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

μPD789479 Subseries

8-Bit Single-Chip Microcontrollers

μPD789477
μPD789478
μPD789479
μPD78F9478
μPD78F9479

Document No. U15400EJ4V0UD00 (4th edition)
Date Published August 2007 NS

Printed in Japan

[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.

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

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

- **The information in this document is current as of August, 2007. 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

[MEMO]

INTRODUCTION

Target Readers

This manual is intended for user engineers who wish to understand the functions of the μPD789479 Subseries and design and develop application systems and programs using these devices.

Target products:

- μPD789479 Subseries: μPD789477, 789478, 789479, 78F9478, 78F9479

Purpose

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

Organization

Two manuals are available for the μPD789479 Subseries:

This manual and the instruction manual (common to the 78K/0S Series).

μPD789479 Subseries User's Manual	78K/0S Series Instructions User's Manual
• Pin functions • Internal block functions • Interrupts • Other on-chip peripheral functions • Electrical specifications	• CPU function • Instruction set • Instruction description

How to Use This Manual

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

- To understand the overall functions of the μPD789479 Subseries
  → Read this manual in the order of the **CONTENTS**. The mark "<R>" shows major revised points. The revised points can be easily searched by copying an "<R>" in the PDF file and specifying it in the "Find what:" field.
- How to read register formats
  → The name of a bit whose number is enclosed with <> is reserved in the assembler and is defined as an sfr variable by the #pragma sfr directive for the C compiler.
- To learn the detailed functions of a register whose register name is known
  → See **APPENDIX C REGISTER INDEX**.
- To learn the details of the instruction functions of the 78K/0S Series
  → Refer to **78K/0S Series Instructions User's Manual (U11047E)** separately available.
- To learn about the electrical specifications of the μPD789479 Subseries
  → Refer to **CHAPTER 22 ELECTRICAL SPECIFICATIONS**.

Conventions

Data significance:	Higher digits on the left and lower digits on the right
Active low representation:	$\overline{xxx}$ (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 ... xxxx or xxxxB
	Decimal ... xxxx
	Hexadecimal ... xxxxH

Related Documents

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

Documents Related to Devices

Document Name	Document No.
μPD789479 Subseries User's Manual	This manual
78K/0S Series Instructions User's Manual	U11047E

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

Document Name		Document No.
RA78K0S Ver.2.00 Assembler Package	Operation	U17391E
	Language	U17390E
	Structured Assembly Language	U17389E
CC78K0S Ver.2.00 C Compiler	Operation	U17416E
	Language	U17415E
SM78K Series Ver. 2.52 System Simulator	Operation	U16768E
	External Part User Open Interface Specification	U15802E
ID78K0S-NS Ver. 2.52 Integrated Debugger	Operation	U16584E
PM+ Ver.6.00		U17178E

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

Document Name	Document No.
IE-78K0S-NS In-Circuit Emulator	U13549E
IE-78K0S-NS-A In-Circuit Emulator	U15207E
IE-789488-NS-EM1 Emulation Board	U16492E

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” webpage (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 (1/6)

LIST OF FIGURES (2/6)

LIST OF FIGURES (3/6)

LIST OF FIGURES (4/6)

LIST OF FIGURES (5/6)

LIST OF FIGURES (6/6)

LIST OF TABLES (1/2)

LIST OF TABLES (2/2)

CHAPTER 1 GENERAL

1.1 Features

- ROM and RAM capacities

Item / Part Number	Program Memory (ROM)		Data Memory	
			Internal RAM	LCD Display RAM
μPD789477	Mask ROM	24 KB	768 bytes	28 × 4 bits
μPD789478		32 KB	1,024 bytes	
μPD789479		48 KB	1,536 bytes	
μPD78F9478	Flash memory	32 KB	1,024 bytes	
μPD78F9479		48 KB	1,536 bytes	

- Minimum instruction execution time can be selected from high speed (0.4 μs: @5.0 MHz operation with main system clock), low speed (1.6 μs: @5.0 MHz operation with main system clock), and ultra low speed (122 μs: @32.768 kHz operation with subsystem clock)
- A circuit to multiply the subsystem clock by 4 is selectable (15.26 μs: @131 kHz operation: 32.768 kHz subsystem clock × 4)
- I/O ports: 45 (N-ch open-drain: 4)
- Timer: 6 channels
- Serial interface: 2 channels
- 8-bit resolution A/D converter: 8 channels
- LCD controller/driver (external resistance division method)
  Segment signals: 28, common signals: 4
- On-chip multiplier: 8 bits × 8 bits = 16 bits
- On-chip infrared remote control reception function
- On-chip key return signal detector
- Supply voltage: V_{DD} = 1.8 to 5.5 V

1.2 Applications

CD radio-cassette players, portable audio, compact cameras, healthcare equipment, etc.

<R> ## 1.3 Ordering Information

Part Number	Package	Internal ROM
μPD789477GC-×××-8BT	80-pin plastic QFP (14 × 14)	Mask ROM
μPD789477GK-×××-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)	Mask ROM
μPD789478GC-×××-8BT	80-pin plastic QFP (14 × 14)	Mask ROM
μPD789478GK-×××-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)	Mask ROM
μPD789479GC-×××-8BT	80-pin plastic QFP (14 × 14)	Mask ROM
μPD789479GK-×××-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)	Mask ROM
μPD78F9478GC-8BT	80-pin plastic QFP (14 × 14)	Flash memory
μPD78F9478GK-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)	Flash memory
μPD78F9479GC-8BT	80-pin plastic QFP (14 × 14)	Flash memory
μPD78F9479GK-9EU	80-pin plastic TQFP (fine pitch) (12 × 12)	Flash memory
μPD789477GC-×××-8BT-A	80-pin plastic QFP (14 × 14)	Mask ROM
μPD789477GK-×××-9EU-A	80-pin plastic TQFP (fine pitch) (12 × 12)	Mask ROM
μPD789478GC-×××-8BT-A	80-pin plastic QFP (14 × 14)	Mask ROM
μPD789478GK-×××-9EU-A	80-pin plastic TQFP (fine pitch) (12 × 12)	Mask ROM
μPD789479GC-×××-8BT-A	80-pin plastic QFP (14 × 14)	Mask ROM
μPD789479GK-×××-9EU-A	80-pin plastic TQFP (fine pitch) (12 × 12)	Mask ROM
μPD78F9478GC-8BT-A	80-pin plastic QFP (14 × 14)	Flash memory
μPD78F9478GK-9EU-A	80-pin plastic TQFP (fine pitch) (12 × 12)	Flash memory
μPD78F9479GC-8BT-A	80-pin plastic QFP (14 × 14)	Flash memory
μPD78F9479GK-9EU-A	80-pin plastic TQFP (fine pitch) (12 × 12)	Flash memory

Remarks 1. ××× indicates ROM code suffix.

2. Products with -A at the end of the part number are lead-free products.

1.4 Pin Configuration (Top View)

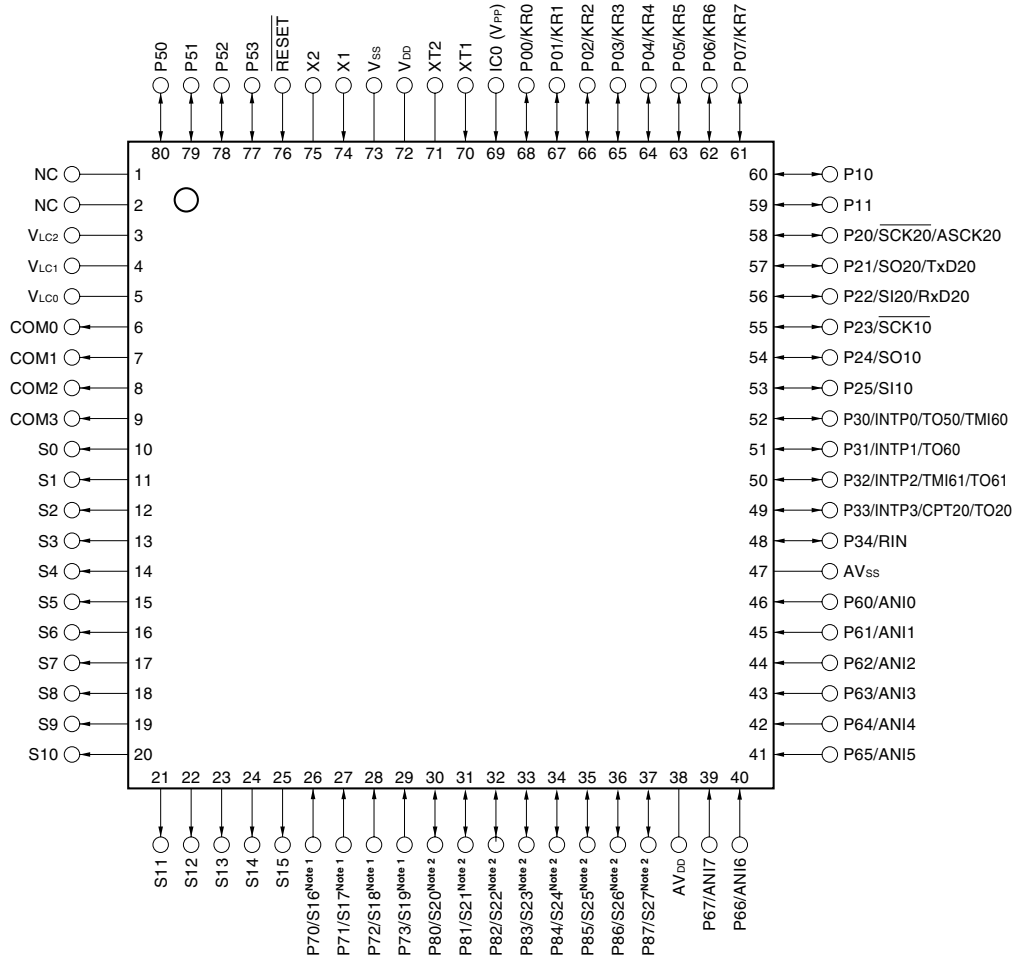
(1) μPD789477, 789478, 78F9478

80-pin plastic QFP (14 × 14)

μPD789477GC-×××-8BT
μPD789478GC-×××-8BT
μPD78F9478GC-8BT
μPD789477GC-×××-8BT-A
μPD789478GC-×××-8BT-A
μPD78F9478GC-8BT-A

80-pin plastic TQFP (fine pitch) (12 × 12)

μPD789477GK-×××-9EU
μPD789478GK-×××-9EU
μPD78F9478GK-9EU
μPD789477GK-×××-9EU-A
μPD789478GK-×××-9EU-A
μPD78F9478GK-9EU-A



Notes 1. Whether to use pins as input port pins (P70 to P73) or segment outputs (S16 to S19) can be selected in 1-bit units by means of a mask option or port function register (refer to **4.3 (3) Port function registers** and **CHAPTER 20 MASK OPTIONS**).

2. Whether to use these pins as I/O port pins (P80 to P87) or segment outputs (S20 to S27) can be selected in 1-bit units by means of a mask option or port function register (refer to **4.3 (3) Port function registers** and **CHAPTER 20 MASK OPTIONS**).

Cautions 1. Connect the IC (Internally Connected) pin directly to VSS.

2. Connect the AVDD pin to VDD.

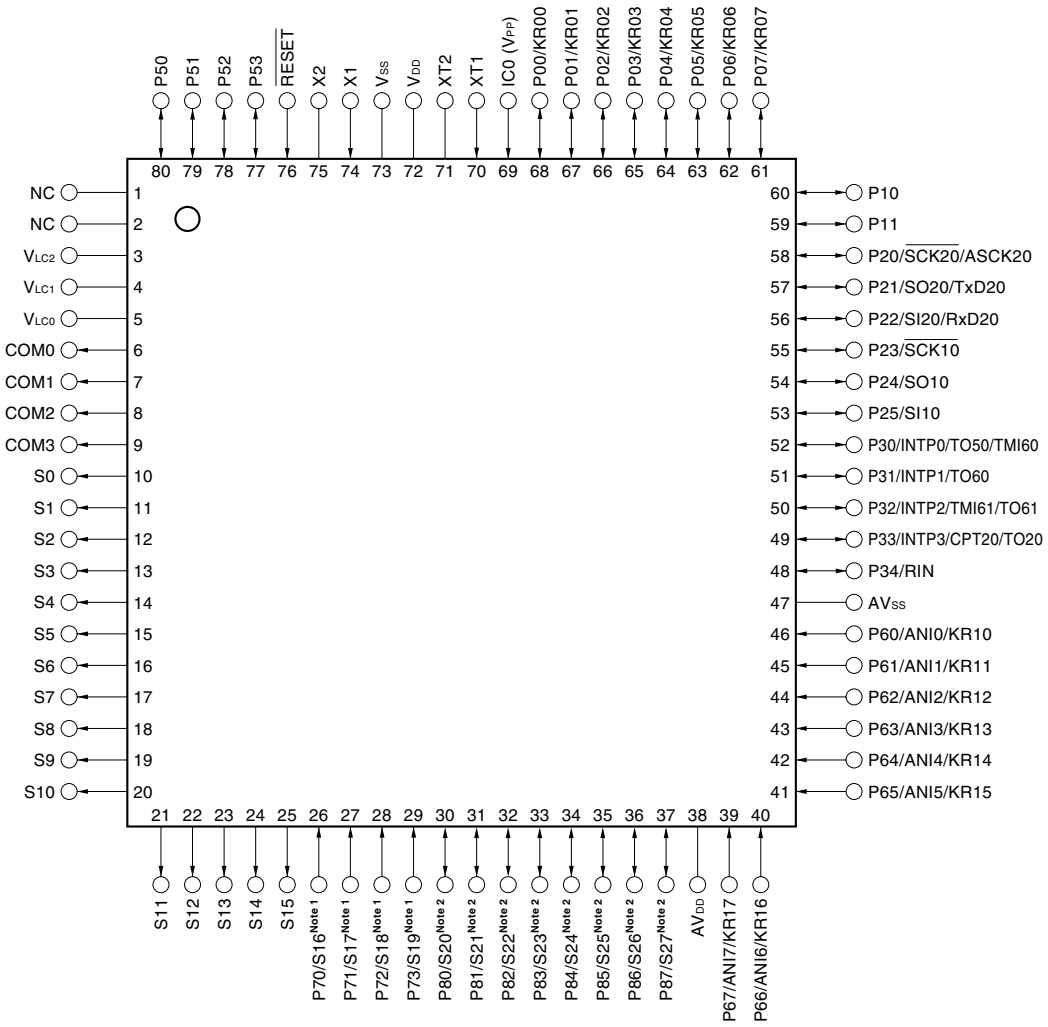
3. Connect the AVSS pin to VSS.

Remark The parenthesized values apply only to the μPD78F9478.

(2) μPD789479, 78F9479

80-pin plastic QFP (14 × 14)
μPD789479GC-×××-8BT
μPD78F9479GC-8BT
μPD789479GC-×××-8BT-A
μPD78F9479GC-8BT-A

80-pin plastic TQFP (fine pitch) (12 × 12)
μPD789479GK-×××-9EU
μPD78F9479GK-9EU
μPD789479GK-×××-9EU-A
μPD78F9479GK-9EU-A



Notes 1. Whether to use pins as input port pins (P70 to P73) or segment outputs (S16 to S19) can be selected in 1-bit units by means of a mask option or port function register (refer to **4.3 (3) Port function registers** and **CHAPTER 20 MASK OPTIONS**).

2. Whether to use these pins as I/O port pins (P80 to P87) or segment outputs (S20 to S27) can be selected in 1-bit units by means of a mask option or port function register (refer to **4.3 (3) Port function registers** and **CHAPTER 20 MASK OPTIONS**).

Cautions 1. Connect the IC (Internally Connected) pin directly to VSS.
2. Connect the AVDD pin to VDD.
3. Connect the AVSS pin to VSS.

