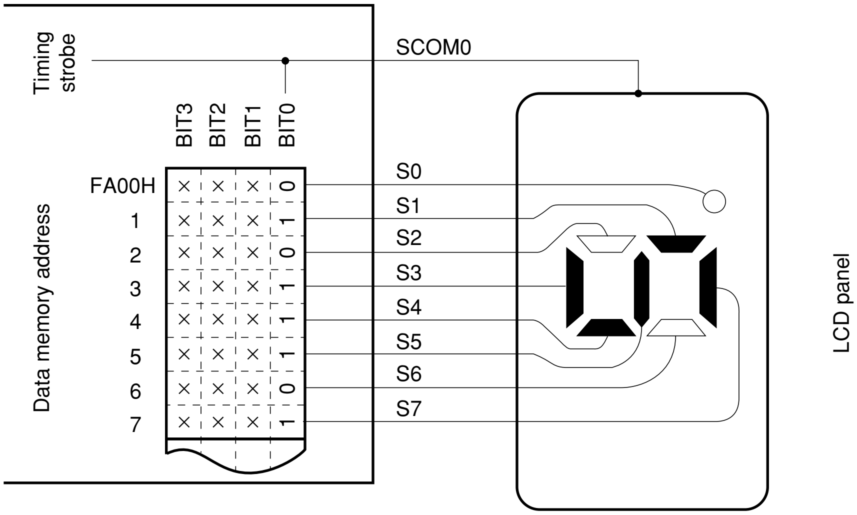
From this, it can be seen that 01011101 must be prepared in bit 0 of the display data memory (addresses FA00H to FA07H) corresponding to S0 to S7.

The LCD drive waveforms for S1, S2, and SCOM0 are shown in Figure 18-15. When S1 is at the selection voltage at the timing for selection with SCOM0, it can be seen that the +VLCD/–VLCD AC square wave, which is the LCD illumination (ON) level, is generated.

Figure 18-13. Static LCD Panel Display Pattern and Electrode Connections

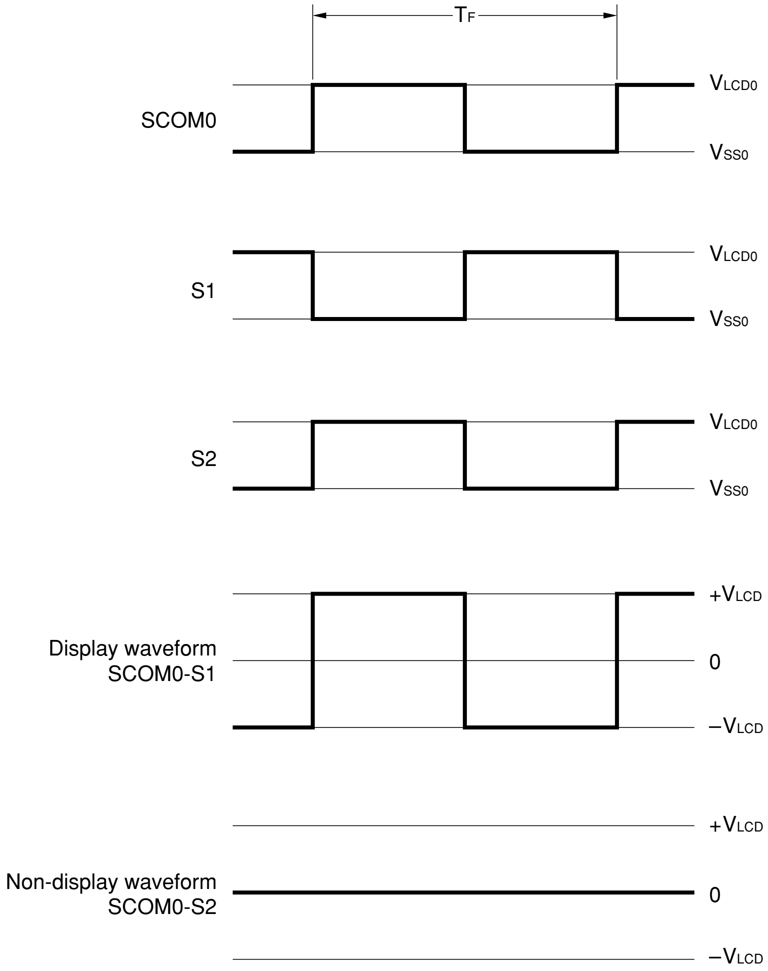


★ Figure 18-14. Static LCD Panel Connection Example (SDSEL3n = 1: n = 0, 1)



Data memory address	BIT3	BIT2	BIT1	BIT0
FA00H	×	×	×	0
1	×	×	×	1
2	×	×	×	0
3	×	×	×	1
4	×	×	×	1
5	×	×	×	1
6	×	×	×	0
7	×	×	×	1

Figure 18-15. Static LCD Drive Waveform Examples



18.8.2 3-time-division display example

Figure 18-17 shows the connection of a 3-time-division type 13-digit LCD panel with the display pattern shown in Figure 18-16 with the segment signals (S0 to S38) and common signals (COM0 to COM2). The display example is “123456.7890123,” and the display data memory contents (addresses FA00H to FA26H) correspond to this.

An explanation is given here taking the example of the eighth digit from the right “6.” (6.). In accordance with the display pattern in Figure 18-16, selection and non-selection voltages must be output to pins S21 to S23 as shown in Table 18-8 at the COM0 to COM2 common signal timings.

Table 18-8. Selection and Non-Selection Voltages (COM0 to COM2)

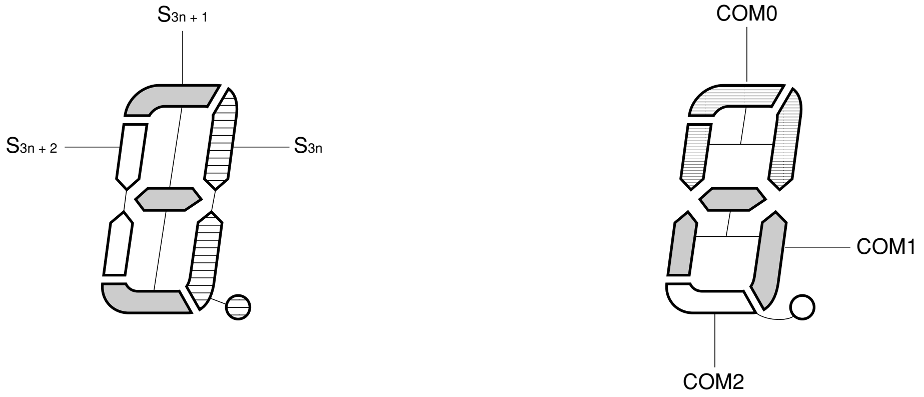
Segment Common	S21	S22	S23
COM0	NS	S	S
COM1	S	S	S
COM2	S	S	—

S: Selection, NS: Non-selection

From this, it can be seen that ×110 must be prepared in the display data memory (address FA15H) corresponding to S21.

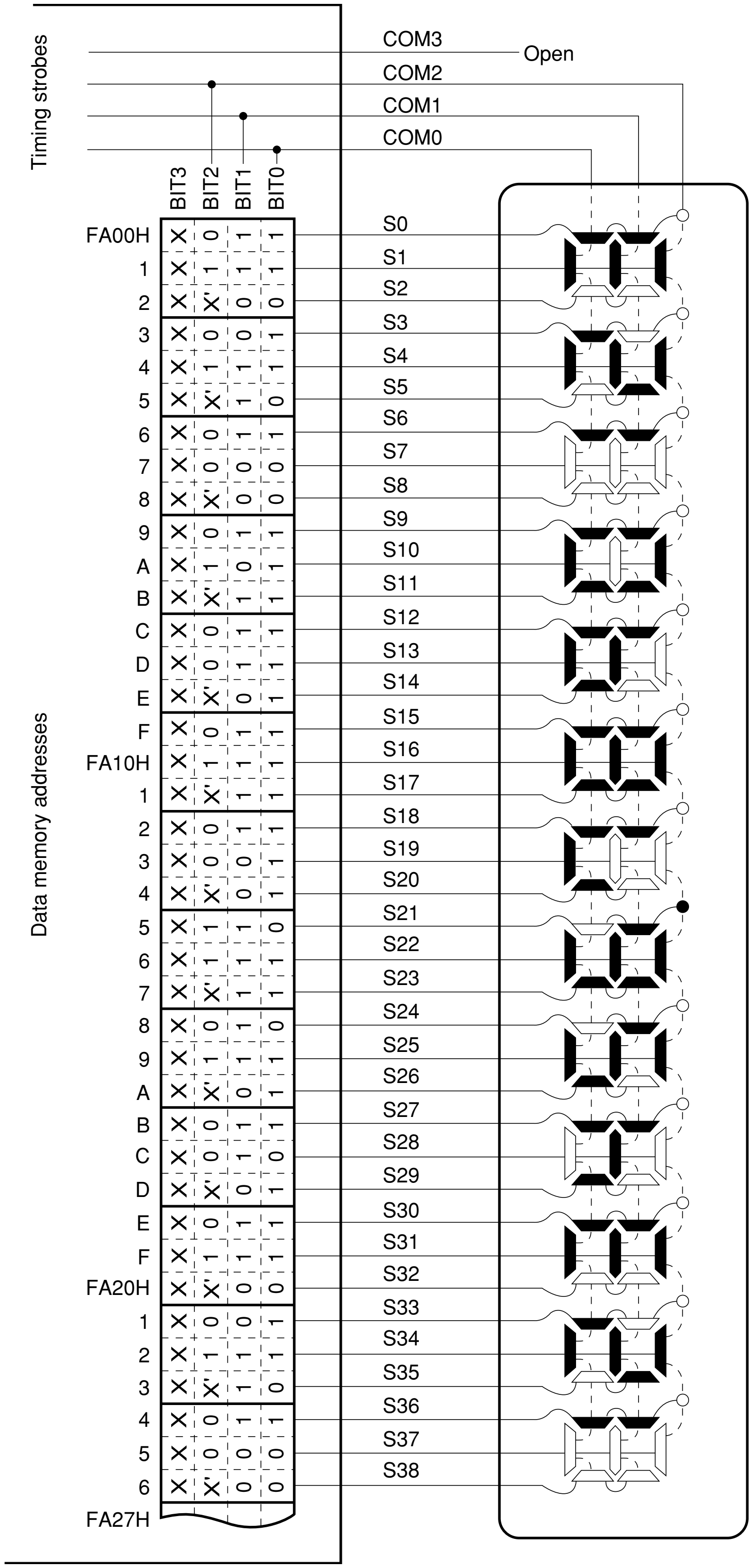
Examples of the LCD drive waveforms between S21 and the common signals are shown in Figure 18-18. When S21 is at the selection voltage at the COM1 selection timing, and S21 is at the selection voltage at the COM2 selection timing, it can be seen that the $+V_{LCD}/-V_{LCD}$ AC square wave, which is the LCD illumination (ON) level, is generated.

Figure 18-16. 3-Time-Division LCD Display Pattern and Electrode Connections



Remark n = 0 to 12

★ **Figure 18-17. 3-Time-Division LCD Panel Connection Example (SDSEL3n = 0: n = 0 to 2)**

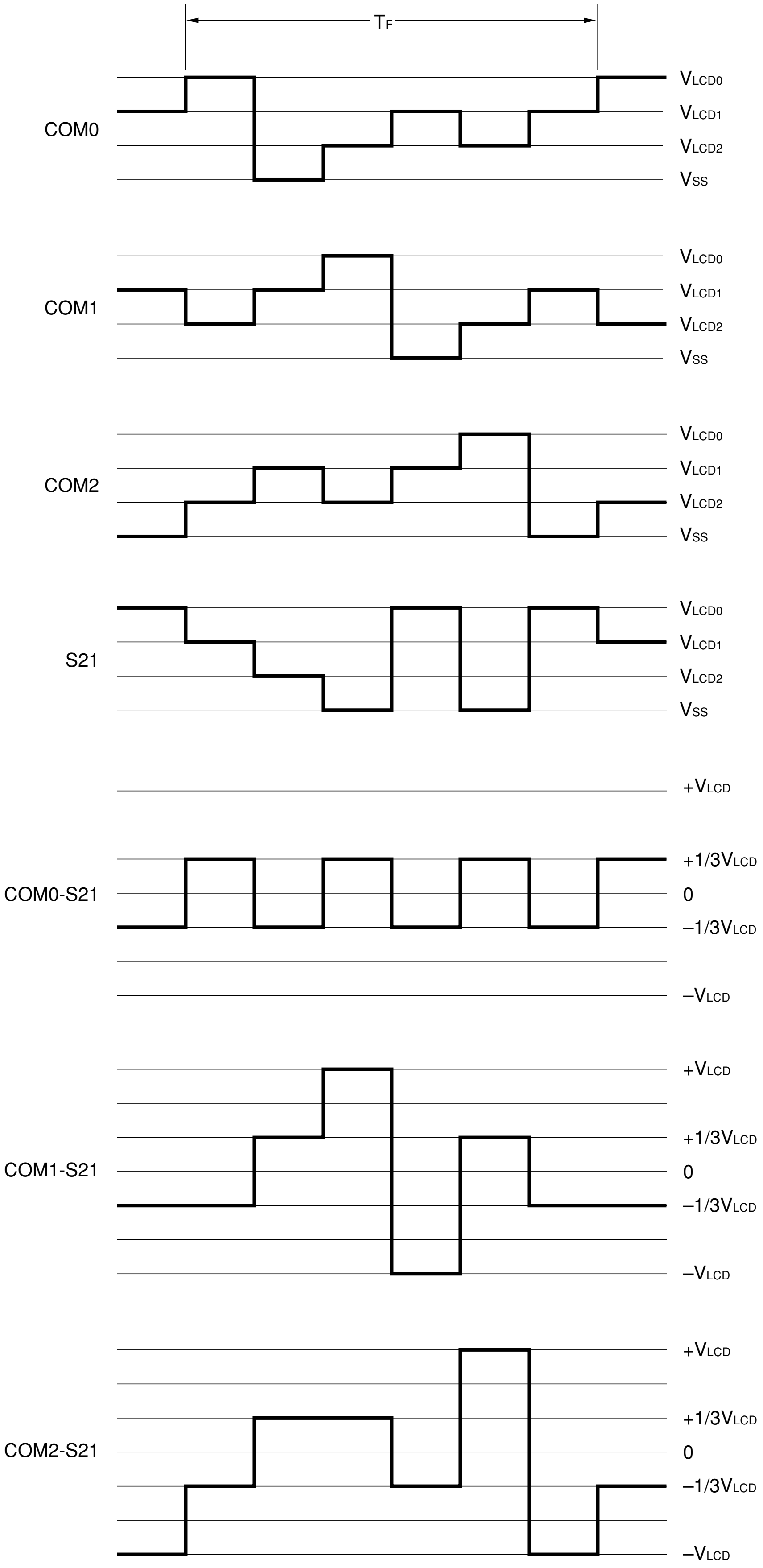


	BIT3	BIT2	BIT1	BIT0
FA00H	X	0	1	1
1	X	1	1	1
2	X	X’	0	0
3	X	0	0	1
4	X	1	1	1
5	X	X’	1	0
6	X	0	1	1
7	X	0	0	0
8	X	X’	0	0
9	X	0	1	1
A	X	1	0	1
B	X	X’	1	1
C	X	0	1	1
D	X	0	1	1
E	X	X’	0	1
F	X	0	1	1
FA10H	X	1	1	1
1	X	X’	1	1
2	X	0	1	1
3	X	0	0	1
4	X	X’	0	1
5	X	1	1	0
6	X	1	1	1
7	X	X’	1	1
8	X	0	1	0
9	X	1	1	1
A	X	X’	0	1
B	X	0	1	1
C	X	0	1	0
D	X	X’	0	1
E	X	0	1	1
F	X	1	1	1
FA20H	X	X’	0	0
1	X	0	0	1
2	X	1	1	1
3	X	X’	1	0
4	X	0	1	1
5	X	0	0	0
6	X	X’	0	0
FA27H				

Remarks 1. X’: Irrelevant bits because they have no corresponding segment in the LCD panel

2. X: Irrelevant bits because this is a 3-time-division display

Figure 18-18. 3-Time-Division LCD Drive Waveform Examples (1/3 Bias Method)



18.8.3 4-time-division display example

Figure 18-20 shows the connection of a 4-time-division type 20-digit LCD panel with the display pattern shown in Figure 18-19 with the segment signals (S0 to S39) and common signals (COM0 to COM3). The display example is “123456.78901234567890,” and the display data memory contents (addresses FA00H to FA27H) correspond to this.

An explanation is given here taking the example of the 15th digit from the right “6.” (6.). In accordance with the display pattern in Figure 18-19, selection and non-selection voltages must be output to pins S28 and S29 as shown in Table 18-9 at the COM0 to COM3 common signal timings.

Table 18-9. Selection and Non-Selection Voltages (COM0 to COM3)

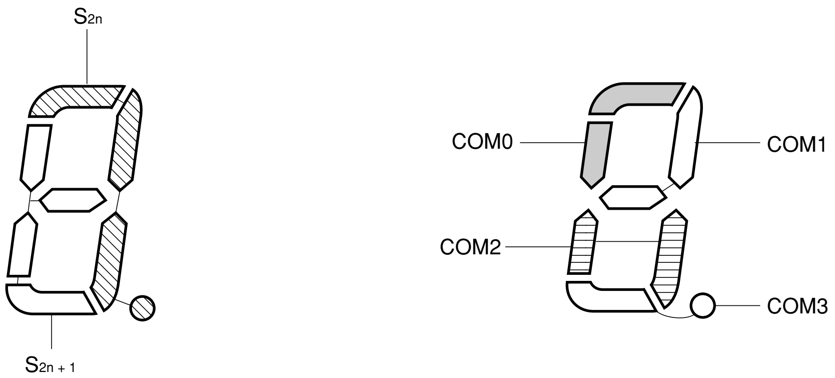
Segment / Common	S28	S29
COM0	S	S
COM1	NS	S
COM2	S	S
COM3	S	S

S: Selection, NS: Non-selection

★ From this, it can be seen that 1101 must be prepared in the display data memory (address FA1CH) corresponding to S28.

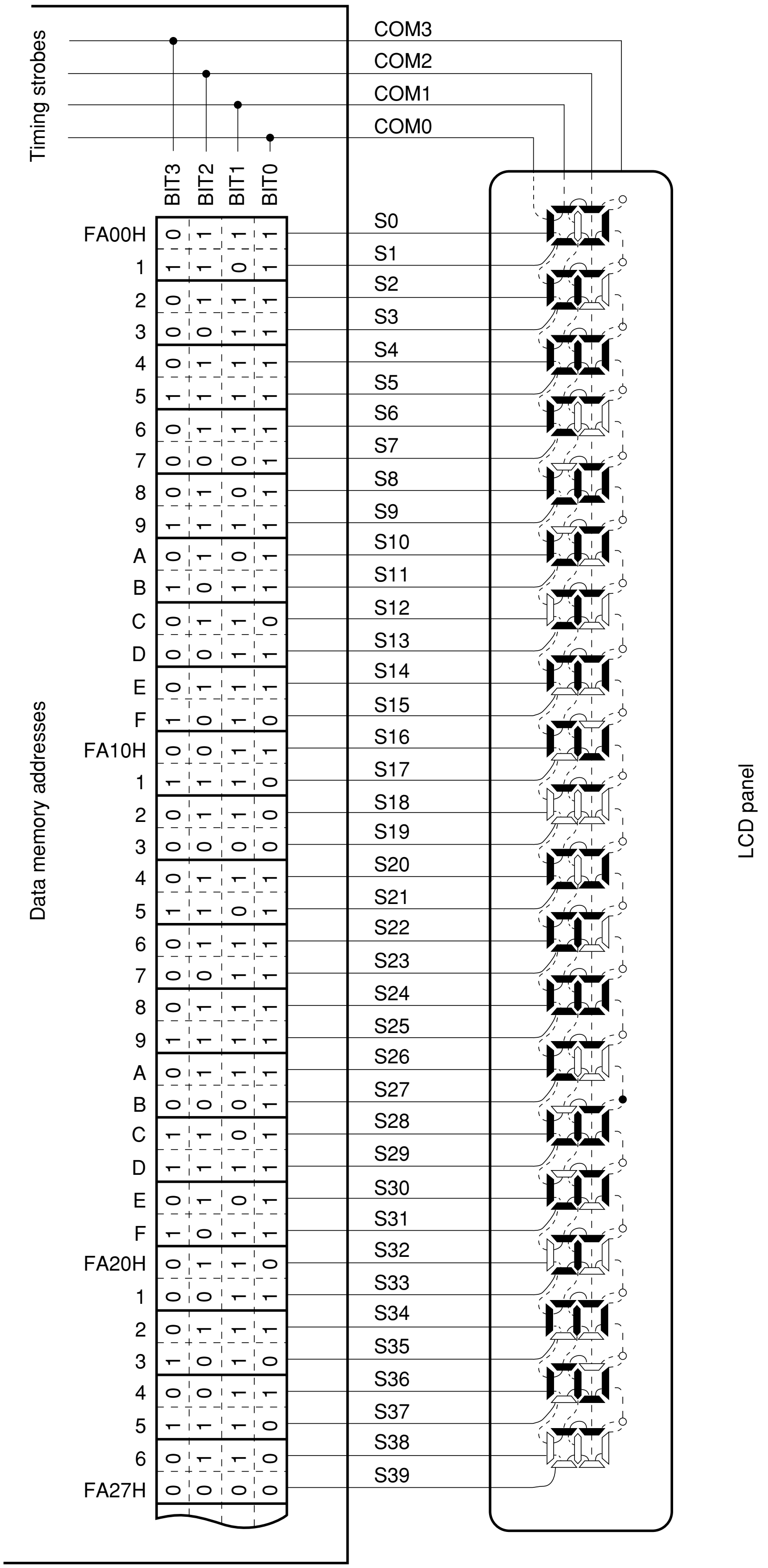
Examples of the LCD drive waveforms between S28 and the common signals are shown in Figure 18-21. When S28 is at the selection voltage at the COM0 selection timing, it can be seen that the +V_{LCD}/–V_{LCD} AC square wave, which is the LCD illumination (ON) level, is generated.

Figure 18-19. 4-Time-Division LCD Display Pattern and Electrode Connections



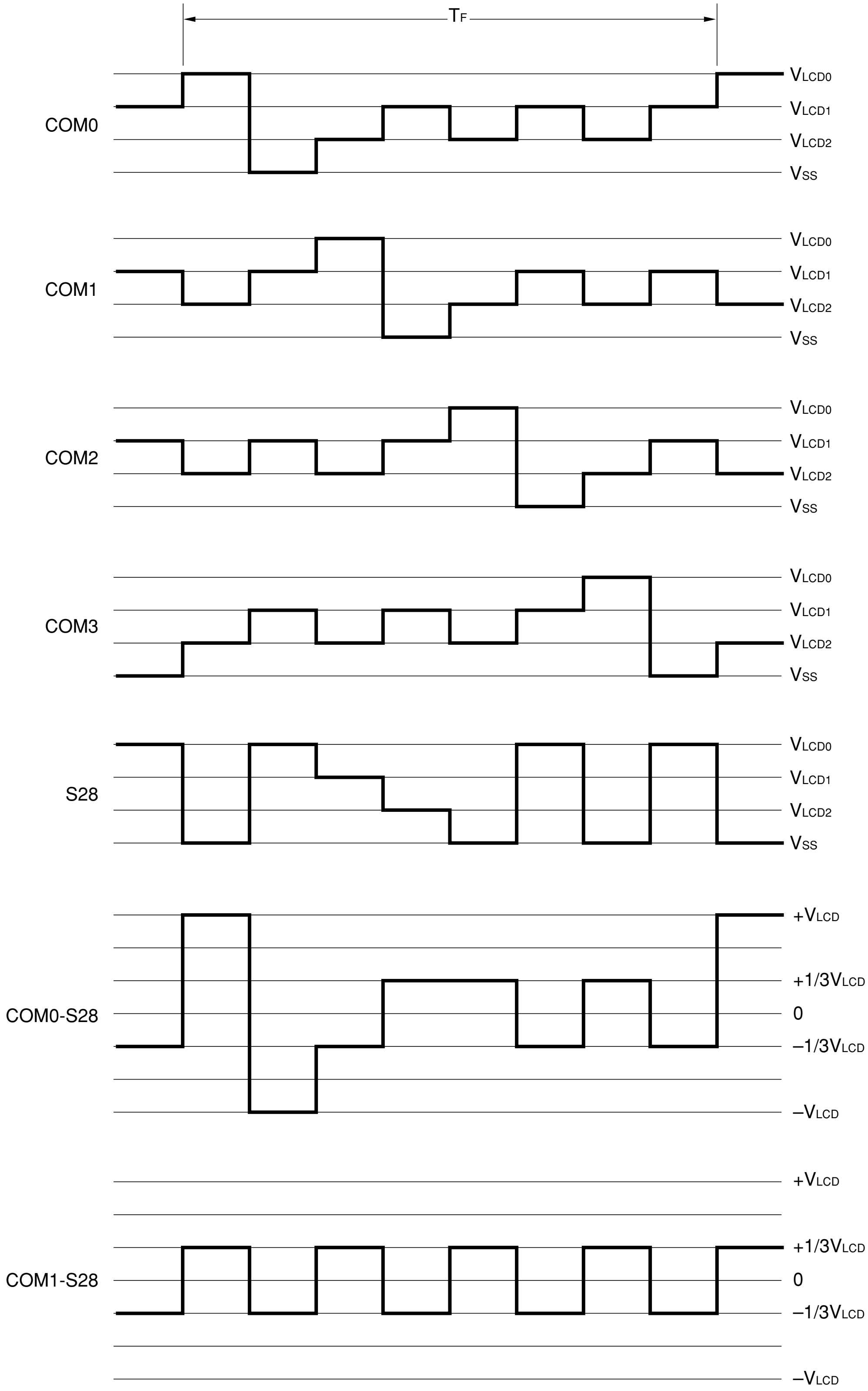
Remark n = 0 to 18

★ **Figure 18-20. 4-Time-Division LCD Panel Connection Example (SDSEL3n = 0, n = 0 to 2)**



	BIT3	BIT2	BIT1	BIT0
FA00H	0	1	1	1
1	1	1	0	1
2	0	1	1	1
3	0	0	1	1
4	0	1	1	1
5	1	1	1	1
6	0	1	1	1
7	0	0	0	1
8	0	1	0	1
9	1	1	1	1
A	0	1	0	1
B	1	0	1	1
C	0	1	1	0
D	0	0	1	1
E	0	1	1	1
F	1	0	1	0
FA10H	0	0	1	1
1	1	1	1	0
2	0	1	1	0
3	0	0	0	0
4	0	1	1	1
5	1	1	0	1
6	0	1	1	1
7	0	0	1	1
8	0	1	1	1
9	1	1	1	1
A	0	1	1	1
B	0	0	0	1
C	1	1	0	1
D	1	1	1	1
E	0	1	0	1
F	1	0	1	1
FA20H	0	1	1	0
1	0	0	1	1
2	0	1	1	1
3	1	0	1	0
4	0	0	1	1
5	1	1	1	0
6	0	1	1	0
FA27H	0	0	0	0

Figure 18-21. 4-Time-Division LCD Drive Waveform Examples (1/3 Bias Method)



Remark The waveforms of COM2-S28 and COM3-S28 are omitted.

18.8.4 Simultaneous driving of static display and dynamic display

Simultaneous driving of static display (S0 to S11) and dynamic display is possible with the μPD780344, 780354, 780344Y, 780354Y Subseries.

★ For register settings, refer to **Figure 18-7**.

CHAPTER 19 INTERRUPT FUNCTIONS

19.1 Interrupt Function Types

The following three types of interrupt functions are used.

(1) Non-maskable interrupt

This interrupt is acknowledged even in an interrupt disabled state. It does not undergo priority control and is
★ given top priority over all other interrupt requests. However, a non-maskable interrupt is held pending during servicing of another non-maskable interrupt.
★ A standby release signal is generated and HALT mode is released.
Only the interrupt request from the watchdog timer is incorporated as a non-maskable interrupt.

(2) Maskable interrupts

These interrupts undergo mask control. Maskable interrupts can be divided into a high interrupt priority group and a low interrupt priority group by setting the priority specification flag registers (PR0L, PR0H, PR1L).
High priority interrupt nesting can be applied to low priority interrupts. If two or more interrupts with the same priority are simultaneously generated, each interrupt has a predetermined priority (see **Table 19-1**).
★ A standby release signal is generated and the STOP and HALT modes are released.
Eight external interrupt requests and 16 internal interrupt requests are incorporated as maskable interrupts.

(3) Software interrupt

This is a vectored interrupt to be generated by executing the BRK instruction. It is acknowledged even in an interrupt disabled state. The software interrupt does not undergo interrupt priority control.

19.2 Interrupt Sources and Configuration

A total of 26 interrupt sources exist among non-maskable, maskable, and software interrupts (see **Table 19-1**).

Remark A non-maskable interrupt or maskable interrupt (internal) can be selected as the watchdog timer interrupt (INTWDT).

Table 19-1. Interrupt Source List

Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Non-maskable	—	INTWDT	Watchdog timer overflow (with watchdog timer mode 1 selected)	Internal	0004H	(A)
Maskable	0	INTWDT	Watchdog timer overflow (with interval timer mode selected)			(B)
	1	INTP0	Pin input edge detection	External	0006H	(C)
	2	INTP1			0008H	
	3	INTP2			000AH	
	4	INTP3			000CH	
	5	INTP4			000EH	
	6	INTP5			0010H	
	7	INTP6			0012H	
	8	INTKR	Detection of port 4 falling edge		0014H	(D)
	9	INTSER0	Serial interface UART0 reception error generation	Internal	0016H	(B)
	10	INTSR0	End of serial interface UART0 reception		0018H	
	11	INTST0	End of serial interface UART0 transmission		001AH	
	12	INTCSI1	End of serial interface CSI1 transfer		001CH	
	13	INTCSI3	End of serial interface SIO3 transfer		001EH	
	14	INTIIC0	End of serial interface IIC0 transfer (μPD780344Y, 780354Y Subseries only)		0020H[Note 3]	
	15	INTWTNI0	Reference time interval signal from watch timer		0022H	
	16	INTTM00	Match between TM00 and CR00 (when CR00 is specified as compare register) Detection of TI01 valid edge (when CR00 is specified as capture register)		0024H	
	17	INTTM01	Match between TM00 and CR01 (when CR01 is specified as compare register) Detection of TI00 valid edge (when CR01 is specified as capture register)		0026H	
	18	INTTMA0	Match between TMA0 and CRA0		0028H	
	19	INTTMB0	Match between TMB0 and CRB0		002AH	
	20	INTTM50	Match between TM50 and CR50		002CH	
	21	INTTM51	Match between TM51 and CR51		002EH	
	22	INTAD0	End of A/D converter conversion		0030H	
	23	INTWTN0	Watch timer overflow		0032H	
Software	—	BRK	BRK instruction execution	—	003EH	(E)

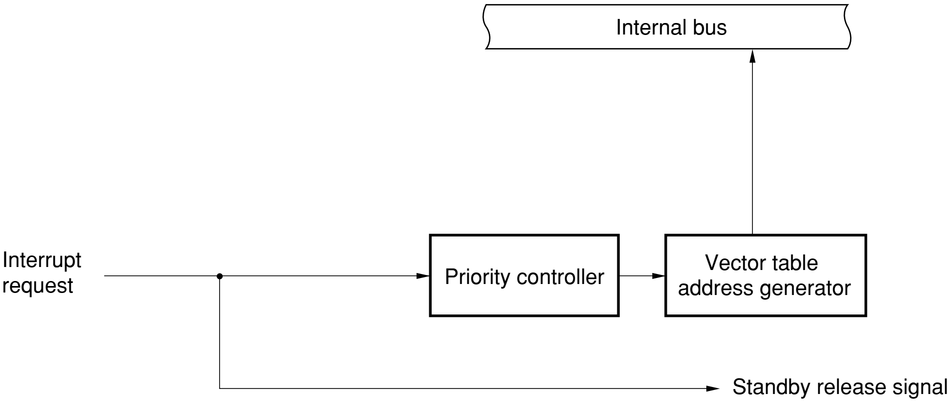
Notes 1. The default priority is the priority applicable when two or more maskable interrupts are generated simultaneously. 0 is the highest priority, and 23 is the lowest.

2. Basic configuration types (A) to (E) correspond to (A) to (E) in Figure 19-1.

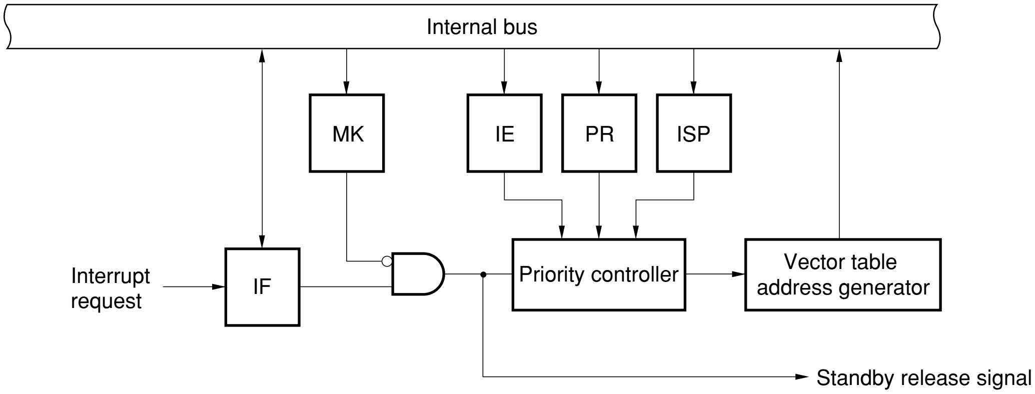
3. The μPD780344 and 780354 Subseries do not have an interrupt source corresponding to vector table address 0020H.

Figure 19-1. Basic Configuration of Interrupt Function (1/2)

(A) Internal non-maskable interrupt



(B) Internal maskable interrupt



(C) External maskable interrupt (INTP0 to INTP6)

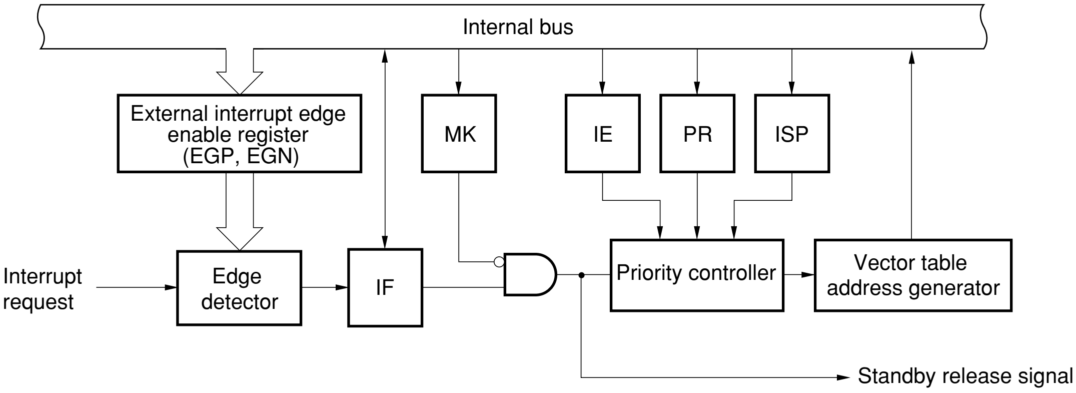
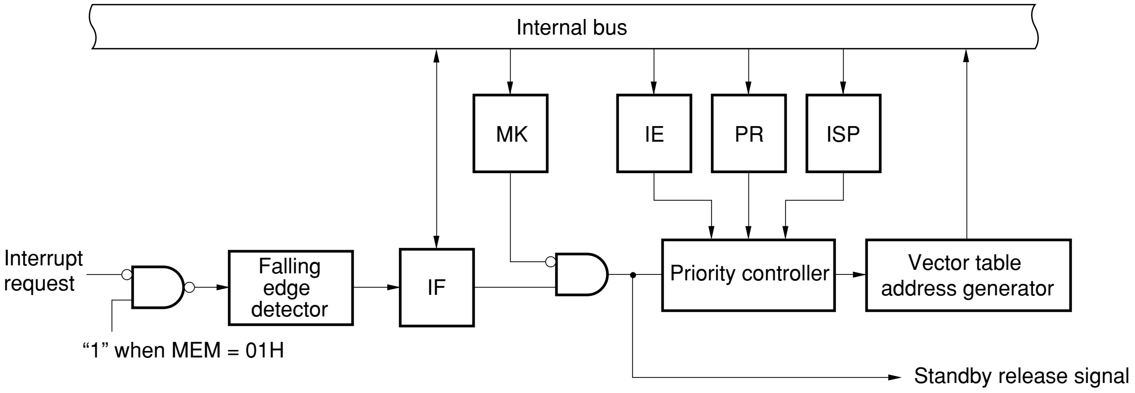
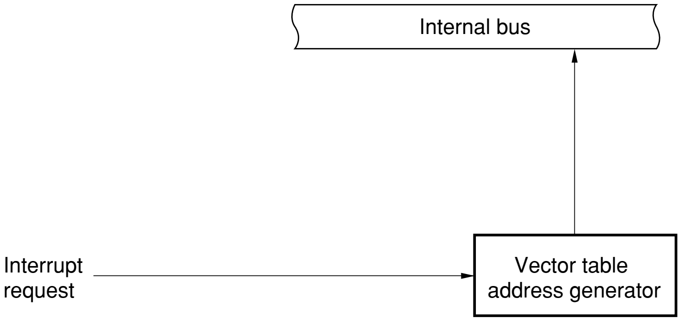


Figure 19-1. Basic Configuration of Interrupt Function (2/2)

(D) External maskable interrupt (INTKR)



★ (E) Software interrupt



IF:	Interrupt request flag
IE:	Interrupt enable flag
ISP:	In-service priority flag
MK:	Interrupt mask flag
PR:	Priority specification flag
MEM:	Memory expansion mode register

19.3 Interrupt Function Control Registers

The following six types of registers are used to control the interrupt functions.