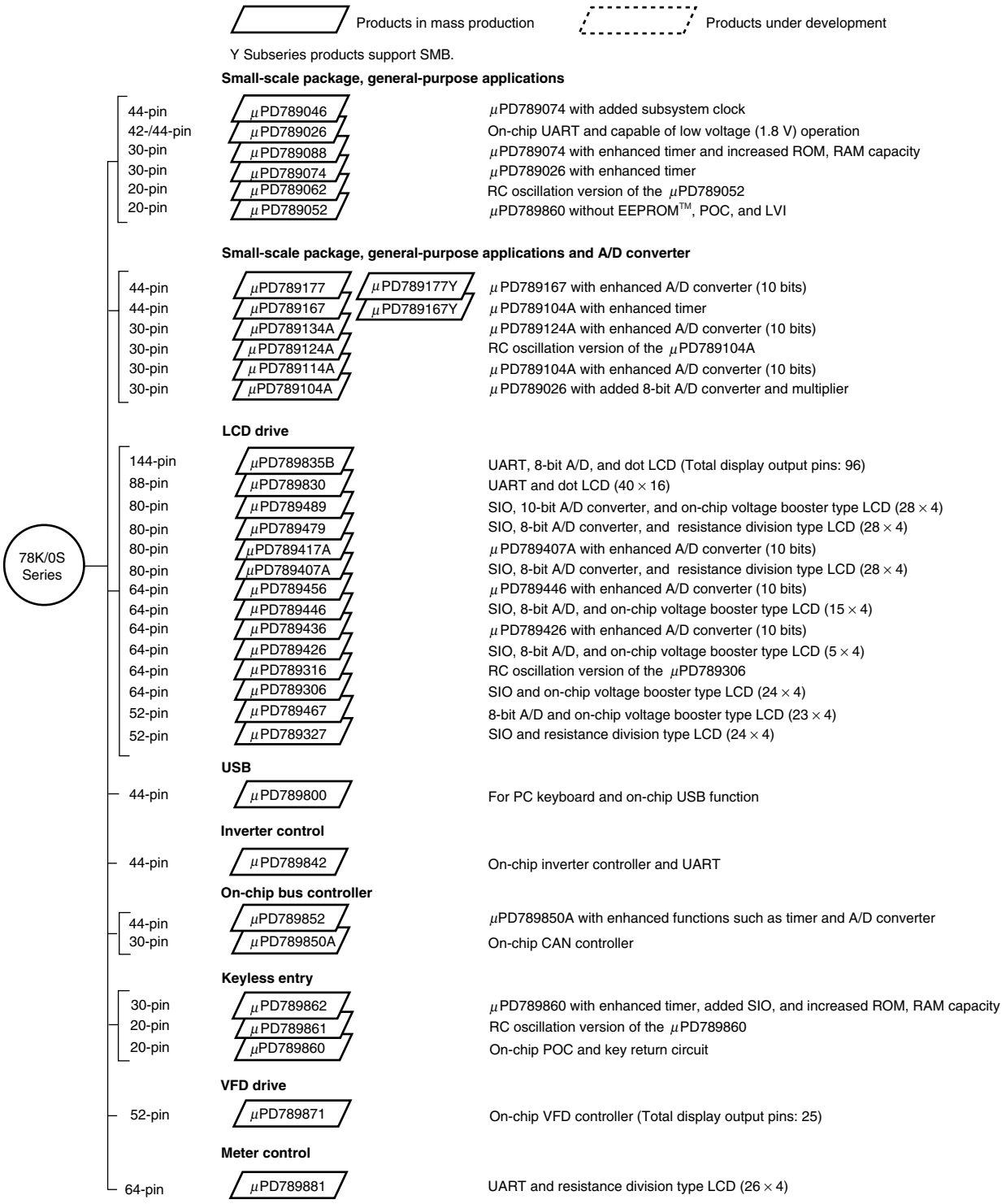
Remark The parenthesized values apply only to the μPD78F9479.

Pin Name

ANI0 to ANI7:	Analog input	$\overline{\text{RESET}}$:	Reset
ASCK20:	Asynchronous serial input	RIN:	Remote control input
AV_{DD}:	Analog power supply	RxD0:	Receive data
AV_{SS}:	Analog ground	S0 to S27:	Segment output
COM0 to COM3:	Common output	$\overline{\text{SCK10}}$:	Serial clock input/output
CPT20:	Capture trigger input	SI10:	Serial data input
IC0:	Internally connected	SO10:	Serial data output
INTP0 to INTP3:	External interrupt input	$\overline{\text{SCK20}}$:	Serial block input/output
KR0 to KR7:	Key return	SI20:	Serial data input
KR00 to KR07:	Key return	SO20:	Serial data output
KR10 to KR17:	Key return	TMI60, 61:	Timer input
NC:	No-connect	TO20, 50, 60, 61:	Timer output
P00 to P07:	Port 0	TxD0:	Transmit Data
P10, P11:	Port 1	V_{DD}:	Power supply
P20 to P25:	Port 2	V_{LC0} to V_{LC2}:	Power supply for LCD
P30 to P34:	Port 3	V_{PP}:	Programming power supply
P60 to P67:	Port 6	V_{SS}:	Ground
P70 to P73:	Port 7	X1, X2:	Crystal (Main system clock)
P80 to P87:	Port 8	XT1, XT2:	Crystal (Subsystem clock)

<R> 1.5 78K/0S Series Lineup

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



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

The major differences between the subseries are shown below.

Series for General-Purpose Applications and LCD Drive

Subseries \ Function		ROM Capacity (Bytes)	Timer				8-Bit A/D	10-Bit A/D	Serial Interface	I/O	V_{DD} MIN.Value	Remarks
			8-Bit	16-Bit	Watch	WDT						
Small-scale package, general-purpose applications	μPD789046	16 K	1 ch	1 ch	1 ch	1 ch	–	–	1 ch (UART: 1ch)	34	1.8 V	–
	μPD789026	4 K to 16 K			–							
	μPD789088	16 K to 32 K	3 ch							24		
	μPD789074	2 K to 8 K	1 ch									
	μPD789062	4 K	2 ch	–					–	14		RC-oscillation version
	μPD789052											–
Small-scale package, general-purpose applications + A/D converter	μPD789177	16 K to 24 K	3 ch	1 ch	1 ch	1ch	–	8 ch	1 ch (UART: 1ch)	31	1.8 V	–
	μPD789167						8 ch	–				
	μPD789134A	2 K to 8 K	1 ch		–		–	4 ch		20		RC-oscillation version
	μPD789124A						4 ch	–				
	μPD789114A						–	4 ch				–
	μPD789104A						4 ch	–				
LCD drive	μPD789835B	24 K to 60 K	6 ch	–	1 ch	1 ch	3 ch	–	1 ch (UART: 1ch)	37	1.8 V[Note]	Dot LCD supported
	μPD789830	24 K	1 ch	1 ch			–			30	2.7 V	
	μPD789489	32 K to 48 K	3 ch					8 ch	2 ch (UART: 1ch)	45	1.8 V	–
	μPD789479	24 K to 48 K					8 ch	–				
	μPD789417A	12 K to 24 K					–	7 ch	1 ch (UART: 1ch)	43		
	μPD789407A						7 ch	–				
	μPD789456	12 K to 16 K	2 ch				–	6 ch		30		
	μPD789446						6 ch	–				
	μPD789436						–	6 ch		40		
	μPD789426						6 ch	–				
	μPD789316	8 K to 16 K					–		2 ch (UART: 1ch)	23		RC-oscillation version
	μPD789306											–
	μPD789467	4 K to 24 K		–			1 ch		–	18		
	μPD789327						–		1 ch	21		

Note Flash memory version: 3.0 V

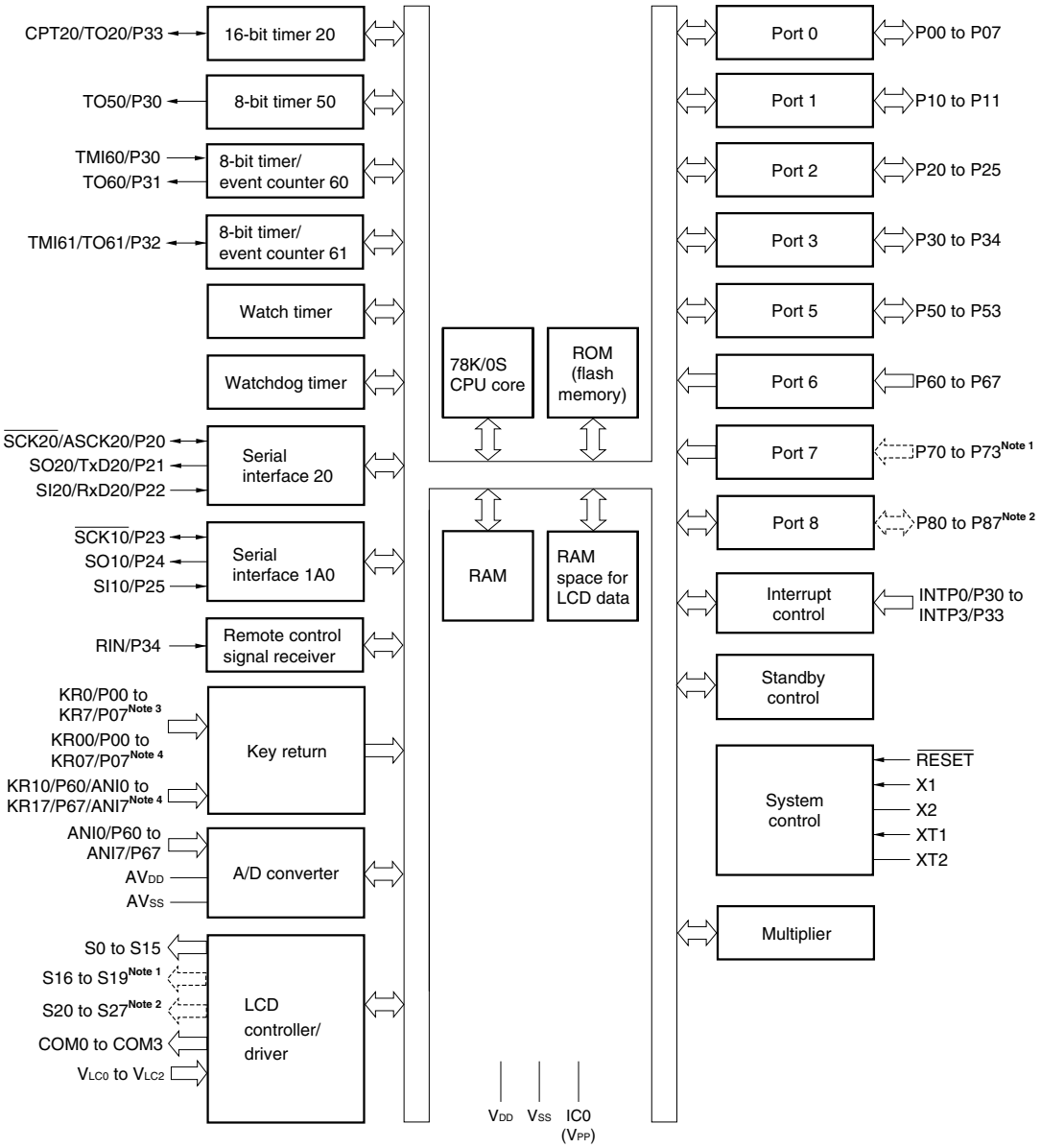
Series for ASSP

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

Notes 1. 10-bit timer: 1 channel

2. Flash memory version: 3.0 V

1.6 Block Diagram



Notes 1. Whether to use these pins as input port pins (P70 to P73) or segment outputs (S16 to S19) can be selected in 1-bit units by means of a mask option in the μPD789477, 789478, and 789479 or a port mode register in the μPD78F9478 and 78F9479 (refer to **4.3 (3) Port function registers** and **CHAPTER 20 MASK OPTIONS**).

2. Whether to use these pins as I/O port pins (P80 to P87) or segment outputs (S20 to S27) can be selected in 1-bit units by means of a mask option in the μPD789477, 789478, and 789479 or a port mode register in the μPD78F9478 and 78F9479 (refer to **4.3 (3) Port function registers** and **CHAPTER 20 MASK OPTIONS**).

3. μPD789477, 789478, and 78F9478 only.

4. μPD789479 and 78F9479 only.

Remark The parenthesized values apply only to the μPD78F9478 and 78F9479.

1.7 Overview of Functions

(1/2)

Item		μPD789477	μPD789478	μPD78F9478	μPD789479	μPD78F9479
Internal memory	ROM	24 KB	32 KB	32 KB (flash memory)	48 KB	48 KB (flash memory)
	High-speed RAM	768 bytes	1024 bytes		1536 bytes	
	Low-speed RAM	–			512 bytes	
	LCD display RAM	28 bytes				
Main system clock (oscillation frequency)		Ceramic/crystal oscillation (1.0 to 5.0 MHz)				
Subsystem clock (oscillation frequency)		Crystal oscillation (32.768 kHz)				
Minimum instruction execution time		0.4 μs/1.6 μs (@5.0 MHz operation with main system clock)				
		122 μs (@32.768 kHz operation with subsystem clock)				
		15.26 μs (@131 kHz operation with ×4 subsystem clock)				
Subsystem clock multiplication function		×4 multiplication circuit (operating supply voltage: V_{DD} = 2.7 to 5.5 V)[Note 1]				
General-purpose registers		8 bits × 8 registers				
Instruction set		• 16-bit operations • Bit manipulation (set, reset, test) etc.				
Multiplier		8 bits × 8 bits = 16 bits				
I/O ports		Total: 45[Note 2] CMOS I/O: 29 CMOS input: 12 N-ch open-drain I/O: 4				
Timers		• 16-bit timer: 1 channel • 8-bit timer: 3 channels • Watch timer: 1 channel • Watchdog timer: 1 channel				
Timer outputs		4				
Serial interface		UART/3-wire serial I/O mode: 1 channel 3-wire serial I/O mode (with automatic transfer function): 1 channel				
A/D converter		8-bit resolution × 8 channels				
LCD controller/driver		• Segment signal outputs: 28[Note 2] • Common signal outputs: 4				
Power supply method for LCD drive		External resistance division method				
Infrared remote control reception function		On-chip				
Key return signal detection function		8 pins			16 pins	
Vectored interrupt sources	Maskable	Internal: 16, External: 5			Internal: 16, External: 6	
	Non-maskable	Internal: 1				

Notes 1. Whether a circuit to multiply the clock by 4 is used or not is selected by a mask option or the subclock selection register.

2. 12 pins are used either as a port function or LCD segment output selected by a mask option or port function register.

(2/2)

Item	μPD789477	μPD789478	μPD78F9478	μPD789479	μPD78F9479
Reset	• Reset by $\overline{\text{RESET}}$ signal input • Internal reset by watchdog timer				
Supply voltage	V_{DD} = 1.8 to 5.5 V				
Operating ambient temperature	T_A = –40 to +85°C				
Package	• 80-pin plastic QFP (14 × 14) • 80-pin plastic TQFP (fine pitch) (12 × 12)				

An outline of the timer is shown below.

		16-Bit Timer 20	8-Bit Timer 50	8-Bit Timer 60	8-Bit Timer 61	Watch Timer	Watchdog Timer
Operation mode	Interval timer	–	1 channel	1 channel	1 channel	1 channel[Note 1]	1 channel[Note 2]
	External event counter	–	–	1 channel	1 channel	–	–
Function	Timer outputs	1 output	1 output	1 output	1 output	–	–
	Square-wave outputs	–	1 output	1 output	1 output	–	–
	Capture	1 input	–	–	–	–	–
	Interrupt sources	1	1	1	1	2	2

Notes 1. The watch timer can perform both watch timer and interval timer functions at the same time.

2. The watchdog timer has watchdog timer and interval timer functions. However, use the watchdog timer by selecting either the watchdog timer function or interval timer function.