- Interrupt request flag registers (IF0L, IF0H, IF1L)
- Interrupt mask flag registers (MK0L, MK0H, MK1L)
- Priority specification flag registers (PR0L, PR0H, PR1L)
- External interrupt rising edge enable register (EGP)
- External interrupt falling edge enable register (EGN)
- Program status word (PSW)

Table 19-2 gives a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 19-2. Flags Corresponding to Interrupt Request Sources

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
		Register		Register		Register
INTWDT	WDTIF[Note 1]	IF0L	WDTMK[Note 1]	MK0L	WDTPR[Note 1]	PR0L
INTP0	PIF0		PMK0		PPR0	
INTP1	PIF1		PMK1		PPR1	
INTP2	PIF2		PMK2		PPR2	
INTP3	PIF3		PMK3		PPR3	
INTP4	PIF4		PMK4		PPR4	
INTP5	PIF5		PMK5		PPR5	
INTP6	PIF6		PMK6		PPR6	
INTKR	KRIF	IF0H	KRMK	MK0H	KRPR	PR0H
INTSER0	SERIF0		SERMK0		SERPR0	
INTSR0	SRIF0		SRMK0		SRPR0	
INTST0	STIF0		STMK0		STPR0	
INTCSI1	CSIIF1		CSIMK1		CSIPR1	
INTCSI3	CSIIF3		CSIMK3		CSIPR3	
INTIIC0[Note 2]	IICIF0[Note 2]		IICMK0[Note 2]		IICPR0[Note 2]	
INTWTNI0	WTNIIF0		WTNIMK0		WTNIPR0	
INTTM00	TMIF00	IF1L	TMMK00	MK1L	TMPR00	PR1L
INTTM01	TMIF01		TMMK01		TMPR01	
INTTMA0	TMIFA0		TMMKA0		TMPRA0	
INTTMB0	TMIFB0		TMMKB0		TMPRB0	
INTTM50	TMIF50		TMMK50		TMPR50	
INTTM51	TMIF51		TMMK51		TMPR51	
INTAD0	ADIF0		ADMK0		ADPR0	
INTWTN0	WTNIF0		WTNMK0		WTNPR0	

Notes 1. Interrupt control flag when the watchdog timer is used as interval timer

2. μPD780344Y, 780354Y Subseries only

(1) Interrupt request flag registers (IF0L, IF0H, IF1L)

The interrupt request flags are set to 1 when the corresponding interrupt request is generated or an instruction is executed. They are cleared to 0 when an instruction is executed upon acknowledgment of an interrupt request or upon application of $\overline{\text{RESET}}$ input.

IF0L, IF0H, and IF1L are set by a 1-bit or 8-bit memory manipulation instruction. When IF0L and IF0H are combined to form 16-bit register IF0, they are set by a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the values of these registers to 00H.

Figure 19-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L)

Address: FFE0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IF0L	PIF6	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	WDTIF

Address: FFE1H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IF0H	WTNIIF0	IICIF0Note	CSIIF3	CSIIF1	STIF0	SRIF0	SERIF0	KRIF

Address: FFE2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IF1L	WTNIF0	ADIF0	TMIF51	TMIF50	TMIFB0	TMIFA0	TMIF01	TMIF00

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request signal is generated, interrupt request status

Note μPD780344Y, 780354Y Subseries only

Cautions 1. The WDTIF flag is R/W enabled only when the watchdog timer is used as the interval timer. If watchdog timer mode 1 is used, set the WDTIF flag to 0.

2. When operating a timer, serial interface, or A/D converter after standby release, run it once after clearing the interrupt request flag, as the 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 started.

(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 by a 16-bit memory manipulation instruction.

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

Figure 19-3. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L)

Address: FFE4H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
MK0L	PMK6	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	WDTMK

Address: FFE5H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
MK0H	WTNIMK0	IICMK0[Note]	CSIMK3	CSIMK1	STMK0	SRMK0	SERMK0	KRMK

Address: FFE6H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
MK1L	WTNMK0	ADMK0	TMMK51	TMMK50	TMMKB0	TMMKA0	TMMK01	TMMK00

XXMKX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Note μPD780344Y, 780354Y Subseries only

Cautions 1. If the watchdog timer is used in watchdog timer mode 1, the contents of the WDTMK flag become undefined when read.

2. Because port 0 pins have an alternate function as external interrupt request 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.

(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 by a 16-bit memory manipulation instruction.

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

Figure 19-4. Format of Priority Specification Flag Registers (PR0L, PR0H, PR1L)

Address: FFE8H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PR0L	PPR6	PPR5	PPR4	PPR3	PPR2	PPR1	PPR0	WDTPR

Address: FFE9H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PR0H	WTNIPR0	IICPR0[Note]	CSIPR3	CSIPR1	STPR0	SRPR0	SERPR0	KRPR

Address: FFEAH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PR1L	WTNPR0	ADPR0	TMPR51	TMPR50	TMPRB0	TMPRA0	TMPR01	TMPR00

XXPRX	Priority level selection
0	High priority level
1	Low priority level

Note μPD780344Y, 780354Y Subseries only

Caution When the watchdog timer is used in the watchdog timer mode 1, set the WDTPR flag 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 INTP6.

EGP and EGN are set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the values of these registers to 00H.

Figure 19-5. Format of External Interrupt Rising Edge Enable Register (EGP), 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	EGP6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

Address: FF49H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN	0	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 6)
0	0	Interrupt disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

★ **Caution When switching from the external interrupt function to the port function, since edge detection may be performed, set EGPn and EGNn to 0 before switching to the port mode.**

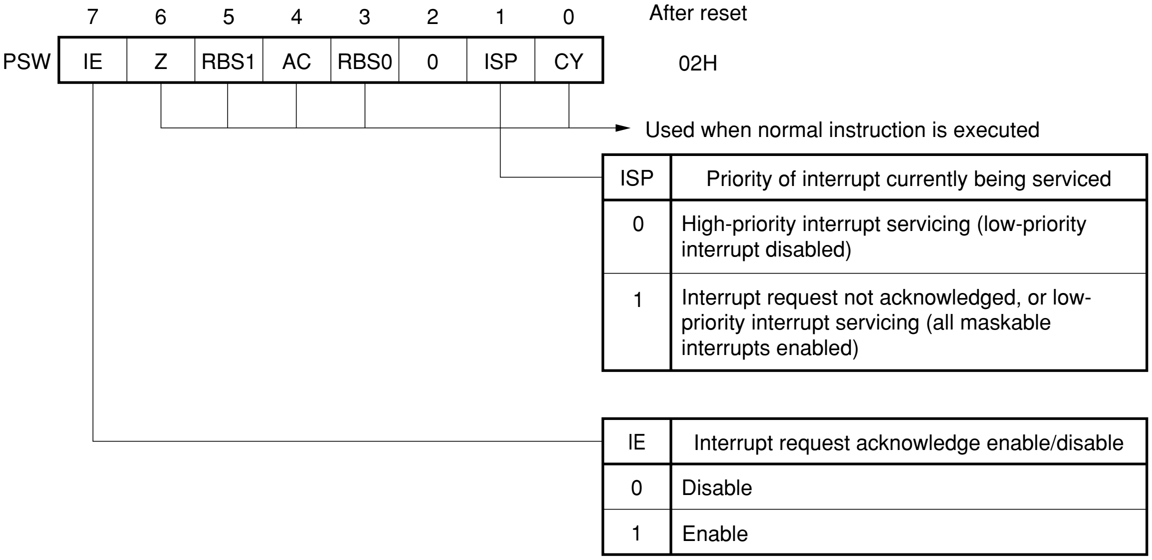
(5) Program status word (PSW)

The program status word is a register used to hold the instruction execution result and the current status for an interrupt request. The IE flag to set maskable interrupt enable/disable and the ISP flag to control nesting processing are mapped here.

Besides 8-bit read/write, this register can carry out operations with a bit manipulation instruction and dedicated instructions (EI and DI). When a vectored interrupt request is acknowledged, if the BRK instruction is executed, the contents of PSW are automatically saved into a stack and the IE flag is reset to 0. If a maskable interrupt request is acknowledged, the contents of the priority specification flag of the acknowledged interrupt are transferred to the ISP flag. The PSW contents are also saved into the stack with the PUSH PSW instruction. They are reset from the stack with the RETI, RETB, and POP PSW instructions.

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

Figure 19-6. Format of Program Status Word



	7	6	5	4	3	2	1	0	After reset
PSW	IE	Z	RBS1	AC	RBS0	0	ISP	CY	02H

ISP	Priority of interrupt currently being serviced
0	High-priority interrupt servicing (low-priority interrupt disabled)
1	Interrupt request not acknowledged, or low-priority interrupt servicing (all maskable interrupts enabled)

IE	Interrupt request acknowledge enable/disable
0	Disable
1	Enable

19.4 Interrupt Servicing Operations

19.4.1 Non-maskable interrupt request acknowledge operation

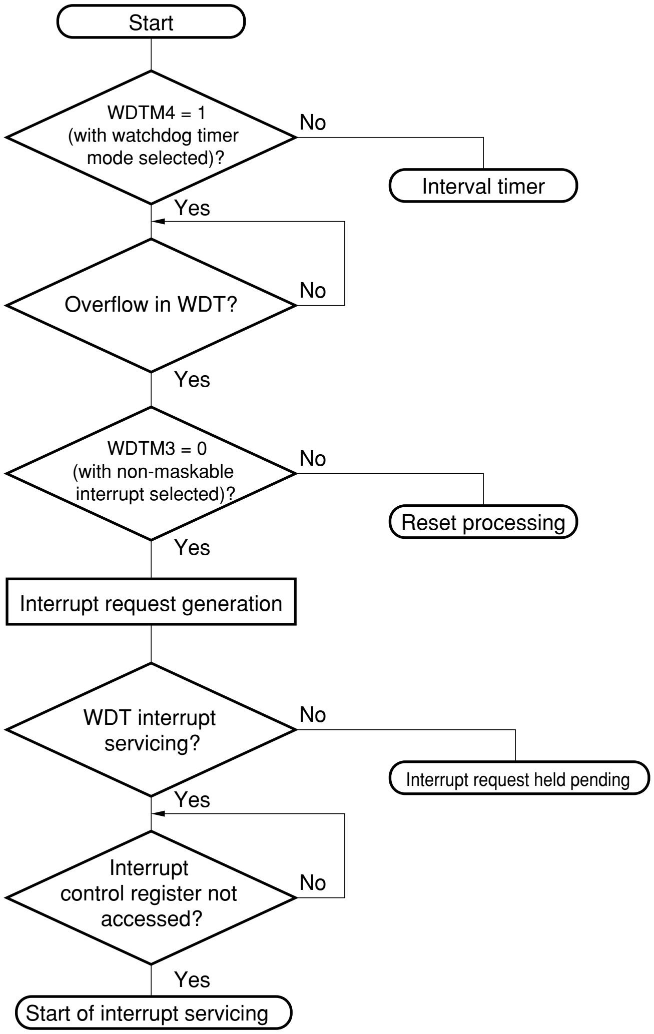
A non-maskable interrupt request is unconditionally acknowledged even if in an interrupt acknowledge disable state. It does not undergo interrupt priority control and has highest priority over all other interrupts.

If a non-maskable interrupt request is acknowledged, the contents are saved into the stacks in the order of PSW, then PC, the IE flag and ISP flag are reset (0), and the contents of the vector table are loaded into PC and branched. ★ At this time, the NMIS flag is set (1) to disable the acknowledgment of multiple interrupts.

A new non-maskable interrupt request generated during execution of a non-maskable interrupt servicing program is acknowledged after the current execution of the non-maskable interrupt servicing program is terminated (following RETI instruction execution) and one main routine instruction is executed. However, if a new non-maskable interrupt request is generated twice or more during non-maskable interrupt servicing program execution, only one non-maskable interrupt request is acknowledged after termination of the non-maskable interrupt servicing program. Figures 19-7, 19-8, and 19-9 show the flowchart of non-maskable interrupt request generation through acknowledgment, acknowledgment timing of non-maskable interrupt request, and acknowledgment operation at multiple non-maskable interrupt request generation, respectively.

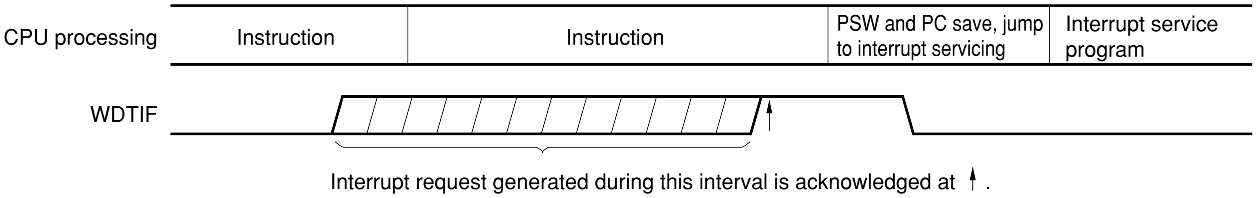
★ **Remark** The NMIS flag is set (1) by acknowledging a non-maskable interrupt and cleared by the RET or RETI instruction. When a non-maskable interrupt occurs, restore from the interrupt using the RETI instruction.

Figure 19-7. Flowchart of Non-Maskable Interrupt Request Generation to Acknowledgment



WDTM: Watchdog timer mode register
WDT: Watchdog timer

Figure 19-8. Non-Maskable Interrupt Request Acknowledgment Timing

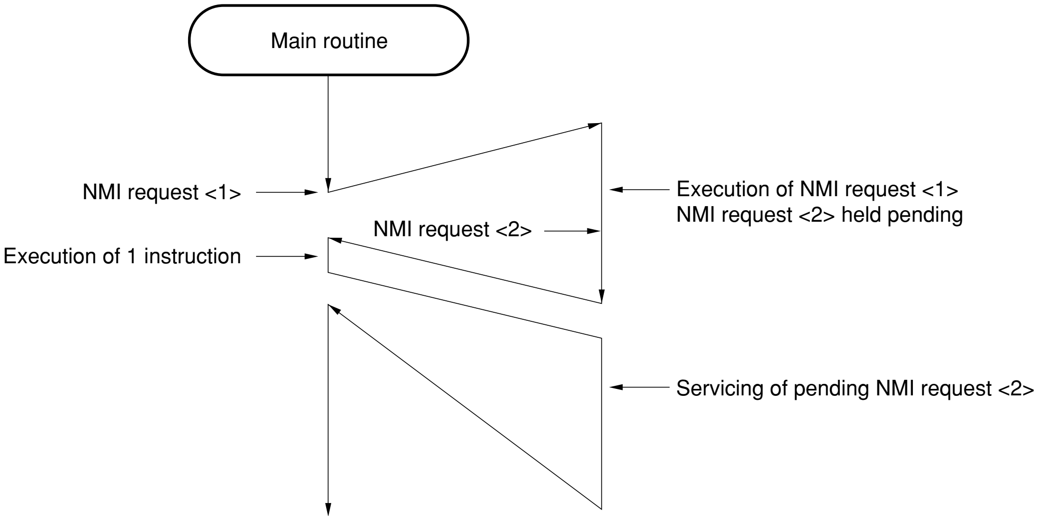


Interrupt request generated during this interval is acknowledged at ↑ .

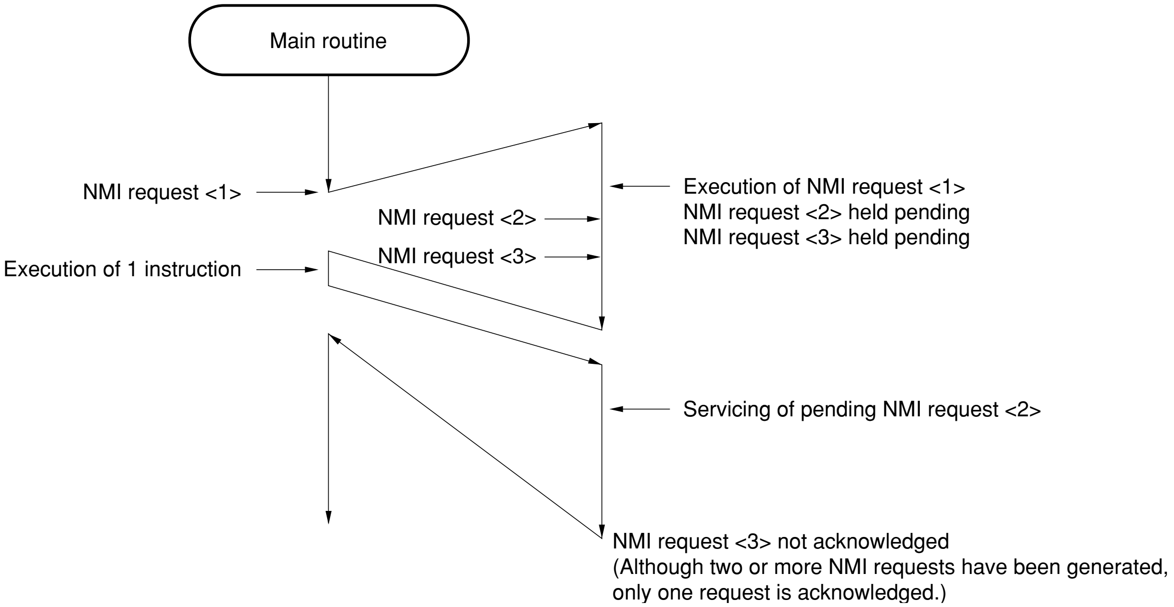
WDTIF: Watchdog timer interrupt request flag

Figure 19-9. Non-Maskable Interrupt Request Acknowledgment Operation

(a) If a non-maskable interrupt request is generated during non-maskable interrupt servicing program execution


Main routine
NMI request <1>
Execution of NMI request <1>
NMI request <2> held pending
NMI request <2>
Execution of 1 instruction
Servicing of pending NMI request <2>

(b) If two non-maskable interrupt requests are generated during non-maskable interrupt servicing program execution


Main routine
NMI request <1>
Execution of NMI request <1>
NMI request <2> held pending
NMI request <3> held pending
NMI request <2>
NMI request <3>
Execution of 1 instruction
Servicing of pending NMI request <2>
NMI request <3> not acknowledged
(Although two or more NMI requests have been generated,
only one request is acknowledged.)

19.4.2 Maskable interrupt request acknowledge operation

A maskable interrupt request 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 in the interrupt enable state (when IE flag is set to 1). However, a low-priority interrupt request is not acknowledged ★ during servicing of a higher priority interrupt request (when the ISP flag is reset to 0). Even if the EI instruction is executed while a non-maskable interrupt servicing program is being executed (NMIS = 1), non-maskable and maskable interrupt requests are not acknowledged. The time from generation of a maskable interrupt request until interrupt servicing is performed is shown in Table 19-3 below.

For the interrupt request acknowledge timing, see **Figures 19-11** and **19-12**.

Table 19-3. Time from Generation of Maskable Interrupt Until Servicing

	Minimum Time	Maximum Time[Note]
When ××PR = 0	7 clocks	32 clocks
When ××PR = 1	8 clocks	33 clocks

Note If an interrupt request is generated just before a divide instruction, the wait time becomes longer.

Remark 1 clock: $1/f_{CPU}$ (f_{CPU}: CPU clock)

If two or more maskable 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 maskable interrupt requests have the same priority level, the request with the highest default priority is acknowledged first.

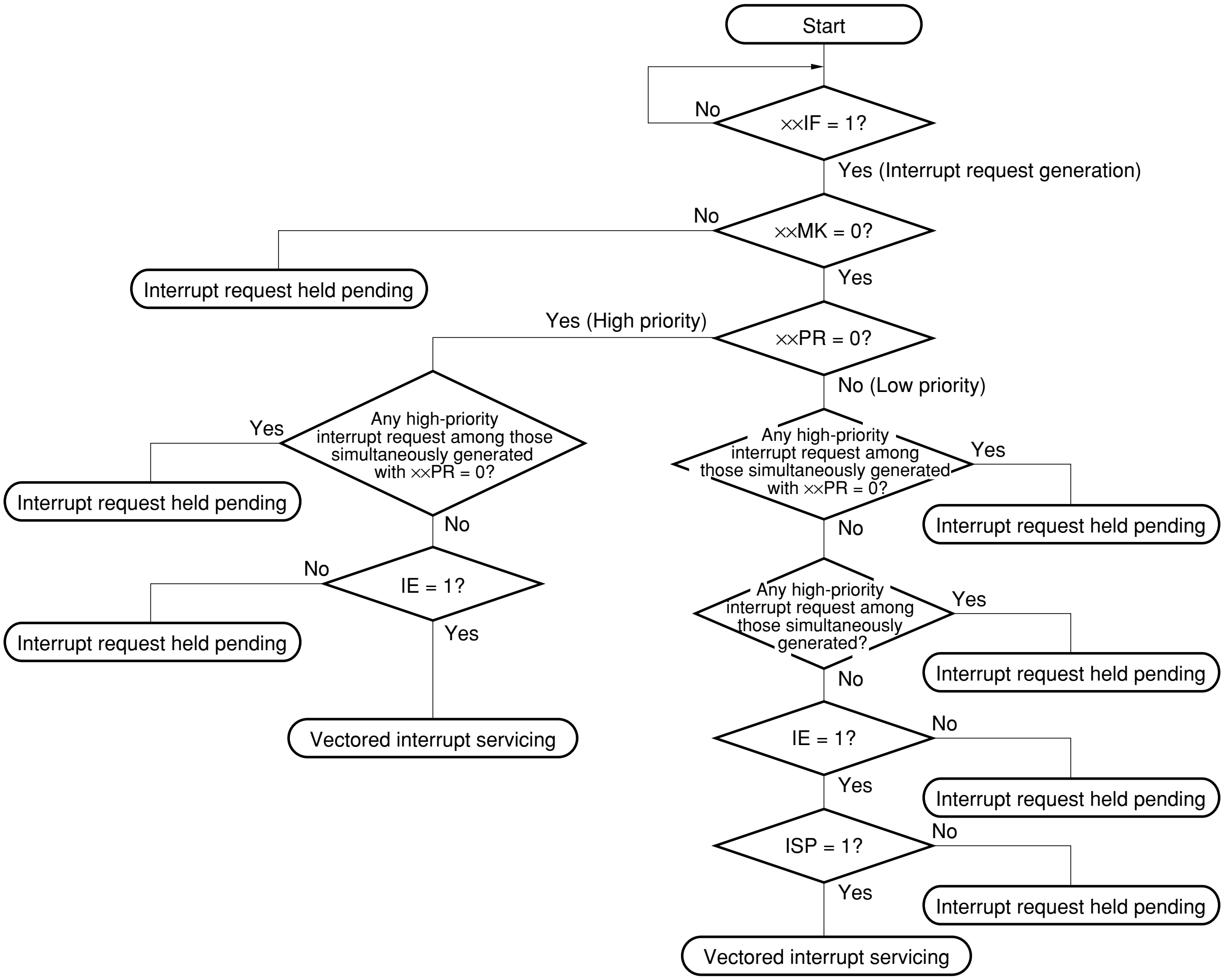
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 19-10 shows the interrupt request acknowledgment algorithm.

If a maskable interrupt request is acknowledged, the contents are saved into the stacks in the order of PSW, then PC, the IE flag is reset (0), and the contents of the priority specification flag corresponding to the acknowledged interrupt are transferred to the ISP flag. Further, the vector table data determined for each interrupt request is loaded into the PC and branched.

Return from an interrupt is possible with the RETI instruction.

Figure 19-10. Interrupt Request Acknowledgment Processing Algorithm


Start
No
××IF = 1?
Yes (Interrupt request generation)
No
××MK = 0?
Interrupt request held pending
Yes
Yes (High priority)
××PR = 0?
No (Low priority)
Yes
Any high-priority interrupt request among those simultaneously generated with ××PR = 0?
Interrupt request held pending
No
No
IE = 1?
Interrupt request held pending
Yes
Vectored interrupt servicing
Any high-priority interrupt request among those simultaneously generated with ××PR = 0?
Yes
Interrupt request held pending
No
Any high-priority interrupt request among those simultaneously generated?
Yes
Interrupt request held pending
No
IE = 1?
No
Interrupt request held pending
Yes
ISP = 1?
No
Interrupt request held pending
Yes
Vectored interrupt servicing

××IF: Interrupt request flag

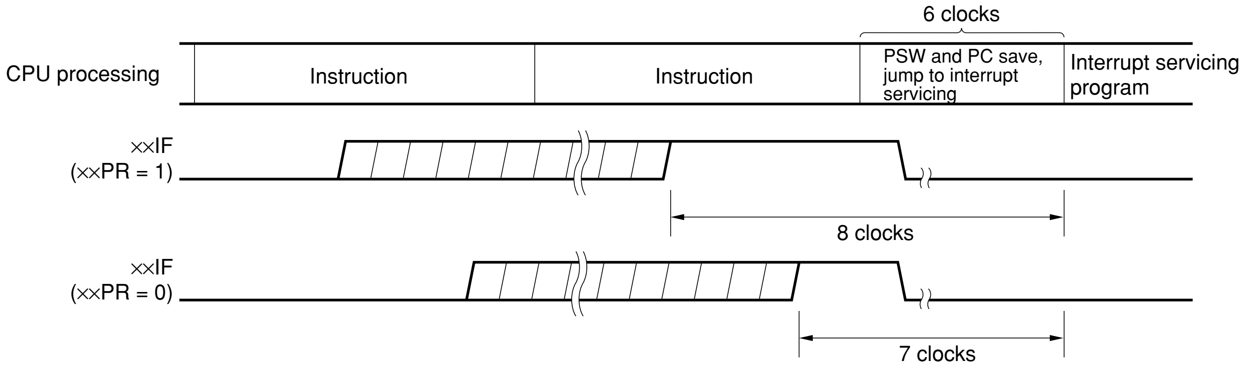
××MK: Interrupt mask flag

××PR: Priority specification flag

IE: Flag that controls acknowledge of maskable interrupt request (1 = enable, 0 = disable)

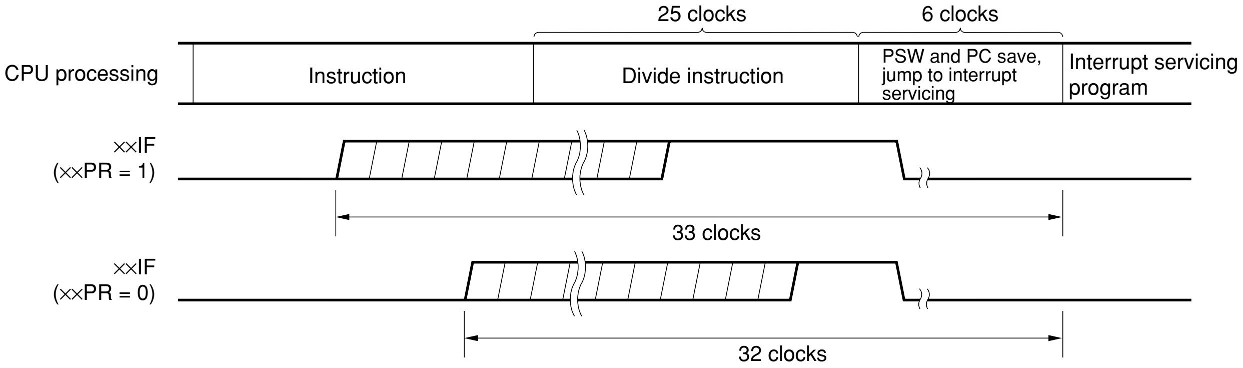
ISP: Flag that indicates the priority level of the interrupt currently being serviced (0 = high-priority interrupt servicing, 1 = no interrupt request acknowledged, or low-priority interrupt servicing)

Figure 19-11. Interrupt Request Acknowledgment Timing (Minimum Time)



Remark 1 clock: 1/fCPU (fCPU: CPU clock)

Figure 19-12. Interrupt Request Acknowledgment Timing (Maximum Time)



Remark 1 clock: 1/fCPU (fCPU: CPU clock)

19.4.3 Software interrupt request acknowledge operation

A software interrupt request is acknowledged by BRK instruction execution. Software interrupts cannot be disabled.

If a software interrupt request is acknowledged, the contents are saved into the 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.

Return from a software interrupt is possible with the RETB instruction.

Caution Do not use the RETI instruction for returning from a software interrupt.

19.4.4 Nesting processing

Nesting occurs when an interrupt request is acknowledged during execution of another interrupt.

Nesting does not occur unless the interrupt request acknowledge enable state is selected (IE = 1) (except non-maskable interrupts). Also, when an interrupt request is acknowledged, interrupt request acknowledgment becomes disabled (IE = 0). Therefore, to enable nesting, it is necessary to set (1) the IE flag with the EI instruction during interrupt servicing to enable interrupt acknowledgment.

Moreover, even if interrupts are enabled, nesting may not be enabled, this being subject to interrupt priority control. Two types of priority control are available: default priority control and programmable priority control. Programmable priority control is used for nesting.

In the interrupt enable state, if an interrupt request with a priority equal to or higher than that of the interrupt currently being serviced is generated, it is acknowledged for nesting. If an interrupt with a priority lower than that of the interrupt currently being serviced is generated during interrupt servicing, it is not acknowledged for nesting.

Interrupt requests that are not enabled because of the interrupt disable state or they have a lower priority are held pending. When servicing of the current interrupt ends, the pending interrupt request is acknowledged following ★ execution of at least one main processing instruction execution.

Nesting is not possible during non-maskable interrupt servicing.

Table 19-4 shows interrupt requests enabled for nesting and Figure 19-13 shows nesting examples.

★ **Table 19-4. Interrupt Request Enabled for Nesting During Interrupt Servicing**

Nesting Request Interrupt Being Serviced		Non-Maskable Interrupt Request	Maskable Interrupt Request				Software Interrupt Request
			PR = 0		PR = 1		
			IE = 1	IE = 0	IE = 1	IE = 0	
Non-maskable interrupt		×	×	×	×	×	○
Maskable interrupt	ISP = 0	○	○	×	×	×	○
	ISP = 1	○	○	×	○	×	○
Software interrupt		○	○	×	○	×	○

Remarks 1. ○: Nesting enabled

2. ×: Nesting disabled

3. ISP and IE are flags contained in PSW.

ISP = 0: An interrupt with higher priority is being serviced.

ISP = 1: No interrupt request has been acknowledged, or an interrupt with a lower priority is being serviced.

IE = 0: Interrupt request acknowledge is disabled.

IE = 1: Interrupt request acknowledge is enabled.

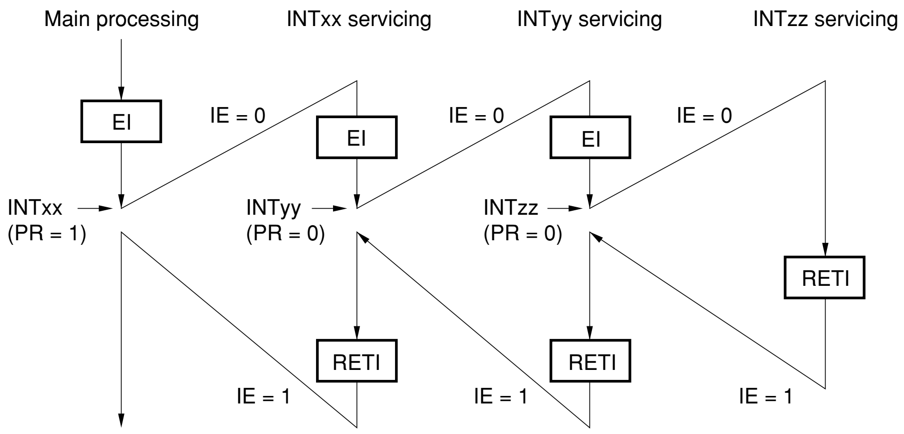
4. PR is a flag contained in PR0L, PR0H, and PR1L.

PR = 0: Higher priority level

PR = 1: Lower priority level

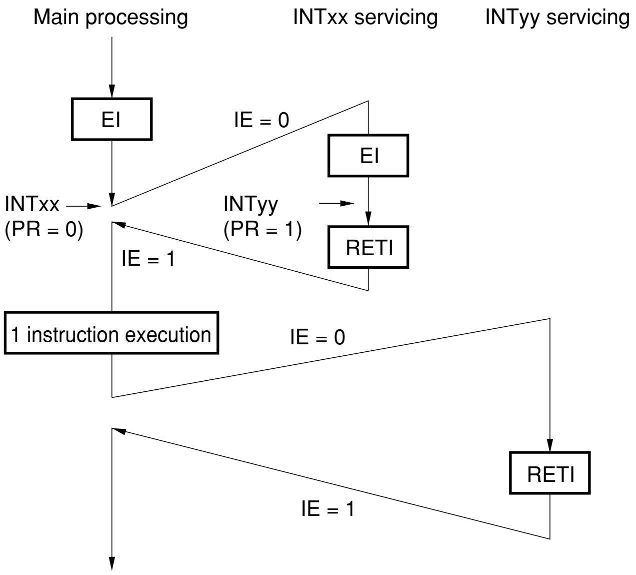
Figure 19-13. Nesting Examples (1/2)

Example 1. Nesting occurs twice



During servicing of interrupt INTxx, two interrupt requests, INTyy and INTzz, are acknowledged, and nesting takes place. Before each interrupt request is acknowledged, the EI instruction must always be issued to enable interrupt request acknowledgment.

Example 2. Nesting does not occur due to priority control

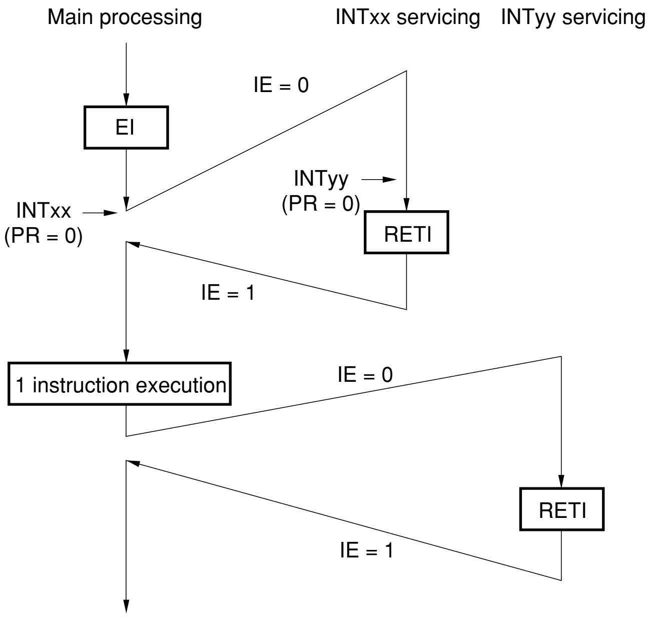


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and nesting does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 0: Higher priority level
PR = 1: Lower priority level
IE = 0: Interrupt request acknowledgment disabled

Figure 19-13. Nesting Examples (2/2)

Example 3. Nesting does not occur because interrupt is not enabled



Interrupt is not enabled during servicing of interrupt INTxx (EI instruction is not issued), therefore, interrupt request INTyy is not acknowledged and nesting does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 0: Higher priority level
IE = 0: Interrupt request acknowledgment disabled

19.4.5 Interrupt request hold

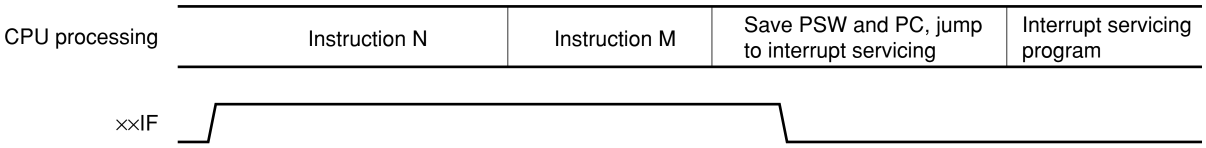
There are instructions where, even if an interrupt request is issued for them while another instruction is being executed, request acknowledgment is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- 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 to 0. Therefore, even if a maskable interrupt request is generated during execution of the BRK instruction, the interrupt request is not acknowledged. However, a non-maskable interrupt request is acknowledged.

Figure 19-14 shows the timing at which interrupt requests are held pending.

Figure 19-14. Interrupt Request Hold


CPU processing
Instruction N
Instruction M
Save PSW and PC, jump to interrupt servicing
Interrupt servicing program
××IF

Remarks 1. Instruction N: Interrupt request hold instruction

2. Instruction M: Instruction other than interrupt request hold instruction

3. The ××PR (priority level) values do not affect the operation of ××IF (interrupt request).

CHAPTER 20 STANDBY FUNCTION

20.1 Standby Function and Configuration

20.1.1 Standby function

The standby function is designed to reduce power consumption of the system. The following two modes are available.

(1) HALT mode

HALT instruction execution sets the HALT mode. The HALT mode is intended to stop the CPU operation clock. The system clock oscillator continues oscillating. In this mode, power consumption is not reduced as much as in the STOP mode. However, the HALT mode is effective to restart operation immediately upon interrupt request and to carry out intermittent operations such as watch applications.

(2) STOP mode

STOP instruction execution sets the STOP mode. In the STOP mode, the main system clock oscillator stops, stopping the whole system, thereby considerably reducing the CPU power consumption.

Data memory low-voltage hold (down to V_{DD} = 1.6 V) is possible. Thus, the STOP mode is effective to hold data memory contents with ultra-low power consumption.

Because this mode can be cleared upon interrupt request, it enables intermittent operations to be carried out. However, because a wait time is required to secure 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.

In either of these two modes, all the contents of registers, flags and data memory just before the standby mode is set are held. The I/O port output latch and output buffer statuses are also held.

Cautions 1. The STOP mode can be used only when the system operates with the main system clock (subsystem clock oscillation cannot be stopped). The HALT mode can be used with either the main system clock or the subsystem clock.

★ **2. When operation is transferred to the STOP mode, be sure to stop the operation of the peripheral hardware that operates on the main system clock and execute the STOP instruction.**

3. The following sequence is recommended for power consumption reduction of the A/D converter when the standby function is used: First clear bit 7 (ADCS0) of A/D converter mode register 0 (ADM0) to 0 to stop the A/D conversion operation, and then execute the HALT or STOP instruction.

20.1.2 Standby function control register

The wait time after the STOP mode is released upon interrupt request is controlled with the oscillation stabilization time select register (OSTS).

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

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

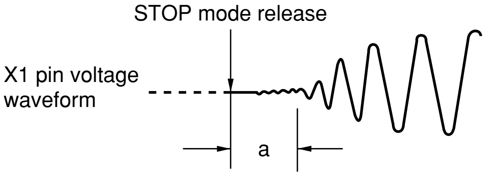
Figure 20-1. Format of Oscillation Stabilization Time Select Register (OSTS)

Address: FFFAH After reset: 04H R/W

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

OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection
0	0	0	2^{12}/fx (410 μs)
0	0	1	2^{14}/fx (1.64 ms)
0	1	0	2^{15}/fx (3.28 ms)
0	1	1	2^{16}/fx (6.55 ms)
1	0	0	2^{17}/fx (13.1 ms)
Other than above			Setting prohibited

Caution The wait time after the STOP mode is released does not include the time (see "a" in the illustration below) from STOP mode release to clock oscillation start. The time is not included either by $\overline{\text{RESET}}$ input or by interrupt request generation.



Remarks 1. fx: Main system clock oscillation frequency

2. Values in parentheses are for operation with fx = 10 MHz.

20.2 Standby Function Operations

20.2.1 HALT mode

(1) HALT mode setting and operating statuses

The HALT mode is set by executing the HALT instruction. It can be set with the main system clock or the subsystem clock.

The operating statuses in the HALT mode are described below.

Table 20-1. HALT Mode Operating Statuses

HALT Mode Setting	During HALT Instruction Execution Using Main System Clock		During HALT Instruction Execution Using Subsystem Clock	
Item	Without Subsystem Clock Note 1	With Subsystem Clock Note 2	With Main System Clock Oscillation	With Main System Clock Oscillation Stopped
Clock generator	Both main system clock and subsystem clock can be oscillated. Clock supply to CPU stops.			
CPU	Operation stops.			
Port (output latch)	Status before HALT mode setting is held.			
×4 subsystem clock multiplication circuit	Operation stops.			
16-bit timer/event counter 0	Operable			Operation stops.
8-bit timer A0	Operable			Operable when INTTMB0, carrier clock, and timer B0 are selected as count clock
8-bit timer/event counter B0	Operable			Operable when TMIB0 is selected as count clock.
8-bit timer/event counters 50, 51	Operable			Operable when TI50 and TI51 are selected as count clock.
Watch timer	Operable when $f_X/2^8$ is selected as count clock.	Operable		Operable when f_{XT} is selected as count clock.
Watchdog timer	Operable		Operation stops.	
Clock output	Operable			Operable when f_{XT} is selected as count clock.
A/D converter	Operation stops.			
Serial interface SIO3	Operable			Operable with external SCK.
Serial interface CSI1				
Serial interface UART0				Operation stops.
Serial interface IIC0 Note 3				
LCD controller/driver	Operable when $f_X/2^6$ to $f_X/2^8$ is selected as count clock.	Operable		Operable when f_{XT} is selected as count clock.

Notes 1. Including case when external clock is not supplied.

2. Including case when external clock is supplied.

3. μPD780344Y, 780354Y Subseries only

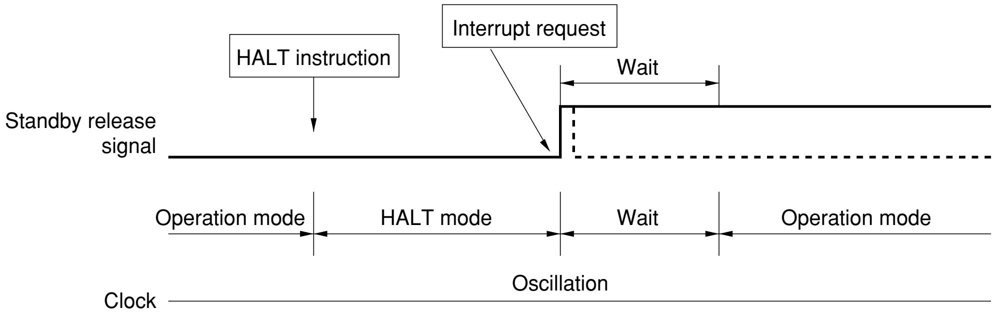
(2) HALT mode release

The HALT mode can be released with the following three types of sources.

(a) Release by unmasked interrupt request

When an unmasked interrupt request is generated, the HALT mode is released. If interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the next address instruction is executed.

Figure 20-2. HALT Mode Release by Interrupt Request Generation



Remarks 1. The broken line indicates the case when the interrupt request which has released the standby mode is acknowledged.

2. Wait times are as follows:

- When vectored interrupt servicing is carried out: 8 or 9 clocks
- When vectored interrupt servicing is not carried out: 2 or 3 clocks

(b) Release by non-maskable interrupt request

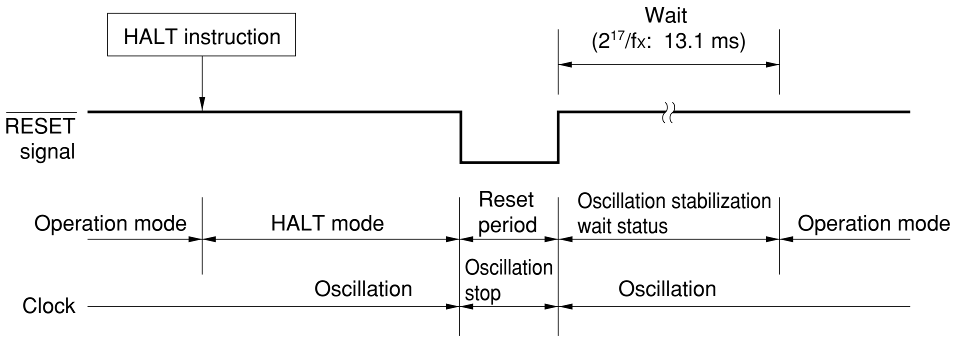
When a non-maskable interrupt request is generated, the HALT mode is released and vectored interrupt servicing is carried out whether interrupt acknowledgment is enabled or disabled.

★ However, a non-maskable interrupt request is not generated during subsystem clock operation.

(c) Release by $\overline{\text{RESET}}$ input

When the $\overline{\text{RESET}}$ signal is input, HALT mode is released. And, as in the case with normal reset operation, the program is executed after branch to the reset vector address.

Figure 20-3. HALT Mode Release by $\overline{\text{RESET}}$ Input



Remarks 1. fx: Main system clock oscillation frequency

2. Values in parentheses are for operation with fx = 10 MHz.

Table 20-2. Operation After HALT Mode Release

Release Source	MK××	PR××	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Next address instruction execution
	0	0	1	×	Interrupt servicing execution
	0	1	0	1	Next address instruction execution
	0	1	×	0	
	0	1	1	1	Interrupt servicing execution
	1	×	×	×	HALT mode hold
Non-maskable interrupt request	—	—	×	×	Interrupt servicing execution
$\overline{\text{RESET}}$ input	—	—	×	×	Reset processing

×: Don't care

★ **Caution In flash memory versions (μPD78F0354, 78F0354A, 78F0354Y, 78F0354AY), when the subclock multiplied by 4 is used, set one NOP instruction immediately after HALT instruction execution to release HALT mode set during servicing of an interrupt with a lower priority (setting is unnecessary for mask ROM versions).**

★ **Table 20-3. HALT Mode Release Condition and Necessity of NOP Instruction Setting When Subclock Multiplied by 4 Is Used (Flash Memory Products Only)**

Clock	Status	Release Condition	NOP Instruction Setting (One Instruction)
Subclock multiplied by 4 is used	HALT mode during main processing	$\overline{\text{RESET}}$ input	Unnecessary
		Interrupt	
	HALT mode during interrupt servicing	$\overline{\text{RESET}}$ input	
		Interrupt with higher priority than the interrupt being processed	
		Interrupt with lower priority than the interrupt being processed	Necessary

20.2.2 STOP mode

(1) STOP mode setting and operating statuses

The STOP mode is set by executing the STOP instruction. It can be set only with the main system clock.

Cautions 1. When the STOP mode is set, the X2 pin is internally connected to V_{DD1} via a pull-up resistor to minimize the leakage current at the crystal oscillator. Thus, do not use the STOP mode in a system where an external clock is used for the main system clock.

2. Because the interrupt request signal is used to clear the standby mode, if there is an interrupt source with the interrupt request flag set and the interrupt mask flag reset, the standby mode is immediately cleared if set. Thus, the STOP mode is reset to the HALT mode immediately after execution of the STOP instruction. After the wait set using the oscillation stabilization time select register (OSTS), the operation mode is set.

The operating statuses in the STOP mode are described below.

Table 20-4. STOP Mode Operating Statuses

STOP Mode Setting Item	With Subsystem Clock	Without Subsystem Clock
Clock generator	Only main system clock oscillation is stopped.	
CPU	Operation stops.	
Port (output latch)	Status before STOP mode setting is held.	
×4 subsystem clock multiplication circuit	Operation stops.	
16-bit timer/event counter 0	Operation stops.	
8-bit timer A0	Operable when INTTMB0, carrier clock, and timer B0 are selected as count clock.	
8-bit timer/event counter B0	Operable when TMIB0 is selected as count clock.	
8-bit timer/event counters 50, 51	Operable when TI50 and TI51 are selected as count clock.	
Watch timer	Operable when f_{XT} is selected as count clock.	Operation stops.
Watchdog timer	Operation stops.	
Clock output	PCL is low	
A/D converter	Operation stops.	
Serial interface SIO3	Operable only when externally input clock is selected as serial clock.	
Serial interface CSI1		
Serial interface UART0	Operation stops (transmit shift register 0 (TXS0), receive shift register 0 (RX0), and receive buffer register 0 (RXB0) hold the value just before the clock stop).	
Serial interface IIC0[Note]	Operation stops.	
LCD controller/driver	Operable when f_{XT} is selected as count clock.	Operation stops.

Note μPD780344Y, 780354Y Subseries only

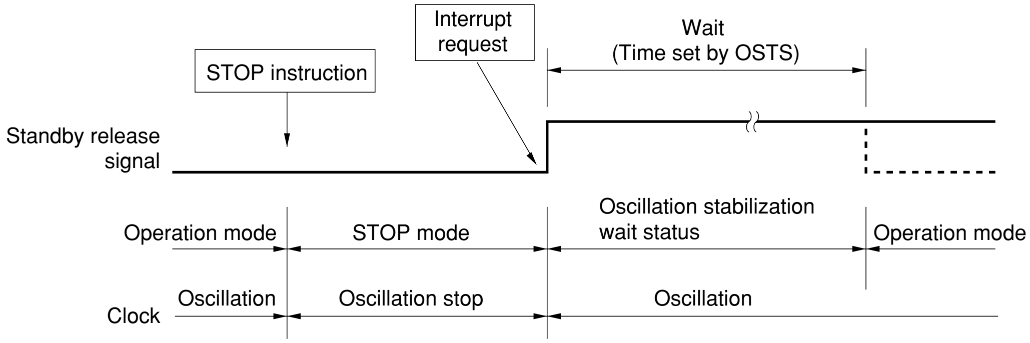
(2) STOP mode release

The STOP mode can be released by the following two types of sources.

(a) Release by unmasked interrupt request

When an unmasked interrupt request is generated, the STOP mode is released. If interrupt acknowledge is enabled after the lapse of oscillation stabilization time, vectored interrupt service is carried out. If interrupt acknowledge is disabled, the next address instruction is executed.

Figure 20-4. STOP Mode Release by Interrupt Request Generation

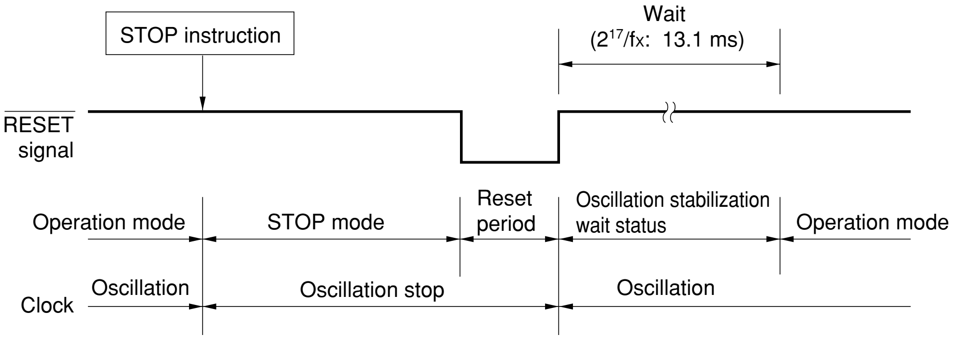


Remark The broken lines indicate the case when the interrupt request which has cleared the standby status is acknowledged.

(b) Release by $\overline{\text{RESET}}$ input

The STOP mode is released when $\overline{\text{RESET}}$ signal is input, and after the lapse of the oscillation stabilization time, the reset operation is carried out.

Figure 20-5. STOP Mode Release by $\overline{\text{RESET}}$ Input



Remarks 1. fx: Main system clock oscillation frequency

2. Values in parentheses are for operation with fx = 10 MHz.

Table 20-5. Operation After STOP Mode Release

Release Source	MK××	PR××	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Next address instruction execution
	0	0	1	×	Interrupt service execution
	0	1	0	1	Next address instruction execution
	0	1	×	0	
	0	1	1	1	Interrupt service execution
	1	×	×	×	STOP mode hold
$\overline{\text{RESET}}$ input	—	—	×	×	Reset processing

×: Don't care

CHAPTER 21 RESET FUNCTION

21.1 Reset Function

The following two operations are available to generate the reset signal.

(1) External reset input via $\overline{\text{RESET}}$ pin

(2) Internal reset by watchdog timer program loop time detection

External reset and internal reset have no functional differences. In both cases, program execution starts at the address at 0000H and 0001H by $\overline{\text{RESET}}$ input. When a low level is input to the $\overline{\text{RESET}}$ pin or the watchdog timer overflows, a reset is applied and each hardware is set to the status shown in Table 21-1. Each pin is high impedance during reset input or during oscillation stabilization time just after reset release.

When a high level is input to the $\overline{\text{RESET}}$ pin, the reset is released and program execution starts after the lapse of oscillation stabilization time 2^{17}/fx. The reset applied by watchdog timer overflow is automatically released after reset and program execution starts after the lapse of oscillation stabilization time 2^{17}/fx (see **Figures 21-2** to **21-4**).

Cautions 1. For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin.

2. During reset input, main system clock oscillation remains stopped but subsystem clock oscillation continues.

3. When the STOP mode is released by reset, the STOP mode contents are held during reset input. However, the port pin becomes high-impedance.

Figure 21-1. Reset Function Block Diagram

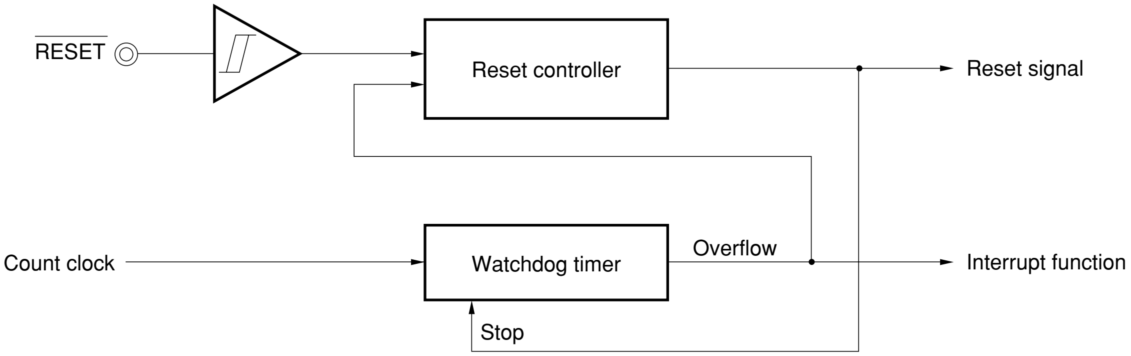


Figure 21-2. Timing of Reset by $\overline{\text{RESET}}$ Input

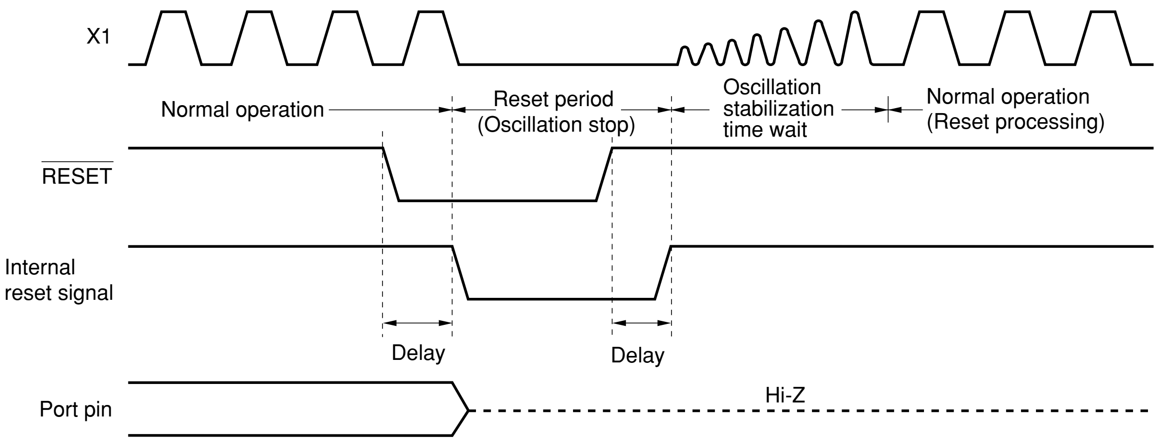


Figure 21-3. Timing of Reset Due to Watchdog Timer Overflow

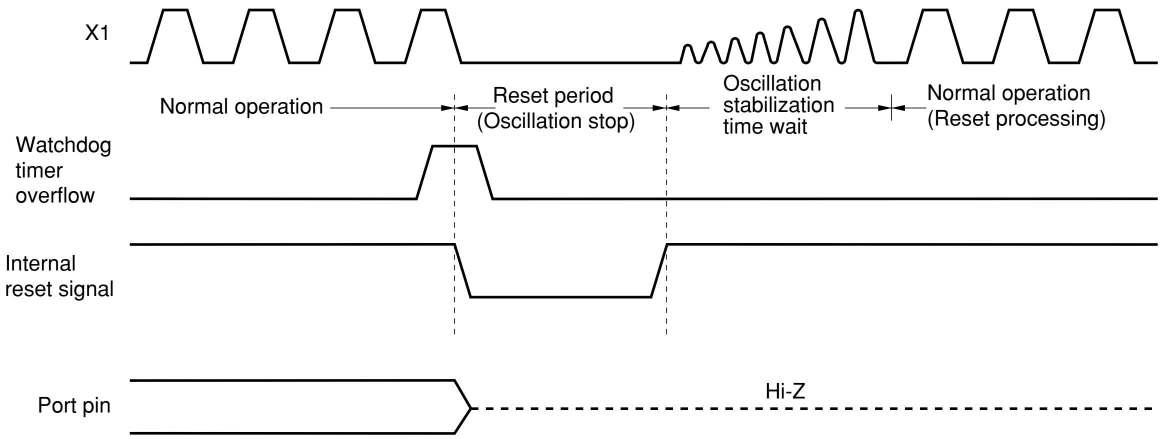


Figure 21-4. Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input

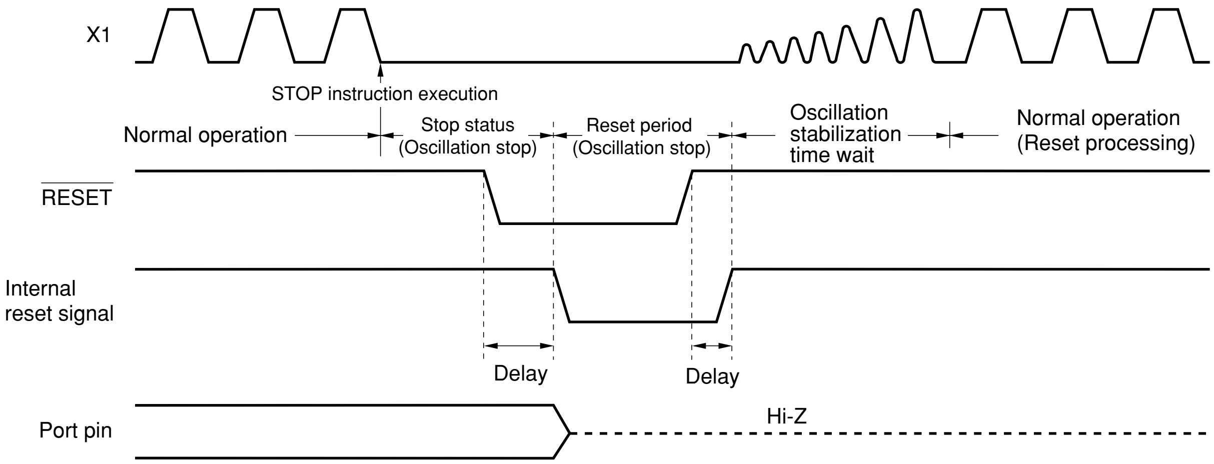


Table 21-1. Hardware Statuses After Reset (1/2)

Hardware		Status After Reset
Program counter (PC)Note 1		Contents of reset vector table (0000H, 0001H) are set.
Stack pointer (SP)		Undefined
Program status word (PSW)		02H
RAM	Data memory	UndefinedNote 2
	General-purpose register	UndefinedNote 2
Port (output latch)		00H
Port mode registers (PM0, PM2 to PM4, PM8 to PM11)		FFH
Pull-up resistor option registers (PU0, PU2 to PU4)		00H
Pin function switching registers (PF8 to PF11)		00H
Processor clock control register (PCC)		04H
Memory size switching register (IMS)		CFHNote 3
Internal expansion RAM size switching register (IXS)		0CHNote 4
Memory expansion mode register (MEM)		00H
Subclock select register (SSCK)		00H
Oscillation stabilization time select register (OSTS)		04H
16-bit timer/event counter 0	Timer counter 0 (TM0)	0000H
	Capture/compare registers 00, 01 (CR00, CR01)	Undefined
	Prescaler mode register 0 (PRM0)	00H
	Mode control register 0 (TMC0)	00H
	Capture/compare control register 0 (CRC0)	00H
	Output control register 0 (TOC0)	00H
8-bit timer/event counters A0, B0	Timer counters (TMA0, TMB0)	00H
	Compare registers (CRA0, CRB0, CRHB0)	Undefined
	Mode control registers (TMCA0, TMCB0)	00H
	Carrier generator output control register B0 (TCAB0)	00H
8-bit timer/event counters 50, 51	Timer counters (TM50, TM51)	00H
	Compare registers (CR50, CR51)	Undefined
	Clock select registers (TCL50, TCL51)	00H
	Mode control registers (TMC50, TMC51)	00H

Notes 1. During reset input or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

2. When a reset is executed in the standby mode, the pre-reset status is held even after reset.

3. Although the initial value is CFH, use the following value to be set for each version.
μPD780343, 780353, 780343Y, 780353Y: 46H
μPD780344, 780354, 780344Y, 780354Y: 48H
μPD78F0354, 78F0354A, 78F0354Y, 78F0354AY: C8H or value for mask ROM versions

4. Although the initial value is 0CH, use this register with a setting of 0BH.

Table 21-1. Hardware Statuses After Reset (2/2)

Hardware		Status After Reset
Watch timer	Operation mode register 0 (WTNM0)	00H
	Interrupt time select register (WTIM)	00H
Watchdog timer	Clock select register (WDCS)	00H
	Mode register (WDTM)	00H
Clock output controller	Clock output select register (CKS)	00H
A/D converter	Conversion result register 0 (ADCR0)Note 1	0000H
	Conversion result register 1 (ADCR1)Note 2	00H
	Mode register 0 (ADM0)	00H
	Analog input channel specification register 0 (ADS0)	00H
Serial interface SIO3	Shift register 3 (SIO3)	Undefined
	Operation mode register 3 (CSIM3)	00H
Serial interface CSI1	Shift register 1 (SIO1)	Undefined
	Transmit buffer register 1 (SOTB1)	Undefined
	Operation mode register 1 (CSIM1)	00H
	Clock select register 1 (CSIC1)	10H
Serial interface UART0	Asynchronous serial interface mode register 0 (ASIM0)	00H
	Asynchronous serial interface status register 0 (ASIS0)	00H
	Baud rate generator control register 0 (BRGC0)	00H
	Transmit shift register 0 (TXS0)	FFH
	Receive buffer register 0 (RXB0)	
Serial interface IIC0Note 3	Transfer clock select register 0 (IICCL0)	00H
	Shift register 0 (IIC0)	00H
	Control register 0 (IICC0)	00H
	Status register 0 (IICS0)	00H
	Slave address register 0 (SVA0)	00H
	Function expansion register 0 (IICX0)	00H
LCD controller/driver	Display mode register 3 (LCDM3)	00H
	Clock control register 3 (LCDC3)	00H
	Gain adjust register 0 (VLCG0)	00H
	Static/dynamic display switching register 3 (SDSEL3)	00H
Interrupt	Request flag registers (IF0L, IF0H, IF1L)	00H
	Mask flag registers (MK0L, MK0H, MK1L)	FFH
	Priority specification flag registers (PR0L, PR0H, PR1L)	FFH
	External interrupt rising edge enable register (EGP)	00H
	External interrupt falling edge enable register (EGN)	00H
ROM correction	Correction address registers (CORAD0, CORAD1)	0000H
	Correction control register (CORCN)	00H

Notes 1. μPD780354, 780354Y Subseries only

2. μPD780344, 780344Y Subseries only

3. μPD780344Y, 780354Y Subseries only

CHAPTER 22 ROM CORRECTION

22.1 ROM Correction Functions

★ In the μPD780344, 780354, 780344Y, 780354Y Subseries, part of a program in the mask ROM or flash memory can be released with a program in the internal expansion RAM.

★ Instruction bugs found in the mask ROM or flash memory can be avoided, and program flow can be changed by using the ROM correction function.

★ ROM correction can correct two places (max.) of the internal ROM or on-chip flash memory (program).

Caution ROM correction cannot be emulated by the in-circuit emulator (IE-78K0-NS, IE-78K0-NS-A).

22.2 ROM Correction Configuration

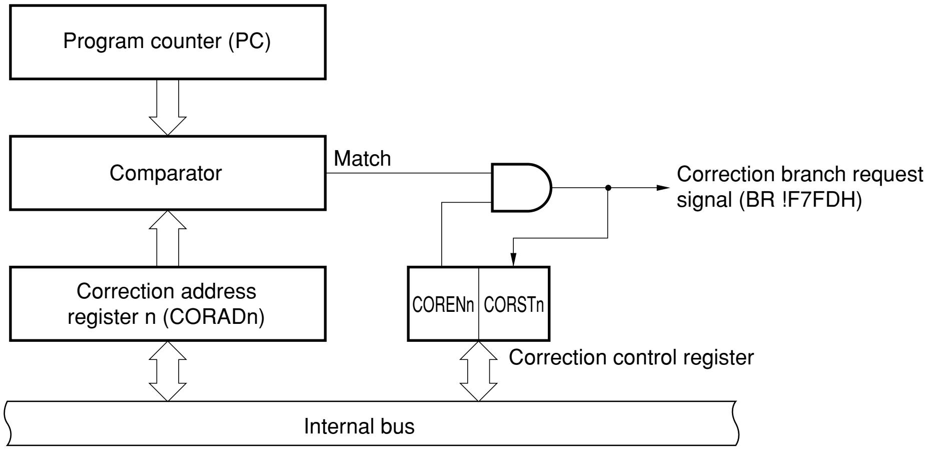
ROM correction consists of the following hardware.

Table 22-1. ROM Correction Configuration

Item	Configuration
Registers	Correction address registers 0 and 1 (CORAD0, CORAD1)
Control register	Correction control register (CORCN)

Figure 22-1 shows a block diagram of ROM correction.

Figure 22-1. ROM Correction Block Diagram


Program counter (PC)
Comparator
Match
Correction branch request signal (BR !F7FDH)
Correction address register n (CORADn)
CORENn
CORSTn
Correction control register
Internal bus

Remark n = 0, 1

(1) Correction address registers 0 and 1 (CORAD0, CORAD1)

★ These registers set the start address (correction address) of the instruction(s) to be corrected in the mask ROM or flash memory.

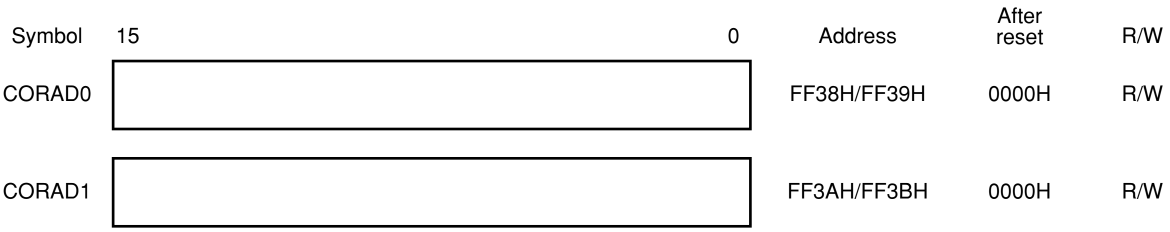
ROM correction corrects two places (max.) of the program. Addresses are set to two registers, CORAD0 and CORAD1. If only one place needs to be corrected, set the address to either of the registers.

ROM correction for the start address specified in CORAD0 and CORAD1 is valid when bit 1 (COREN0) and bit 3 (COREN1) of the correction control register (CORCN) is 1.

CORAD0 and CORAD1 are set by a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CORAD0 and CORAD1 to 0000H.

Figure 22-2. Format of Correction Address Registers 0 and 1



Symbol	15 … 0	Address	After reset	R/W
CORAD0		FF38H/FF39H	0000H	R/W
CORAD1		FF3AH/FF3BH	0000H	R/W

Cautions 1. Set the CORAD0 and CORAD1 when bit 1 (COREN0) and bit 3 (COREN1) of the correction control register (CORCN) are 0.

2. Only start addresses where operation codes are stored can be set in CORAD0 and CORAD1.

3. Do not set the following addresses to CORAD0 and CORAD1.

- **Address value in table area of table reference instruction (CALLT instruction): 0040H to 007FH**
- **Address value in vector table area: 0000H to 003FH**

(2) Comparator

The comparator always compares the correction address value set in correction address registers 0 and 1 (CORAD0, CORAD1) with the fetch address value. When bit 1 (COREN0) or bit 3 (COREN1) of the correction control register (CORCN) is 1 and the correction address matches the fetch address value, the correction branch request signal (BR !F7FDH) is generated from the ROM correction circuit.

22.3 ROM Correction Control Register

ROM correction is controlled by the correction control register (CORCN).

(1) Correction control register (CORCN)

This register controls whether or not the correction branch request signal is generated when the fetch address matches the correction address set in correction address registers 0 and 1. The correction control register consists of correction enable flags (COREN0, COREN1) and correction status flags (CORST0, CORST1). The correction enable flags enable or disable the comparator match detection signal, and correction status flags show the values are matched.

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

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

Figure 22-3. Format of Correction Control Register (CORCN)

Address: FF8AH After reset: 00H R/W[Note]

Symbol	7	6	5	4	$\boxed{3}$	2	$\boxed{1}$	0
CORCN	0	0	0	0	COREN1	CORST1	COREN0	CORST0

COREN1	Correction address register 1 and fetch address match detection control
0	Disabled
1	Enabled

CORST1	Correction address register 1 and fetch address match detection flag
0	Not detected
1	Detected

COREN0	Correction address register 0 and fetch address match detection control
0	Disabled
1	Enabled

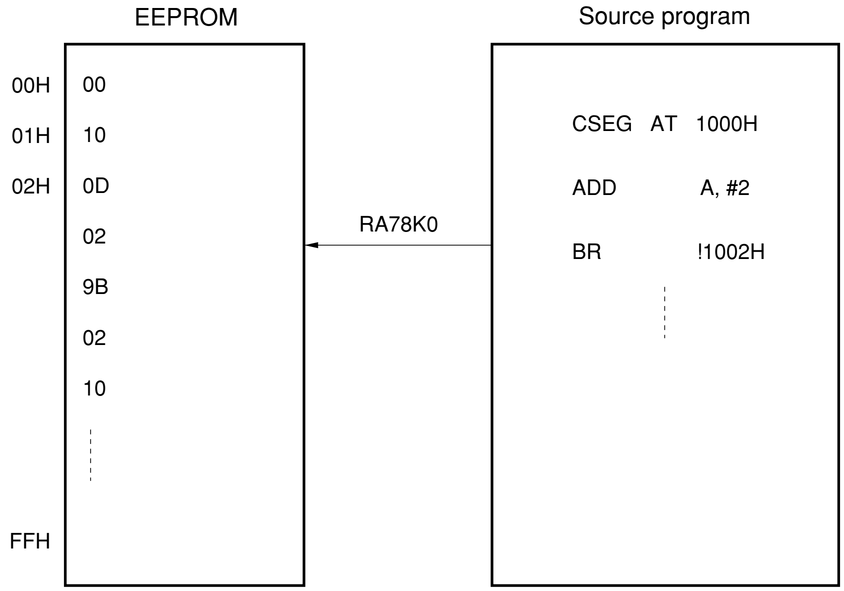
CORST0	Correction address register 0 and fetch address match detection flag
0	Not detected
1	Detected

Note Bits 0 and 2 are read-only bits. Bits 0 and 2 are set (1) only when a match is detected by the comparator. Do not set these bits to 1 by software.

22.4 ROM Correction Application

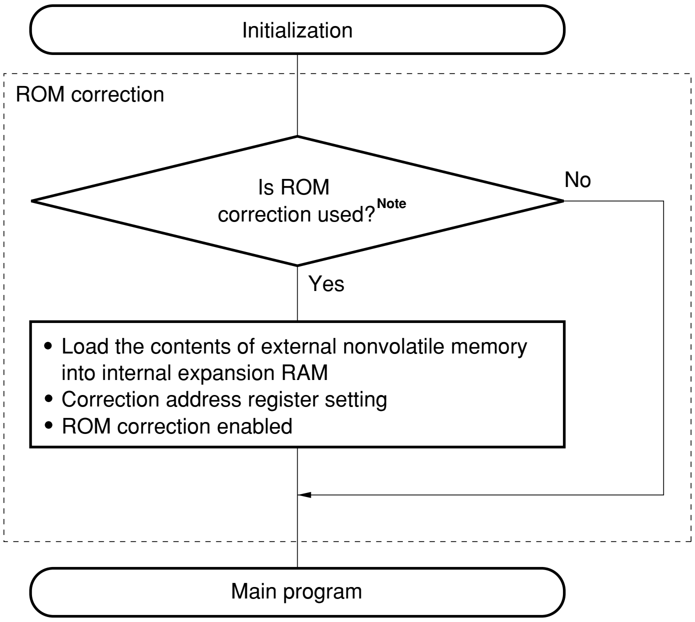
(1) Store the correction address and instruction after correction (patch program) to nonvolatile memory (such as EEPROM^{TM}) outside the microcontroller.
When two places should be corrected, store the branch destination judgment program as well. The branch destination judgment program checks which one of the addresses set to correction address register 0 and 1 (CORAD0 and CORAD1) generates the correction branch.

Figure 22-4. Storing Example to EEPROM (When One Place Is Corrected)



(2) Assemble in advance the initialization routine as shown in Figure 22-5 to correct the program.

Figure 22-5. Initialization Routine


Initialization
ROM correction
Is ROM correction used?Note
No
Yes
• Load the contents of external nonvolatile memory into internal expansion RAM
• Correction address register setting
• ROM correction enabled
Main program

Note Whether ROM correction is used or not should be judged by the port input level. For example, when the P20 input level is high, ROM correction is used; otherwise, it is not used.

(3) After reset, store the contents that were previously stored in the external nonvolatile memory in initialization routine for ROM correction of the user to internal expansion RAM (see **Figure 22-5**).
Set the start address of the instruction to be corrected to CORAD0 and CORAD1, and set bits 1 and 3 (COREN0, COREN1) of the correction control register (CORCN) to 1.