2.1 List of Pin Functions

(1) Port pins (1/2)

Pin Name	I/O	Function	After Reset	Alternate Function
P00 to P07	I/O	Port 0. 8-bit I/O port. Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be specified in 1-bit units by means of pull-up resistor option register B0 (PUB0) or the key return mode register (KRM00).	Input	KR0 to KR7[Note 1] KR00 to KR07[Note 2]
P10, P11	I/O	Port 1. 2-bit I/O port. Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be specified in 1-bit units by means of pull-up resistor option register B1 (PUB1).	Input	–
P20	I/O	Port 2. 6-bit I/O port. Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be specified in 1-bit units by means of pull-up resistor option register B2 (PUB2).	Input	$\overline{\text{SCK20}}$/ASCK20
P21				SO20/TxD20
P22				SI20/RxD20
P23				$\overline{\text{SCK10}}$
P24				SO10
P25				SI10
P30	I/O	Port 3. 5-bit I/O port. Input/output can be specified in 1-bit units. When used as an input port, on-chip pull-up resistors can be specified in 1-bit units by means of pull-up resistor option register B3 (PUB3).	Input	INTP0/TO50/TMI60
P31				INTP1/TO60
P32				INTP2/TMI61/TO61
P33				INTP3/CPT20/TO20
P34				RIN
P50 to P53	I/O	Port 5. 4-bit N-ch open-drain I/O port. Input/output can be specified in 1-bit units. For mask ROM version, an on-chip pull-up resistor can be specified by means of mask option.	Input	–
P60 to P67	Input	Port 6. 8-bit input port.	Input	ANI0 to ANI7[Note 1] ANI0/KR10- ANI7/KR17[Note 2]

Notes 1. μPD789477, 789478, and 78F9478 only

2. μPD789479 and 78F9479 only

(1) Port pins (2/2)

Pin Name	I/O	Function	After Reset	Alternate Function
P70 to P73[Note 1]	Input	Port 7. 4-bit input port. (Only when input port is selected by mask option or port function register)	Input	–
P80 to P87[Note 2]	I/O	Port 8. 8-bit I/O port. (Only when I/O port is selected by mask option or port function register)	Input	–

Notes 1. Whether to use these pins as input port pins (P70 to P73) or segment outputs (S16 to S19) can be selected in 1-bit units by means of a mask option in the μPD789477, 789478, and 789479 or a port mode register in the μPD78F9478 and 78F9479 (refer to **4.3 (3) Port function registers** and **CHAPTER 20 MASK OPTIONS**).

2. Whether to use these pins as I/O port pins (P80 to P87) or segment outputs (S20 to S27) can be selected in 1-bit units by means of a mask option in the μPD789477 789478, and 789479 or a port mode register in the μPD78F9478 and 78F9479 (refer to **4.3 (3) Port function registers** and **CHAPTER 20 MASK OPTIONS**).

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

Pin Name	I/O	Function	After Reset	Alternate Function
INTP0	Input	External interrupt input for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.	Input	P30/TO50/TMI60
INTP1				P31/TO60
INTP2				P32/TMI61/TO61
INTP3				P33/CPT20/TO20
KR0 to KR7[Note 1]	Input	Key return signal detection	Input	P00 to P07
KR00 to KR07[Note 2]	Input	Key return signal detection	Input	P00 to P07
KR10 to KR17[Note 2]				P60/ANI0 to P67/ANI7
TO20	Output	16-bit timer 20 output	Input	P33/INTP3/CPT20
CPT20	Output	Capture edge input of 16-bit timer 20	Input	P33/INTP3/TO20
TO50	Output	8-bit timer 50 output	Input	P30/INTP0/TMI60
TO60	Output	8-bit timer 60 output	Input	P31/INTP1
TO61	Output	8-bit timer 61 output	Input	P32/INTP2/TMI61
TMI60	Input	External count clock input to 8-bit timer 60	Input	P30/INTP0/TO50
TMI61	Input	External count clock input to 8-bit timer 61	Input	P32/INTP2/TO61
$\overline{\text{SCK20}}$	I/O	Serial clock input/output of serial interface	Input	P20/ASCK20
$\overline{\text{SCK10}}$				P23
SO20	Output	Serial data output of serial interface	Input	P21/TxD20
SO10				P24
SI20	Input	Serial data input of serial interface	Input	P22/RxD20
SI10				P25
ASCK20	Input	Serial clock input of asynchronous serial interface	Input	P20/$\overline{\text{SCK20}}$

Notes 1. μPD789477, 789478, and 78F9478 only

2. μPD789479 and 78F9479 only

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

Pin Name	I/O	Function	After Reset	Alternate Function
TxD20	Output	Serial data output of asynchronous serial interface	Input	P21/SO20
RxD20	Input	Serial data input of asynchronous serial interface	Input	P22/SI20
RIN	Input	Remote control receive data input	Input	P34
S0 to S15	Output	LCD controller/driver segment signal outputs	Low-level output	–
S16 to S19[Note 1]		Only when segment output is selected by mask option		–
S20 to S27[Note 2]		Only when segment output is selected by mask option		–
COM0 to COM3	Output	LCD controller/driver common signal outputs	Low-level output	–
V_{LC0} to V_{LC2}	–	LCD drive voltage	–	–
ANI0 to ANI7	–	A/D converter analog input	–	P60 to P67[Note 3] P60/KR10 to P67/KR17[Note 4]
AV_{SS}	–	A/D converter ground potential	–	–
AV_{DD}	–	A/D converter analog power supply	–	–
X1	Input	Connecting crystal resonator for main system clock oscillation	–	–
X2	–		–	–
XT1	Input	Connecting crystal resonator for subsystem clock oscillation	–	–
XT2	–		–	–
$\overline{\text{RESET}}$	Input	System reset input	Input	–
V_{DD}	–	Positive power supply	–	–
V_{SS}	–	Ground potential	–	–
IC0	–	Internally connected. Connect directly to V_{SS}.	–	–
NC	–	Not internally connected. Leave open.	–	–
V_{PP}	–	Sets flash memory programming mode. Used to apply high voltage when a program is written or verified.	–	–

Notes 1. Whether to use these pins as input port pins (P70 to P73) or segment outputs (S16 to S19) can be selected in 1-bit units by means of a mask option in the μPD789477, 789478, and 789479 or a port mode register in the μPD78F9478 and 78F9479 (refer to **4.3 (3) Port function registers** and **CHAPTER 20 MASK OPTIONS**).

2. Whether to use these pins as I/O port pins (P80 to P87) or segment outputs (S20 to S27) can be selected in 1-bit units by means of a mask option in the μPD789477 789478, and 789479 or a port mode register in the μPD78F9478 and 78F9479 (refer to **4.3 (3) Port function registers** and **CHAPTER 20 MASK OPTIONS**).

3. μPD789477, 789478, and 78F9478 only

4. μPD789479 and 78F9479 only

2.2 Description of Pin Functions

2.2.1 P00 to P07 (Port 0)

These pins constitute an 8-bit I/O port. In addition, these pins enable key return signal detection.
Port 0 can be specified in the following operation modes in 1-bit units.

(1) Port mode

These pins constitute an 8-bit I/O port and can be set in the input or output port mode in 1-bit units by port mode register 0 (PM0). When used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register B0 (PUB0) in 1-bit units.

(2) Control mode

In this mode, P00 to P07 function as key return signal detection pins (KR0 to KR7 (μPD789477, 789478, 78F9478), KR00 to KR07 (μPD789479, 78F9479)).

2.2.2 P10, P11 (Port 1)

These pins constitute a 2-bit I/O port and can be set in the input or output port mode in 1-bit units by port mode register 1 (PM1). When used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register B1 (PUB1) in 1-bit units.

2.2.3 P20 to P25 (Port 2)

These pins constitute a 6-bit I/O port. In addition, these pins enable serial interface data I/O and serial clock I/O.
Port 2 can be specified in the following operation modes in 1-bit units.

(1) Port mode

In this mode, P20 to P25 function as a 6-bit I/O port. Port 2 can be set in the input or output port mode in 1-bit units by port mode register 2 (PM2). When used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register B2 (PUB2) in 1-bit units.

(2) Control mode

In this mode, P20 to P25 function as the serial interface data I/O and serial clock I/O.

(a) SI20, SO20, SI10, SO10

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

(b) $\overline{\text{SCK20}}$, $\overline{\text{SCK10}}$

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

(c) RxD20, TxD20

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

(d) ASCK20

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

Caution When using P20 to P25 as serial interface pins, the I/O mode and output latch must be set according to the functions to be used. For the details of the setting, refer to Table 11-2 Serial Interface 20 Operation Mode Setting and 12.3 (1) Serial operation mode register 1A0 (CSIM1A0).

2.2.4 P30 to P34 (Port 3)

These pins constitute a 5-bit I/O port. In addition, they also function as timer I/O, external interrupt inputs, and input of remote control receive data.

Port 3 can be specified in the following operation modes in 1-bit units.

(1) Port mode

In this mode, P30 to P34 function as a 5-bit I/O port. Port 3 can be set in the input or output port mode in 1-bit units by port mode register 3 (PM3). When used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register B3 (PUB3) in 1-bit units.

(2) Control mode

In this mode, P30 to P34 function as timer I/O, external interrupt inputs, and input of remote control receive data.

(a) TMI60, TMI61

These are the external clock input pins of timers 60 and 61.

(b) TO20, TO50, TO60, TO61

These are the timer output pins of timers 20, 50, 60, and 61.

(c) CPT20

This is the capture edge input pin of 16-bit timer 20.

(d) INTP0 to INTP3

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

(e) RIN

This is the data input pin of the remote control receiver.

2.2.5 P50 to P53 (Port 5)

These pins constitute a 4-bit N-ch open-drain I/O port. Port 5 can be set in the input or output port mode in 1-bit units by port mode register 5 (PM5). In the mask ROM version, use of an on-chip pull-up resistor can be specified by a mask option in 1-bit units.

2.2.6 P60 to P67 (Port 6)

This is an 8-bit input-only port. In addition to a general-purpose input port function, it has an A/D converter input function and key return signal detection function[Note].

(1) Port mode

In this mode, P60 to P67 function as an 8-bit input-only port.

(2) Control mode

In this mode, P60 to P67 function as the analog inputs of the A/D converter and key return signal detection pins[Note].

(a) ANI0 to ANI7

These are the analog input pins of the A/D converter.

(b) KR10 to KR17[Note]

These are the key return signal detection pins.

Note μPD789479 and 78F9479 only

2.2.7 P70 to P73 (Port 7)

These pins constitute a 4-bit input-only port. This port can be used only when the port function is selected by a mask option in the μPD789477, 789478, and 789479 or by a port function register in the μPD78F9478 and 78F9479.

2.2.8 P80 to P87 (Port 8)

These pins constitute an 8-bit I/O port. Port 8 can be set in the input or output mode in 1-bit units by port mode register 8 (PM8). This port can be used only when the port function is selected by a mask option in the μPD789477, 789478 and 789479 or by a port function register in the μPD78F9478 and 78F9479.

2.2.9 S0 to S27[Note]

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

Note Pins S16 through S27 can be used only when segment output is selected by a mask option in the μPD789477, 789478, and 789479 or by a port function register in the μPD78F9478 and 78F9479.

2.2.10 COM0 to COM3

These pins are the common signal output pins for the LCD controller/driver.

2.2.11 V_{LC0} to V_{LC2}

These pins are power supply voltage pins for driving the LCD.

2.2.12 NC

The NC (No-Connect) pin is not internally connected. Connect this pin to V_{SS}. (It can also be left open.)

2.2.13 $\overline{\text{RESET}}$

This pin inputs an active-low system reset signal.

2.2.14 X1, X2

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

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

2.2.15 XT1, XT2

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

To supply an external clock, input the clock to XT1 and input the inverted signal to XT2.

2.2.16 AVDD

This is the analog power supply pin of the A/D converter. Always use the same potential as that of the V_{DD} pin even when the A/D converter is not used.

2.2.17 AVSS

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

2.2.18 VDD

This is the positive power supply pin.

2.2.19 VSS

This is the ground pin.

2.2.20 VPP (flash memory version only)

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

Handle the pins in either of the following ways.

- Independently connect a 10 kΩ pull-down resistor.
- Switch this pin to be directly connected to the dedicated flash programmer in programming mode or to V_{SS} in normal operation mode using a jumper on the board.

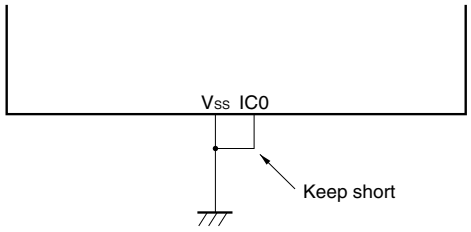
If there is a potential difference between the V_{PP} pin and V_{SS} pin due to a long wiring length or external noise superimposed on the V_{PP} pin, the user program may not run correctly.

2.2.21 IC0 (mask ROM version only)

The IC0 (Internally Connected) pin is used to set the μPD789479 Subseries in the test mode before shipment. In the normal operation mode, directly connect this pin to the V_{SS} pin with as short a wiring length as possible.

If there is a potential difference between the IC0 pin and V_{SS} pin due to a long wiring length or external noise superimposed on the IC0 pin, the user program may not run correctly.

- Directly connect the IC0 pin to the V_{SS} pin.



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

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

Table 2-1. Types of Pin I/O Circuits (1/2)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00/KR0 to P07/KR7 Note 1	8-A	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P00/KR00 to P07/KR07 Note 2			
P10, P11	5-A		
P20/$\overline{SCK20}$/ASCK20	8-A		
P21/SO20/TxD20	5-A		
P22/SI20/RxD20	8-A		
P23/$\overline{SCK10}$			
P24/SO10	5-A		
P25/SI10	8-A		
P30/INTP0/TO50/TMI60			Input: Independently connect to V_{SS} via a resistor. Output: Leave open.
P31/INTP1/TO60			
P32/INTP2/TO61/TMI61			
P33/INTP3/CPT20/TO20			
P34/RIN			
P50 to P53 (mask ROM version)	13-W		Input: Independently connect to V_{DD} via a resistor. Output: Leave open.
P50 to P53 (flash memory version)	13-V		
P60/ANI0 to P67/ANI7 Note 1	9-C	Input	Connect directly to V_{DD} or V_{SS}.
P60/ANI10/KR10 to P67/ANI17/KR17 Note 2			
P70 to P73 Note 3	2-H		
P80 to P87 Note 3	5-K	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
COM0 to COM3	18	Output	Leave open.
S0 to S15	17		
S16 to S19 Note 4			
S20 to S27 Note 4			
NC	–	–	
V_{LC0} to V_{LC2}			
AV_{DD}			Connect directly to V_{DD}.
AV_{SS}			Connect directly to V_{SS}.

Notes 1. μPD789477, 789478, and 78F9478 only

2. μPD789479 and 78F9479 only

3. Only when port pin is selected by mask option or port function register.

4. Only when segment output pin is selected by mask option or port function register.

Table 2-1. Types of Pin I/O Circuits (2/2)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
XT1	–	Input	Connect directly to V_{SS}.
XT2		–	Leave open.
$\overline{\text{RESET}}$	2	Input	–
IC0	–	–	Connect directly to V_{SS}.
V_{PP}			Independently connect a 10 kΩ pull-down resistor, or connect directly to V_{SS}.