(4) Set the entire-space branch instruction (BR !addr16) to the specified address (F7FDH) of the internal expansion RAM in the main program.

(5) After the main program is started, the fetch address value and the values set in CORAD0 and CORAD1 are always compared by the comparator in the ROM correction circuit. When these values match, the correction branch request signal is generated. Simultaneously, the corresponding correction status flag (CORST0 or CORST1) is set to 1.

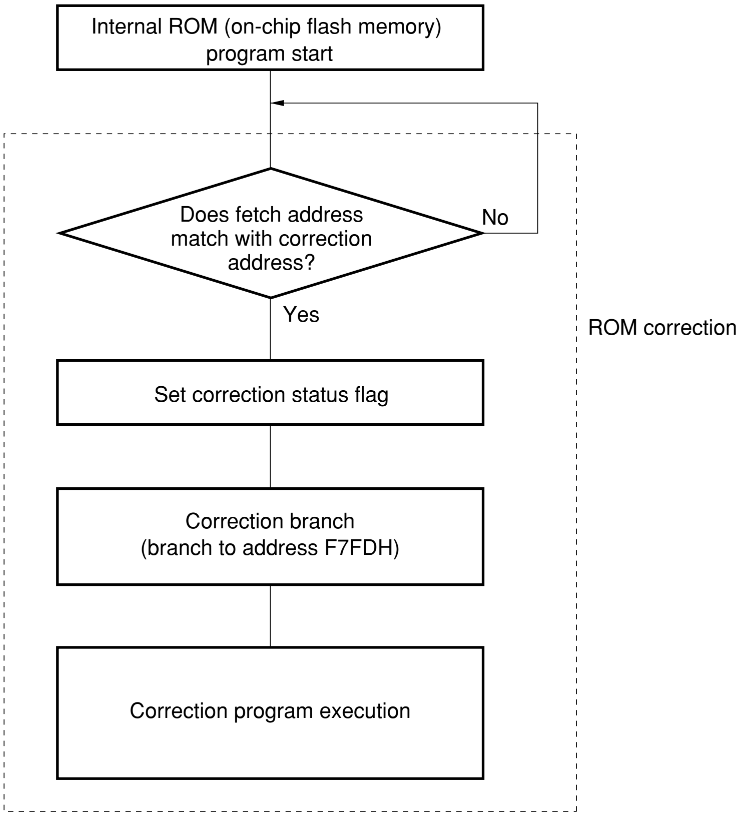
(6) Branch to the address F7FDH by the correction branch request signal.

(7) Branch to the internal expansion RAM address set in the main program by the entire-space branch instruction of the address F7FDH.

(8) When one place is corrected, the correction program is executed.
When two places are corrected, the correction status flag is checked with the branch destination judgment program, and branches to the correction program.

★

Figure 22-6. ROM Correction Operation

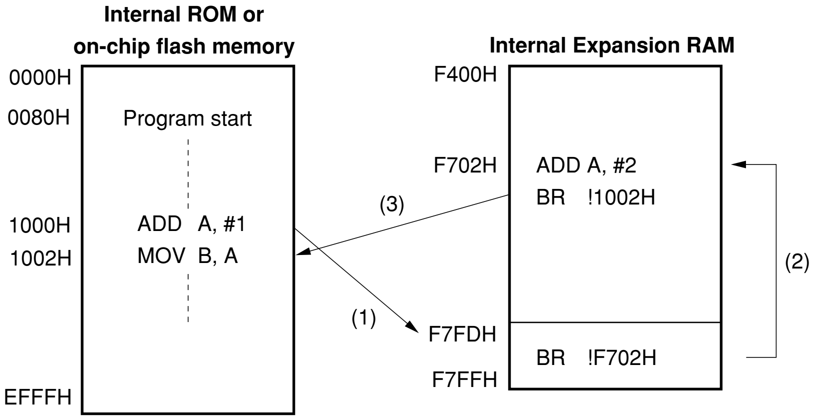

Internal ROM (on-chip flash memory) program start
Does fetch address match with correction address?
No
Yes
ROM correction
Set correction status flag
Correction branch (branch to address F7FDH)
Correction program execution

22.5 ROM Correction Example

An example of ROM correction when the instruction at address 1000H “ADD A, #1” is changed to “ADD A, #2” is as follows.

Figure 22-7. ROM Correction Example

★



(1) Branches to address F7FDH when the preset value 1000H in correction address register 0, 1 (CORAD0, CORAD1) matches the fetch address value after the main program is started.

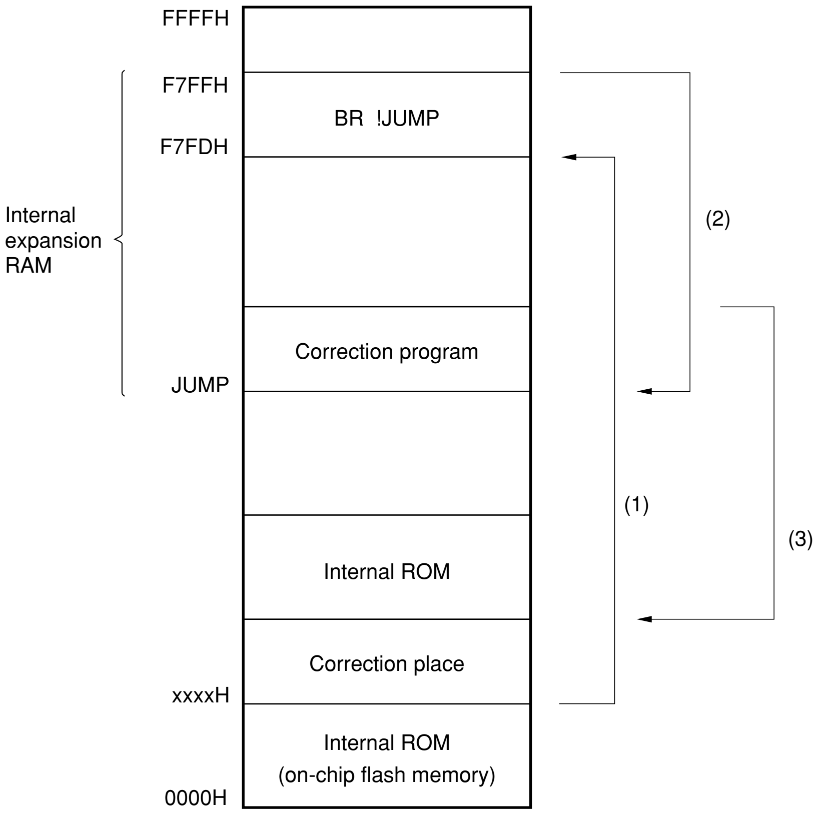
(2) Branches to any address (address F702H in this example) by setting the entire-space branch instruction (BR !addr16) to address F7FDH in the main program.

★ (3) Returns to the internal ROM (on-chip flash memory) program after executing the substitute instruction ADD A, #2.

22.6 Program Execution Flow

Figures 22-8 and 22-9 show the program transition diagrams when ROM correction is used.

Figure 22-8. Program Transition Diagram (When One Place Is Corrected)



(1) Branches to address F7FDH when fetch address matches correction address

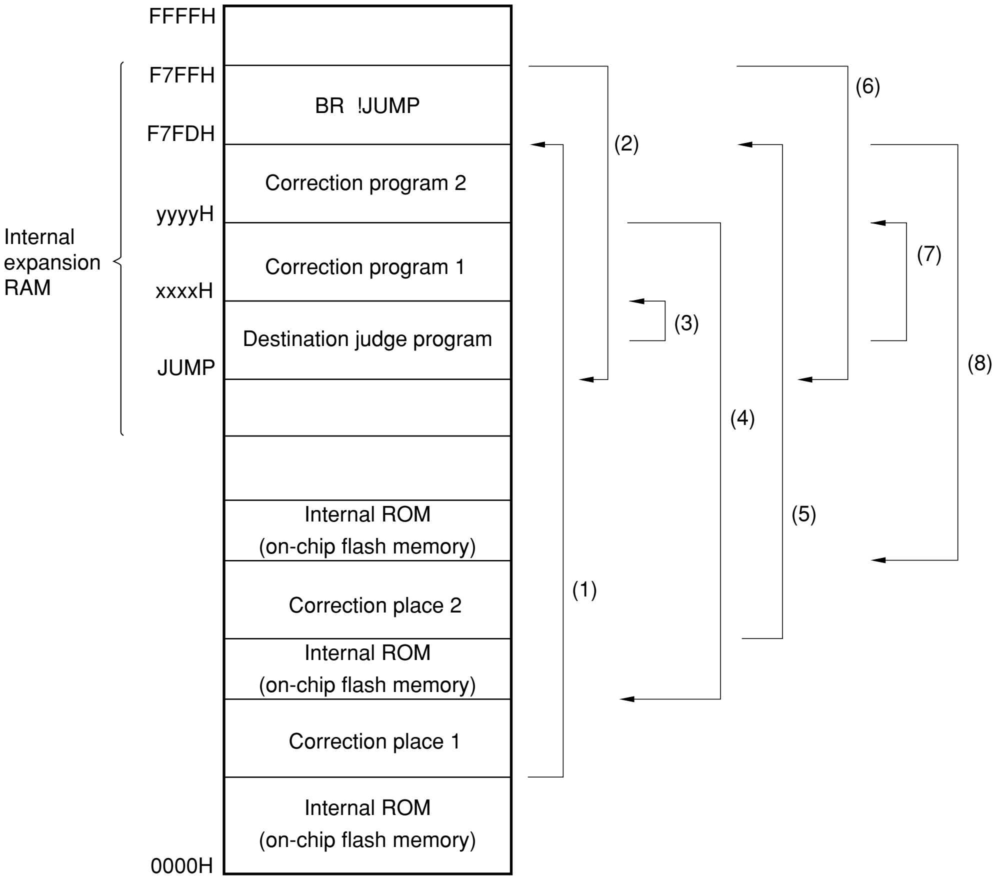
(2) Branches to correction program

★ (3) Returns to internal ROM program (on-chip flash memory)

Caution Do not use internal high-speed RAM and LCD display RAM for ROM correction area.

Remark JUMP: Correction program start address

Figure 22-9. Program Transition Diagram (When Two Places Are Corrected)



(1) Branches to address F7FDH when fetch address matches correction address

(2) Branches to branch destination judgment program

(3) Branches to correction program 1 by branch destination judgment program (BTCLR !CORST0, $xxxxH)

★ (4) Returns to internal ROM (on-chip flash memory) program

(5) Branches to address F7FDH when fetch address matches correction address

(6) Branches to branch destination judgment program

(7) Branches to correction program 2 by branch destination judgment program (BTCLR !CORST1, $yyyyH)

★ (8) Returns to internal ROM (on-chip flash memory) program

Caution Do not use internal high-speed RAM and LCD display RAM for ROM correction area.

Remark JUMP: Correction program start address

22.7 Cautions on ROM Correction

(1) Address values set in correction address registers 0 and 1 (CORAD0 and CORAD1) must be addresses where instruction codes are stored. In addition, address values to be set must be the start address of the instruction code.

(2) Correction address registers 0 and 1 (CORAD0 and CORAD1) should be set when the correction enable flags (COREN0, COREN1) are “0” (when correction branch processing is disabled). If an address is set to CORAD0 or CORAD1 when COREN0 or COREN1 is 1 (when the correction branch is in an enabled state), the correction branch may start with the different address from the set address value.

(3) Do not set the address value of instruction immediately after the instruction that sets the correction enable flag (COREN0, COREN1) to 1, to correction address register 0 or 1 (CORAD0, CORAD1); otherwise the correction branch may not start.

(4) Do not set the address value in table area of table reference instruction (CALLT instruction) (0040H to 007FH), and the address value in vector table area (0000H to 003FH) to correction address registers 0 and 1 (CORAD0, CORAD1).

(5) Do not set two addresses immediately after the instructions shown below to correction address registers 0 and 1 (CORAD0, CORAD1) (that is, when the mapped terminal address of these instructions is N, do not set the address values of N+1 and N+2).

- RET
- RETI
- RETB
- BR $addr16
- STOP
- HALT

(6) Do not set the address value set to correction address registers 0 and 1 (CORAD0 and CORAD1) to F7FDH.

CHAPTER 23 μPD78F0354, 78F0354A, 78F0354Y, 78F0354AY

The μPD78F0354, 78F0354A, 78F0354Y, and 78F0354AY are provided as the flash memory versions of the μPD780344, 780354, 780344Y, 780354Y Subseries.

The μPD78F0354, 78F0354A, 78F0354Y, and 78F0354AY incorporate flash memory on which a program can be written, erased and overwritten while mounted on the board.

Data can be written to the flash memory with the memory mounted on the target system (on-board). To do this, connect the dedicated flash programmer to the target system.

Using flash memory in a development environment or application enables the following.

- Software can be modified after soldering the μPD78F0354, 78F0354A, 78F0354Y, and 78F0354AY to the target system.
- Many products can be produced in small quantities by distinguishing the software of each.
- Data can be easily adjusted when mass production is started.

Table 23-1 lists the differences between the μPD78F0354, 78F0354A, 78F0354Y, 78F0354AY, and the mask ROM versions.

Table 23-1. Differences Between μPD78F0354, 78F0354A, 78F0354Y, 78F0354AY, and Mask ROM Versions

Item		Flash Memory Versions		Mask ROM Versions			
		μPD78F0354, 78F0354A	μPD78F0354Y 78F0354AY	μPD780343, μPD780344	μPD780353, μPD780354	μPD780343Y, μPD780344Y	μPD780353Y, μPD780354Y
Internal memory	ROM structure	Flash memory		Mask ROM			
	ROM capacity	32 KB[Note]		μPD780343, 780353, 780343Y, 780353Y: 24 KB μPD780344, 780354, 780344Y, 780354Y: 32 KB			
	High-speed RAM capacity	1,024 bytes[Note]		512 bytes			
	Expansion RAM capacity	512 bytes					
	RAM capacity for LCD display	40 × 8 bits					
Alternate function of pins P30 and P31		Not provided	Provided (SCL0, SDA0)	Not provided		Provided (SCL0, SDA0)	
Pull-up resistors of pins P30 and P31		Not provided		On-chip pull-up resistor can be specified by mask option in 1-bit units		Not provided	
Pull-up resistors of pins P70 to P73		Not provided		On-chip pull-up resistor can be specified by mask option in 1-bit units			
A/D converter resolution		10 bits		8 bits	10 bits	8 bits	10 bits
Serial interface IIC0		Not provided	Provided	Not provided		Provided	
★ Restrictions in HALT mode when ×4 multiplication subclock is used		Refer to **Caution** in **20.2.1 (2) HALT mode release**		None			
IC pin		Not provided		Provided			
V_{PP} pin		Provided		Not provided			
Electrical specifications		Refer to **CHAPTER 25 ELECTRICAL SPECIFICATIONS**					

Note The same capacity as the mask ROM versions can be specified by means of the 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 version.

23.1 Memory Size Switching Register

The μPD78F0354, 78F0354A, 78F0354Y, and 78F0354AY allow users to select the internal memory capacity using the memory size switching register (IMS) so that the same memory map as that of 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 the value of this register to CFH.

Caution Be sure to set IMS to C8H, 46H, or 48H as the initial setting of the program. Reset input initializes IMS to CFH. Be sure to set IMS to C8H, 46H, or 48H after reset.

Figure 23-1. Format of Memory Size Switching Register (IMS)

Address: FFF0H After reset: CFH R/W

Symbol	7	6	5	4	3	2	1	0
IMS	RAM2	RAM1	RAM0	0	ROM3	ROM2	ROM1	ROM0

RAM2	RAM1	RAM0	Internal high-speed RAM capacity selection
0	1	0	512 bytes
1	1	0	1,024 bytes
Other than above			Setting prohibited

ROM3	ROM2	ROM1	ROM0	Internal ROM capacity selection
0	1	1	0	24 KB
1	0	0	0	32 KB
Other than above				Setting prohibited

The IMS settings to obtain the same memory map as mask ROM versions are shown in Table 23-2.

Table 23-2. Memory Size Switching Register Settings

Target Mask ROM Versions	IMS Setting
μPD780343, 780353, 780343Y, 780353Y	46H
μPD780344, 780354, 780344Y, 780354Y	48H

Caution When using the mask ROM versions, be sure to set the value indicated in Table 23-2 to IMS.

23.2 Internal Expansion RAM Size Switching Register

The internal expansion RAM size switching register (IXS) is used to set the internal expansion RAM capacity.
IXS is set by an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets the value of this register to 0CH.

Caution Be sure to set IXS to 0BH as the initial setting of the program. Reset input initializes IXS to 0CH. Be sure to set IXS to 0BH after reset. Set the mask ROM versions in the same manner.

Figure 23-2. Format of Internal Expansion RAM Size Switching Register (IXS)

Address: FFF4H After reset: 0CH R/W

Symbol	7	6	5	4	3	2	1	0
IXS	0	0	0	0	IXRAM3	IXRAM2	IXRAM1	IXRAM0

IXRAM3	IXRAM2	IXRAM1	IXRAM0	Internal expansion RAM capacity selection
1	0	1	1	512 bytes
Other than above				Setting prohibited

23.3 Flash Memory Characteristics

Flash memory programming is performed by connecting a dedicated flash programmer (Flashpro III (part no. FL-PR3, PG-FP3)/Flashpro IV (part no. FL-PR4, PG-FP4)) to the target system with the flash memory mounted on the target system (on-board). A flash memory writing adapter (program adapter), which is a target board used exclusively for programming, is also provided.

★

Remark FL-PR3, FL-PR4, and the program adapter are products made by Naito Densei Machida Mfg. Co., Ltd. (TEL +81-45-475-4191).

Programming using flash memory has the following advantages.

- Software can be modified after the microcontroller is solder-mounted on the target system.
- Distinguishing software facilities low-quantity, varied model production
- Easy data adjustment when starting mass production

23.3.1 Programming environment

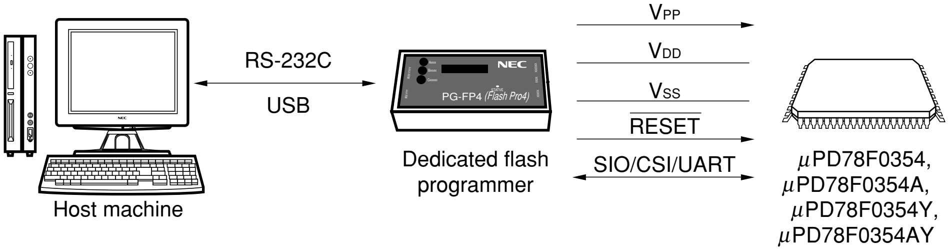
The following shows the environment required for μPD78F0354, 78F0354A, 78F0354Y, 78F0354AY flash memory programming.

When Flashpro III or Flashpro IV is used as a dedicated flash programmer, a host machine is required to control the dedicated flash programmer. Communication between the host machine and flash programmer is performed via RS-232C/USB (Rev. 1.1).

For details, refer to the manuals for Flashpro III/Flashpro IV.

Remark USB is supported by Flashpro IV only.

Figure 23-3. Environment for Writing Program to Flash Memory



23.3.2 Communication mode

Use the communication mode shown in Table 23-3 to perform communication between the dedicated flash programmer and μPD78F0354, 78F0354A, 78F0354Y, 78F0354AY.

★ **Table 23-3. Communication Mode List**

Communication Mode	Standard (TYPE) Setting[Note 1]					Pins Used	Number of V_{PP} Pulses
	Port (COMM PORT)	Speed (SIO CLOCK)	On Target (CPU CLOCK)	Frequency (Flash Clock)	Multiply Rate (Multiple Rate)		
3-wire serial I/O (SIO3)	SIO-ch0 (SIO ch-0)	2.4 to 625 kHz[Note 2] (100 Hz to 1.25 MHz) Note 2	Optional	1 to 10 MHz Note 2	1.0	SI3/P20 SO3/P21 $\overline{\text{SCK3}}$/P22	0
3-wire serial I/O (SIO3) with handshake	SIO-HS (SIO ch-3 + handshake)					SI3/P20 SO3/P21 $\overline{\text{SCK3}}$/P22 P33 (HS)	3
3-wire serial I/O (CSI1)	SIO-ch1 (SIO ch-1)	2.4 to 625 kHz[Note] (100 Hz to 2 MHz)[Note 2]	Optional	1 to 10 MHz Note 2	1.0	SI1/P23 SO1/P24 SCK1/P25	1
UART (UART0)	UART-ch0 (UART ch-0)	4,800 to 76,800 bps Notes 2, 3	Optional	1 to 10 MHz Note 2	1.0	RxD0/P26 TxD0/P27	8

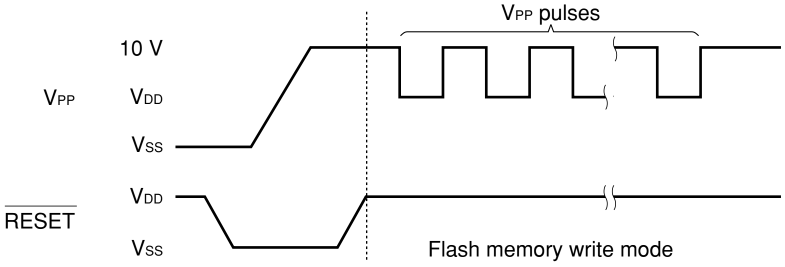
Notes 1. Selection items for standard settings (TYPE settings on Flashpro III) on the dedicated flash programmer Flashpro IV.

2. The possible setting range differs depending on the voltage. For details, refer to **CHAPTER 25 ELECTRICAL SPECIFICATIONS**.

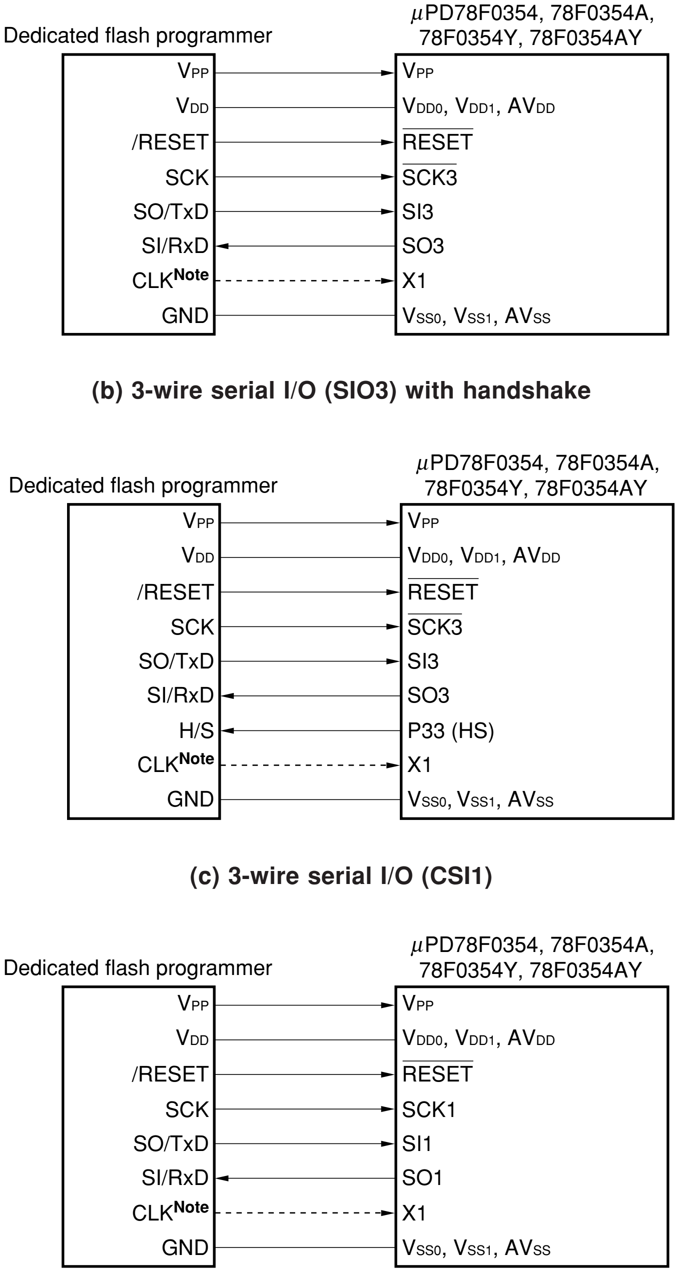
3. Because factors other than the baud rate error, such as the signal waveform slew, also affect UART communication, thoroughly evaluate the slew as well as the baud rate error.

Remark The items in parentheses in the Standard Setting column indicate the setting value or setting item of Flashpro III when it differs from Flashpro IV.

Figure 23-4. Communication Mode Selection Format



★ **Figure 23-5. Example of Connection with Dedicated Flash Programmer (1/2)**



(a) 3-wire serial I/O (SIO3)

(b) 3-wire serial I/O (SIO3) with handshake

(c) 3-wire serial I/O (CSI1)

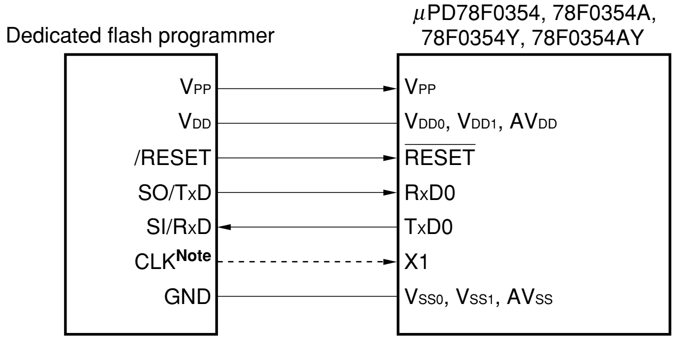
Note Connect this pin when the system clock is supplied from the dedicated flash programmer. If a resonator is already connected to the X1 pin, the CLK pin does not need to be connected.

Caution The V_{DD0} and V_{DD1} pins, if already connected to the power supply, must be connected to the V_{DD} pin of the dedicated flash programmer. When using the power supply connected to the V_{DD0} and V_{DD1} pins, supply voltage before starting programming.

★

Figure 23-5. Example of Connection with Dedicated Flash Programmer (2/2)

(d) UART (UART0)


Dedicated flash programmer
μPD78F0354, 78F0354A, 78F0354Y, 78F0354AY
V_{PP}
V_{DD}
/RESET
SO/TxD
SI/RxD
CLKNote
GND
V_{PP}
V_{DD0}, V_{DD1}, AV_{DD}
$\overline{RESET}$
RxD0
TxD0
X1
V_{SS0}, V_{SS1}, AV_{SS}

Note Connect this pin when the system clock is supplied from the dedicated flash programmer. If a resonator is already connected to the X1 pin, the CLK pin does not need to be connected.

Caution The V_{DD0} and V_{DD1} pins, if already connected to the power supply, must be connected to the V_{DD} pin of the dedicated flash programmer. When using the power supply connected to the V_{DD0} and V_{DD1} pins, supply voltage before starting programming.

If Flashpro III/Flashpro IV is used as a dedicated flash programmer, the following signals are generated for the μPD78F0354, 78F0354A, 78F0354Y, and 78F0354AY. For details, refer to the manual of Flashpro III/Flashpro IV.

★

Table 23-4. Pin Connection List

Signal Name	I/O	Pin Function	Pin Name	SIO3	SIO3 (HS)	CSI1	UART0
V_{PP}	Output	Write voltage	V_{PP}	◎	◎	◎	◎
V_{DD}	I/O	V_{DD} voltage generation/ voltage monitoring	V_{DD0}, V_{DD1}, AV_{DD}	◎Note	◎Note	◎Note	◎Note
GND	–	Ground	V_{SS0}, V_{SS1}, AV_{SS}	◎	◎	◎	◎
CLK	Output	Clock output	X1	○	○	○	○
/RESET	Output	Reset signal	$\overline{RESET}$	◎	◎	◎	◎
SI/RxD	Input	Reception signal	SO3/SO1/TxD0	◎	◎	◎	◎
SO/TxD	Output	Transmit signal	SI3/SI1/RxD0	◎	◎	◎	◎
SCK	Output	Transfer clock	$\overline{SCK3}$/SCK1	◎	◎	◎	×
H/S	Input	Handshake signal	P33 (HS)	×	◎	×	×

Note V_{DD} voltage must be supplied before programming is started.

Remark ◎: Pin must be connected.
○: If the signal is supplied on the target board, pin need not be connected.
×: Pin need not be connected.

23.3.3 On-board pin processing

When performing programming on the target system, provide a connector on the target system to connect the dedicated flash programmer.

An on-board function that allows switching between normal operation mode and flash memory programming mode may be required in some cases.

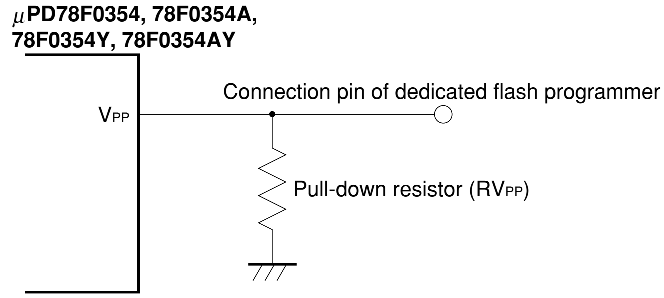
<VPP pin>

In normal operation mode, input 0 V to the VPP pin. In flash memory programming mode, a write voltage of 10.0 V (TYP.) is supplied to the VPP pin, so perform either of the following.

(1) Connect a pull-down resistor (RVPP = 10 kΩ) to the VPP pin.
(2) Use the jumper on the board to switch the VPP pin input to either the programmer or directly to GND.

A VPP pin connection example is shown below.

Figure 23-6. VPP Pin Connection Example


μPD78F0354, 78F0354A,
78F0354Y, 78F0354AY
VPP
Connection pin of dedicated flash programmer
Pull-down resistor (RVPP)

<Serial interface pin>

The following shows the pins used by the serial interface.

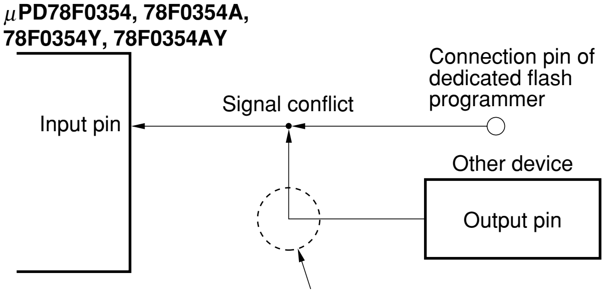
Serial Interface	Pins Used
3-wire serial I/O (SIO3)	SI3, SO3, $\overline{\text{SCK3}}$
3-wire serial I/O (SIO3) with handshake	SI3, SO3, $\overline{\text{SCK3}}$, P33 (HS)
3-wire serial I/O (CSI1)	SI1, SO1, SCK1
UART (UART0)	RxD0, TxD0

When connecting the dedicated flash programmer to a serial interface pin that is connected to another device on-board, signal conflict or abnormal operation of the other device may occur. Care must therefore be taken with such connections.

(1) Signal conflict

If the dedicated flash programmer (output) is connected to a serial interface pin (input) that is connected to another device (output), a signal conflict occurs. To prevent this, isolate the connection with the other device or set the other device to the output high impedance status.

Figure 23-7. Signal Conflict (Input Pin of Serial Interface)

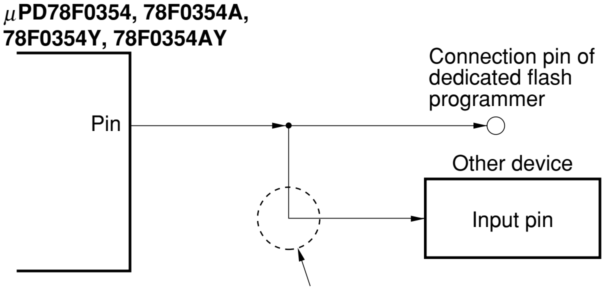


In the flash memory programming mode, the signal output by another device and the signal sent by the dedicated flash programmer conflict, therefore, isolate the signal of the other device.

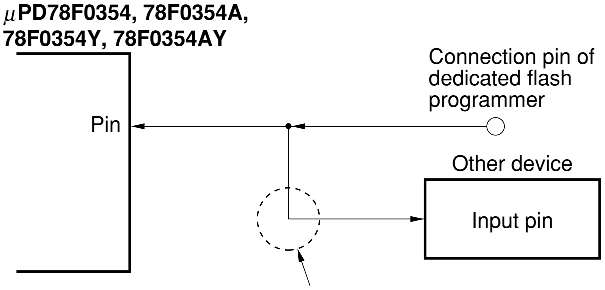
(2) Abnormal operation of other device

If the dedicated flash programmer (output or input) is connected to a serial interface pin (input or output) that is connected to another device (input), a signal is output to the device, and this may cause an abnormal operation. To prevent this abnormal operation, isolate the connection with the other device or set so that the input signals to the other device are ignored.

Figure 23-8. Abnormal Operation of Other Device



If the signal output by the μPD78F0354, 78F0354A, 78F0354Y, and 78F0354AY affects another device in the flash memory programming mode, isolate the signals of the other device.

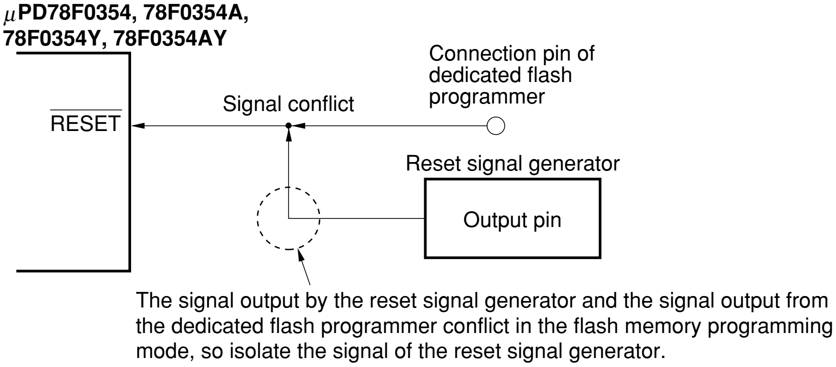


If the signal output by the dedicated flash programmer affects another device in the flash memory programming mode, isolate the signals of the other device.

<$\overline{\text{RESET}}$ pin>

If the reset signal of the dedicated flash programmer is connected to the $\overline{\text{RESET}}$ pin connected to the reset signal generator on-board, a signal conflict occurs. To prevent this, isolate the connection with the reset signal generator. If the reset signal is input from the user system in the flash memory programming mode, a normal programming operation cannot be performed. Therefore, do not input reset signals from other than the dedicated flash programmer.

Figure 23-9. Signal Conflict ($\overline{\text{RESET}}$ Pin)


μPD78F0354, 78F0354A,
78F0354Y, 78F0354AY
$\overline{\text{RESET}}$
Signal conflict
Connection pin of dedicated flash programmer
Reset signal generator
Output pin
The signal output by the reset signal generator and the signal output from the dedicated flash programmer conflict in the flash memory programming mode, so isolate the signal of the reset signal generator.

<Port pins>

★ When the μPD78F0354 and 78F0354Y enter the flash memory programming mode, all the pins other than those that communicate with the flash programmer are in the same status as immediately after reset.

If the external device does not recognize initial statuses such as the output high impedance status, therefore, connect the external device to V_{DD0} or V_{SS0} via a resistor.

<Oscillator>

When using the on-board clock, connect X1, X2, XT1, and XT2 as required in the normal operation mode.

When using the clock output of the flash programmer, connect it directly to X1, disconnecting the main oscillator on-board, and leave the X2 pin open. The subclock conforms to the normal operation mode.

<Power supply>

To use the power output from the flash programmer, connect the V_{DD0} and V_{DD1} pins to V_{DD} of the flash programmer, and the V_{SS0} and V_{SS1} pins to GND of the flash programmer.

To use the on-board power supply, make connections that accord with the normal operation mode. However, because the voltage is monitored by the flash programmer, be sure to connect V_{DD} of the flash programmer.

★ Supply the same power as in the normal operation mode to the other power supply pins (AV_{DD} and AV_{SS}).

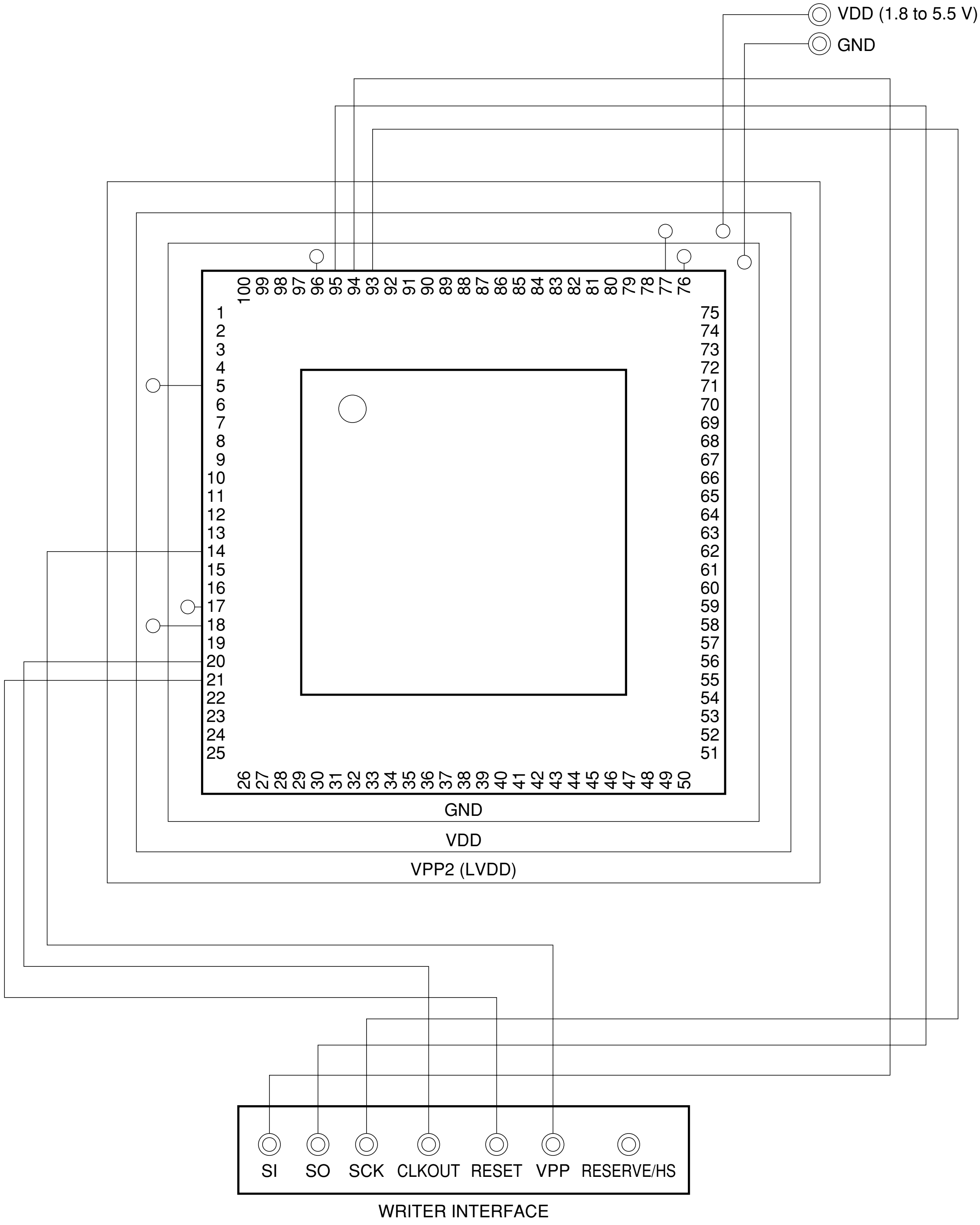
<Other pins>

★ Process the other pins (S0 to S39, COM0 to COM3, SCOMO, V_{LC0} to V_{LC2}, CAPH, and CAPL) in the same manner as in the normal operation mode.

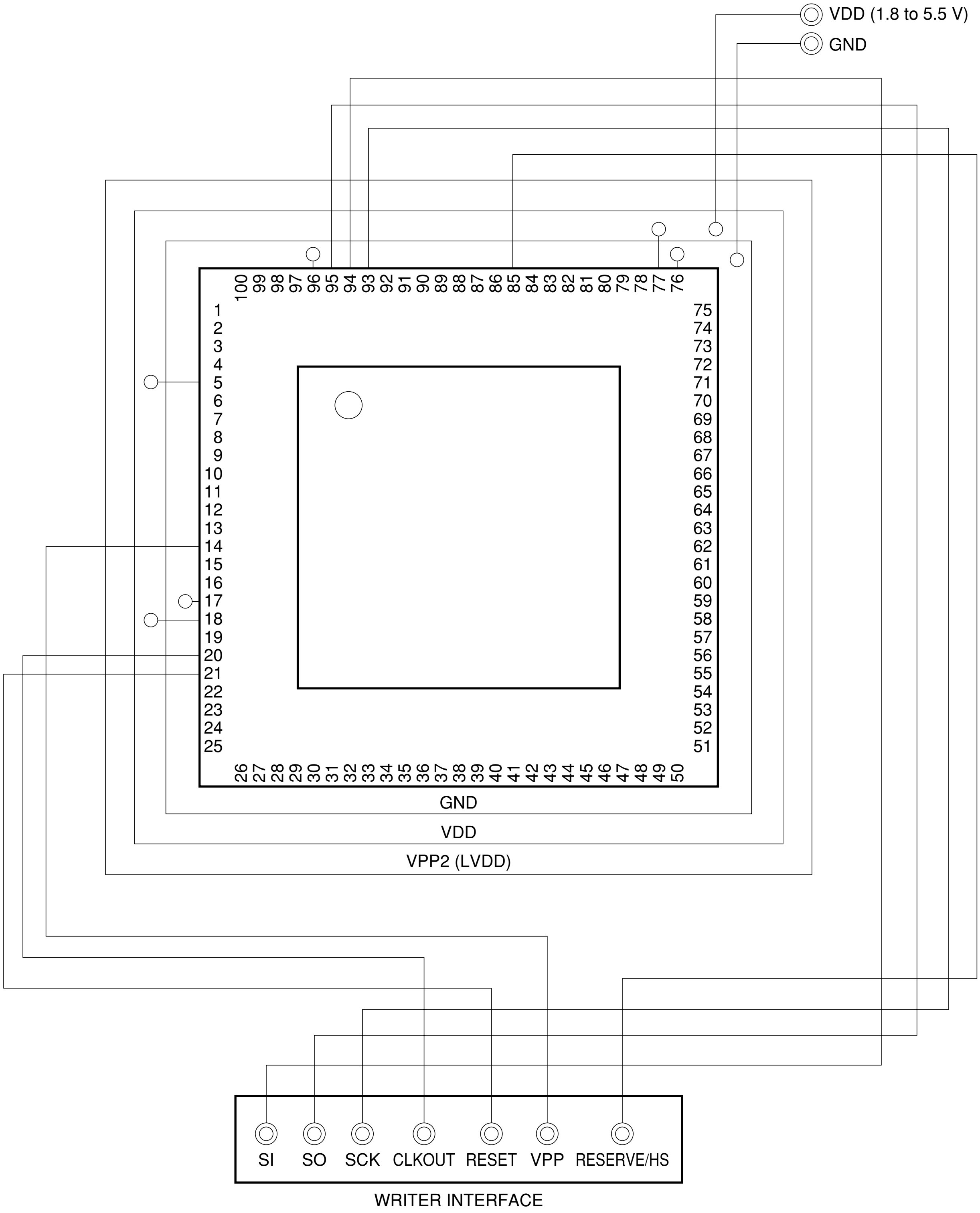
23.3.4 Connection of adapter for flash writing

The following figures show the examples of recommended connection when the adapter for flash writing is used.

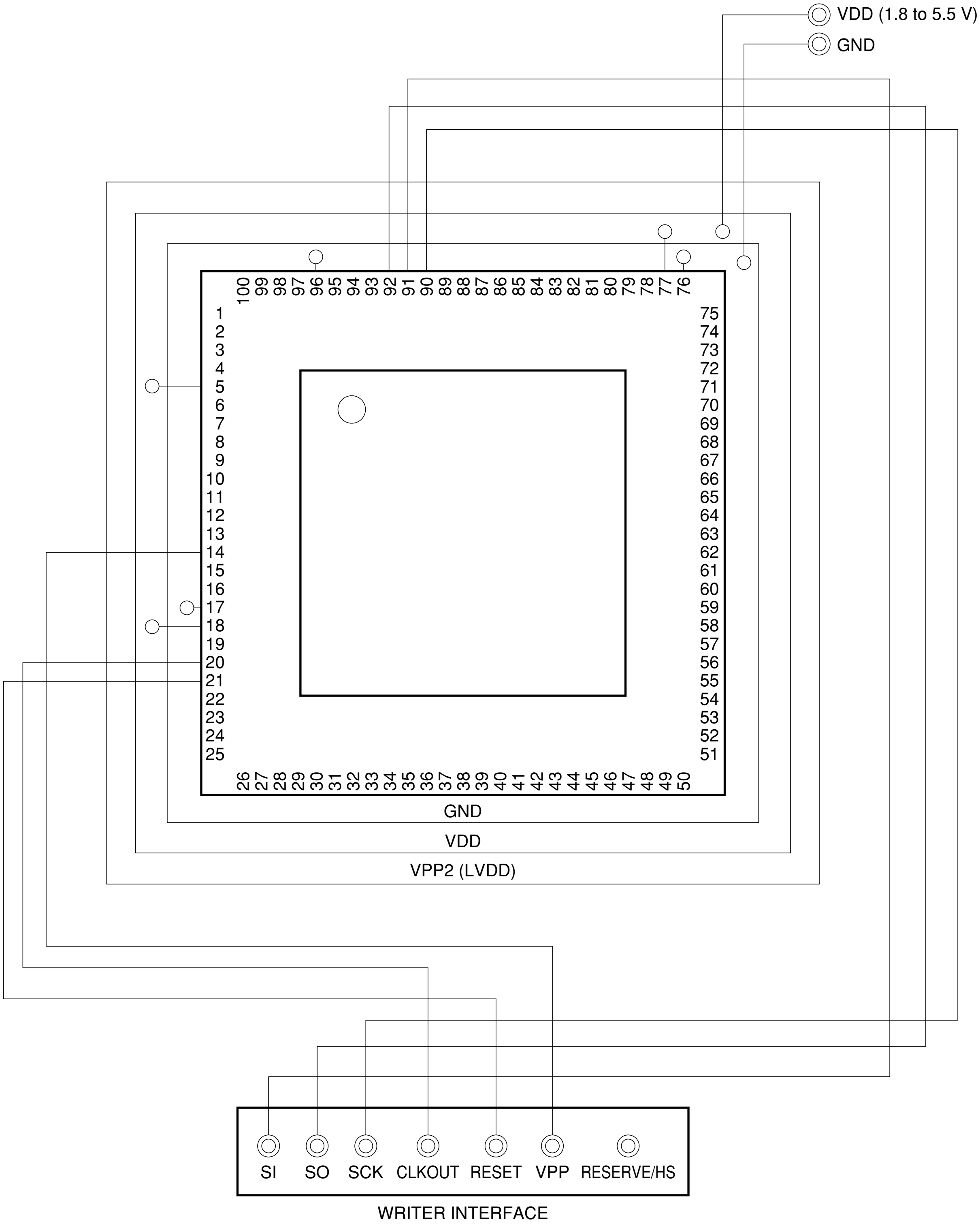
★ Figure 23-10. Wiring Example for Flash Writing Adapter with 3-Wire Serial I/O (SIO3)



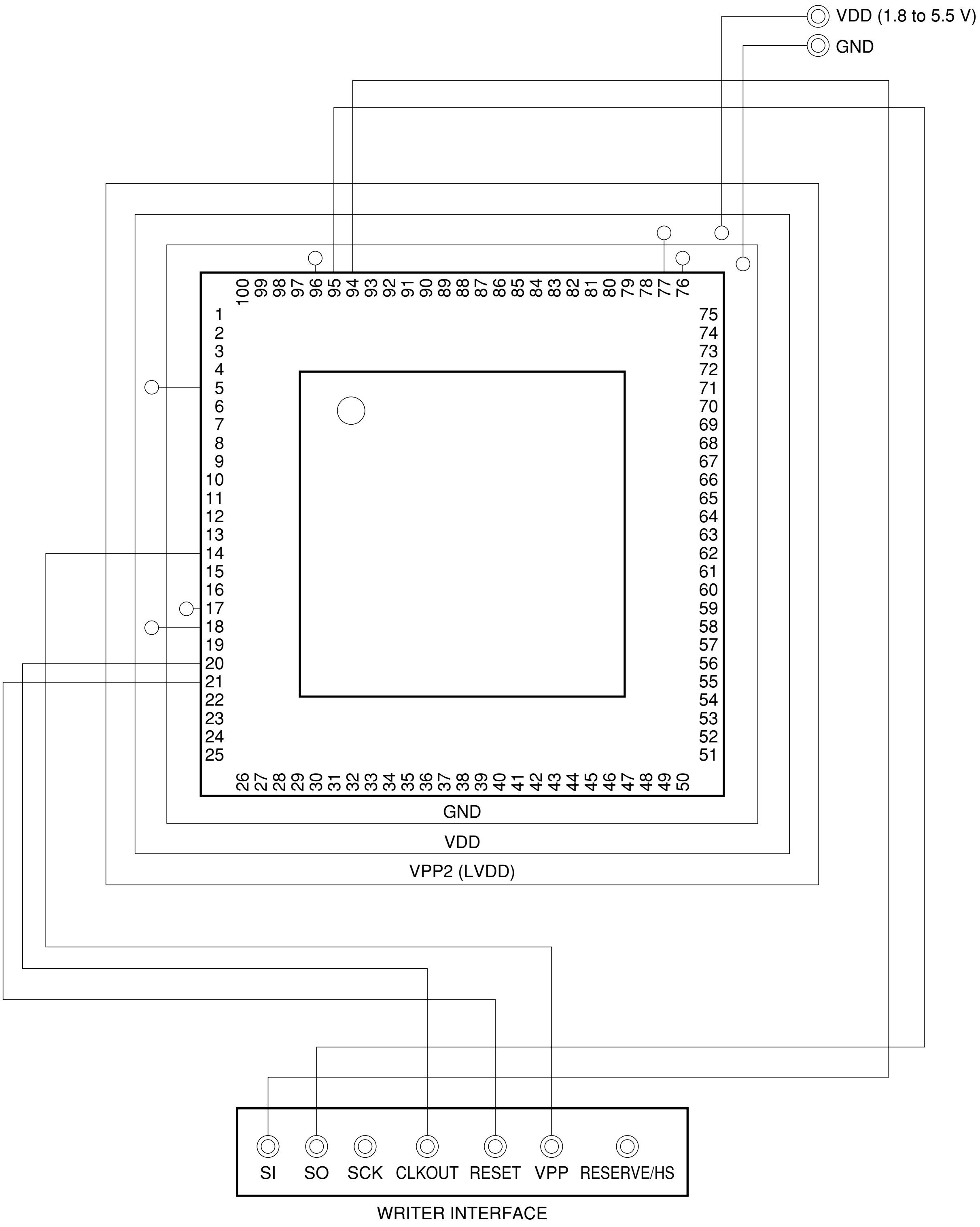
★ **Figure 23-11. Wiring Example for Flash Writing Adapter with 3-Wire Serial I/O (SIO3) with Handshake**



★ **Figure 23-12. Wiring Example for Flash Writing Adapter with 3-Wire Serial I/O (CSI1)**



★ **Figure 23-13. Wiring Example for Flash Writing Adapter with UART (UART0)**



CHAPTER 24 INSTRUCTION SET

This chapter lists each instruction set of the μPD780344, 780354, 780344Y, 780354Y Subseries in table form. For details of each instruction's operation and operation code, refer to the separate document **78K/0 Series Instructions User's Manual (U12326E)**.

24.1 Conventions

24.1.1 Operand identifiers and specification methods

Operands are written in the "Operand" column of each instruction in accordance with the specification method of the instruction operand identifier (refer to the assembler specifications for details). When there are two or more methods, select one of them. Uppercase letters and the symbols #, !, $, and [ ] are 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 24-1. Operand Identifiers and Specification Methods

Identifier	Specification Method
r rp sfr sfrp	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7) AX (RP0), BC (RP1), DE (RP2), HL (RP3) Special function register symbolNote Special function register symbol (16-bit manipulatable register even addresses only)Note
saddr saddrp	FE20H to FF1FH Immediate data or labels FE20H to FF1FH Immediate data or labels (even address only)
addr16 addr11 addr5	0000H to FFFFH Immediate data or labels (only even addresses for 16-bit data transfer instructions) 0800H to 0FFFH Immediate data or labels 0040H to 007FH Immediate data or labels (even address only)
word byte bit	16-bit immediate data or label 8-bit immediate data or label 3-bit immediate data or label
RBn	RB0 to RB3

Note Addresses from FFD0H to FFDFH cannot be accessed with these operands.

Remark For special function register symbols, refer to **Table 3-3 Special Function Register List**.

24.1.2 Description of “operation” column

A:	A register; 8-bit accumulator
X:	X register
B:	B register
C:	C register
D:	D register
E:	E register
H:	H register
L:	L register
AX:	AX register pair; 16-bit accumulator
BC:	BC register pair
DE:	DE register pair
HL:	HL register pair
PC:	Program counter
SP:	Stack pointer
PSW:	Program status word
CY:	Carry flag
AC:	Auxiliary carry flag
Z:	Zero flag
RBS:	Register bank select flag
IE:	Interrupt request enable flag
NMIS:	Non-maskable interrupt servicing flag
( ):	Memory contents indicated by address or register contents in parentheses
X_H, X_L:	Higher 8 bits and lower 8 bits of 16-bit register
$\wedge$:	Logical product (AND)
$\vee$:	Logical sum (OR)
$\veebar$:	Exclusive logical sum (exclusive OR)
——:	Inverted data
addr16:	16-bit immediate data or label
jdisp8:	Signed 8-bit data (displacement value)

24.1.3 Description of “flag operation” column

(Blank):	Not affected
0:	Cleared to 0
1:	Set to 1
×:	Set/cleared according to the result
R:	Previously saved value is restored

24.2 Operation List

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	**MOV**	r, #byte	2	4	–	r ← byte			
		saddr, #byte	3	6	7	(saddr) ← byte			
		sfr, #byte	3	–	7	sfr ← byte			
		A, r Note 3	1	2	–	A ← r			
		r, A Note 3	1	2	–	r ← A			
		A, saddr	2	4	5	A ← (saddr)			
		saddr, A	2	4	5	(saddr) ← A			
		A, sfr	2	–	5	A ← sfr			
		sfr, A	2	–	5	sfr ← A			
		A, !addr16	3	8	9	A ← (addr16)			
		!addr16, A	3	8	9	(addr16) ← A			
		PSW, #byte	3	–	7	PSW ← byte	×	×	×
		A, PSW	2	–	5	A ← PSW			
		PSW, A	2	–	5	PSW ← A	×	×	×
		A, [DE]	1	4	5	A ← (DE)			
		[DE], A	1	4	5	(DE) ← A			
		A, [HL]	1	4	5	A ← (HL)			
		[HL], A	1	4	5	(HL) ← A			
		A, [HL + byte]	2	8	9	A ← (HL + byte)			
		[HL + byte], A	2	8	9	(HL + byte) ← A			
		A, [HL + B]	1	6	7	A ← (HL + B)			
		[HL + B], A	1	6	7	(HL + B) ← A			
		A, [HL + C]	1	6	7	A ← (HL + C)			
		[HL + C], A	1	6	7	(HL + C) ← A			
	XCH	A, r Note 3	1	2	–	A ↔ r			
		A, saddr	2	4	6	A ↔ (saddr)			
		A, sfr	2	–	6	A ↔ sfr			
		A, !addr16	3	8	10	A ↔ (addr16)			
		A, [DE]	1	4	6	A ↔ (DE)			
		A, [HL]	1	4	6	A ↔ (HL)			
		A, [HL + byte]	2	8	10	A ↔ (HL + byte)			
		A, [HL + B]	2	8	10	A ↔ (HL + B)			
		A, [HL + C]	2	8	10	A ↔ (HL + C)			

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed.

3. Except "r = A"

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit data transfer	**MOVW**	rp, #word	3	6	–	rp ← word			
		saddrp, #word	4	8	10	(saddrp) ← word			
		sfrp, #word	4	–	10	sfrp ← word			
		AX, saddrp	2	6	8	AX ← (saddrp)			
		saddrp, AX	2	6	8	(saddrp) ← AX			
		AX, sfrp	2	–	8	AX ← sfrp			
		sfrp, AX	2	–	8	sfrp ← AX			
		AX, rp **Note 3**	1	4	–	AX ← rp			
		rp, AX **Note 3**	1	4	–	rp ← AX			
		AX, !addr16	3	10	12	AX ← (addr16)			
		!addr16, AX	3	10	12	(addr16) ← AX			
	XCHW	AX, rp **Note 3**	1	4	–	AX ↔ rp			
8-bit operation	**ADD**	A, #byte	2	4	–	A, CY ← A + byte	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) + byte	×	×	×
		A, r **Note 4**	2	4	–	A, CY ← A + r	×	×	×
		r, A	2	4	–	r, CY ← r + A	×	×	×
		A, saddr	2	4	5	A, CY ← A + (saddr)	×	×	×
		A, !addr16	3	8	9	A, CY ← A + (addr16)	×	×	×
		A, [HL]	1	4	5	A, CY ← A + (HL)	×	×	×
		A, [HL + byte]	2	8	9	A, CY ← A + (HL + byte)	×	×	×
		A, [HL + B]	2	8	9	A, CY ← A + (HL + B)	×	×	×
		A, [HL + C]	2	8	9	A, CY ← A + (HL + C)	×	×	×
	ADDC	A, #byte	2	4	–	A, CY ← A + byte + CY	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) + byte + CY	×	×	×
		A, r **Note 4**	2	4	–	A, CY ← A + r + CY	×	×	×
		r, A	2	4	–	r, CY ← r + A + CY	×	×	×
		A, saddr	2	4	5	A, CY ← A + (saddr) + CY	×	×	×
		A, !addr16	3	8	9	A, CY ← A + (addr16) + CY	×	×	×
		A, [HL]	1	4	5	A, CY ← A + (HL) + CY	×	×	×
		A, [HL + byte]	2	8	9	A, CY ← A + (HL + byte) + CY	×	×	×
		A, [HL + B]	2	8	9	A, CY ← A + (HL + B) + CY	×	×	×
		A, [HL + C]	2	8	9	A, CY ← A + (HL + C) + CY	×	×	×

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Only when rp = BC, DE or HL

4. Except “r = A”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	**SUB**	A, #byte	2	4	–	A, CY ← A – byte	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) – byte	×	×	×
		A, r Note 3	2	4	–	A, CY ← A – r	×	×	×
		r, A	2	4	–	r, CY ← r – A	×	×	×
		A, saddr	2	4	5	A, CY ← A – (saddr)	×	×	×
		A, !addr16	3	8	9	A, CY ← A – (addr16)	×	×	×
		A, [HL]	1	4	5	A, CY ← A – (HL)	×	×	×
		A, [HL + byte]	2	8	9	A, CY ← A – (HL + byte)	×	×	×
		A, [HL + B]	2	8	9	A, CY ← A – (HL + B)	×	×	×
		A, [HL + C]	2	8	9	A, CY ← A – (HL + C)	×	×	×
	SUBC	A, #byte	2	4	–	A, CY ← A – byte – CY	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) – byte – CY	×	×	×
		A, r Note 3	2	4	–	A, CY ← A – r – CY	×	×	×
		r, A	2	4	–	r, CY ← r – A – CY	×	×	×
		A, saddr	2	4	5	A, CY ← A – (saddr) – CY	×	×	×
		A, !addr16	3	8	9	A, CY ← A – (addr16) – CY	×	×	×
		A, [HL]	1	4	5	A, CY ← A – (HL) – CY	×	×	×
		A, [HL + byte]	2	8	9	A, CY ← A – (HL + byte) – CY	×	×	×
		A, [HL + B]	2	8	9	A, CY ← A – (HL + B) – CY	×	×	×
		A, [HL + C]	2	8	9	A, CY ← A – (HL + C) – CY	×	×	×
	AND	A, #byte	2	4	–	A ← A∧byte	×		
		saddr, #byte	3	6	8	(saddr) ← (saddr)∧byte	×		
		A, r Note 3	2	4	–	A ← A∧r	×		
		r, A	2	4	–	r ← r∧A	×		
		A, saddr	2	4	5	A ← A∧(saddr)	×		
		A, !addr16	3	8	9	A ← A∧(addr16)	×		
		A, [HL]	1	4	5	A ← A∧(HL)	×		
		A, [HL + byte]	2	8	9	A ← A∧(HL + byte)	×		
		A, [HL + B]	2	8	9	A ← A∧(HL + B)	×		
		A, [HL + C]	2	8	9	A ← A∧(HL + C)	×		

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Except “r = A”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	**OR**	A, #byte	2	4	–	A ← A∨byte	×		
		saddr, #byte	3	6	8	(saddr) ← (saddr)∨byte	×		
		A, r Note 3	2	4	–	A ← A∨r	×		
		r, A	2	4	–	r ← r∨A	×		
		A, saddr	2	4	5	A ← A∨(saddr)	×		
		A, !addr16	3	8	9	A ← A∨(addr16)	×		
		A, [HL]	1	4	5	A ← A∨(HL)	×		
		A, [HL + byte]	2	8	9	A ← A∨(HL + byte)	×		
		A, [HL + B]	2	8	9	A ← A∨(HL + B)	×		
		A, [HL + C]	2	8	9	A ← A∨(HL + C)	×		
	XOR	A, #byte	2	4	–	A ← A∀byte	×		
		saddr, #byte	3	6	8	(saddr) ← (saddr)∀byte	×		
		A, r Note 3	2	4	–	A ← A∀r	×		
		r, A	2	4	–	r ← r∀A	×		
		A, saddr	2	4	5	A ← A∀(saddr)	×		
		A, !addr16	3	8	9	A ← A∀(addr16)	×		
		A, [HL]	1	4	5	A ← A∀(HL)	×		
		A, [HL + byte]	2	8	9	A ← A∀(HL + byte)	×		
		A, [HL + B]	2	8	9	A ← A∀(HL + B)	×		
		A, [HL + C]	2	8	9	A ← A∀(HL + C)	×		
	CMP	A, #byte	2	4	–	A – byte	×	×	×
		saddr, #byte	3	6	8	(saddr) – byte	×	×	×
		A, r Note 3	2	4	–	A – r	×	×	×
		r, A	2	4	–	r – A	×	×	×
		A, saddr	2	4	5	A – (saddr)	×	×	×
		A, !addr16	3	8	9	A – (addr16)	×	×	×
		A, [HL]	1	4	5	A – (HL)	×	×	×
		A, [HL + byte]	2	8	9	A – (HL + byte)	×	×	×
		A, [HL + B]	2	8	9	A – (HL + B)	×	×	×
		A, [HL + C]	2	8	9	A – (HL + C)	×	×	×

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Except “r = A”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit operation	**ADDW**	AX, #word	3	6	–	AX, CY ← AX + word	×	×	×
	SUBW	AX, #word	3	6	–	AX, CY ← AX – word	×	×	×
	CMPW	AX, #word	3	6	–	AX – word	×	×	×
Multiply/ divide	**MULU**	X	2	16	–	AX ← A × X			
	DIVUW	C	2	25	–	AX (Quotient), C (Remainder) ← AX ÷ C			
Increment/ decrement	**INC**	r	1	2	–	r ← r + 1	×	×	
		saddr	2	4	6	(saddr) ← (saddr) + 1	×	×	
	DEC	r	1	2	–	r ← r – 1	×	×	
		saddr	2	4	6	(saddr) ← (saddr) – 1	×	×	
	INCW	rp	1	4	–	rp ← rp + 1			
	DECW	rp	1	4	–	rp ← rp – 1			
Rotate	**ROR**	A, 1	1	2	–	(CY, A_7 ← A_0, A_{m-1} ← A_m) × 1 time			×
	ROL	A, 1	1	2	–	(CY, A_0 ← A_7, A_{m+1} ← A_m) × 1 time			×
	RORC	A, 1	1	2	–	(CY ← A_0, A_7 ← CY, A_{m-1} ← A_m) × 1 time			×
	ROLC	A, 1	1	2	–	(CY ← A_7, A_0 ← CY, A_{m+1} ← A_m) × 1 time			×
	ROR4	[HL]	2	10	12	A_{3-0} ← $(HL)_{3-0}$, $(HL)_{7-4}$ ← A_{3-0}, $(HL)_{3-0}$ ← $(HL)_{7-4}$			
	ROL4	[HL]	2	10	12	A_{3-0} ← $(HL)_{7-4}$, $(HL)_{3-0}$ ← A_{3-0}, $(HL)_{7-4}$ ← $(HL)_{3-0}$			
BCD adjust	**ADJBA**		2	4	–	Decimal Adjust Accumulator after Addition	×	×	×
	ADJBS		2	4	–	Decimal Adjust Accumulator after Subtract	×	×	×
Bit manipulate	**MOV1**	CY, saddr.bit	3	6	7	CY ← (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← sfr.bit			×
		CY, A.bit	2	4	–	CY ← A.bit			×
		CY, PSW.bit	3	–	7	CY ← PSW.bit			×
		CY, [HL].bit	2	6	7	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	(HL).bit ← CY			

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Bit manipulate	**AND1**	CY, saddr.bit	3	6	7	CY ← CY∧(saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← CY∧sfr.bit			×
		CY, A.bit	2	4	–	CY ← CY∧A.bit			×
		CY, PSW.bit	3	–	7	CY ← CY∧PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← CY∧(HL).bit			×
	OR1	CY, saddr.bit	3	6	7	CY ← CY∨(saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← CY∨sfr.bit			×
		CY, A.bit	2	4	–	CY ← CY∨A.bit			×
		CY, PSW.bit	3	–	7	CY ← CY∨PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← CY∨(HL).bit			×
	XOR1	CY, saddr.bit	3	6	7	CY ← CY∀(saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← CY∀sfr.bit			×
		CY, A.bit	2	4	–	CY ← CY∀A.bit			×
		CY, PSW.bit	3	–	7	CY ← CY∀PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← CY∀(HL).bit			×
	SET1	saddr.bit	2	4	6	(saddr.bit) ← 1			
		sfr.bit	3	–	8	sfr.bit ← 1			
		A.bit	2	4	–	A.bit ← 1			
		PSW.bit	2	–	6	PSW.bit ← 1	×	×	×
		[HL].bit	2	6	8	(HL).bit ← 1			
	CLR1	saddr.bit	2	4	6	(saddr.bit) ← 0			
		sfr.bit	3	–	8	sfr.bit ← 0			
		A.bit	2	4	–	A.bit ← 0			
		PSW.bit	2	–	6	PSW.bit ← 0	×	×	×
		[HL].bit	2	6	8	(HL).bit ← 0			
	SET1	CY	1	2	–	CY ← 1			1
	CLR1	CY	1	2	–	CY ← 0			0
	NOT1	CY	1	2	–	CY ← $\overline{\text{CY}}$			×

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

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	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Call/return	**CALL**	!addr16	3	7	–	(SP – 1) ← $(PC + 3)_H$, (SP – 2) ← $(PC + 3)_L$, PC ← addr16, SP ← SP – 2			
	CALLF	!addr11	2	5	–	(SP – 1) ← $(PC + 2)_H$, (SP – 2) ← $(PC + 2)_L$, PC_{15-11} ← 00001, PC_{10-0} ← addr11, SP ← SP – 2			
	CALLT	[addr5]	1	6	–	(SP – 1) ← $(PC + 1)_H$, (SP – 2) ← $(PC + 1)_L$, PC_H ← (00000000, addr5 + 1), PC_L ← (00000000, addr5), SP ← SP – 2			
	BRK		1	6	–	(SP – 1) ← PSW, (SP – 2) ← $(PC + 1)_H$, (SP – 3) ← $(PC + 1)_L$, PC_H ← (003FH), PC_L ← (003EH), SP ← SP – 3, IE ← 0			
	RET		1	6	–	PC_H ← (SP + 1), PC_L ← (SP), SP ← SP + 2			
	RETI		1	6	–	PC_H ← (SP + 1), PC_L ← (SP), PSW ← (SP + 2), SP ← SP + 3, NMIS ← 0	R	R	R
	RETB		1	6	–	PC_H ← (SP + 1), PC_L ← (SP), PSW ← (SP + 2), SP ← SP + 3	R	R	R
Stack manipulate	**PUSH**	PSW	1	2	–	(SP – 1) ← PSW, SP ← SP – 1			
		rp	1	4	–	(SP – 1) ← rp_H, (SP – 2) ← rp_L, SP ← SP – 2			
	POP	PSW	1	2	–	PSW ← (SP), SP ← SP + 1	R	R	R
		rp	1	4	–	rp_H ← (SP + 1), rp_L ← (SP), SP ← SP + 2			
	MOVW	SP, #word	4	–	10	SP ← word			
		SP, AX	2	–	8	SP ← AX			
		AX, SP	2	–	8	AX ← SP			
Unconditional branch	**BR**	!addr16	3	6	–	PC ← addr16			
		$addr16	2	6	–	PC ← PC + 2 + jdisp8			
		AX	2	8	–	PC_H ← A, PC_L ← X			
Conditional branch	**BC**	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if CY = 1			
	BNC	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if CY = 0			
	BZ	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if Z = 1			
	BNZ	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if Z = 0			

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Conditional branch	**BT**	saddr.bit, $addr16	3	8	9	PC ← PC + 3 + jdisp8 if (saddr.bit) = 1			
		sfr.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if sfr.bit = 1			
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 1			
		PSW.bit, $addr16	3	–	9	PC ← PC + 3 + jdisp8 if PSW.bit = 1			
		[HL].bit, $addr16	3	10	11	PC ← PC + 3 + jdisp8 if (HL).bit = 1			
	BF	saddr.bit, $addr16	4	10	11	PC ← PC + 4 + jdisp8 if (saddr.bit) = 0			
		sfr.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if sfr.bit = 0			
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 0			
		PSW.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if PSW.bit = 0			
		[HL].bit, $addr16	3	10	11	PC ← PC + 3 + jdisp8 if (HL).bit = 0			
	BTCLR	saddr.bit, $addr16	4	10	12	PC ← PC + 4 + jdisp8 if (saddr.bit) = 1 then reset (saddr.bit)			
		sfr.bit, $addr16	4	–	12	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit			
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit			
		PSW.bit, $addr16	4	–	12	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	×	×	×
		[HL].bit, $addr16	3	10	12	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit			
	DBNZ	B, $addr16	2	6	–	B ← B – 1, then PC ← PC + 2 + jdisp8 if B ≠ 0			
		C, $addr16	2	6	–	C ← C –1, then PC ← PC + 2 + jdisp8 if C ≠ 0			
		saddr, $addr16	3	8	10	(saddr) ← (saddr) – 1, then PC ← PC + 3 + jdisp8 if (saddr) ≠ 0			
CPU control	**SEL**	RBn	2	4	–	RBS1, 0 ← n			
	NOP		1	2	–	No Operation			
	EI		2	–	6	IE ← 1 (Enable Interrupt)			
	DI		2	–	6	IE ← 0 (Disable Interrupt)			
	HALT		2	6	–	Set HALT Mode			
	STOP		2	6	–	Set STOP Mode			

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

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.

24.3 Instructions Listed by Addressing Type

(1) 8-bit instructions

MOV, XCH, ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP, MULU, DIVUW, INC, DEC, ROR, ROL, RORC, ROLC, ROR4, ROL4, PUSH, POP, DBNZ

Second Operand First Operand	#byte	A	r[Note]	sfr	saddr	!addr16	PSW	[DE]	[HL]	[HL + byte] [HL + B] [HL + C]	$addr16	1	None
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP		ROR ROL RORC ROLC	
r	MOV	MOV ADD ADDC SUB SUBC AND OR XOR CMP											INC DEC
B, C											DBNZ		
sfr	MOV	MOV											
saddr	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV									DBNZ		INC DEC
!addr16		MOV											
PSW	MOV	MOV											PUSH POP
[DE]		MOV											
[HL]		MOV											ROR4 ROL4
[HL + byte] [HL + B] [HL + C]		MOV											
X													MULU
C													DIVUW

Note Except r = A

(2) 16-bit instructions

MOVW, XCHW, ADDW, SUBW, CMPW, PUSH, POP, INCW, DECW

Second Operand / First Operand	#word	AX	rp[Note]	sfrp	saddrp	!addr16	SP	None
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	MOVW	MOVW	
rp	MOVW	MOVW[Note]						INCW DECW PUSH POP
sfrp	MOVW	MOVW						
saddrp	MOVW	MOVW						
!addr16		MOVW						
SP	MOVW	MOVW						

Note Only when rp = BC, DE, HL

(3) Bit manipulation instructions

MOV1, AND1, OR1, XOR1, SET1, CLR1, NOT1, BT, BF, BTCLR

Second Operand / First Operand	A.bit	sfr.bit	saddr.bit	PSW.bit	[HL].bit	CY	$addr16	None
A.bit						MOV1	BT BF BTCLR	SET1 CLR1
sfr.bit						MOV1	BT BF BTCLR	SET1 CLR1
saddr.bit						MOV1	BT BF BTCLR	SET1 CLR1
PSW.bit						MOV1	BT BF BTCLR	SET1 CLR1
[HL].bit						MOV1	BT BF BTCLR	SET1 CLR1
CY	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1			SET1 CLR1 NOT1

(4) Call instructions/branch instructions

CALL, CALLF, CALLT, BR, BC, BNC, BZ, BNZ, BT, BF, BTCLR, DBNZ

Second Operand First Operand	AX	!addr16	!addr11	[addr5]	$addr16
Basic instruction	BR	CALL BR	CALLF	CALLT	BR BC BNC BZ BNZ
Compound instruction					BT BF BTCLR DBNZ

(5) Other instructions

ADJBA, ADJBS, BRK, RET, RETI, RETB, SEL, NOP, EI, DI, HALT, STOP

★

CHAPTER 25 ELECTRICAL SPECIFICATIONS

Absolute maximum ratings (T_A = 25°C)

Parameter	Symbol	Conditions		Ratings	Unit
Supply voltage	V_{DD}	V_{DD0}, V_{DD1}		–0.3 to +6.5	V
	V_{PP}	μPD78F0354, 78F0354Y only$^{\text{Note 1}}$		–0.5 to +10.5	V
	AV_{DD}			–0.3 to V_{DD} + 0.3$^{\text{Note 2}}$	V
	AV_{SS}			–0.3 to +0.3	V
Input voltage	V_{I1}	P00 to P07, P10 to P17$^{\text{Note 3}}$, P20 to P27, P32 to P35, P40 to P43, P80 to P87, P90 to P97, P100 to P107, P110 to P113, X1, X2, XT1, XT2, $\overline{\text{RESET}}$		–0.3 to V_{DD} + 0.3$^{\text{Note 2}}$	V
	V_{I2}	P30, P31 (N-ch open-drain)		–0.3 to V_{DD} + 0.3$^{\text{Note 2}}$	V
	V_{I3}	P70 to P73 (N-ch open-drain)	Mask option	–0.3 to V_{DD} + 0.3$^{\text{Note 2}}$	V
			No mask option	–0.3 to +16	V
Output voltage	V_O			–0.3 to V_{DD} + 0.3$^{\text{Notes 2, 4}}$	V
Analog input voltage	V_{AN}	P10 to P17	Analog input pin	AV_{SS} – 0.3 to AV_{DD} + 0.3 and –0.3 to V_{DD} + 0.3	V
Output current, high	I_{OH}	Per pin for P00 to P07, P20 to P27, P30 to P35, P40 to P43, P80 to P87, P90 to P97, P100 to P107, P110 to P113		–10	mA
		Total for P80 to P87, P90 to P97, P100 to P107, P110 to P113		–15	mA
		Total for P00 to P07, P20 to P27, P30 to P35, P40 to P43		–15	mA
Output current, low	I_{OL}	Per pin for P00 to P07, P20 to P27, P32 to P35, P40 to P43, P80 to P87, P90 to P97, P100 to P107, P110 to P113		20	mA
		Per pin for P30, P31, P70 to P73		30	mA
		Total for P80 to P87, P90 to P97, P100 to P107, P110 to P113		50	mA
		Total for P00 to P07, P20 to P27, P32 to P35, P40 to P43		30	mA
		Total for P30, P31		40	mA
		Total for P70 to P73		80	mA
Operating ambient temperature	T_A			–40 to +85	°C
Storage temperature	T_{stg}	μPD78F0354, 78F0354Y		–40 to +125	°C
		Mask ROM version		–40 to +150	°C

Notes 1. The explanation is shown on the next page.