Figure 2-1. I/O Circuit Types (1/2)

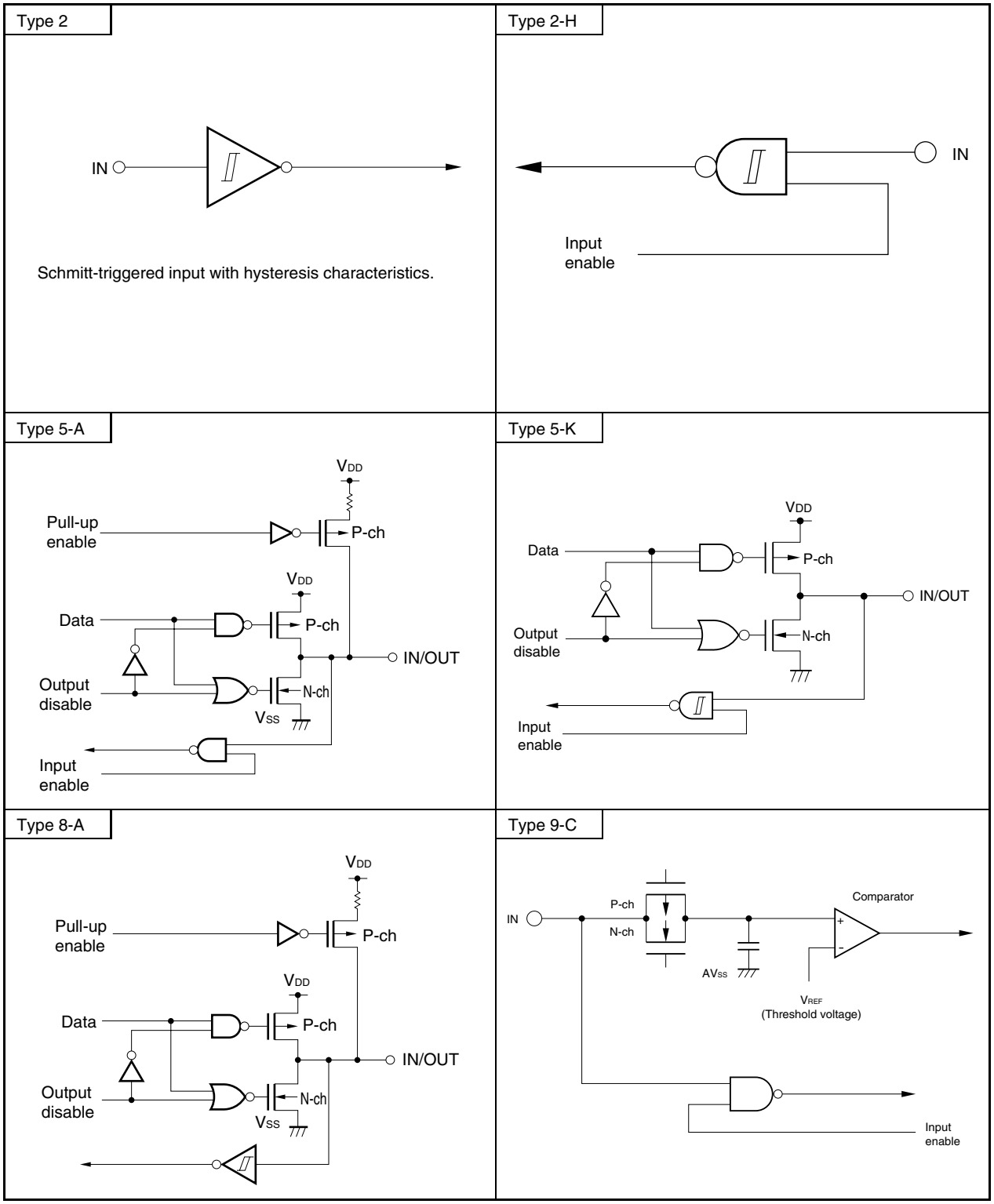
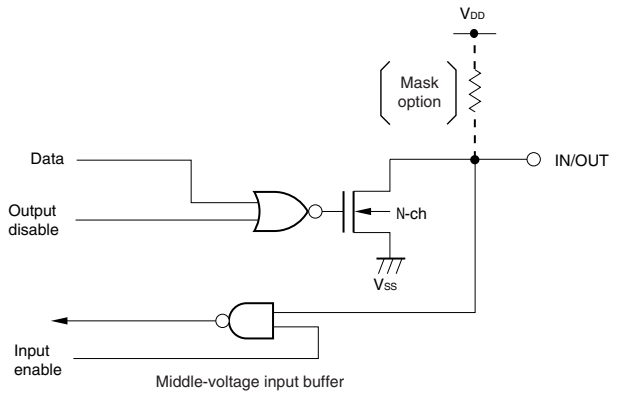
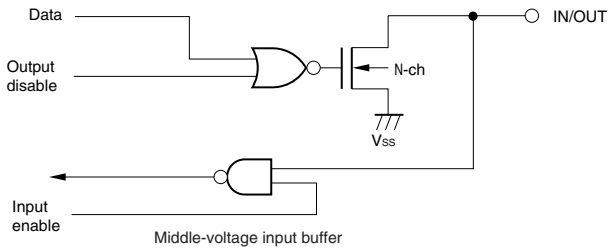


Figure 2-1. I/O Circuit Types (2/2)

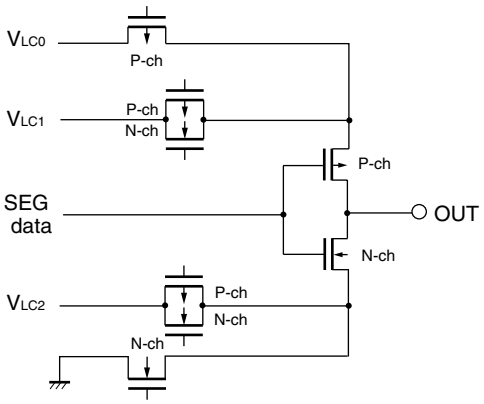
Type 13-W



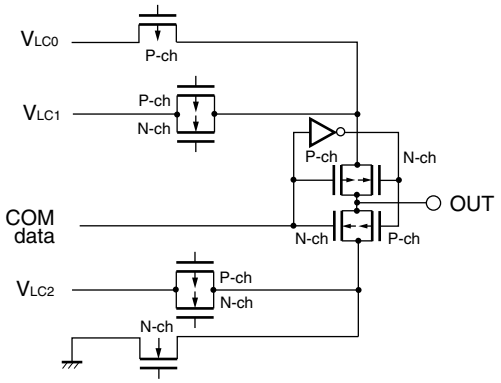
Type 13-V



Type 17



Type 18



CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

The μPD789479 Subseries can access 64 KB of memory space. Figures 3-1 to 3-5 show the memory maps.

Figure 3-1. Memory Map (μPD789477)

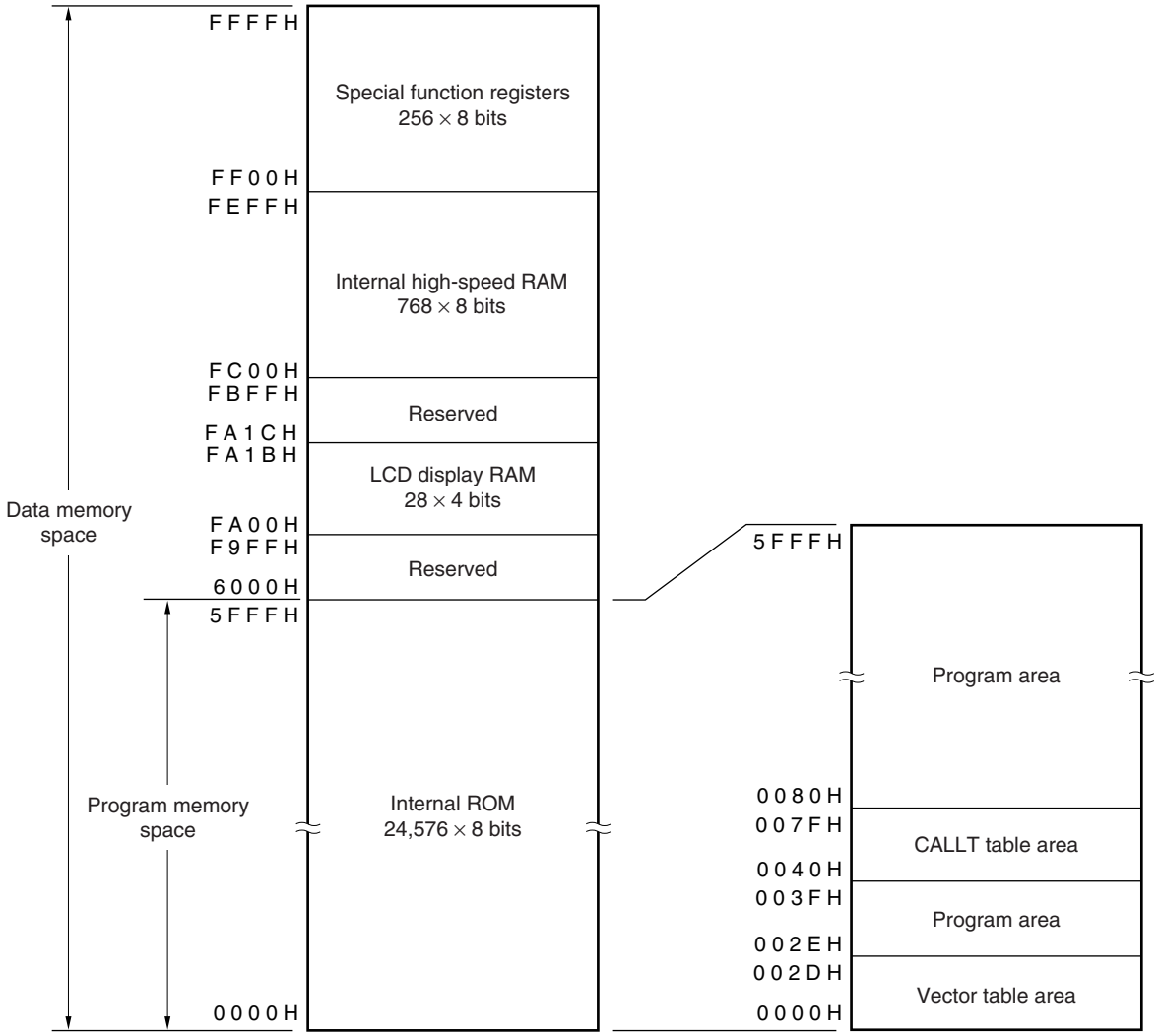


Figure 3-2. Memory Map (μPD789478)

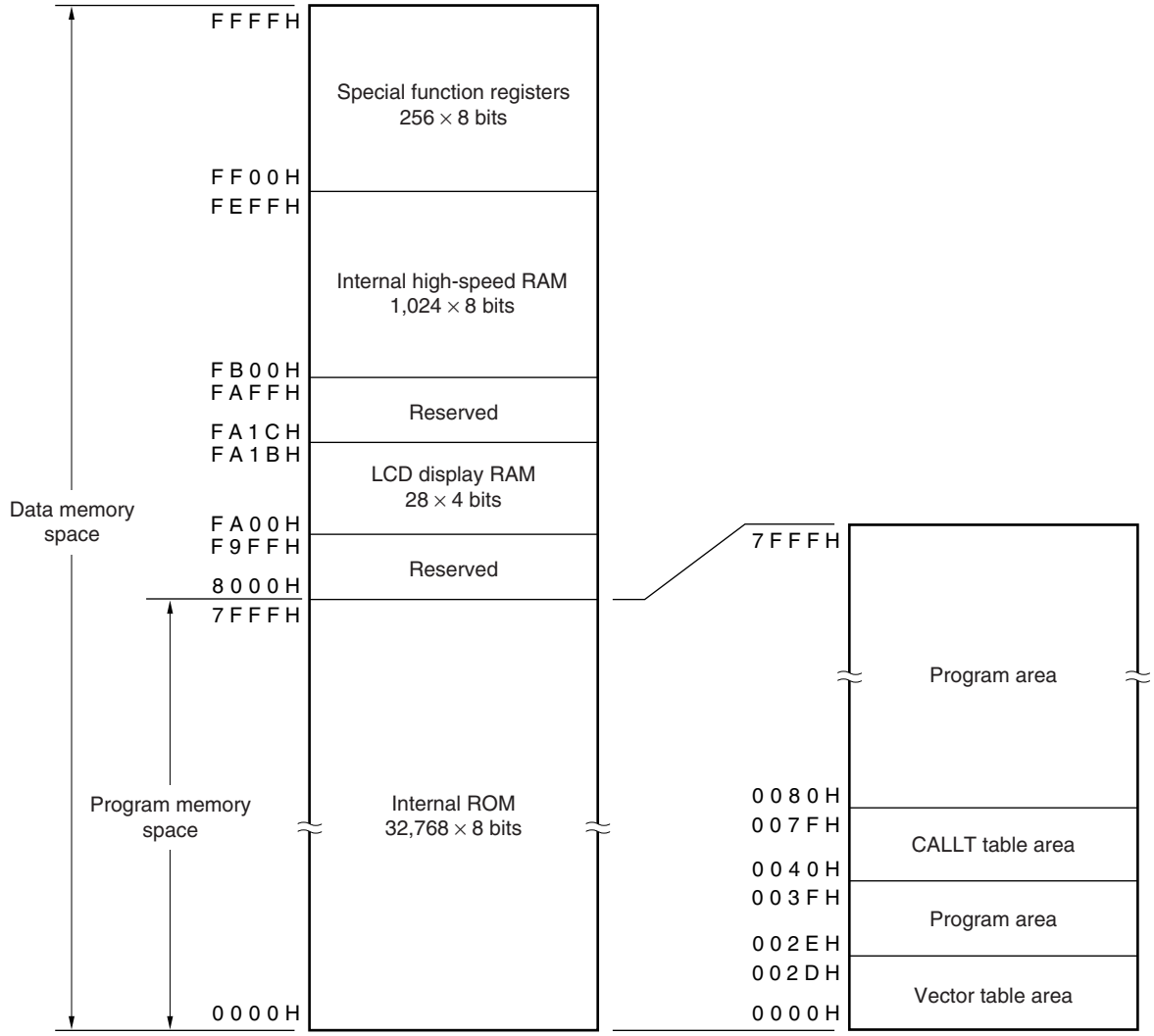


Figure 3-3. Memory Map (μPD78F9478)

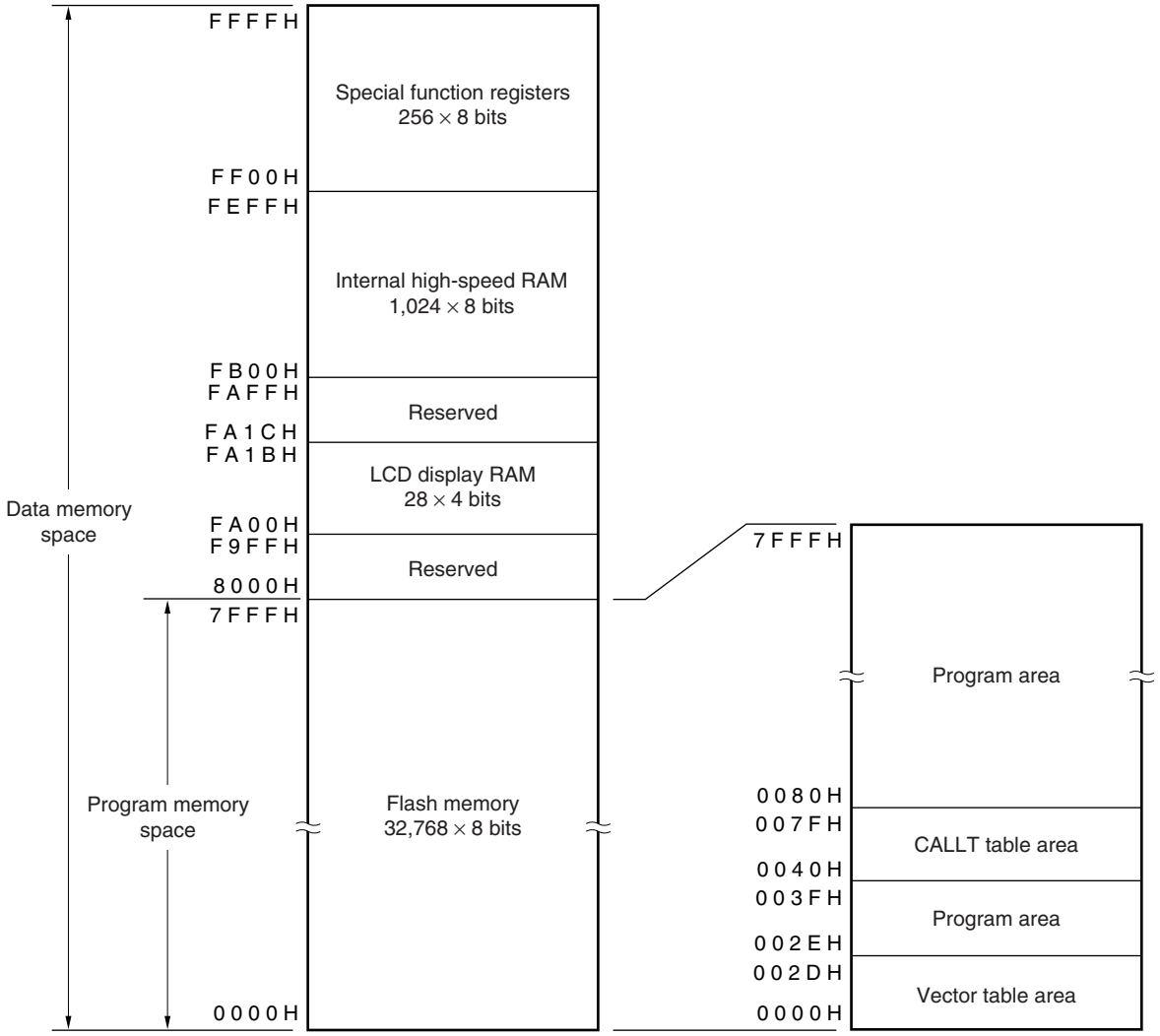


Figure 3-4. Memory Map (μPD789479)