2. 6.5 V or less

3. V_{DD} = AV_{DD}

4. –0.3 to V_{LC0} + 0.3 V for common and segment pins

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.

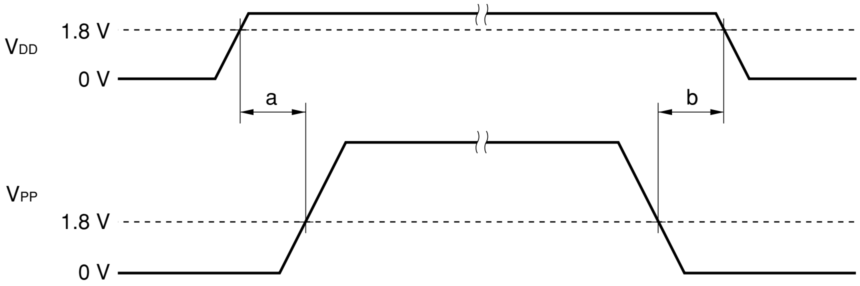
★ **Note 1.** Make sure that the following conditions of the V_{PP} voltage application timing are satisfied when the flash memory is written.

- **When supply voltage rises**

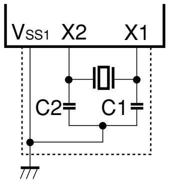
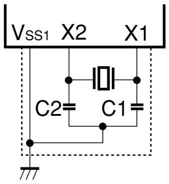
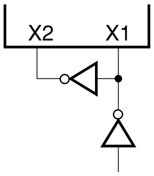
  V_{PP} must exceed V_{DD} 10 μs or more after V_{DD} has reached the lower-limit value (1.8 V) of the operating voltage range (see a in the figure below).

- **When supply voltage drops**

  V_{DD} must be lowered 10 μs or more after V_{PP} falls below the lower-limit value (1.8 V) of the operating voltage range of V_{DD} (see b in the figure below).



Main system clock oscillator characteristics (T_A = −40 to +85°C, V_{DD} = 1.8 to 5.5 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic resonator		Oscillation frequency (f_X)Note 1	V_{DD} = 1.8 to 5.5 V	2.0		10	MHz
		Oscillation stabilization timeNote 2	After V_{DD} reaches oscillation voltage range MIN.			4	ms
Crystal resonator		Oscillation frequency (f_X)Note 1	V_{DD} = 1.8 to 5.5 V	2.0		10	MHz
		Oscillation stabilization timeNote 2	V_{DD} = 1.8 to 5.5 V			10	ms
External clock		X1 input frequency (f_X)Note 1	V_{DD} = 4.5 to 5.5 V	2.0		10	MHz
			V_{DD} = 1.8 to 5.5 V	2.0		5.0	MHz
		X1 input high-/low-level width (t_{XH}, t_{XL})	V_{DD} = 4.5 to 5.5 V	42.5		500	ns
			V_{DD} = 1.8 to 5.5 V	85		500	ns

Notes 1. Indicates only oscillator characteristics.

2. Time required to stabilize oscillation after reset or STOP mode release.

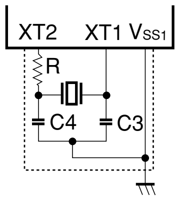
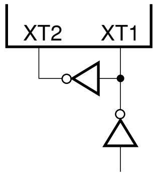
Cautions 1. When using the main system clock oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines.**
- **Do not route the wiring near a line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as V_{SS1}.**
- **Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

2. When the main system clock is stopped and the system is operated by the subsystem clock, the subsystem clock should be switched again to the main system clock after the oscillation stabilization time is secured by the program.

Remark For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

Subsystem clock oscillator characteristics (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Crystal resonator		Oscillation frequency (f_{XT})Note 1	V_{DD} = 1.8 to 5.5 V	32	32.768	35	kHz
		Oscillation stabilization timeNote 2	V_{DD} = 4.5 to 5.5 V		1.2	2	s
			V_{DD} = 1.8 to 5.5 V			10	s
External clock		XT1 input frequency (f_{XT})Note 1		32		35	kHz
		XT1 input high-/low-level width (t_{XTH}, t_{XTL})		5		15	μs

Notes 1. Indicates only oscillator characteristics.

2. Time required to stabilize oscillation after reset or STOP mode release.

Cautions 1. When using the subsystem clock oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines.**
- **Do not route the wiring near a line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as V_{SS1}.**
- **Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

2. The subsystem clock oscillator is designed as a low-amplitude circuit for reducing current consumption, and is more prone to malfunction due to noise than the main system clock oscillator. Particular care is therefore required with the wiring method when the subsystem clock is used.

Remark For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

Capacitance (T_A = 25°C, V_{DD} = V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input capacitance	C_{IN}	f = 1 MHz Unmeasured pins returned to 0 V.				15	pF
I/O capacitance	C_{IO}	f = 1 MHz Unmeasured pins returned to 0 V.	P00 to P07, P20 to P27, P32 to P35, P40 to P43, P80 to P87, P90 to P97, P100 to P107, P110 to P113			15	pF
			P30, P31, P70 to P73			20	pF

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC characteristics (T_A = −40 to +85°C, V_{DD} = 1.8 to 5.5 V)

	Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
	Output current, high	I_{OH}	Per pin				−1	mA
			All pins				−20	mA
	Output current, low	I_{OL}	Per pin for P00 to P07, P20 to P27, P32 to P35, P40 to P43, P80 to P87, P90 to P97, P100 to P107, P110 to P113				10	mA
			Per pin for P30, P31, P70 to P73				15	mA
			Total for P80 to P87, P90 to P97, P100 to P107, P110 to P113				40	mA
			Total for P00 to P07, P20 to P27, P32 to P35, P40 to P43				20	mA
			Total for P30, P31				30	mA
			Total for P70 to P73				60	mA
	Input voltage, high	V_{IH1}	P10 to P17, P21, P24, P27, P40 to P43	2.7 V ≤ V_{DD} ≤ 5.5 V	0.7V_{DD}		V_{DD}	V
★				1.8 V ≤ V_{DD} < 2.7 V	0.8V_{DD}		V_{DD}	V
		V_{IH2}	P00 to P07, P20, P22, P23, P25, P26, P32 to P35, P80 to P87, P90 to P97, P100 to P107, P110 to P113, $\overline{\text{RESET}}$	2.7 V ≤ V_{DD} ≤ 5.5 V	0.8V_{DD}		V_{DD}	V
★				1.8 V ≤ V_{DD} < 2.7 V	0.85V_{DD}		V_{DD}	V
		V_{IH3}	P30, P31 (N-ch open-drain)	2.7 V ≤ V_{DD} ≤ 5.5 V	0.7V_{DD}		5.5	V
				1.8 V ≤ V_{DD} < 2.7 V	0.8V_{DD}		5.5	V
			P70 to P73 (N-ch open-drain)	2.7 V ≤ V_{DD} ≤ 5.5 V	0.7V_{DD}		12	V
				1.8 V ≤ V_{DD} < 2.7 V	0.8V_{DD}		12	V
		V_{IH4}	X1, X2	2.7 V ≤ V_{DD} ≤ 5.5 V	V_{DD} − 0.5		V_{DD}	V
				1.8 V ≤ V_{DD} < 2.7 V	V_{DD} − 0.2		V_{DD}	V
		V_{IH5}	XT1, XT2	4.5 V ≤ V_{DD} ≤ 5.5 V	0.8V_{DD}		V_{DD}	V
				1.8 V ≤ V_{DD} < 4.5 V	0.9V_{DD}		V_{DD}	V
	Input voltage, low	V_{IL1}	P10 to P17, P21, P24, P27, P40 to P43	2.7 V ≤ V_{DD} ≤ 5.5 V	0		0.3V_{DD}	V
★				1.8 V ≤ V_{DD} < 2.7 V	0		0.2V_{DD}	V
		V_{IL2}	P00 to P07, P20, P22, P23, P25, P26, P32 to P35, P80 to P87, P90 to P97, P100 to P107, P110 to P113, $\overline{\text{RESET}}$	2.7 V ≤ V_{DD} ≤ 5.5 V	0		0.2V_{DD}	V
★				1.8 V ≤ V_{DD} < 2.7 V	0		0.15V_{DD}	V
		V_{IL3}	P30, P31, P70 to P73 (N-ch open-drain)	4.0 V ≤ V_{DD} ≤ 5.5 V	0		0.3V_{DD}	V
				2.7 V ≤ V_{DD} < 4.0 V	0		0.2V_{DD}	V
				1.8 V ≤ V_{DD} < 2.7 V	0		0.1V_{DD}	V
		V_{IL4}	X1, X2	2.7 V ≤ V_{DD} ≤ 5.5 V	0		0.4	V
				1.8 V ≤ V_{DD} < 2.7 V	0		0.2	V
		V_{IL5}	XT1, XT2	4.5 V ≤ V_{DD} ≤ 5.5 V	0		0.2V_{DD}	V
				1.8 V ≤ V_{DD} < 4.5 V	0		0.1V_{DD}	V
	Output voltage, high	V_{OH}	I_{OH} = −1 mA	4.0 V ≤ V_{DD} ≤ 5.5 V	V_{DD} − 1.0		V_{DD}	V
			I_{OH} = −100 μA	1.8 V ≤ V_{DD} ≤ 5.5 V	V_{DD} − 0.5		V_{DD}	V

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC characteristics (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output voltage, low	V_{OL1}	P30, P31	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL} = 15 mA			2.0	V
	V_{OL2}	P70 to P73	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL} = 15 mA		0.4	1.0	V
	V_{OL3}	P00 to P07, P20 to P27, P32 to P35, P40 to P43, P80 to P87, P90 to P97, P100 to P107, P110 to P113	4.5 V ≤ V_{DD} ≤ 5.5 V, I_{OL} = 1.6 mA			0.4	V
	V_{OL4}	1.8 V ≤ V_{DD} ≤ 5.5 V, I_{OL} = 400 μA				0.5	V
Input leakage current, high	I_{LIH1}	V_{IN} = V_{DD}	P00 to P03, P10 to P17, P20 to P27, P32 to P35, P40 to P43, P80 to P87, P90 to P97, P100 to P107, P110 to P113, $\overline{\text{RESET}}$			3	μA
	I_{LIH2}		X1, X2, XT1, XT2			20	μA
	I_{LIH3}	V_{IN} = 5.5 V	P30, P31			3	μA
		V_{IN} = 12 V	P70 to P73			10	μA
Input leakage current, low	I_{LIL1}	V_{IN} = 0 V	P00 to P03, P10 to P17, P20 to P27, P32 to P35, P40 to P43, P80 to P87, P90 to P97, P100 to P107, P110 to P113, $\overline{\text{RESET}}$			–3	μA
	I_{LIL2}		X1, X2, XT1, XT2			–20	μA
	I_{LIL3}		P30, P31, P70 to P73			–3	μA
Output leakage current, high	I_{LOH}	V_{OUT} = V_{DD}				3	μA
Output leakage current, low	I_{LOL}	V_{OUT} = 0 V				–3	μA
Mask option pull-up resistor**Note 1**	R_1	V_{IN} = 0 V	P30**Note 2**, P31**Note 2**, P70 to P73	20	40	90	kΩ
Software pull-up resistor	R_2	V_{IN} = 0 V	P00 to P07, P20 to P27, P32 to P35, P40 to P43	15	30	90	kΩ

Notes 1. Mask ROM version only

2. μPD780343, 780344, 780353, 780354 (mask ROM version without I^2C bus) only

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC characteristics (T_A = −40 to +85°C, V_{DD} = 1.8 to 5.5 V) (flash memory version)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Power supply current[Note 1]	I_{DD1}[Note 2]	10 MHz crystal oscillation operating mode	V_{DD} = 5.0 V ±10%[Note 3]	When A/D converter stopped		15.0	30.0	mA
				When A/D converter is operating		16.0	32.0	mA
		5.0 MHz crystal oscillation operating mode	V_{DD} = 3.0 V ±10%[Note 3]	When A/D converter stopped		4.5	9.0	mA
				When A/D converter is operating		5.5	11.0	mA
			V_{DD} = 2.0 V ±10%[Note 4]	When A/D converter stopped		2.8	5.6	mA
				When A/D converter is operating		3.8	7.6	mA
	I_{DD2}	10 MHz crystal oscillation HALT mode	V_{DD} = 5.0 V ±10%[Note 3]	When peripheral function stopped		1.25	2.5	mA
				When peripheral function is operating			5.7	mA
		5.0 MHz crystal oscillation HALT mode	V_{DD} = 3.0 V ±10%[Note 3]	When peripheral function stopped		0.4	0.8	mA
				When peripheral function is operating			1.7	mA
			V_{DD} = 2.0 V ±10%[Note 4]	When peripheral function stopped		0.2	0.4	mA
				When peripheral function is operating			1.1	mA
	I_{DD3}	32.768 kHz crystal oscillation operating mode[Note 5]	V_{DD} = 5.0 V ±10%			115	230	μA
			V_{DD} = 3.0 V ±10%			95	190	μA
			V_{DD} = 2.0 V ±10%			75	150	μA
★		32.768 kHz crystal oscillation ×4 multiplication operating mode[Note 5]	V_{DD} = 5.0 V ±10%			280	560	μA
★			V_{DD} = 3.0 V ±10%			200	400	μA
	I_{DD4}	32.768 kHz crystal oscillation HALT mode[Note 5]	V_{DD} = 5.0 V ±10%	When LCD stopped[Note 6]		25	45	μA
				Only when LCD boost function is operating[Notes 7, 9]		27	51	μA
				When LCD is operating[Notes 8, 9]		30	60	μA
			V_{DD} = 3.0 V ±10%	When LCD stopped[Note 6]		6	18	μA
				Only when LCD boost function is operating[Notes 7, 9]		7.5	23	μA
				When LCD is operating[Notes 8, 9]		10	30	μA
			V_{DD} = 2.0 V ±10%	When LCD stopped[Note 6]		3	10	μA
				Only when LCD boost function is operating[Notes 7, 9]		4	12	μA
				When LCD is operating[Notes 8, 9]		6	18	μA
	I_{DD5}	STOP mode	V_{DD} = 5.0 V ±10%			0.1	30	μA
			V_{DD} = 3.0 V ±10%			0.05	10	μA
			V_{DD} = 2.0 V ±10%			0.05	10	μA

Notes 1. Total current flowing through the internal power supply (V_{DD0}, V_{DD1}).

2. Includes the peripheral operating current. However, the current flowing through the pull-up resistors on the ports is not included.

3. When the processor clock control register (PCC) is set to 00H.

4. When PCC is set to 02H.

5. When the main system clock is stopped.

6. When the LCD is stopped (LCDON = 0, SCOC = 0, VLCON = 0).

7. Only when the LCD boost function is operating (LCDON = 0, SCOC = 0, VLCON = 1).

8. When the LCD is operating (LCDON = 1, SCOC = 1, VLCON = 1).

9. With no load and without the LCD display panel connected. Also with capacitors for boost C1 to C4 = 0.47 μF connected, and boosting stabilized.

DC characteristics (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V) (mask ROM version)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Power supply current[Note 1]	I_{DD1}[Note 2]	10 MHz crystal oscillation operating mode	V_{DD} = 5.0 V ±10%[Note 3]	When A/D converter stopped		6.3	12.6	mA
				When A/D converter is operating		7.3	14.6	mA
		5.0 MHz crystal oscillation operating mode	V_{DD} = 3.0 V ±10%[Note 3]	When A/D converter stopped		2.0	4.0	mA
				When A/D converter is operating		3.0	6.0	mA
			V_{DD} = 2.0 V ±10%[Note 4]	When A/D converter stopped		0.4	1.5	mA
				When A/D converter is operating		1.4	4.2	mA
	I_{DD2}	10 MHz crystal oscillation HALT mode	V_{DD} = 5.0 V ±10%[Note 3]	When peripheral function stopped		1.15	2.3	mA
				When peripheral function is operating			5.7	mA
		5.0 MHz crystal oscillation HALT mode	V_{DD} = 3.0 V ±10%[Note 3]	When peripheral function stopped		0.35	0.7	mA
				When peripheral function is operating			1.7	mA
			V_{DD} = 2.0 V ±10%[Note 4]	When peripheral function stopped		0.15	0.4	mA
				When peripheral function is operating			1.1	mA
	I_{DD3}	32.768 kHz crystal oscillation operating mode[Note 5]	V_{DD} = 5.0 V ±10%			40	80	μA
			V_{DD} = 3.0 V ±10%			20	40	μA
			V_{DD} = 2.0 V ±10%			10	20	μA
		32.768 kHz crystal oscillation ×4 multiplication operating mode[Note 5]	V_{DD} = 5.0 V ±10%			85	185	μA
			V_{DD} = 3.0 V ±10%			62	106	μA
	I_{DD4}	32.768 kHz crystal oscillation HALT mode[Note 5]	V_{DD} = 5.0 V ±10%	When LCD stopped[Note 6]		25	45	μA
				Only when LCD boost function is operating[Notes 7, 9]		27	51	μA
				When LCD is operating[Notes 8, 9]		30	60	μA
			V_{DD} = 3.0 V ±10%	When LCD stopped[Note 6]		6	18	μA
				Only when LCD boost function is operating[Notes 7, 9]		7.5	23	μA
				When LCD is operating[Notes 8, 9]		10	30	μA
			V_{DD} = 2.0 V ±10%	When LCD stopped[Note 6]		3	10	μA
				Only when LCD boost function is operating[Notes 7, 9]		4	12	μA
				When LCD is operating[Notes 8, 9]		6	18	μA
	I_{DD5}	STOP mode	V_{DD} = 5.0 V ±10%			0.1	30	μA
			V_{DD} = 3.0 V ±10%			0.05	10	μA
			V_{DD} = 2.0 V ±10%			0.05	10	μA

Notes
1. Total current flowing through the internal power supply (V_{DD0}, V_{DD1}).
2. Includes the peripheral operating current. However, the current flowing through the pull-up resistors on the ports is not included.
3. When the processor clock control register (PCC) is set to 00H.
4. When PCC is set to 02H.
5. When the main system clock is stopped.
6. When the LCD is stopped (LCDON = 0, SCOC = 0, VLCON = 0).
7. Only when the LCD boost function is operating (LCDON = 0, SCOC = 0, VLCON = 1).
8. When the LCD is operating (LCDON = 1, SCOC = 1, VLCON = 1).
9. With no load and without the LCD display panel connected. Also with capacitors for boost C1 to C4 = 0.47 μF connected, and boosting stabilized.

AC characteristics

(1) Basic operation (T_A = −40 to +85°C, V_{DD} = 1.8 to 5.5 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Cycle time (minimum instruction execution time)	T_{CY}	Operating with main system clock	4.5 V ≤ V_{DD} ≤ 5.5 V	0.2		16	μs
			2.7 V ≤ V_{DD} < 4.5 V	0.4		16	μs
			1.8 V ≤ V_{DD} < 2.7 V	1.6		16	μs
		Operating with subsystem clock	When source oscillation is operating (1.8 V ≤ V_{DD} ≤ 5.5 V)	103.9 **Note 1**	122	125	μs
			When ×4 multiplication is operating (2.7 V ≤ V_{DD} ≤ 5.5 V)	28.57	30.52	31.25	μs
TI00, TI01 input high-/low-level width	t_{TIH0}, t_{TIL0}	4.5 V ≤ V_{DD} ≤ 5.5 V		2/f_{sam} + 0.1**Note 2**			μs
		2.7 V ≤ V_{DD} < 4.5 V		2/f_{sam} + 0.2**Note 2**			μs
		1.8 V ≤ V_{DD} < 2.7 V		2/f_{sam} + 0.5**Note 2**			μs
TMIB0 input frequency	f_{TIB}	2.7 V ≤ V_{DD} ≤ 5.5 V		0		4	MHz
		1.8 V ≤ V_{DD} < 2.7 V		0		275	kHz
TMIB0 input high-/low-level width	t_{TIHB}, t_{TILB}	2.7 V ≤ V_{DD} ≤ 5.5 V		125			ns
		1.8 V ≤ V_{DD} < 2.7 V		1.8			μs
TI50, TI51 input frequency	f_{TI5}	2.7 V ≤ V_{DD} ≤ 5.5 V		0		4	MHz
		1.8 V ≤ V_{DD} < 2.7 V		0		275	kHz
TI50, TI51 input high-/low-level width	t_{TIH5}, t_{TIL5}	2.7 V ≤ V_{DD} ≤ 5.5 V		100			ns
		1.8 V ≤ V_{DD} < 2.7 V		1.8			μs
Interrupt request input high-/low-level width	t_{INTH}, t_{INTL}	INTP0 to INTP6, P40 to P43	2.7 V ≤ V_{DD} ≤ 5.5 V	1			μs
			1.8 V ≤ V_{DD} < 2.7 V	2			μs
$\overline{\text{RESET}}$ low-level width	t_{RSL}	2.7 V ≤ V_{DD} ≤ 5.5 V		10			μs
		1.8 V ≤ V_{DD} < 2.7 V		20			μs

Notes 1. Value when using an external clock. When using a crystal resonator, the value becomes 114 μs (MIN.).

2. Selection of f_{sam} = f_X, f_X/4, f_X/64 is possible using bits 0 and 1 (PRM00, PRM01) of prescaler mode register 0 (PRM0). However, if the TI00 valid edge is selected as the count clock, the value becomes f_{sam} = f_X/8.

(2) Serial interface (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

(a) SIO3 3-wire serial I/O mode ($\overline{\text{SCK3}}$ ... internal clock output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{SCK3}}$ cycle time	t_{KCY1}	4.5 V ≤ V_{DD} ≤ 5.5 V	800			ns
		2.7 V ≤ V_{DD} < 4.5 V	1,600			ns
		1.8 V ≤ V_{DD} < 2.7 V	3,200			ns
$\overline{\text{SCK3}}$ high-/low-level width	t_{KH1}, t_{KL1}	4.5 V ≤ V_{DD} ≤ 5.5 V	t_{KCY1}/2 – 50			ns
		1.8 V ≤ V_{DD} < 4.5 V	t_{KCY1}/2 – 100			ns
SI3 setup time (to $\overline{\text{SCK3}}$↑)	t_{SIK1}	4.5 V ≤ V_{DD} ≤ 5.5 V	100			ns
		2.7 V ≤ V_{DD} < 4.5 V	150			ns
		1.8 V ≤ V_{DD} < 2.7 V	300			ns
SI3 hold time (from $\overline{\text{SCK3}}$↑)	t_{KSI1}		400			ns
Delay time from $\overline{\text{SCK3}}$↓ to SO3 output	t_{KSO1}	C = 100 pF^{Note}			300	ns

Note C is the load capacitance of the $\overline{\text{SCK3}}$ and SO3 output lines.

(b) SIO3 3-wire serial I/O mode ($\overline{\text{SCK3}}$ ... external clock input)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{SCK3}}$ cycle time	t_{KCY2}	4.5 V ≤ V_{DD} ≤ 5.5 V	800			ns
		2.7 V ≤ V_{DD} < 4.5 V	1,600			ns
		1.8 V ≤ V_{DD} < 2.7 V	3,200			ns
$\overline{\text{SCK3}}$ high-/low-level width	t_{KH2}, t_{KL2}	4.5 V ≤ V_{DD} ≤ 5.5 V	400			ns
		2.7 V ≤ V_{DD} < 4.5 V	800			ns
		1.8 V ≤ V_{DD} < 2.7 V	1,600			ns
SI3 setup time (to $\overline{\text{SCK3}}$↑)	t_{SIK2}		100			ns
SI3 hold time (from $\overline{\text{SCK3}}$↑)	t_{KSI2}		400			ns
Delay time from $\overline{\text{SCK3}}$↓ to SO3 output	t_{KSO2}	C = 100 pF^{Note}			300	ns

Note C is the load capacitance of the SO3 output line.

(c) CSI1 3-wire serial I/O mode (SCK1 ... internal clock output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
★ SCK1 cycle time	t_{KCY3}	4.5 V ≤ V_{DD} ≤ 5.5 V	200			ns
		2.7 V ≤ V_{DD} < 4.5 V	500			ns
		1.8 V ≤ V_{DD} < 2.7 V	1			μs
SCK1 high-/low-level width	t_{KH3}, t_{KL3}	4.5 V ≤ V_{DD} ≤ 5.5 V	$t_{KCY3}/2 - 5$			ns
		2.7 V ≤ V_{DD} < 4.5 V	$t_{KCY3}/2 - 20$			ns
		1.8 V ≤ V_{DD} < 2.7 V	$t_{KCY3}/2 - 30$			ns
SI1 setup time (to SCK1↑)	t_{SIK3}		20			ns
SI1 hold time (from SCK1↑)	t_{KSI3}		110			ns
Delay time from SCK1↓ to SO1 output	t_{KSO3}	C = 100 pF[Note]			150	ns

Note C is the load capacitance of the SCK1 and SO1 output lines.

(d) CSI1 3-wire serial I/O mode (SCK1 ... external clock input)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
SCK1 cycle time	t_{KCY4}	4.5 V ≤ V_{DD} ≤ 5.5 V	200			ns
		2.7 V ≤ V_{DD} < 4.5 V	500			ns
		1.8 V ≤ V_{DD} < 2.7 V	1			μs
SCK1 high-/low-level width	t_{KH4}, t_{KL4}	4.5 V ≤ V_{DD} ≤ 5.5 V	100			ns
		2.7 V ≤ V_{DD} < 4.5 V	250			ns
		1.8 V ≤ V_{DD} < 2.7 V	500			ns
SI1 setup time (to SCK1↑)	t_{SIK4}		25			ns
SI1 hold time (from SCK1↑)	t_{KSI4}		110			ns
Delay time from SCK1↓ to SO1 output	t_{KSO4}	C = 100 pF[Note]			150	ns

Note C is the load capacitance of the SO1 output line.

(e) UART0 (dedicated baud rate generator output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		4.5 V ≤ V_{DD} ≤ 5.5 V			156,250	bps
		2.7 V ≤ V_{DD} < 4.5 V			78,125	bps
		1.8 V ≤ V_{DD} < 2.7 V			39,063	bps

(f) I^{2}C bus mode

Parameter		Symbol	Standard Mode		High-Speed Mode		Unit
			MIN.	MAX.	MIN.	MAX.	
SCL0 clock frequency		f_{SCL}	0	100	0	400	kHz
Bus free time (between stop and start condition)		t_{BUF}	4.7	–	1.3	–	μs
Hold time[Note 1]		$t_{HD:STA}$	4.0	–	0.6	–	μs
SCL0 clock low-level width		t_{LOW}	4.7	–	1.3	–	μs
SCL0 clock high-level width		t_{HIGH}	4.0	–	0.6	–	μs
Start/restart condition setup time		$t_{SU:STA}$	4.7	–	0.6	–	μs
Data hold time	CBUS compatible master	$t_{HD:DAT}$	5.0	–	–	–	μs
	I^{2}C bus		0[Note 2]	–	0[Note 2]	0.9[Note 3]	μs
Data setup time		$t_{SU:DAT}$	250	–	100[Note 4]	–	ns
SDA0 and SCL0 signal rise time		t_R	–	1,000	20 + 0.1Cb[Note 5]	300	ns
SDA0 and SCL0 signal fall time		t_F	–	300	20 + 0.1Cb[Note 5]	300	ns
Stop condition setup time		$t_{SU:STO}$	4.0	–	0.6	–	μs
Capacitive load per each bus line		Cb	–	400	–	400	pF
Spike pulse width controlled by input filter		t_{SP}	–	–	0	50	ns

Notes **1.** On the start condition, the first clock pulse is generated after the hold period.

2. To fulfill the undefined area of the SCL0 falling edge, it is necessary for the device to provide internally an SDA0 signal (on $V_{IHmin.}$ of the SCL0 signal) with at least 300 ns of hold time.

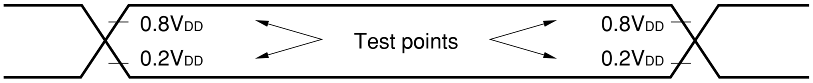
3. If the device does not extend the SCL0 signal low hold time (t_{LOW}), only maximum data hold time $t_{HD:DAT}$ needs to be fulfilled.

4. The high-speed mode I^{2}C bus is available in the standard mode I^{2}C bus system. At this time, the conditions described below must be satisfied.

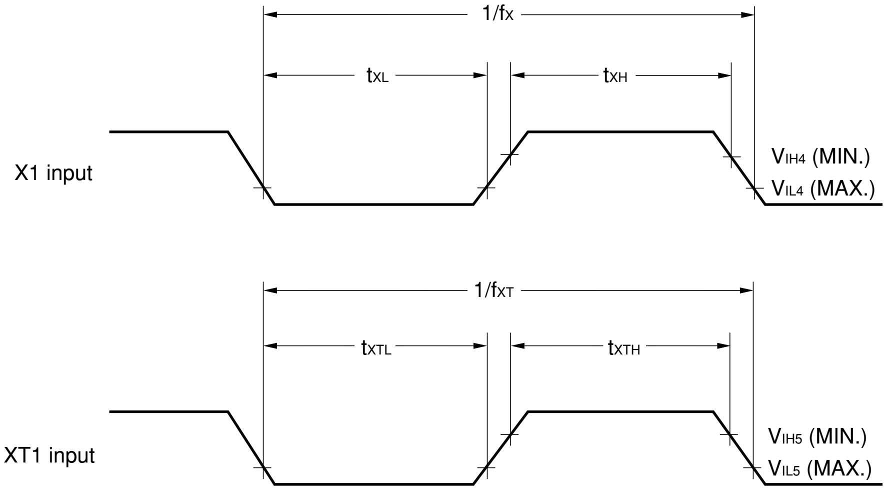
- If the device does not extend the SCL0 signal low state hold time
  $t_{SU:DAT} \geq 250$ ns
- If the device extends the SCL0 signal low state hold time
  Be sure to transmit the next data bit to the SDA0 line before the SCL0 line is released ($t_{Rmax.} + t_{SU:DAT} = 1{,}000 + 250 = 1{,}250$ ns by standard mode I^{2}C bus specification).

5. Cb: Total capacitance per bus line (unit: pF)

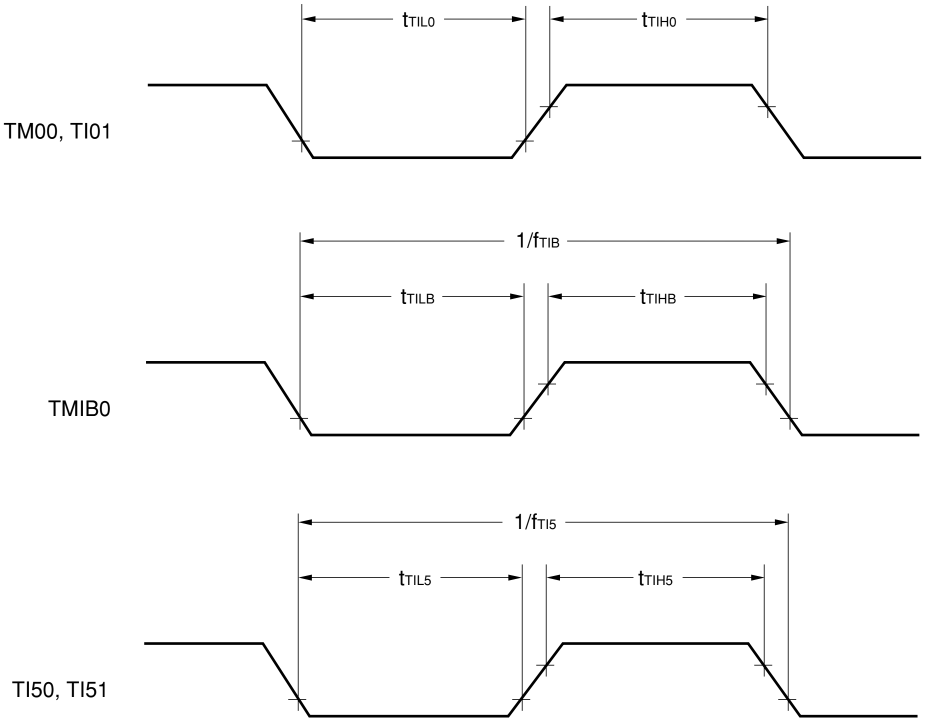
AC timing test points (excluding X1, XT1 input)



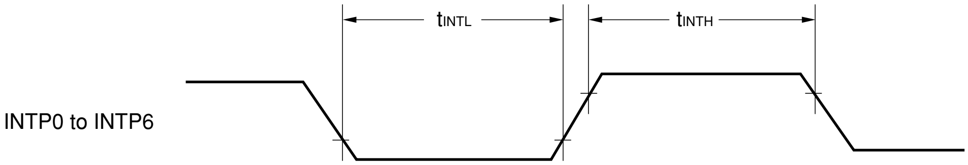
Clock timing



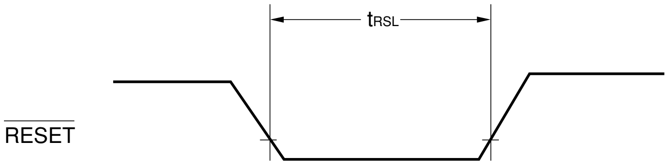
TI timing



Interrupt request input timing

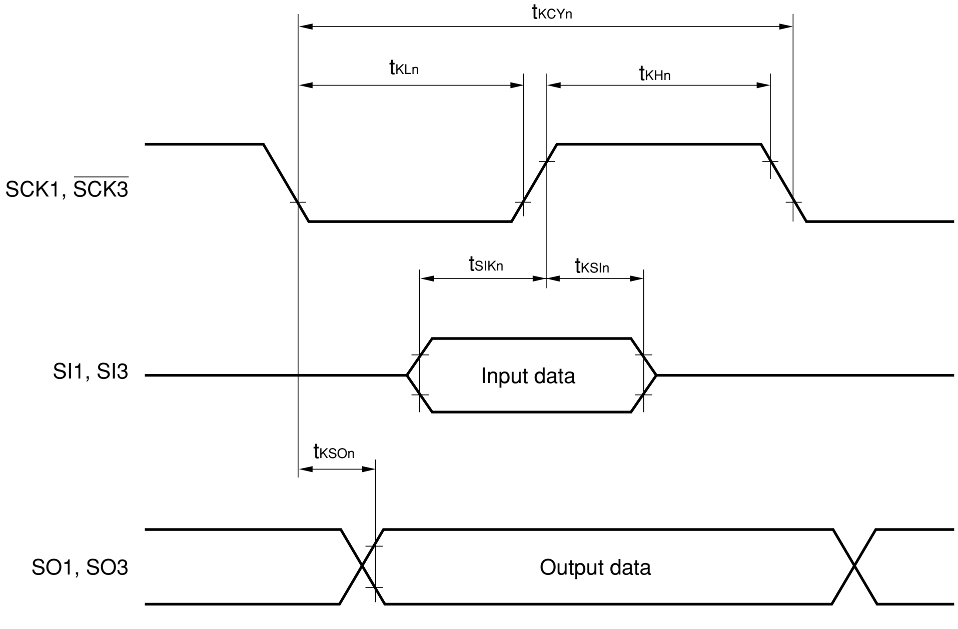


$\overline{\text{RESET}}$ input timing



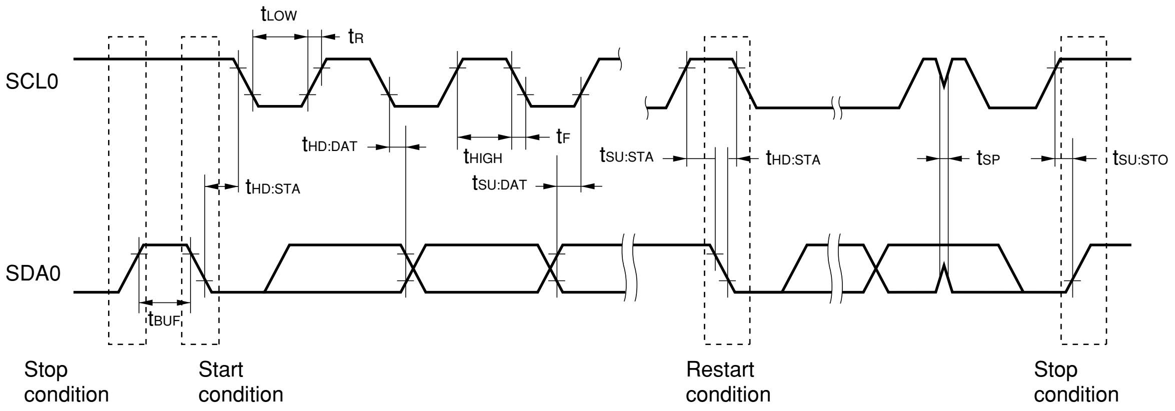
Serial transfer timing

3-wire serial I/O mode (SIO3, CSI1)



n = 1 to 4

I^{2}C bus mode



★ 8-bit A/D converter characteristics (μPD780344, 780344Y Subseries only)
(T_A = –40 to +85°C, AV_{DD} = V_{DD} = 2.2 to 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution			8	8	8	bit
Overall error[Note]		4.5 V ≤ AV_{DD} ≤ 5.5 V		±0.2	±0.4	%FSR
		2.7 V ≤ AV_{DD} < 4.5 V		±0.3	±0.6	%FSR
		2.2 V ≤ AV_{DD} < 2.7 V		±0.6	±1.2	%FSR
Conversion time	t_{CONV}	4.5 V ≤ AV_{DD} ≤ 5.5 V	14		100	μs
		2.7 V ≤ AV_{DD} < 4.5 V	19		100	μs
		2.2 V ≤ AV_{DD} < 2.7 V			100	μs
Analog input voltage	V_{AIN}		0		AV_{DD}	V
Series resistor string resistance	R_{REF}	At A/D conversion operation	20	40		kΩ

Note Excluding quantization error (±1/2 LSB). It is indicated as a ratio (%FSR) to the full-scale value.

★ 10-bit A/D converter characteristics (μPD780354, 780354Y Subseries only)
(T_A = –40 to +85°C, AV_{DD} = V_{DD} = 2.2 to 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution			10	10	10	bit
Overall error[Note]		4.5 V ≤ AV_{DD} ≤ 5.5 V		±0.2	±0.4	%FSR
		2.7 V ≤ AV_{DD} < 4.5 V		±0.3	±0.6	%FSR
		2.2 V ≤ AV_{DD} < 2.7 V		±0.6	±1.2	%FSR
Conversion time	t_{CONV}	4.5 V ≤ AV_{DD} ≤ 5.5 V	14		100	μs
		2.7 V ≤ AV_{DD} < 4.5 V	19		100	μs
		2.2 V ≤ AV_{DD} < 2.7 V	28		100	μs
Zero-scale error[Note]		4.5 V ≤ AV_{DD} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV_{DD} < 4.5 V			±0.6	%FSR
		2.2 V ≤ AV_{DD} < 2.7 V			±1.2	%FSR
Full-scale error[Note]		4.5 V ≤ AV_{DD} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV_{DD} < 4.5 V			±0.6	%FSR
		2.2 V ≤ AV_{DD} < 2.7 V			±1.2	%FSR
Integral linearity error		4.5 V ≤ AV_{DD} ≤ 5.5 V			±2.5	LSB
		2.7 V ≤ AV_{DD} < 4.5 V			±4.5	LSB
		2.2 V ≤ AV_{DD} < 2.7 V			±8.5	LSB
Differential linearity error		4.5 V ≤ AV_{DD} ≤ 5.5 V			±1.5	LSB
		2.7 V ≤ AV_{DD} < 4.5 V			±2.0	LSB
		2.2 V ≤ AV_{DD} < 2.7 V			±3.5	LSB
Analog input impedance		Sampling			100	kΩ
		Non-sampling		10		MΩ
Analog input voltage	V_{AIN}		0		AV_{DD}	V
Series resistor string resistance	R_{REF}	At A/D conversion operation	20	40		kΩ

Note Excluding quantization error (±1/2 LSB). It is indicated as a ratio (%FSR) to the full-scale value.

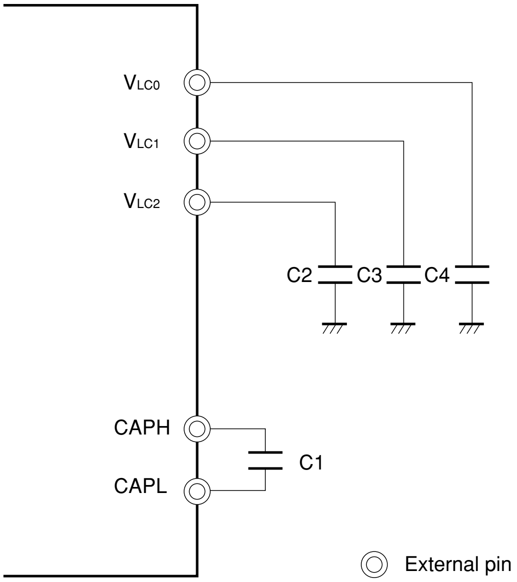
LCD controller/driver characteristics (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
LCD reference voltage	V_{LCD2}	C1 to C4 = 0.47 μF	Gain = 1	0.84	1	1.165	V
			Gain = 1.5	1.26	1.5	1.74	V
Gain adjustment				1.0		1.5	times
Doubler output voltage	V_{LCD1}	C1 to C4 = 0.47 μF		$2V_{LCD2}$ – 0.1	$2V_{LCD2}$	$2V_{LCD2}$	V
Tripler output voltage	V_{LCD0}	C1 to C4 = 0.47 μF		$3V_{LCD2}$ – 0.15	$3V_{LCD2}$	$3V_{LCD2}$	V
Boost wait time**Note 1**	t_{VAWAIT}	Gain = 1	4.5 V ≤ V_{DD} ≤ 5.5 V	4			s
			1.8 V ≤ V_{DD} < 4.5 V	0.5			s
		Gain = 1.5	1.8 V ≤ V_{DD} ≤ 5.5 V	0.5			s
LCD output resistance**Note 2** (common)	R_{ODC}					40	kΩ
LCD output resistance**Note 2** (segment)	R_{ODS}					200	kΩ

Notes 1. The boost wait time is the wait time from when boosting is started to when display is enabled.

2. The output resistance is the resistance between the segment/common pin and the V_{LC0}, V_{LC1}, V_{LC2}, or V_{SS} pin.

Remark C1, C2, C3, and C4 are the capacitors connected between CAPH and CAPL, V_{LC2} and GND, V_{LC1} and GND, and V_{LC0} and GND, respectively.



- C1 = C2 = C3 = C4 = 0.47 [μF]

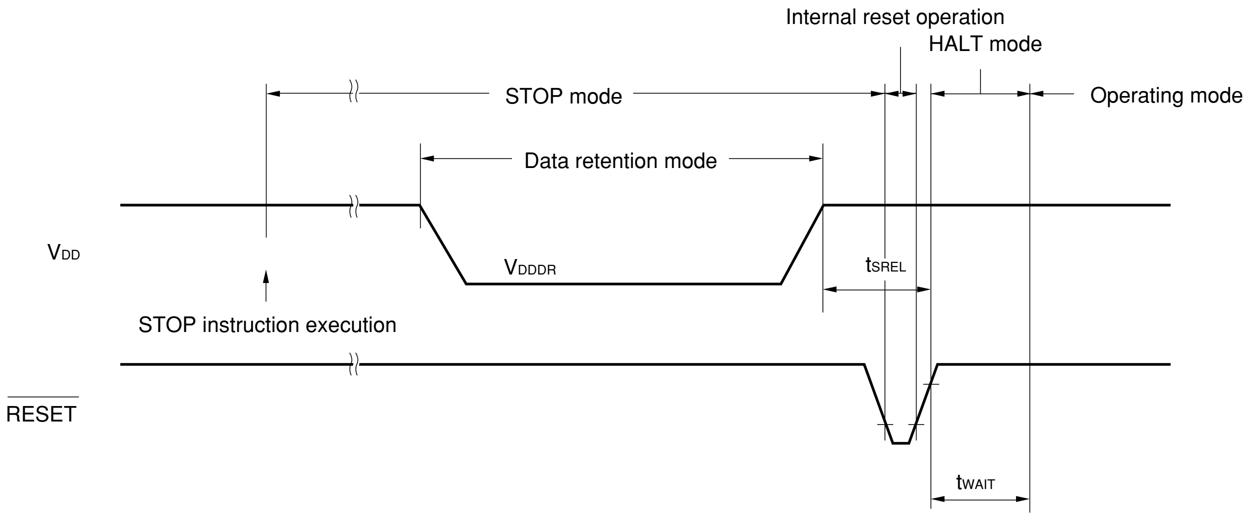
	V_{LCD2} (V)	V_{LCD1} (V)	V_{LCD0} (V)
V_{LCD0} = 3 V (gain = 1)	1	2	3
V_{LCD0} = 4.5 V (gain = 1.5)	1.5	3	4.5

Data memory STOP mode low power supply voltage data retention characteristics (T_A = –40 to +85°C)

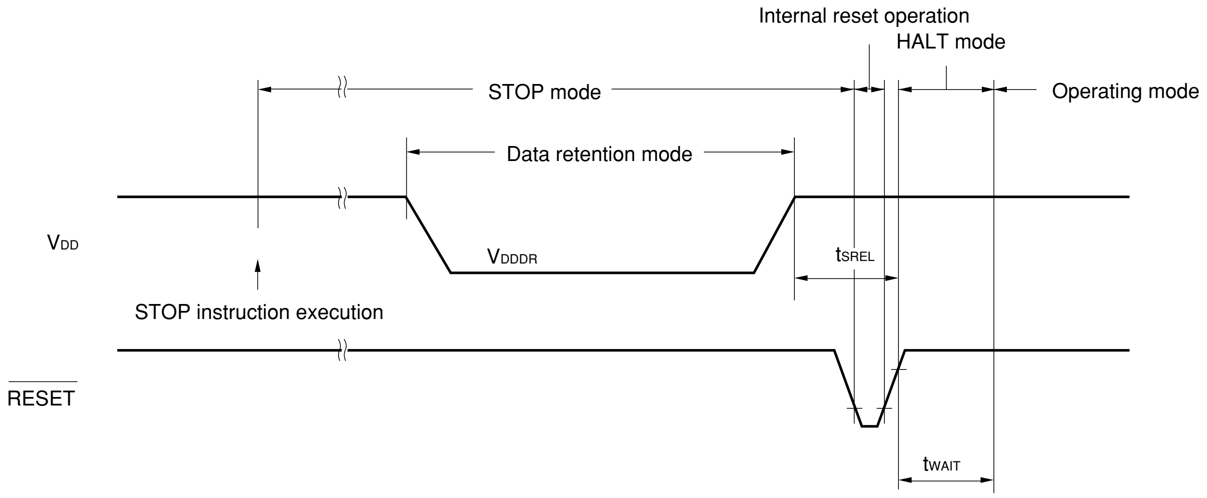
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention power supply voltage	V_{DDDR}		1.6		5.5	V
Data retention power supply current	I_{DDDR}	V_{DDDR} = 2.7 V		0.1	30	μA
Release signal set time	t_{SREL}		0			μs
Oscillation stabilization wait time	t_{WAIT}	Release by $\overline{\text{RESET}}$		2^{17}/fx		s
		Release by interrupt request		**Note**		s

Note Selection of 2^{12}/fx, 2^{14}/fx, 2^{15}/fx, 2^{16}/fx, and 2^{17}/fx is possible using bits 0 to 2 (OSTS0 to OSTS2) of the oscillation stabilization time select register (OSTS).

Data retention timing (STOP mode release by $\overline{\text{RESET}}$)



Data retention timing (standby release signal: STOP mode release by interrupt request signal)



Flash memory programming characteristics: μPD78F0354, 78F0354A, 78F0354Y, and 78F0354AY only (V_{DD} = 1.8 to 5.5 V, V_{SS} = 0 V, V_{PP} = 9.7 to 10.3 V)

(1) Basic characteristics

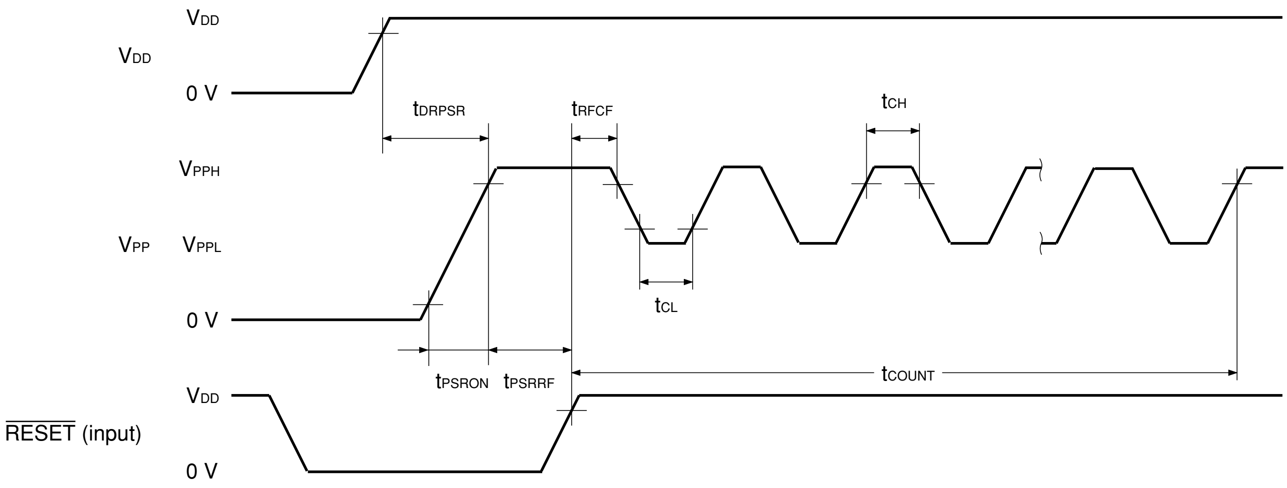
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Operating frequency	f_X	4.5 V ≤ V_{DD} ≤ 5.5 V	2.0		10.0	MHz
		2.7 V ≤ V_{DD} < 4.5 V	2.0		5.0	MHz
★		1.8 V ≤ V_{DD} < 2.7 V	2.0		1.25	MHz
Supply voltage	V_{DD}	Operating voltage during write operation	2.7		1.25	V
	V_{PPL}	When detecting V_{PP} low level	0		$0.2V_{DD}$	V
	V_{PP}	When detecting V_{PP} high level	$0.8V_{DD}$	V_{DD}	$1.2V_{DD}$	V
	V_{PPH}	When detecting V_{PP} high voltage**Note**	9.7	10.0	10.3	V
V_{DD} supply current	I_{DD}				10	mA
V_{PP} supply current	I_{PP}	V_{PP} = 10 V		75	100	mA
Write time (per byte)	T_{WRT}		50		500	μs
Number of rewriting times	C_{WRT}				20	times
Erase time	T_{ERASE}		0.2		20	s
Programming temperature	T_{PRG}		10		40	°C

Note For details of the input/output voltage and input/output leakage current, refer to DC characteristics.

(2) Write operation characteristics

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
V_{PP} set time	t_{PSRON}	V_{PP} high voltage	1.0			μs
★ V_{PP}↑ set time from V_{DD}↑	t_{DRPSR}	V_{PP} high voltage	10			μs
$\overline{\text{RESET}}$↑ set time from V_{PP}↑	t_{PSRRF}	V_{PP} high voltage	1.0			μs
V_{PP} count start time from $\overline{\text{RESET}}$↑	t_{RFCF}		1.0			μs
Count execution time	t_{COUNT}				2.0	ms
V_{PP} counter high-/low-level width	t_{CH}, t_{CL}		8.0			μs
V_{PP} counter noise elimination width	t_{NFW}			40		ns

Flash write mode setting timing

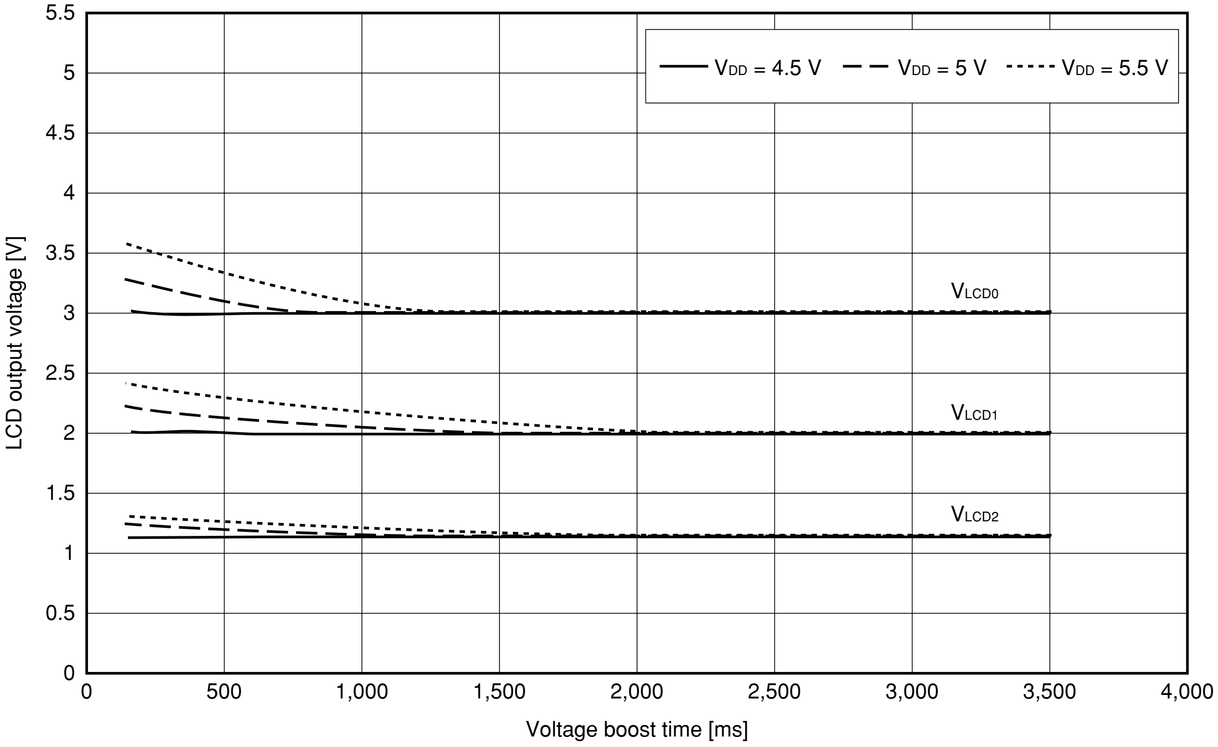


CHAPTER 26 CHARACTERISTICS CURVES OF LCD CONTROLLER/DRIVER (REFERENCE VALUES)

(1) Characteristics curves of voltage boost stabilization time

★ The following shows the characteristics curves of the time from the start of voltage boost (VLCON = 1) and the changes in the LCD output voltage (when GAIN is set to 0 (using the 3 V display panel)).

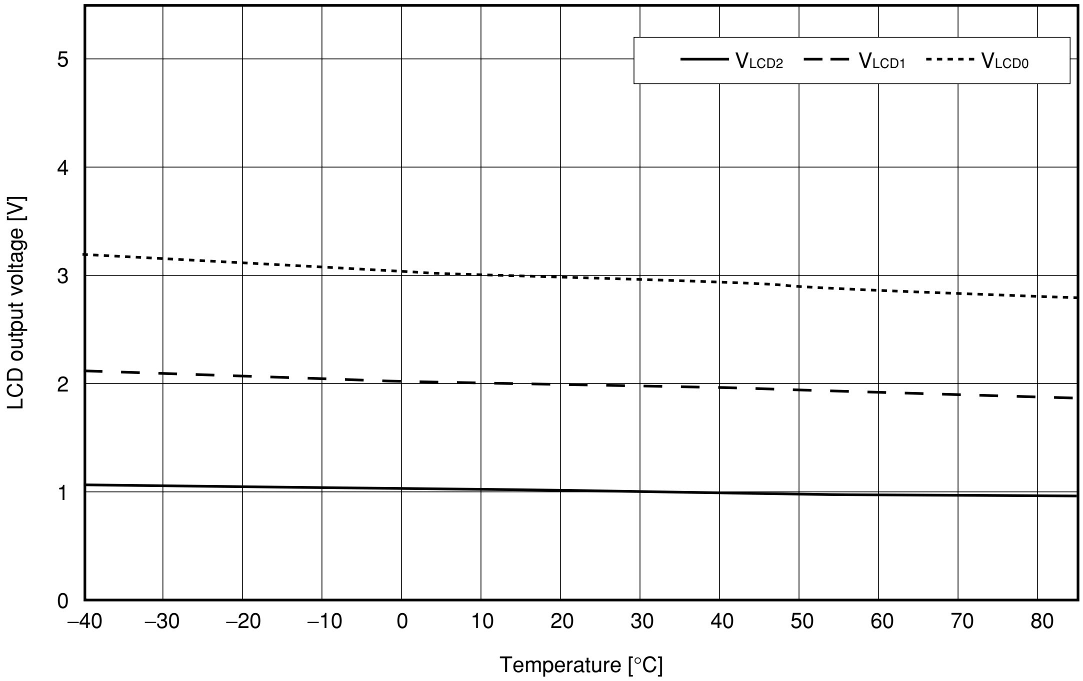
LCD output voltage/voltage boost time



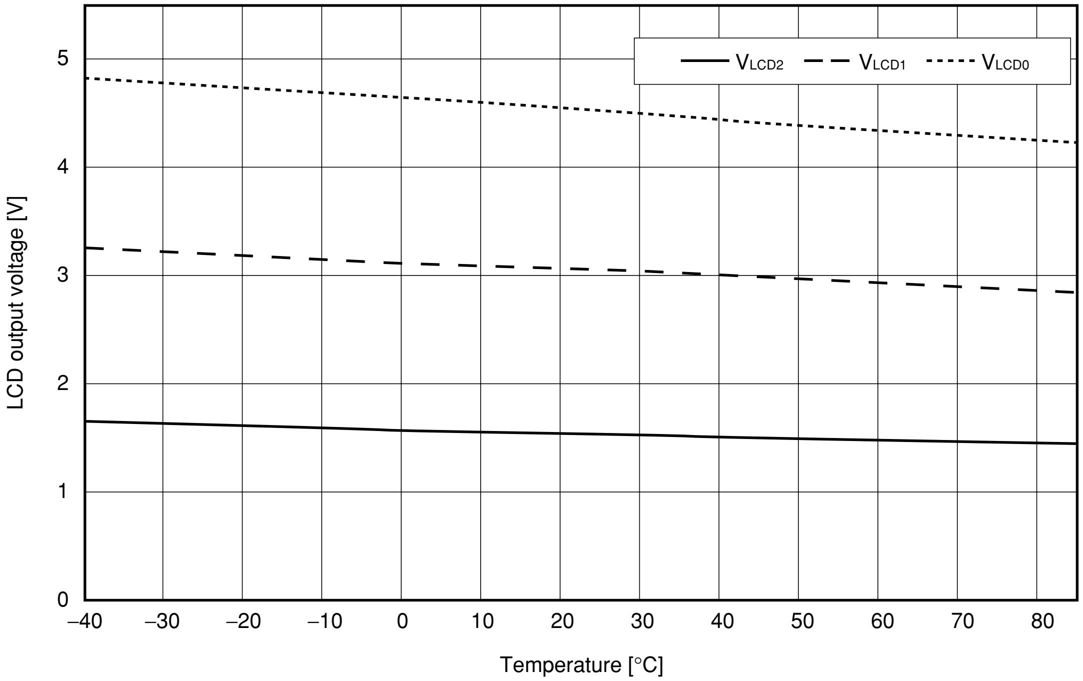
(2) Temperature characteristics of LCD output voltage

The following shows the temperature characteristics curves of LCD output voltage.

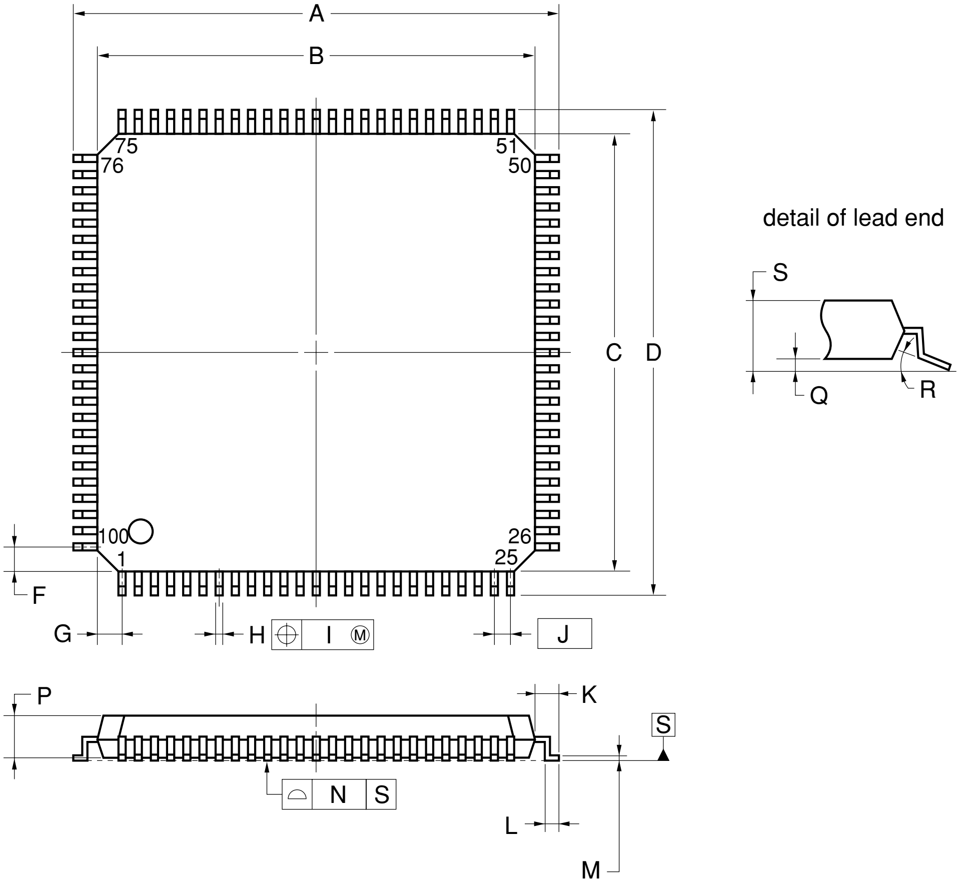
★ **LCD output voltage/temperature (when GAIN = 0)**



★ **LCD output voltage/temperature (when GAIN = 1)**



100-PIN PLASTIC LQFP (FINE PITCH) (14x14)



NOTE

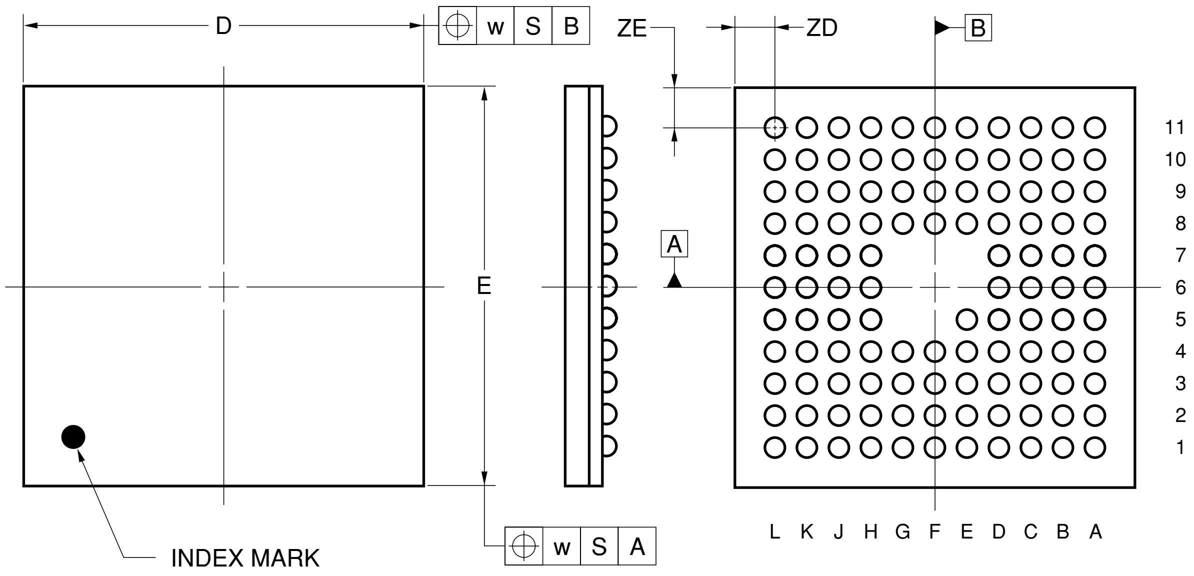
Each lead centerline is located within 0.08 mm of its true position (T.P.) at maximum material condition.

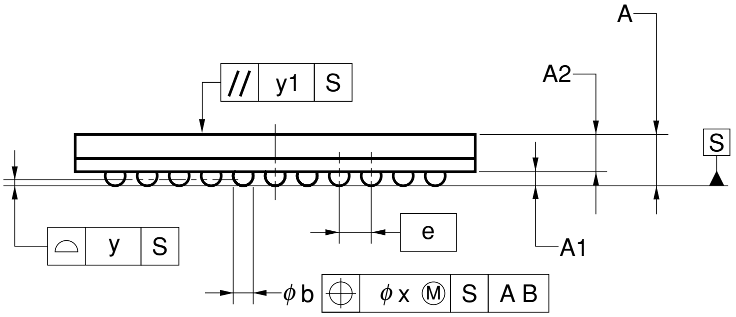
ITEM	MILLIMETERS
A	16.00±0.20
B	14.00±0.20
C	14.00±0.20
D	16.00±0.20
F	1.00
G	1.00
H	$0.22^{+0.05}_{-0.04}$
I	0.08
J	0.50 (T.P.)
K	1.00±0.20
L	0.50±0.20
M	$0.17^{+0.03}_{-0.07}$
N	0.08
P	1.40±0.05
Q	0.10±0.05
R	$3^{\circ}{}^{+7^{\circ}}_{-3^{\circ}}$
S	1.60 MAX.

S100GC-50-8EU, 8EA-2

Remark The dimensions and materials of the ES version are the same as those of the mass-produced version.

★ 113-PIN PLASTIC FBGA (10x10)





(UNIT:mm)

ITEM	DIMENSIONS
D	10.00±0.10
E	10.00±0.10
w	0.20
A	1.28±0.10
A1	0.35±0.06
A2	0.93
$\boxed{e}$	0.80
b	$0.50^{+0.05}_{-0.10}$
x	0.08
y	0.10
y1	0.20
ZD	1.00
ZE	1.00

P113F1-80-DA3

Remark The dimensions and materials of the ES version are the same as those of the mass-produced version.

★

CHAPTER 28 RECOMMENDED SOLDERING CONDITIONS

This product should be soldered and mounted under the following recommended conditions.

For soldering methods and conditions other than those recommended below, contact an NEC Electronics sales representative.

For technical information, see the following website.

Semiconductor Device Mount Manual (http://www.necel.com/pkg/en/mount/index.html)

Table 28-1. Surface Mounting Type Soldering Conditions (1/3)

(1) μPD780343GC-×××-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD780344GC-×××-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD780353GC-×××-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD780354GC-×××-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD780343YGC-×××-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD780344YGC-×××-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD780353YGC-×××-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD780354YGC-×××-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD78F0354GC-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD78F0354YGC-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 235°C, Time: 30 seconds max. (at 210°C or higher), Count: Two times or less, Exposure limit: 7 days**Note** (after that, prebake at 125°C for 10 hours)	IR35-107-2
VPS	Package peak temperature: 215°C, Time: 40 seconds max. (at 200°C or higher), Count: Two times or less, Exposure limit: 7 days**Note** (after that, prebake at 125°C for 10 hours)	VP15-107-2
Partial heating	Pin temperature: 350°C max., Time: 3 seconds max. (per pin row)	—

Note After opening the dry pack, store it at 25°C or less and 65%RH or less for the allowable storage period.

Caution Do not use different soldering methods together (except for partial heating).

Table 28-1. Surface Mounting Type Soldering Conditions (2/3)

(2) μPD780343F1-×××-DA3: 113-pin plastic FBGA (10 × 10)
μPD780344F1-×××-DA3: 113-pin plastic FBGA (10 × 10)
μPD780353F1-×××-DA3: 113-pin plastic FBGA (10 × 10)
μPD780354F1-×××-DA3: 113-pin plastic FBGA (10 × 10)
μPD780343YF1-×××-DA3: 113-pin plastic FBGA (10 × 10)
μPD780344YF1-×××-DA3: 113-pin plastic FBGA (10 × 10)
μPD780353YF1-×××-DA3: 113-pin plastic FBGA (10 × 10)
μPD780354YF1-×××-DA3: 113-pin plastic FBGA (10 × 10)
μPD78F0354F1-DA3: 113-pin plastic FBGA (10 × 10)
μPD78F0354YF1-DA3: 113-pin plastic FBGA (10 × 10)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 235°C, Time: 30 seconds max. (at 210°C or higher), Count: Two times or less, Exposure limit: 7 days**Note** (after that, prebake at 125°C for 10 hours)	IR35-107-2
VPS	Package peak temperature: 215°C, Time: 40 seconds max. (at 200°C or higher), Count: Two times or less, Exposure limit: 7 days**Note** (after that, prebake at 125°C for 10 hours)	VP15-107-2

Note After opening the dry pack, store it at 25°C or less and 65%RH or less for the allowable storage period.

Caution Do not use different soldering methods together.

Table 28-1. Surface Mounting Type Soldering Conditions (3/3)

(3) μPD780343GC-××× -8EU-A: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD780344GC-×××-8EU-A: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD780353GC-×××-8EU-A: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD780354GC-×××-8EU-A: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD78F0354GC-8EU-A: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD780343YGC-×××-8EU-A: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD780344YGC-×××-8EU-A: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD780353YGC-×××-8EU-A: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD780354YGC-×××-8EU-A: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD78F0354YGC-8EU-A: 100-pin plastic LQFP (fine pitch) (14 × 14)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 260°C, Time: 60 seconds max. (at 220°C or higher), Count: Three times or less, Exposure limit: 7 days[Note] (after that, prebake at 125°C for 20 to 72 hours)	IR60-207-3
Partial heating	Pin temperature: 350°C max., Time: 3 seconds max. (per pin row)	—

Note After opening the dry pack, store it at 25°C or less and 65%RH or less for the allowable storage period.

Remark Products that have the part numbers suffixed by “-A” are lead-free products.

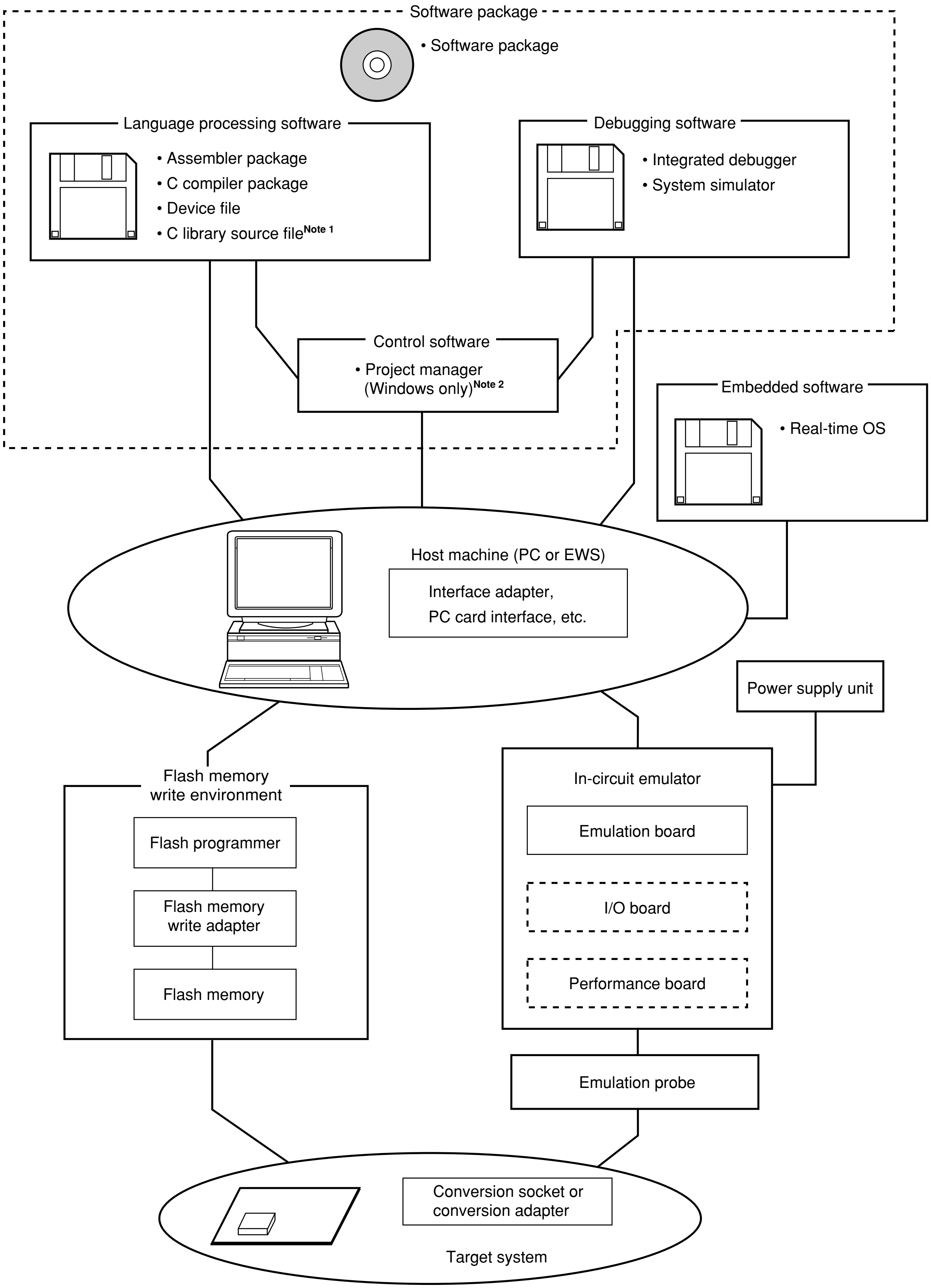
APPENDIX A DEVELOPMENT TOOLS

The following development tools are available for the development of systems which employ the μPD780344, 780354, 780344Y, 780354Y Subseries. Figure A-1 shows the configuration example of the tools.