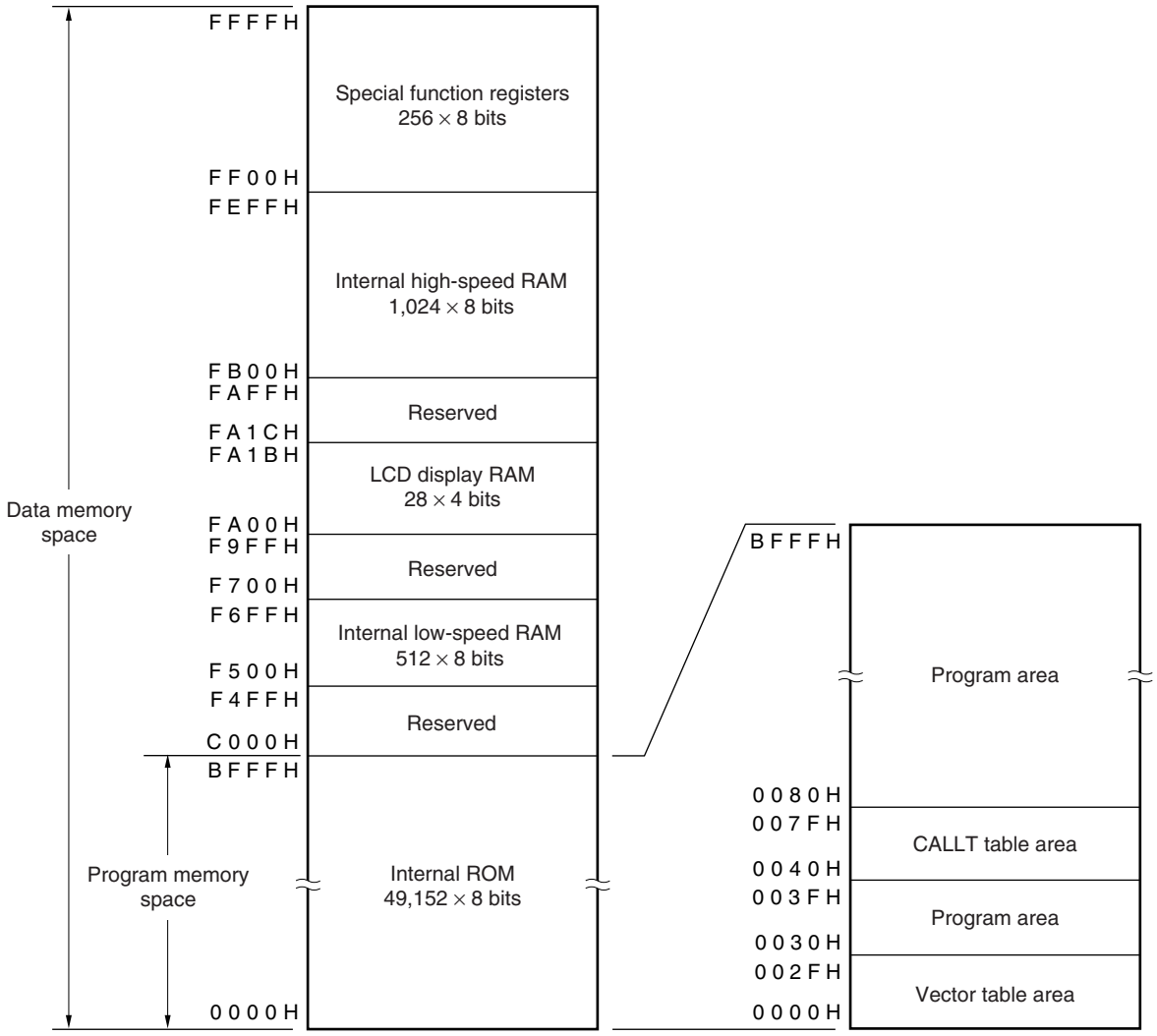
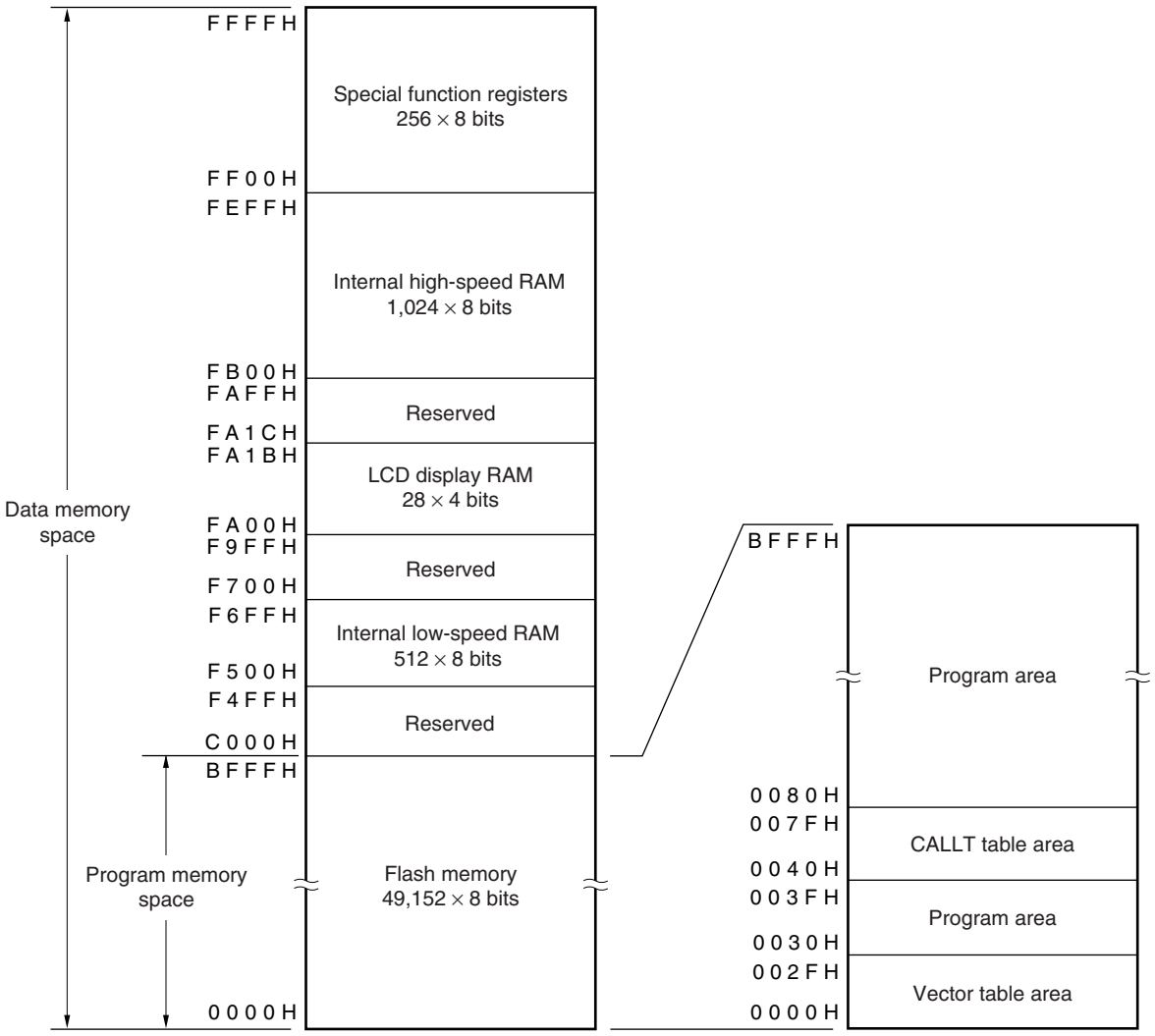


Figure 3-5. Memory Map (μPD78F9479)



3.1.1 Internal program memory space

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

Internal ROM (or flash memory) with the following capacity is provided for each product in the μPD789479 Subseries.

Table 3-1. Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
μPD789477	Mask ROM	24,576 × 8 bits
μPD789478		32,768 × 8 bits
μPD789479		49,152 × 8 bits
μPD78F9478	Flash memory	32,768 × 8 bits
μPD78F9479		49,152 × 8 bits

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

(1) Vector table area

The 46-byte area of addresses 0000H to 002DH in the μPD789477, 789478, and 78F9478 and the 48-byte area of addresses 0000H to 002FH in the μPD789479 and 78F9479 is reserved as a vector table area. This area stores program start addresses to be used when branching by $\overline{\text{RESET}}$ input or interrupt request generation. Of a 16-bit program address, the lower 8 bits are stored in an even address, and the higher 8 bits are stored in an odd address.

Table 3-2. Vector Table

Vector Table Address	Interrupt Request	Vector Table Address	Interrupt Request
0000H	$\overline{\text{RESET}}$ input	0018H	INTTM20
0004H	INTWDT	001AH	INTTM50
0006H	INTP0	001CH	INTTM60
0008H	INTP1	001EH	INTTM61
000AH	INTP2	0020H	INTAD0
000CH	INTP3	0022H	INTWT
000EH	INTRIN	0024H	INTKR00
0010H	INTSR20/INTCSI20	0026H	INTRERR
0012H	INTCSI10	0028H	INTGP
0014H	INTST20	002AH	INTREND
0016H	INTWTI	002CH	INTDFULL
		002EH	INTKR01[Note]

Note μPD789479 and 78F9479 only

(2) CALLT instruction table area

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

3.1.2 Internal data memory space

(1) Internal high-speed RAM and internal low-speed RAM

The μPD789479 Subseries products incorporate internal high-speed RAM and internal low-speed RAM of the following capacity for each product.

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

The internal low-speed RAM cannot be used as a stack.

Table 3-3. Internal High-Speed RAM, Internal Low-Speed RAM Capacity

Part Number	Structure	Capacity
μPD789477	768 × 8 bits	–
μPD789478	1,024 × 8 bits	
μPD789479		512 × 8 bits
μPD78F9478		–
μPD78F9479		512 × 8 bits

(2) LCD display RAM

LCD display RAM is incorporated in the area between FA00H and FA1BH.

The LCD display RAM can also be used as ordinary RAM.

3.1.3 Special function register (SFR) area

Special function registers (SFRs) of on-chip peripheral hardware are allocated in the area between FF00H and FFFFH (see **Table 3-4**).

3.1.4 Data memory addressing

The μPD789479 Subseries is provided with a variety of addressing modes to make memory manipulation as efficient as possible. At the addresses corresponding to data memory area especially, specific addressing modes that correspond to the particular function of an area such as the special function registers are available. Figures 3-6 to 3-10 show the data memory addressing modes.

Figure 3-6. Data Memory Addressing (μPD789477)

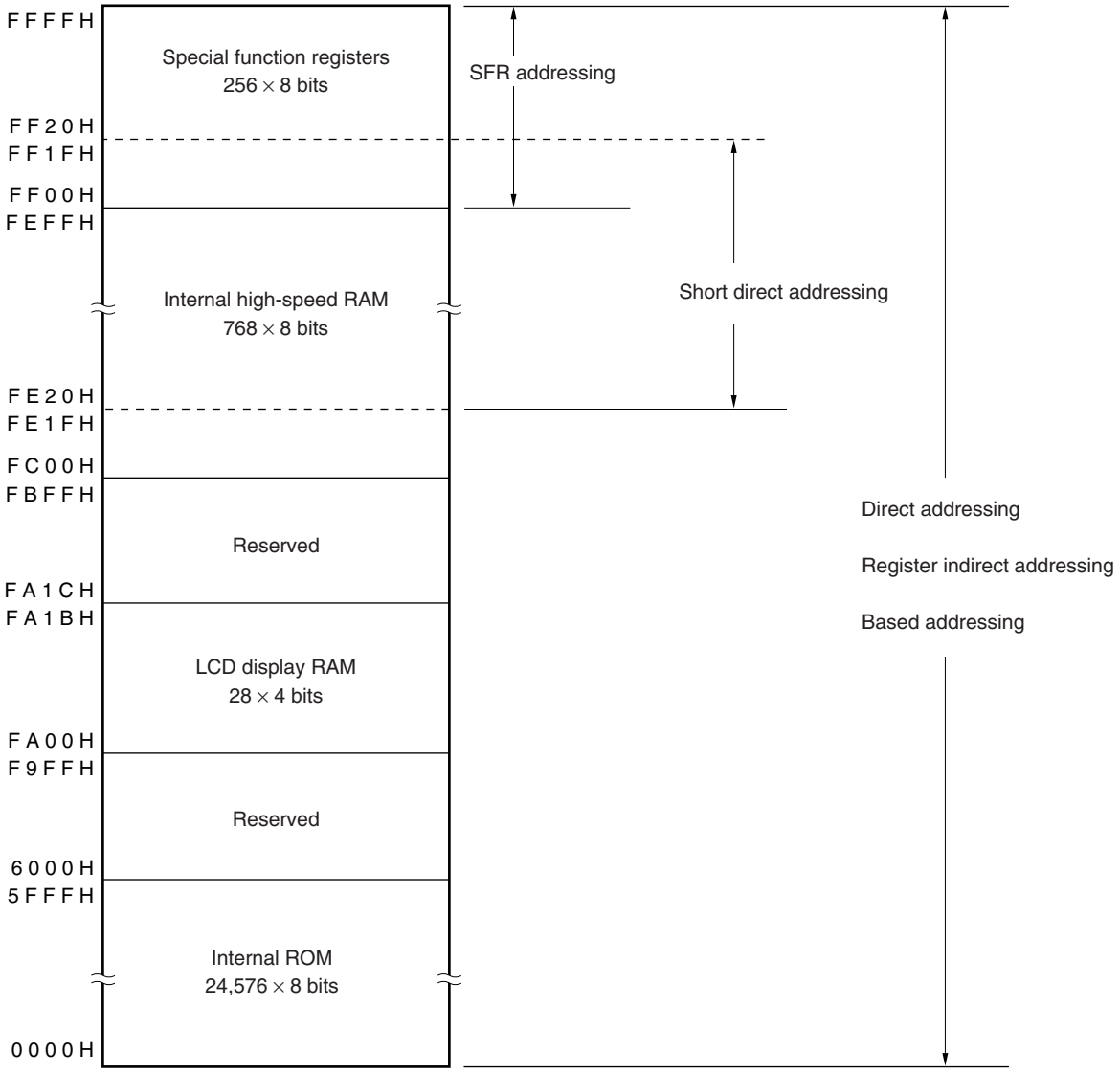


Figure 3-7. Data Memory Addressing (μPD789478)