- **Support for PC98-NX series**
  Unless otherwise specified, products supported by IBM PC/AT™ compatibles can be used for PC98-NX series computers. When using PC98-NX series computers, refer to the description for IBM PC/AT compatibles.

- **Windows**
  Unless otherwise specified, "Windows" means the following OSs.
  - Windows 3.1
  - Windows 95
  - Windows 98
  - Windows 2000
  - Windows NT™ Ver. 4.0

Figure A-1. Configuration of Development Tools



★ **Notes 1.** The C library source file is not included in the software package.

2. The project manager is included in the assembler package.
The project manager is only used for Windows.

A.1 Software Package

SP78K0 Software package	This package contains various software tools for 78K/0 Series development. The following tools are included. RA78K0, CC78K0, ID78K0-NS, SM78K0, and various device files
	Part Number: μS××××SP78K0

Remark ×××× in the part number differs depending on the OS used.

μS××××SP78K0

××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	

A.2 Language Processing Software

RA78K0 Assembler package	This assembler converts programs written in mnemonics into object codes executable with a microcontroller. Further, this assembler is provided with functions capable of automatically creating symbol tables and branch instruction optimization. This assembler should be used in combination with a device file (DF780354) (sold separately). **<Caution when using RA78K0 in PC environment>** This assembler package is a DOS-based application. It can also be used in Windows, however, by using the project manager (included in assembler package) on Windows.
	Part Number: μS××××RA78K0
CC78K0 C compiler package	This compiler converts programs written in C language into object codes executable with a microcontroller. This compiler should be used in combination with an assembler package and device file (both sold separately). **<Caution when using CC78K0 in PC environment>** This C compiler package is a DOS-based application. It can also be used in Windows, however, by using the project manager (included in the assembler package) on Windows.
	Part Number: μS××××CC78K0
DF780354**Note 1** Device file	This file contains information peculiar to the device. This device file should be used in combination with a tool (RA78K0, CC78K0, SM78K0, ID78K0-NS, and RX78K0) (all sold separately). The corresponding OS and host machine differ depending on the tool used.
	Part Number: μS××××DF780354
CC78K0-L**Note 2** C library source file	This is a source file of functions configuring the object library included in the C compiler package. This file is required to match the object library included in C compiler package to the user's specifications. It does not depend on the operating environment because it is a source file.
	Part Number: μS××××CC78K0-L

Notes 1. The DF780354 can be used in common with the RA78K0, CC78K0, SM78K0, ID78K0-NS, and RX78K0.

2. CC78K0-L is not included in the software package (SP78K0).

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××RA78K0
μS××××CC78K0

××××	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$^{\text{TM}}$	HP-UX$^{\text{TM}}$ (Rel. 10.10)	
★ 3K17	SPARCstation$^{\text{TM}}$	SunOS$^{\text{TM}}$ (Rel. 4.1.4), Solaris$^{\text{TM}}$ (Rel. 2.5.1)	

μS××××DF780354
μS××××CC78K0-L

××××	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), Solaris (Rel. 2.5.1)	3.5-inch 2HD FD
3K15			1/4-inch CGMT

A.3 Control Software

Project manager	This is control software designed to enable efficient user program development in the Windows environment. All operations used in development of a user program, such as starting the editor, building, and starting the debugger, can be performed from the project manager. **<Caution>** The project manager is included in the assembler package (RA78K0). It can only be used in Windows.

★ A.4 Flash Memory Writing Tools

Flashpro III (Part number: FL-PR3, PG-FP3) Flashpro IV (Part number: FL-PR4, PG-FP4) Flash programmer	Flash programmer dedicated to microcontrollers with on-chip flash memory.
FA-100GC-8EU Flash memory writing adapter	Flash memory writing adapter used connected to the Flashpro III/Flashpro IV. • FA-100GC-8EU: 100-pin plastic LQFP (GC-8EU type)

Remark FL-PR3, FL-PR4, and FA-100GC-8EU are products of Naito Densei Machida Mfg. Co., Ltd.
Contact: +81-45-475-4191 Naito Densei Machida Mfg. Co., Ltd.

A.5 Debugging Tools (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 an integrated debugger (ID78K0-NS). This emulator should be used in combination with a power supply unit, emulation probe, and interface adapter which is required to connect this emulator to the host machine.
IE-78K0-NS-PA Performance board	This board is used for extending the IE-78K0-NS functions, and is used connected to the IE-78K0-NS. With the addition of this board, the addition of a coverage function, enhancement of tracer and timer functions, and other such debugging function enhancements are possible.
IE-78K0-NS-A In-circuit emulator	In-circuit emulator 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 receptacle of 100 to 240 V AC.
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 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 host machine (PCMCIA socket compatible).
IE-70000-PC-IF-C Interface adapter	This adapter is required when using an IBM PC/AT compatible computer as the IE-78K0-NS host machine (ISA bus compatible).
IE-70000-PCI-IF-A Interface adapter	This adapter is required when using a PC with a PCI bus as the IE-78K0-NS host machine.
IE-780354-NS-EM1 Emulation board	This board emulates the operations of the peripheral hardware peculiar to a device. It should be used in combination with an in-circuit emulator.
NP-100GC NP-H100GC-TQ Emulation probe	This probe is used to connect the in-circuit emulator to the target system and is designed for a 100-pin plastic LQFP (GC-8EU type).
TGC-100SDW Conversion adapter (Refer to **Figure A-2**)	This conversion socket connects the NP-100GC or NP-H100GC-TQ to a target system board designed to mount a 100-pin plastic LQFP (GC-8EU type).
★ NP-113F1-DA3 Emulation probe	This probe is used to connect the in-circuit emulator and target system. For 113-pin plastic FBGA (F1-DA3 type). This includes LSPACK113A1110N01 and CSSOCKET113A1110N01.
★ LSPACK113A1110N01, CSSOCKET113A1110N01 Conversion socket	This conversion socket is used to connect the target system board created for mounting the 113-pin plastic FBGA (F1-DA3 type) and the NP-113F1-DA3.

Remarks 1. NP-100GC, NP-H100GC-TQ, and NP-113F1-DA3 are products of Naito Densei Machida Mfg. Co., Ltd.

Contact: +81-45-475-4191 Naito Densei Machida Mfg. Co., Ltd.

2. TGC-100SDW, LSPACK113A1110N01, and CSSOCKET113A1110N01 are products of TOKYO ELETECH CORPORATION.

Inquiry: Daimaru Kogyo, Ltd. Phone: Tokyo +81-3-3820-7112 Electronics Dept.
Osaka +81-6-6244-6672 Electronics 2nd Dept.

A.6 Debugging Tools (Software)

SM78K0 System simulator	This is a system simulator for the 78K/0 Series. The SM78K0 is Windows-based software. It is used to perform debugging at the C source level or assembler level while simulating the operation of the target system on a host machine. Use of the SM78K0 allows the execution of application logical testing and performance testing on an independent basis from hardware development, thereby providing higher development efficiency and software quality. The SM78K0 should be used in combination with the device file (DF780354) (sold separately).
	Part Number: µS××××SM78K0
ID78K0-NS Integrated debugger (supporting in-circuit emulators IE-78K0-NS and IE-78K0-NS-A)	This debugger supports the in-circuit emulators for the 78K/0 Series. The ID78K0-NS is Windows-based software. It has improved C-compatible debugging functions and can display the results of tracing with the source program using an integrating window function that associates the source program, disassemble display, and memory display with the trace result. It should be used in combination with the device file (sold separately).
	Part Number: µS××××ID78K0-NS

Remark ×××× in the part number differs depending on the host machine and OS used.

µS××××SM78K0
µS××××ID78K0-NS

××××	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)	

A.7 Embedded Software

RX78K0 Real-time OS	RX78K0 is a real-time OS conforming to the μITRON specifications. A tool (configurator) for generating the nucleus of RX78K0 and multiple information tables is supplied. Used in combination with an assembler package (RA78K0) and device file (DF780354) (both sold separately). **<Caution when using RX78K0 in PC environment>** The real-time OS is a DOS-based application. It should be used in the DOS Prompt when using in Windows.
	Part number: μS××××RX78013-ΔΔΔΔ

Caution When purchasing the RX78K0, fill in the purchase application form in advance and sign the user agreement.

Remark ×××× and ΔΔΔΔ in the part number differ depending on the host machine and OS used.

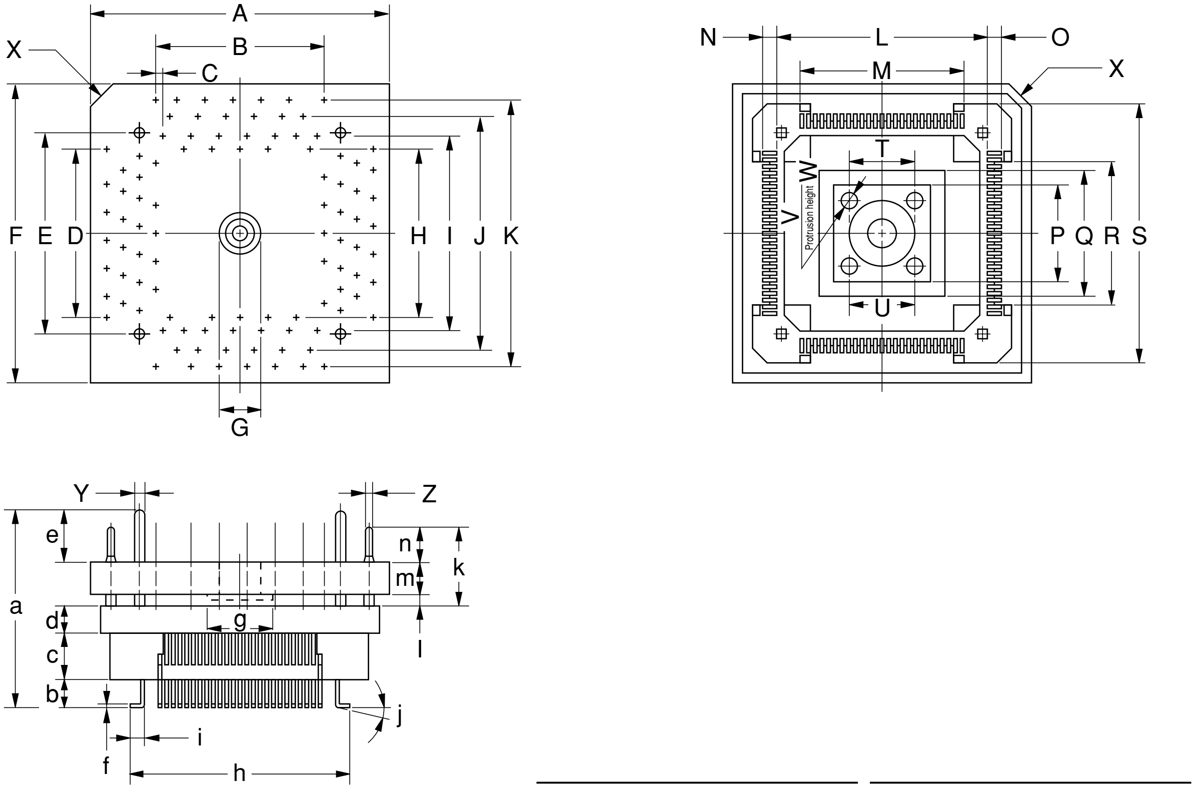
μS××××RX78013-ΔΔΔΔ

ΔΔΔΔ	Product Outline	Maximum Number for Use in Mass Production
001	Evaluation object	Do not use for mass-produced product.
100K	Mass-production object	0.1 million units
001M		1 million units
010M		10 million units
S01	Source program	Source program for mass-produced object

××××	Host Machine	OS	Supply Medium
AA13	PC-9800 series	Windows (Japanese version)	3.5-inch 2HD FD
AB13	IBM PC/AT compatibles	Windows (Japanese version)	
BB13		Windows (English version)	

A.8 Package Drawing for Conversion Adapter (TGC-100SDW)

Figure A-2. TGC-100SDW Package Drawing (for Reference Only)



ITEM	MILLIMETERS	INCHES
A	21.55	0.848
B	0.5x24=12	0.020x0.945=0.472
C	0.5	0.020
D	0.5x24=12	0.020x0.945=0.472
E	15.0	0.591
F	21.55	0.848
G	ϕ3.55	ϕ0.140
H	10.9	0.429
I	13.3	0.524
J	15.7	0.618
K	18.1	0.713
L	13.75	0.541
M	0.5x24=12.0	0.020x0.945=0.472
N	1.125±0.3	0.044±0.012
O	1.125±0.2	0.044±0.008
P	7.5	0.295
Q	10.0	0.394
R	11.3	0.445
S	18.1	0.713
T	ϕ5.0	ϕ0.197
U	5.0	0.197
V	4-ϕ1.3	4-ϕ0.051
W	1.8	0.071
X	C 2.0	C 0.079
Y	ϕ0.9	ϕ0.035
Z	ϕ0.3	ϕ0.012

ITEM	MILLIMETERS	INCHES
a	14.45	0.569
b	1.85±0.25	0.073±0.010
c	3.5	0.138
d	2.0	0.079
e	3.9	0.154
f	0.25	0.010
g	ϕ4.5	ϕ0.177
h	16.0	0.630
i	1.125±0.3	0.044±0.012
j	0~5°	0.000~0.197°
k	5.9	0.232
l	0.8	0.031
m	2.4	0.094
n	2.7	0.106

TGC-100SDW-G1E

note: Product by TOKYO ELETECH CORPORATION.

★

APPENDIX B NOTES ON TARGET SYSTEM DESIGN

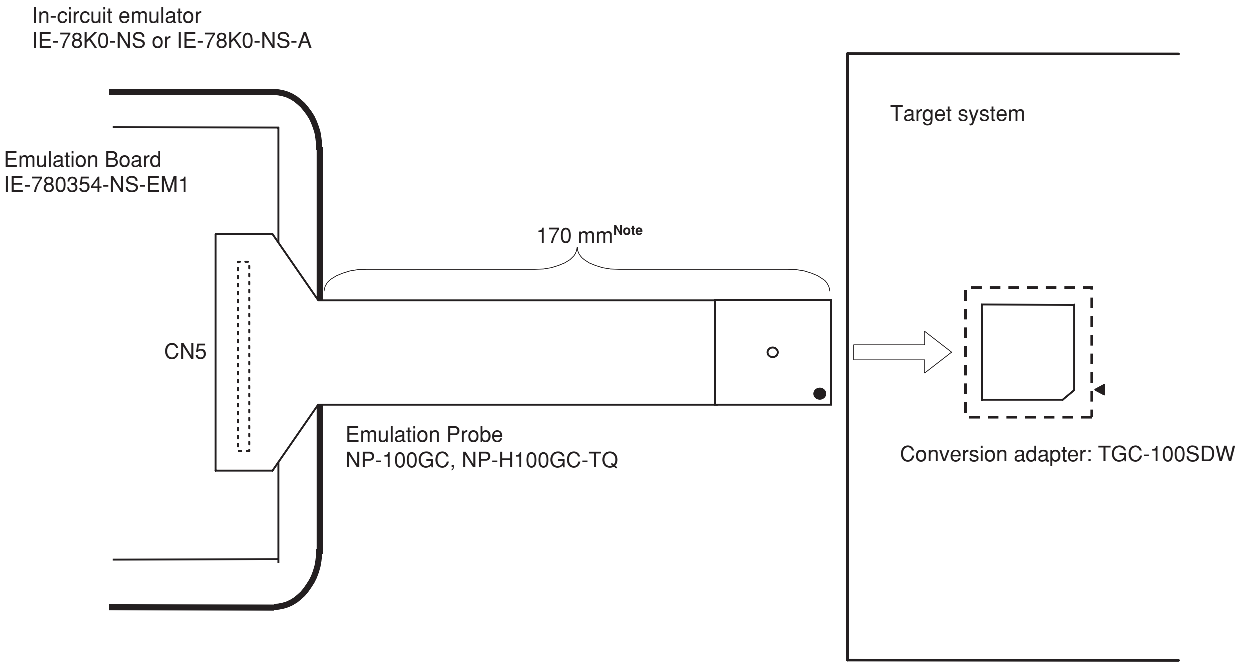
The following shows the conditions when connecting the emulation probe to the conversion adapter. Follow the configuration below and consider the shape of parts to be mounted on the target system when designing a system.

Among the products described in this appendix, NP-100GC and NP-H100GC-TQ are products of Naito Densei Machida Mfg. Co., Ltd., and TGC-100SDW is a product of TOKYO ELETECH CORPORATION.

Table B-1. Distance Between IE System and Conversion Adapter (When 100-Pin Plastic LQFP Is Used)

Emulation Probe	Conversion Adapter	Distance Between IE System and Conversion Adapter
NP-100GC	TGC-100SDW	170 mm
NP-H100GC-TQ		370 mm

Figure B-1. Distance Between IE System and Conversion Adapter (When 100-Pin Plastic LQFP Is Used)



Note Distance when the NP-100GC is used. When the NP-H100GC-TQ is used, the distance is 370 mm.

Figure B-2. Connection Conditions of Target System (When NP-100GC Is Used)

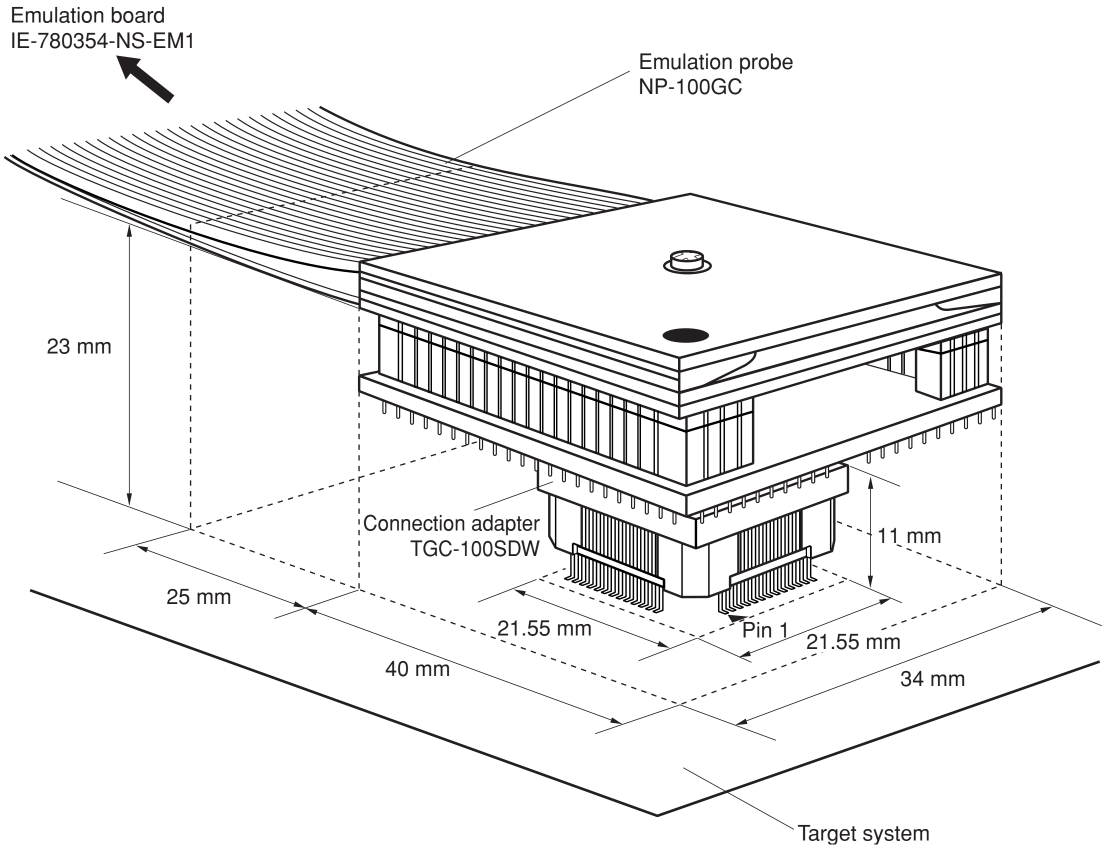
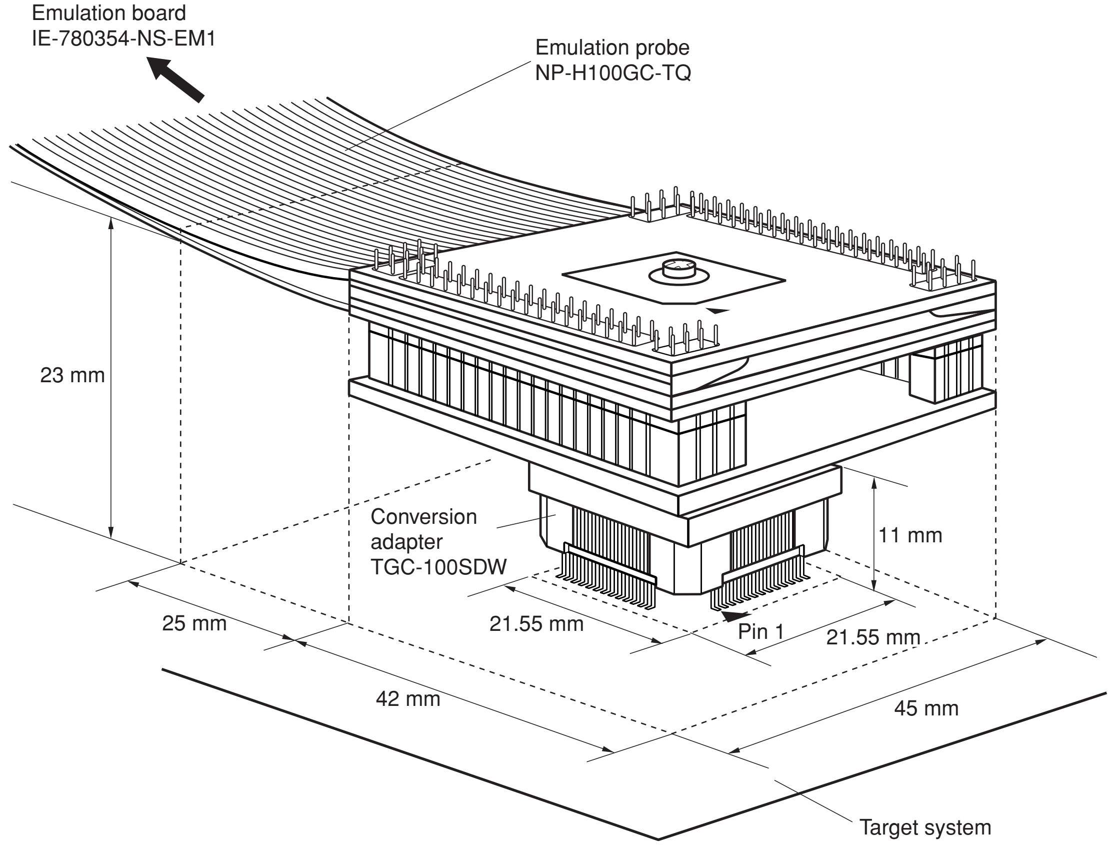


Figure B-3. Connection Conditions of Target System (When NP-H100GC-TQ Is Used)



APPENDIX C REGISTER INDEX

C.1 Register Name Index

C.2 Register Symbol Index

★

APPENDIX D REVISION HISTORY

The following table shows the revision history up to this edition. The “Applied to:” column indicates the chapters of each edition in which the revision was applied.

(1/4)

Edition	Major Revision from Previous Edition	Applied to:
2nd edition	Deletion of indication “under development” for all target products	Throughout
	AV_{REF} pin → AV_{DD} pin	
	A/D converter operation enable voltage AV_{REF} = 2.7 to 5.5 V → AV_{DD} = 2.2 to 5.5 V	
	Change of 113-pin plastic FBGA package in **1.4 Pin Configuration (Top View)**	**CHAPTER 1 OUTLINE**
	Modification of **Table 2-1 Pin I/O Circuit Types**	**CHAPTER 2 PIN FUNCTIONS**
	Addition of description on program area in **3.1.2 (1) Internal high-speed RAM** and **(2) Internal expansion RAM**	**CHAPTER 3 CPU ARCHITECTURE**
	Change of **Figure 3-10 Data To Be Saved to Stack Memory** and **Figure 3-11 Data To Be Restored from Stack Memory**	
	Modification of **[Description example]** in **3.4.4 Short direct addressing**	
	Addition of **[Illustration]** in **3.4.7 Based addressing, 3.4.8 Based indexed addressing**, and **3.4.9 Stack addressing**	
	Modification of description of port 1 and port 4 in **Table 4-1 Port Functions**	**CHAPTER 4 PORT FUNCTIONS**
	Modification of **Figure 4-4 P10 to P17 Block Diagram**	
	Addition of **Caution** in **4.2.3 Port 2**	
	Modification of **Note** in **Format of Figure 4-18 Port Mode Registers (PM0, PM2 to PM4, PM7 to PM11)**	
	Modification of setting and addition of **Caution 2** in **Format of Figure 4-21 Pin Function Switching Registers (PF8 to PF11)**	
	Addition of description in **5.5.1 Main system clock operations**	**CHAPTER 5 CLOCK GENERATOR**
	Modification of **Figure 6-1 Block Diagram of 16-Bit Timer/Event Counter 0**	**CHAPTER 6 16-BIT TIMER/ EVENT COUNTER 0**
	Addition of **Figure 6-11 Configuration of PPG Output** and **Figure 6-12 PPG Output Operation Timing**	
	Modification of **6.6 (4) Capture register data retention timing** and addition of **(13) STOP mode and main system clock stop mode settings**	
	Deletion of <1> in **6.6 (7) Conflicting operations** in 1st edition	
	Modification of Figure **7-6 Format of Carrier Generator Output Control Register B0**	**CHAPTER 7 8-BIT TIMERS A0, B0**
	Addition of input frequency from TMIB0 pin in **Table 7-7 Square-Wave Output Range with 16-Bit Resolution**	
	Addition of description in **8.3 (2) 8-bit timer compare register 5n (CR5n: n = 0, 1)**	**CHAPTER 8 8-BIT TIMER/ EVENT COUNTERS 50, 51**
	Addition of **[Setting]** in **8.5.2 External event counter operation**	
	Addition of description on frequencies in **[Setting]** in **8.5.3 Square-wave output operation**	
	Modification of description of **[Setting]** in **8.5.4 PWM output operation**	

(2/4)

Edition	Major Revision from Previous Edition	Applied to:
2nd edition	Correction of **Figure 9-2 Format of Watch Timer Operation Mode Register 0 (WTNM0)**	**CHAPTER 9 WATCH TIMER**
	Correction of **12.2 (3) Sample & hold circuit** and **(4) Voltage comparator**	**CHAPTER 12 8-BIT A/D CONVERTER (μPD780344, 780344Y SUBSERIES)**
	Modification of description of **Note 3** in **Figure 12-2 Format of A/D Converter Mode Register 0 (ADM0)**, and addition of **Table 12-2 Settings of ADCS0 and ADCE0** and **Figure 12-3 Timing Chart When Boost Reference Voltage Generator Is Used**	
	Modification of **Figure 12-12 Analog Input Pin Connection**	
	Addition of the followings in **12.6 A/D Converter Cautions** **(6) Input impedance of ANI0 to ANI7 pins** **(14) AV_{DD} pin**	
	Change of **Figure 12-16 Example of Connecting Capacitor to V_{DD1} and AV_{REF} Pins** in 1st edition to **Figure 12-16 Example of Connecting Capacitor to AV_{DD} Pin**	
	Modification of **Table 12-3 Resistances and Capacitances of Equivalent Circuit (Reference Values)**	
	Addition and modification of description in **13.2 (2) A/D conversion result register 0 (ADCR0), (3) Sample & hold circuit**, and **(4) Voltage comparator**	**CHAPTER 13 10-BIT A/D CONVERTER (μPD780354, 780354Y SUBSERIES)**
	Modification of description of **Note 3** in **Figure 13-2 Format of A/D Converter Mode Register 0 (ADM0)**, and addition of **Table 13-2 Settings of ADCS0 and ADCE0** and **Figure 13-3 Timing Chart When Boost Reference Voltage Generator Is Used**	
	Modification of **Figure 13-16 Analog Input Pin Connection**	
	Addition of the followings in **13.6 A/D Converter Cautions** **(6) Input impedance of ANI0 to ANI7 pins** **(14) AV_{DD} pin**	
	Change of **Figure 13-20 Example of Connecting Capacitor to V_{DD1} and AV_{REF} Pins** in 1st edition to **Figure 13-20 Example of Connecting Capacitor to AV_{DD} Pin**	
	Modification of **Table 13-3 Resistances and Capacitances of Equivalent Circuit (Reference Values)**	
	Modification of description of MODE flag in **Figure 14-2 Format of Serial Operation Mode Register 3 (CSIM3)**	**CHAPTER 14 SERIAL INTERFACE SIO3**
	Modification of **Caution 1** in **Figure 15-3 Format of Serial Clock Select Register 1 (CSIC1)**	**CHAPTER 15 SERIAL INTERFACE CSI1**
	Deletion of **15.4.2 (6) SCK1 pin** and **(7) SO1 pin** in 1st edition	
	Change of **Caution** in **Figure 16-3 Format of Asynchronous Serial Interface Mode Register 0 (ASIM0)**	**CHAPTER 16 SERIAL INTERFACE UART0**
	Addition of baud rate calculation in **Remarks** in **Figure 16-5 Format of Baud Rate Generator Control Register 0 (BRGC0)**	
	Modification of description in **16.4.2 (2) (d) Reception**	
	Change of **Caution** in **Figure 16-9 Timing of Asynchronous Serial Interface Receive Completion Interrupt Request**	
	Modification of **Caution 2** in **Figure 16-10 Receive Error Timing**	

(3/4)

Edition	Major Revision from Previous Edition	Applied to:
2nd edition	Combination of **17.2 (1) IIC shift register 0 (IIC0), (2) Slave address register 0 (SVA0)**, and **17.3 (5) IIC shift register 0 (IIC0), (6) Slave address register 0 (SVA0)** in 1st edition	**CHAPTER 17 SERIAL INTERFACE IIC0 (μPD780344Y, 780354Y SUBSERIES ONLY)**
	Correction of address value in **Figure 17-3 Format of IIC Control Register 0 (IICC0)**, **Figure 17-4 Format of IIC Status Register 0 (IICS0)**, and **Figure 17-5 Format of IIC Transfer Clock Select Register 0 (IICCL0)**	
	Addition of description on **“Transfer lines”** in **Figure 17-14 Wait Signal**	
	Correction of **17.5.7 (3) (d) (ii) When WTIM0 = 1 (after restart, does not match with address (= not extension code))**	
	Addition of description in **Notes 1** and **2** in **Table 17-2 INTIIC0 Timing and Wait Control**	
	Correction of **Figure 17-21 Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (1) Start condition ~ address** and **(2) Data**	
	Correction of **Figure 17-22 Example of Slave to Master Communication (When 9-Clock Wait Is Selected for Both Master and Slave)**	
	Correction of **Figure 18-1 LCD Controller/Driver Block Diagram**	**CHAPTER 18 LCD CONTROLLER/DRIVER**
	Modification of **Note** in **Table 18-4 Frame Frequency**	
	Modification of description of GAIN bit in **Figure 18-6 Format of LCD Gain Adjust Register 0 (VLCG0)**	
	Modification of **Figure 18-7 Format of Static/Dynamic Display Switching Register 3 (SDSEL3)**	
	Modification of **Figure 18-8 Format of Pin Function Switching Registers (PF8 to PF11)** and addition of **Caution 2**.	
	Replace **18.4 LCD Controller/Driver Settings** and 1**8.5 LCD Display RAM** of 1st edition	
	Deletion of **Table 18-7 LCD Drive Voltages** of 1st edition	
	Standardization of symbols • V_{LC0} pin output voltage: V_{LCD0} • V_{LC1} pin output voltage: V_{LCD1} • V_{LC2} pin output voltage: V_{LCD2}	
	Change of **Table 18-6 Output Voltages of V_{LC0} to V_{LC2} Pins**	
	Addition of description in **18.8.1 Static display example**	
	Change of LCD panel connection examples • **Figure 18-14 Static LCD Panel Connection Example (SDSEL3n = 1: n = 0, 1)** • **Figure 18-17 3-Time-Division LCD Panel Connection Example (SDSEL3n = 0: n = 0 to 2)** • **Figure 18-20 4-Time-Division LCD Panel Connection Example (SDSEL3n = 0, n = 0 to 2)**	

(4/4)

Edition	Major Revision from Previous Edition	Applied to:
2nd edition	Correction of **Figure 19-1 Basic Configuration of Interrupt Function (E) Software interrupt**	**CHAPTER 19 INTERRUPT FUNCTIONS**
	Addition of **Caution** in **Figure 19-5 Format of External Interrupt Rising Edge Enable Register (EGP), External Interrupt Falling Edge Enable Register (EGN)**	
	Addition of description and **Remark** in **19.4.1 Non-maskable interrupt request acknowledgement operation**	
	Addition of description in **19.4.2 Maskable interrupt request acknowledgement operation**	
	Addition of items in **Table 19-4 Interrupt Request Enabled for Nesting During Interrupt Servicing**	
	Addition of **Caution** and **Table 20-3 HALT Mode Release Condition and Necessity of NOP Instruction Setting When Subclock Multiplied by 4 Is Used (μPD78F0354, 78F0354Y Only)** in **20.2.1 (2) HALT mode release**	**CHAPTER 20 STANDBY FUNCTION**
	Addition of description on flash memory in **CHAPTER 22 ROM CORRECTION**	**CHAPTER 22 ROM CORRECTION**
	Correction of **Figure 23-1 Format of Memory Size Switching Register (IMS)**	**CHAPTER 23 μPD78F0354, 78F0354Y**
	Modification of **Table 23-3 Communication Mode List**	
	Change of pin names and signal names in **Figure 23-5 Example of Connection with Dedicated Flash Programmer** and **Table 23-4 Pin Connection List**	
	Correction of flash writing adapter wiring examples • **Figure 23-10 Wiring Example for Flash Writing Adapter with 3-Wire Serial I/O (SIO3)** • **Figure 23-11 Wiring Example for Flash Writing Adapter with 3-Wire Serial I/O (SIO3) with Handshake** • **Figure 23-12 Wiring Example for Flash Writing Adapter with 3-Wire Serial I/O (CSI1)** • **Figure 23-13 Wiring Example for Flash Writing Adapter with UART (UART0)**	
	Revision of **CHAPTER 25 ELECTRICAL SPECIFICATIONS**	**CHAPTER 25 ELECTRICAL SPECIFICATIONS**
	Addition of 113-pin plastic FBGA package in **CHAPTER 27 PACKAGE DRAWINGS**	**CHAPTER 27 PACKAGE DRAWINGS**
	Addition of **CHAPTER 28 RECOMMENDED SOLDERING CONDITIONS**	**CHAPTER 28 RECOMMENDED SOLDERING CONDITIONS**
	Addition of emulation probe NP-113F1-DA3 and conversion sockets LSPACK113A1110N01 and CSSOCKET113A1110N01 in **A.5 Debugging Tools (Hardware)**	**APPENDIX A DEVELOPMENT TOOLS**
	Addition of **APPENDIX B NOTES ON TARGET SYSTEM DESIGN**	**APPENDIX B NOTES ON TARGET SYSTEM DESIGN**
	Addition of **APPENDIX D REVISION HISTORY**	**APPENDIX D REVISION HISTORY**
2nd edition (Modification version)	Addition of μPD78F0354A, 78F0354AY	Throughout
	Modification of **1.3 Ordering Information**	**CHAPTER 1 OUTLINE**
	Modification of CHAPTER	**CHAPTER 28 RECOMMENDED SOLDERING CONDITIONS**