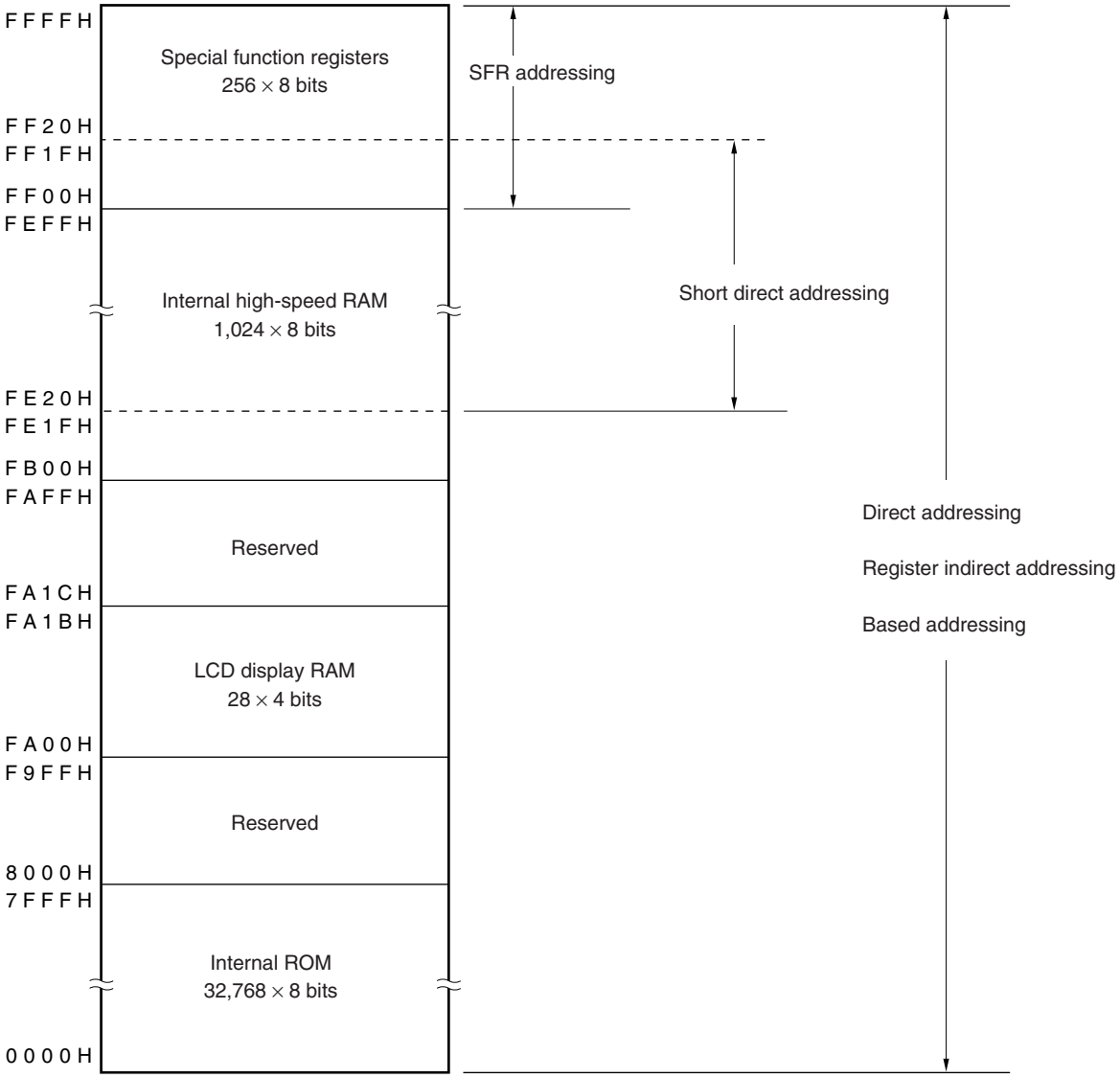


Figure 3-8. Data Memory Addressing (μPD78F9478)

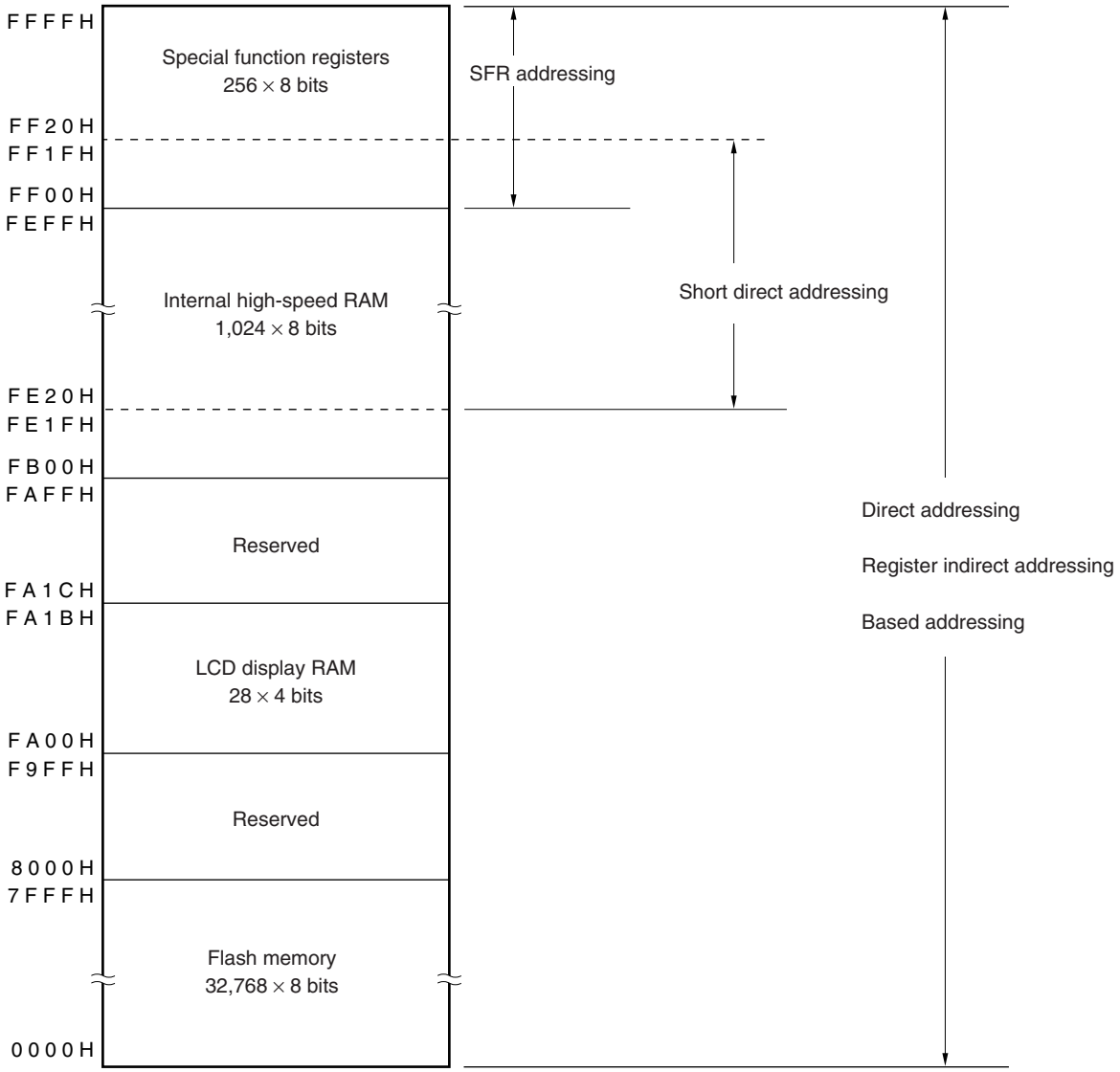


Figure 3-9. Data Memory Addressing (μPD789479)

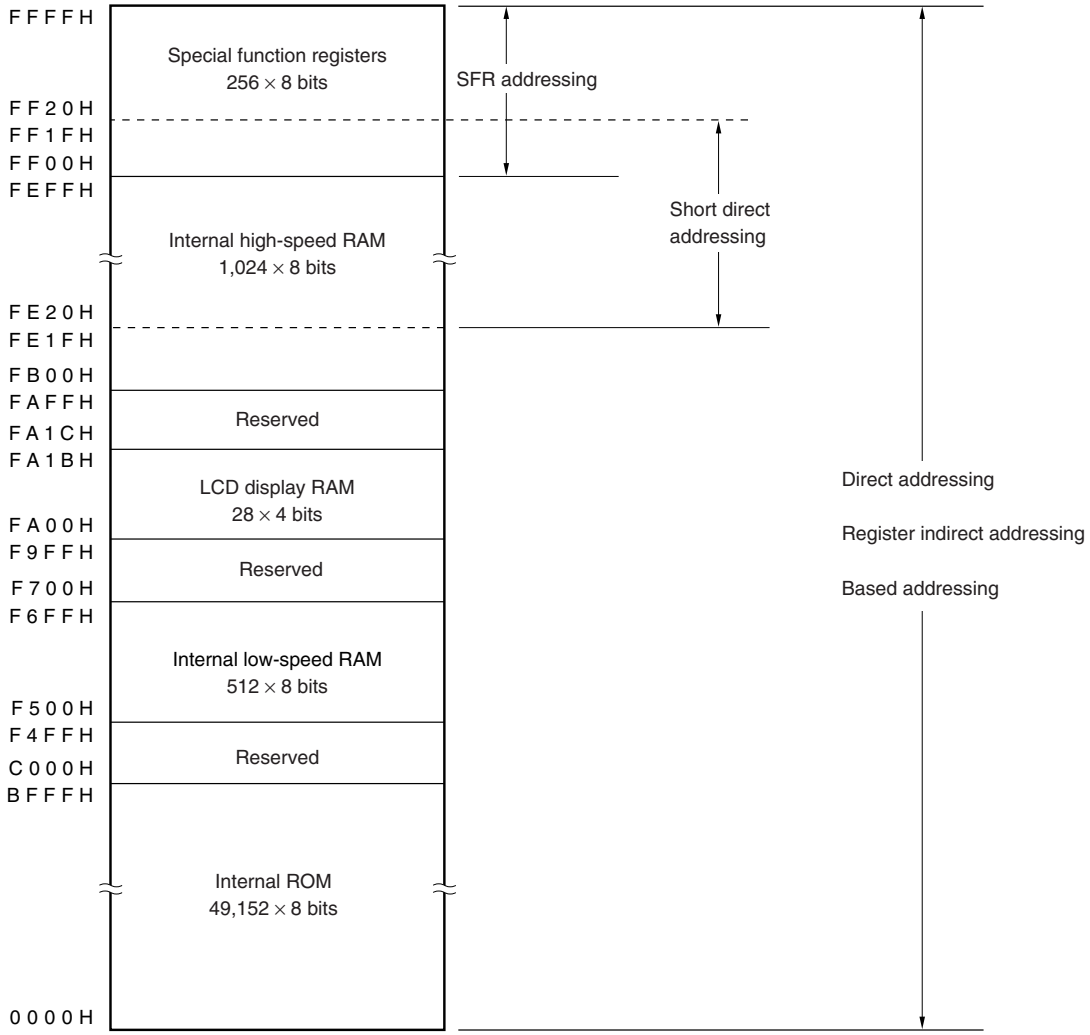
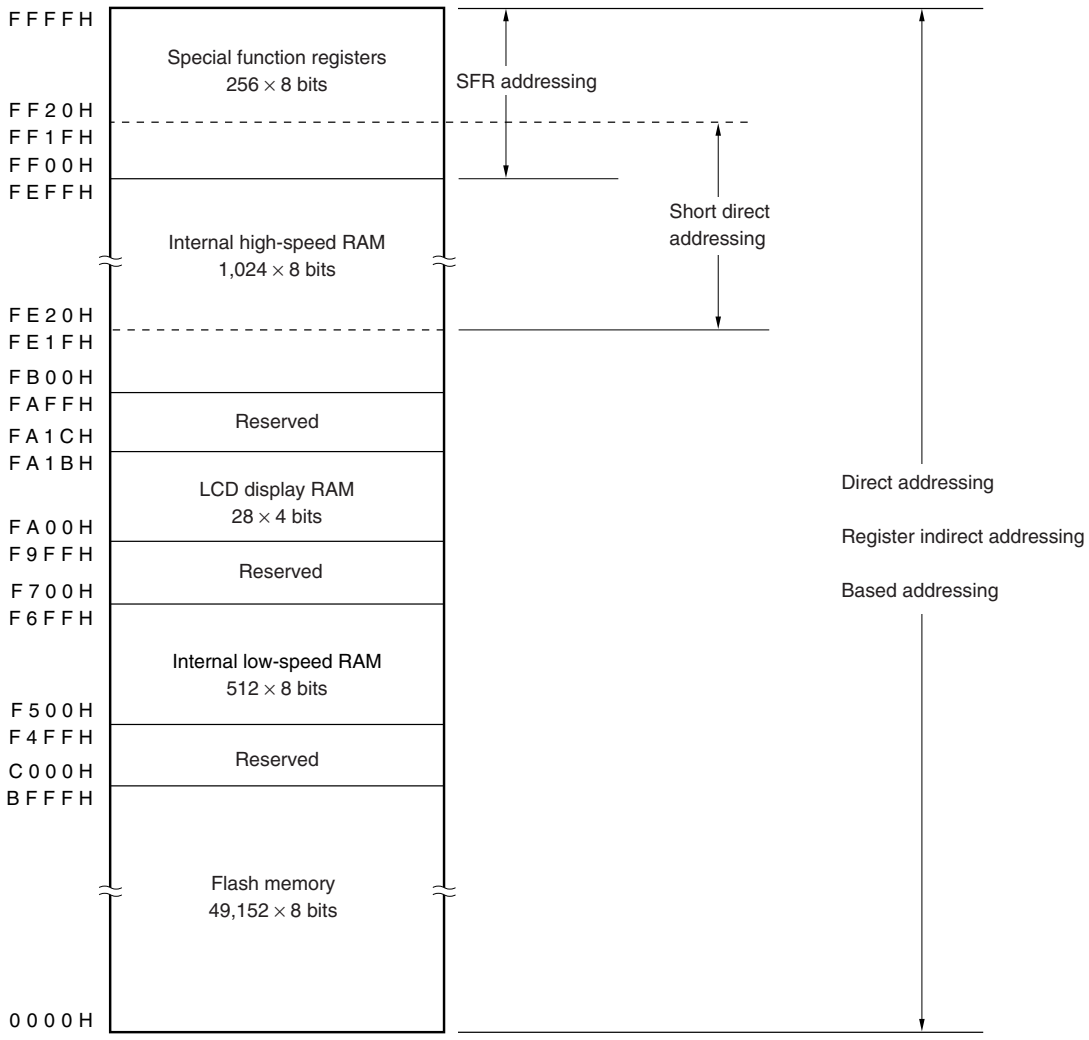


Figure 3-10. Data Memory Addressing (μPD78F9479)



3.2 Processor Registers

The μPD789479 Subseries is provided with the following on-chip processor registers.

3.2.1 Control registers

The control registers contain special functions to control the program sequence status and stack memory. The program counter, program status word, and stack pointer are control registers.

(1) Program counter (PC)

The program counter is a 16-bit register that holds the address information of the next program to be executed.

In normal operation, the PC is automatically incremented according to the number of bytes of the instruction to be fetched. When a branch instruction is executed, immediate data or register contents are set.

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

Figure 3-11. Program Counter Configuration

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

(2) Program status word (PSW)

The program status word is an 8-bit register consisting of various flags to be set/reset by instruction execution.

The program status word contents are automatically stacked upon interrupt request generation or PUSH PSW instruction execution and are automatically restored upon execution of the RETI and POP PSW instructions.

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

Figure 3-12. Program Status Word Configuration

	7							0
PSW	IE	Z	0	AC	0	0	1	CY

(a) Interrupt enable flag (IE)

This flag controls interrupt request acknowledgement operations of the CPU.

When 0, IE is set to the interrupt disabled status (DI), and interrupt requests other than non-maskable interrupts are all disabled.

When 1, IE is set to the interrupt enabled status (EI). Interrupt request acknowledgement enable is controlled by the interrupt mask flag for the corresponding interrupt source.

IE is reset (0) upon DI instruction execution or interrupt acknowledgment and is set (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) 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.

(d) Carry flag (CY)

This flag stores an overflow or 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 that holds the start address of the memory stack area. Only the internal high-speed RAM area can be set as the stack area.

Figure 3-13. Stack Pointer Configuration

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

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

Each stack operation saves/restores data as shown in Figures 3-14 and 3-15.

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

Figure 3-14. Data to Be Saved to Stack Memory

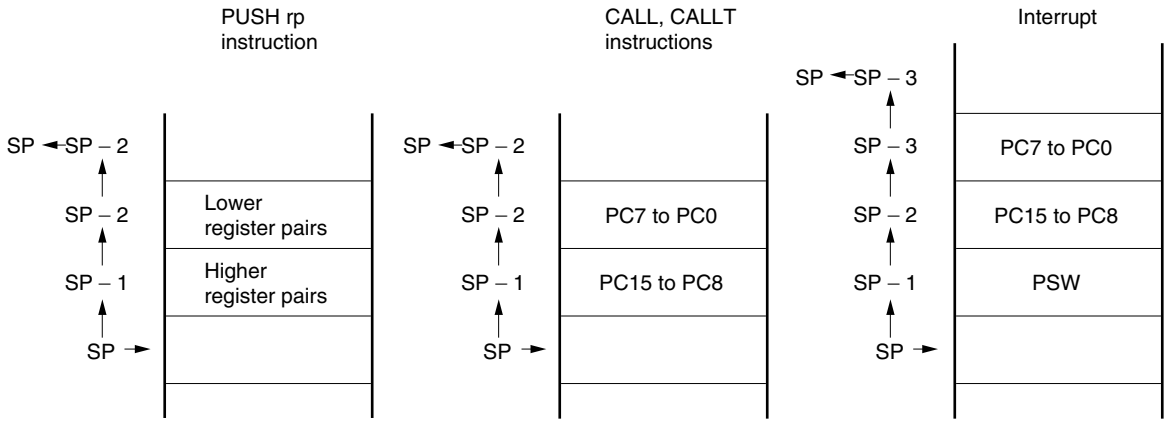
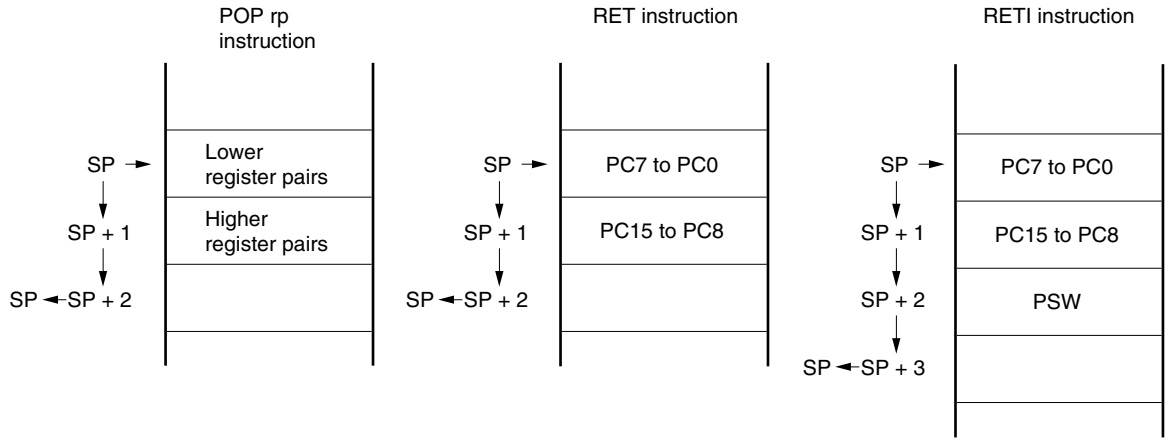


Figure 3-15. Data to Be Restored from Stack Memory



3.2.2 General-purpose registers

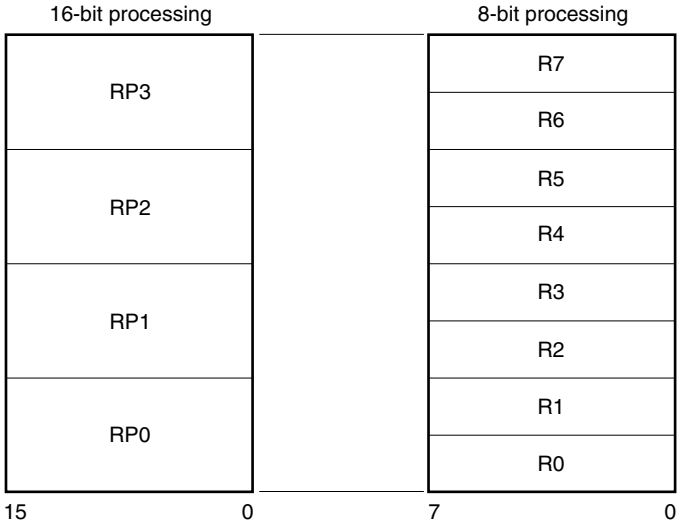
The general-purpose registers consist of eight 8-bit registers (X, A, C, B, E, D, L, and H).

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

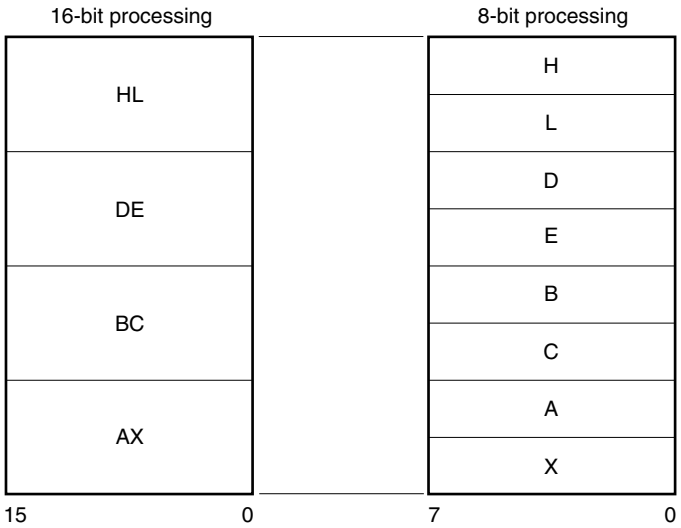
General-purpose registers can be described in terms of function names (X, A, C, B, E, D, L, H, AX, BC, DE, or HL) or absolute names (R0 to R7 and RP0 to RP3).

Figure 3-16. General-Purpose Register Configuration

(a) Absolute names



(b) Function names



3.2.3 Special function registers (SFRs)

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

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

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

The manipulatable bits can be specified as follows.

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

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

- 16-bit manipulation
  Describe the symbol reserved by the assembler for the 16-bit manipulation instruction operand. When addressing an address, describe an even address.

Table 3-4 lists the special function registers. The meanings of the symbols in this table are as follows.

- Symbol
  Indicates the addresses of the incorporated special function registers. It is a reserved word in the assembler, and is defined as an sfr variable using the #pragma sfr directive in the C compiler. Therefore, these symbols can be used as instruction operands if an assembler or integrated debugger is used.

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

- Bit unit for manipulation
  Indicates the bit units (1, 8, 16) in which the special function register in question can be manipulated.

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

Table 3-4. Special Function Registers (1/3)

Address	Special Function Register (SFR) Name	Symbol		R/W	Bit Unit for Manipulation			After Reset
					1 Bit	8 Bits	16 Bits	
FF00H	Port 0	P0		R/W	√	√	–	00H
FF01H	Port 1	P1			√	√	–	
FF02H	Port 2	P2			√	√	–	
FF03H	Port 3	P3			√	√	–	
FF05H	Port 5	P5			√	√	–	
FF06H	Port 6	P6		R	√	√	–	
FF07H	Port 7[Note]	P7			√	√	–	
FF08H	Port 8[Note]	P8		R/W	√	√	–	
FF0AH	8-bit compare register 61	CR61		W	–	√	–	Undefined
FF0BH	8-bit timer counter 61	TM61		R	–	√	–	00H
FF0CH	8-bit compare register 60	CR60	CR6	W	–	√	√	Undefined
FF0DH	8-bit compare register 50	CR50			–	√		
FF0EH	8-bit timer counter 60	TM60	TM6	R	–	√	√	00H
FF0FH	8-bit timer counter 50	TM50			–	√		
FF11H	Serial I/O shift register 1A0	SIO1A0		R/W	–	√	–	
FF12H	16-bit multiplication result store register L	MUL0L	MUL 0	R	–	√	√	Undefined
FF13H	16-bit multiplication result store register H	MUL0H			–	√		
FF15H	A/D conversion result register 0	ADCRL0			–	√	–	00H
FF16H	16-bit compare register 20	CR20		W	–	–	√	FFFFH
FF17H								
FF18H	16-bit timer counter 20	TM20		R	–	–	√	0000H
FF19H								
FF1AH	16-bit capture register 20	TCP20			–	–	√	Undefined
FF1BH								
FF20H	Port mode register 0	PM0		R/W	√	√	–	FFH
FF21H	Port mode register 1	PM1			√	√	–	
FF22H	Port mode register 2	PM2			√	√	–	
FF23H	Port mode register 3	PM3			√	√	–	
FF25H	Port mode register 5	PM5			√	√	–	
FF28H	Port mode register 8[Note]	PM8			√	√	–	
FF30H	Pull-up resistor option register B0	PUB0			√	√	–	00H
FF31H	Pull-up resistor option register B1	PUB1			√	√	–	
FF32H	Pull-up resistor option register B2	PUB2			√	√	–	
FF33H	Pull-up resistor option register B3	PUB3			√	√	–	
FF40H	8-bit H width compare register 61	CRH61		W	–	√	–	Undefined
FF41H	8-bit timer mode control register 61	TMC61		R/W	√	√	–	00H
FF42H	Watchdog timer clock selection register	WDCS			–	√	–	
FF46H	Subclock selection register[Note]	SSCK			√	√	–	

Note When used as a port by a mask option or port function register.

Table 3-4. Special Function Registers (2/3)

Address	Special Function Register (SFR) Name	Symbol	R/W	Bit Unit for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
FF48H	16-bit timer mode control register 20	TMC20	R/W	√	√	–	00H
FF4AH	Watch timer mode control register	WTM		√	√	–	
FF4BH	Watch timer interrupt time selection register	WTIM		√	√	–	
FF4CH	8-bit H width compare register 60	CRH60	W	–	√	–	Undefined
FF4DH	8-bit timer mode control register 50	TMC50	R/W	√	√	–	00H
FF4EH	8-bit timer mode control register 60	TMC60		√	√	–	
FF4FH	Carrier generator output control register 60	TCA60		√	√	–	
FF57H	Port function register 7[Note]	PF7	W	–	√	–	
FF58H	Port function register 8[Note]	PF8		–	√	–	
FF60H	Remote controller receive control register	RMCN	R/W	√	√	–	
FF61H	Remote controller receive data register	RMDR	R	–	√	–	
FF62H	Remote controller shift register receive counter register	RMSCR		–	√	–	
FF63H	Remote controller receive shift register	RMSR		–	√	–	
FF64H	Remote controller receive GPLS compare register	RMGPLS	R/W	–	√	–	
FF65H	Remote controller receive GPLL compare register	RMGPLL		–	√	–	
FF66H	Remote controller receive GPHS compare register	RMGPHS		–	√	–	
FF67H	Remote controller receive GPHL compare register	RMGPHL		–	√	–	
FF68H	Remote controller receive DLS compare register	RMDLS		–	√	–	
FF69H	Remote controller receive DLL compare register	RMDLL		–	√	–	
FF6AH	Remote controller receive DH0S compare register	RMDH0S		–	√	–	
FF6BH	Remote controller receive DH0L compare register	RMDH0L		–	√	–	
FF6CH	Remote controller receive DH1S compare register	RMDH1S		–	√	–	
FF6DH	Remote controller receive DH1L compare register	RMDH1L		–	√	–	
FF6EH	Remote controller receive end width selection register	RMER		–	√	–	
FF70H	Asynchronous serial interface mode register 20	ASIM20		√	√	–	
FF71H	Asynchronous serial interface status register 20	ASIS20	R	√	√	–	
FF72H	Serial operation mode register 20	CSIM20	R/W	√	√	–	
FF73H	Baud rate generator control register 20	BRGC20		–	√	–	
FF74H	Transmit shift register 20	TXS20, SIO20	W	–	√	–	FFH
	Receive buffer register 20	RXB20	R	–	√	–	Undefined
FF78H	Serial operation mode register 1A0	CSIM1A0	R/W	√	√	–	00H
FF79H	Automatic data transmit/receive control register 0	ADTC0		√	√	–	
FF7AH	Automatic data transmit/receive address pointer 0	ADTP0		–	√	–	Undefined
FF7BH	Automatic data transmit/receive interval specification register 0	ADTI0		√	√	–	00H

Note These registers function only in the μPD78F9478 and 78F9479; however, writing to these registers in the μPD789477, 789478, and 789479 will not affect the operation.

Table 3-4. Special Function Registers (3/3)

Address	Special Function Register (SFR) Name	Symbol	R/W	Bit Unit for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
FF80H	A/D converter mode register 0	ADML0	R/W	√	√	–	00H
FF81H	A/D converter mode register 1	ADML1		√	√	–	
FF84H	Analog input channel specification register 0	ADS0		√	√	–	
FFA0H	Serial interface buffer memory 0	SBMEM0		–	√	–	Undefined
FFA1H	Serial interface buffer memory 1	SBMEM1		–	√	–	
FFA2H	Serial interface buffer memory 2	SBMEM2		–	√	–	
FFA3H	Serial interface buffer memory 3	SBMEM3		–	√	–	
FFA4H	Serial interface buffer memory 4	SBMEM4		–	√	–	
FFA5H	Serial interface buffer memory 5	SBMEM5		–	√	–	
FFA6H	Serial interface buffer memory 6	SBMEM6		–	√	–	
FFA7H	Serial interface buffer memory 7	SBMEM7		–	√	–	
FFA8H	Serial interface buffer memory 8	SBMEM8		–	√	–	
FFA9H	Serial interface buffer memory 9	SBMEM9		–	√	–	
FFAAH	Serial interface buffer memory A	SBMEMA		–	√	–	
FFABH	Serial interface buffer memory B	SBMEMB		–	√	–	
FFACH	Serial interface buffer memory C	SBMEMC		–	√	–	
FFADH	Serial interface buffer memory D	SBMEMD		–	√	–	
FFAEH	Serial interface buffer memory E	SBMEME		–	√	–	
FFAFH	Serial interface buffer memory F	SBMEMF		–	√	–	
FFB0H	LCD display mode register 0	LCDM0		√	√	–	00H
FFB2H	LCD clock control register 0	LCDC0		√	√	–	
FFD0H	Multiplication data register A0	MRA0	W	–	√	–	Undefined
FFD1H	Multiplication data register B0	MRB0		–	√	–	
FFD2H	Multiplier control register 0	MULC0	R/W	√	√	–	00H
FFE0H	Interrupt request flag register 0	IF0		√	√	–	
FFE1H	Interrupt request flag register 1	IF1		√	√	–	
FFE2H	Interrupt request flag register 2	IF2		√	√	–	
FFE4H	Interrupt mask flag register 0	MK0		√	√	–	FFH
FFE5H	Interrupt mask flag register 1	MK1		√	√	–	
FFE6H	Interrupt mask flag register 2	MK2		√	√	–	
FFECH	External interrupt mode register 0	INTM0		–	√	–	00H
FFEDH	External interrupt mode register 1	INTM1		–	√	–	
FFF0H	Subclock oscillation mode register	SCKM		√	√	–	
FFF2H	Subclock control register	CSS		√	√	–	
FFF4H	Key return mode register 01[Note]	KRM01		√	√	–	
FFF5H	Key return mode register 00	KRM00		√	√	–	
FFF9H	Watchdog timer mode register	WDTM		√	√	–	
FFFAH	Oscillation stabilization time selection register	OSTS		–	√	–	04H
FFFBH	Processor clock control register	PCC		√	√	–	02H

Note μPD789479, 78F9479 only

3.3 Instruction Address Addressing

An instruction address is determined by the program counter (PC) contents. The PC contents are normally incremented (+1 for each byte) automatically according to the number of bytes of an instruction to be fetched each time another instruction is executed. When a branch instruction is executed, the branch destination information is set to the PC and branched by the following addressing (for details of each instruction, refer to **78K/0S Series Instructions User's Manual (U11047E)**).

3.3.1 Relative addressing

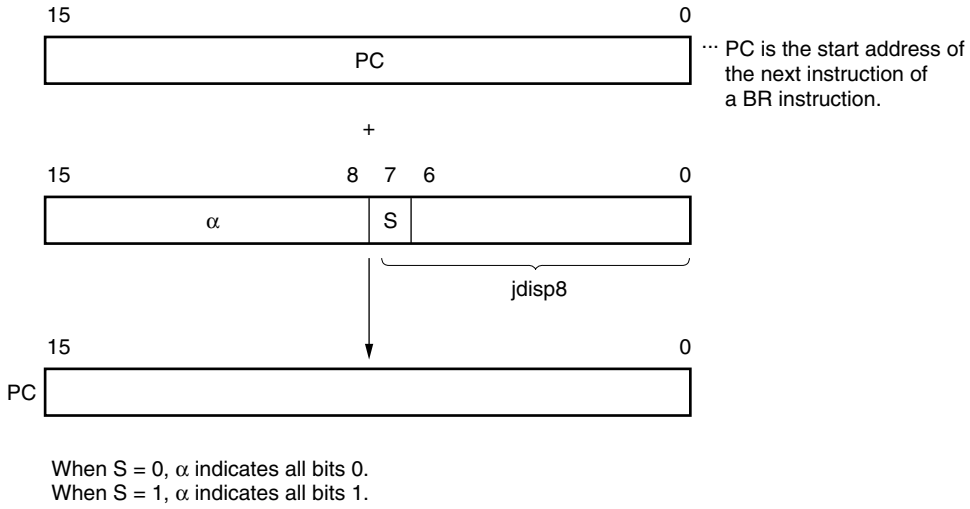
[Function]

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

This means that information is relatively branched to a location between –128 and +127, from the start address of the next instruction when relative addressing is used.

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

[Illustration]



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

3.3.2 Immediate addressing

[Function]

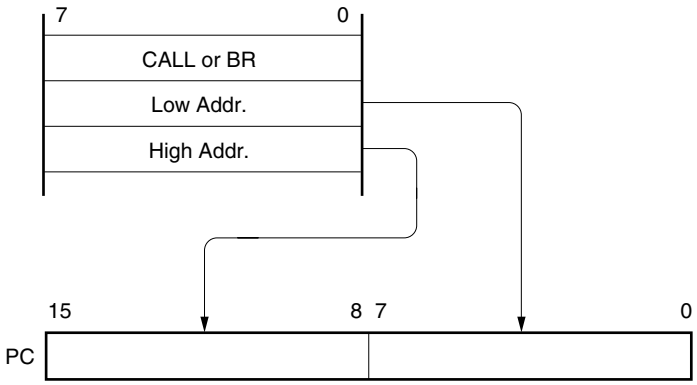
Immediate data in the instruction word is transferred to the program counter (PC) and branched.

This function is carried out when the CALL !addr16 or BR !addr16 instruction is executed.

CALL !addr16 and BR !addr16 instructions can be branched to any location in the memory space.

[Illustration]

In case of CALL !addr16 and BR !addr16 instructions



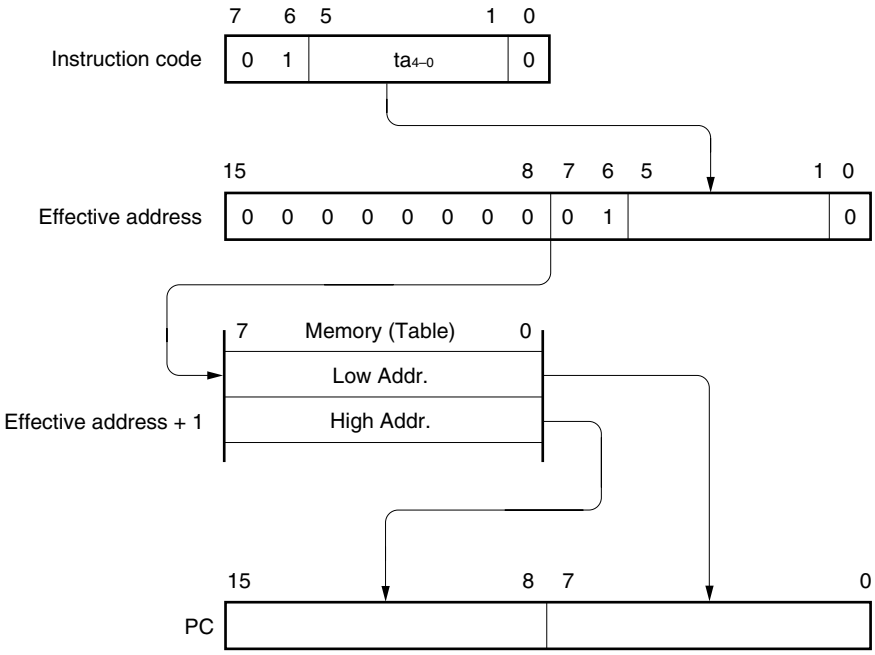
3.3.3 Table indirect addressing

[Function]

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

This function is carried out when the CALLT [addr5] instruction is executed. The instruction enables a branch to any location in the memory space by referring to the addresses stored in the memory table at 40H to 7FH.

[Illustration]



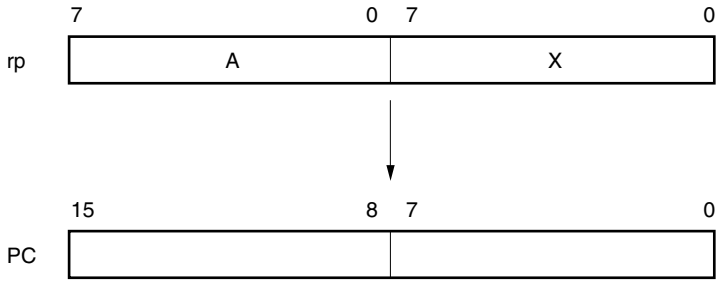
3.3.4 Register addressing

[Function]

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

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

[Illustration]



3.4 Operand Address Addressing

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

3.4.1 Direct addressing

[Function]

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

[Operand format]

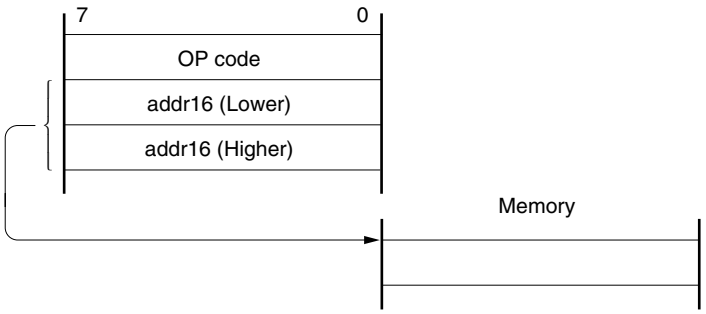
Identifier	Description
addr16	Label or 16-bit immediate data

[Description example]

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

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

[Illustration]



3.4.2 Short direct addressing

[Function]

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

The fixed space is the 256-byte space FE20H to FF1FH where the addressing is applied. Internal high-speed RAM and special function registers (SFRs) are mapped at FE20H to FEFFH and FF00H to FF1FH, respectively.

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

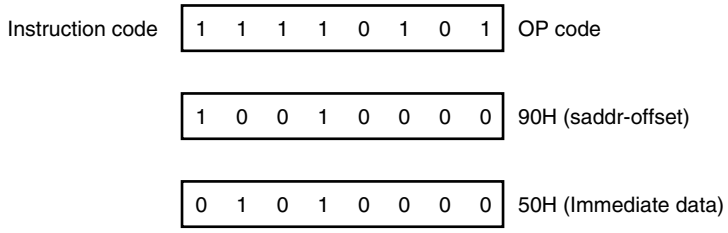
When 8-bit immediate data is at 20H to FFH, bit 8 of an effective address is set to 0. When it is at 00H to 1FH, bit 8 is set to 1. See **[Illustration]** below.

[Operand format]

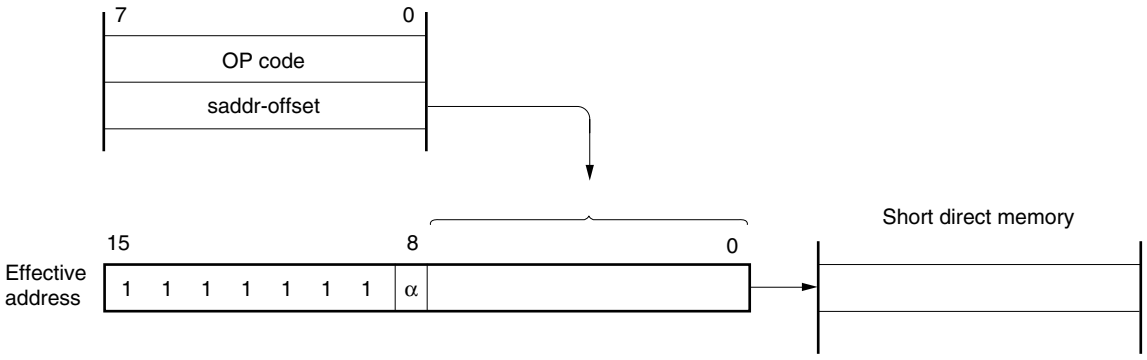
Identifier	Description
saddr	Label or FE20H to FF1FH immediate data
saddrp	Label or FE20H to FF1FH immediate data (even address only)

[Description example]

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



[Illustration]



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

3.4.3 Special function register (SFR) addressing

[Function]

The memory-mapped special function registers (SFRs) are addressed with 8-bit immediate data in an instruction word.

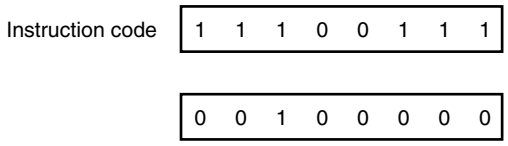
This addressing is applied to the 256-byte space FF00H to FFFFH. However, the SFRs mapped at FF00H to FF1FH can also be accessed with short direct addressing.

[Operand format]

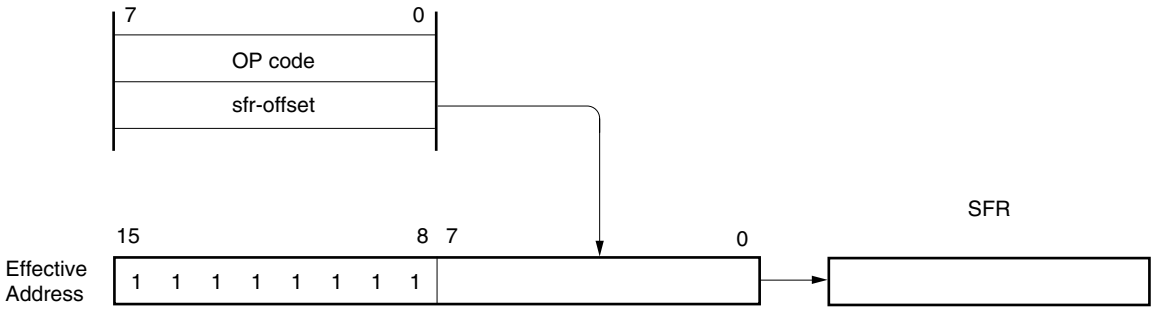
Identifier	Description
sfr	Special function register name

[Description example]

MOV PM0, A; When selecting PM0 for sfr



[Illustration]



3.4.4 Register addressing

[Function]

In the register addressing mode, general-purpose registers are accessed as operands. The general-purpose register to be accessed is specified by a register specification code or functional name in the instruction code. Register addressing is carried out when an instruction with the following operand format is executed. When an 8-bit register is specified, one of the eight registers is specified with 3 bits in the instruction code.

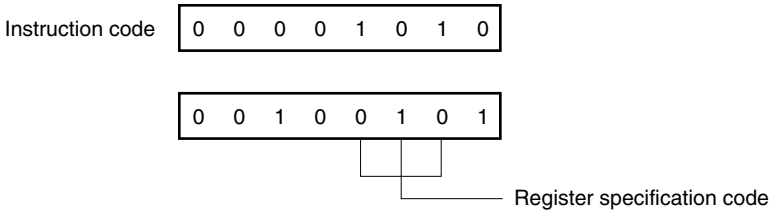
[Operand format]

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

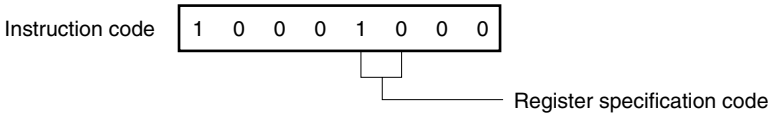
r and rp can be described with absolute names (R0 to R7 and RP0 to RP3) as well as function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL).

[Description example]

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



INCW DE; When selecting the DE register pair for rp



3.4.5 Register indirect addressing

[Function]

In the register indirect addressing mode, memory is manipulated according to the contents of a register pair specified as an operand. The register pair to be accessed is specified by the register pair specification code in an instruction code.

This addressing can be carried out for all the memory spaces.

[Operand format]

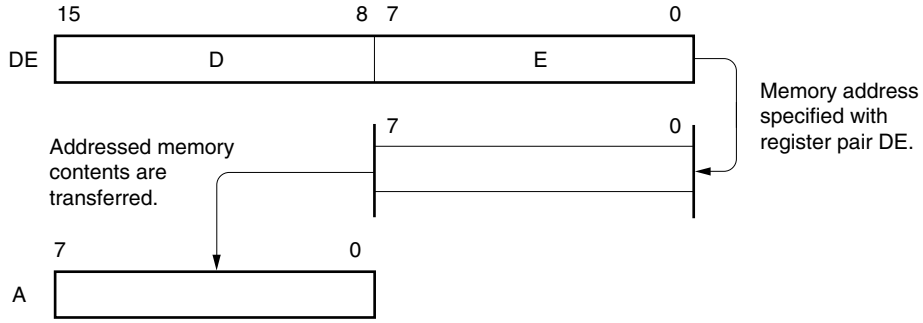
Identifier	Description
–	[DE], [HL]

[Description example]

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

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

[Illustration]



3.4.6 Based addressing

[Function]

8-bit immediate data is added to the contents of the base register, that is, the HL register pair, and the sum is used to address the memory. Addition is performed by expanding the offset data as a positive number to 16 bits. A carry from the 16th bit is ignored. This addressing can be carried out for all the memory spaces.

[Operand format]

Identifier	Description
–	[HL+byte]

[Description example]

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

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

3.4.7 Stack addressing

[Function]

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

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

Only the internal high-speed RAM area can be addressed using stack addressing.

[Description example]

In the case of PUSH DE

Instruction code	1	0	1	0	1	0	1	0

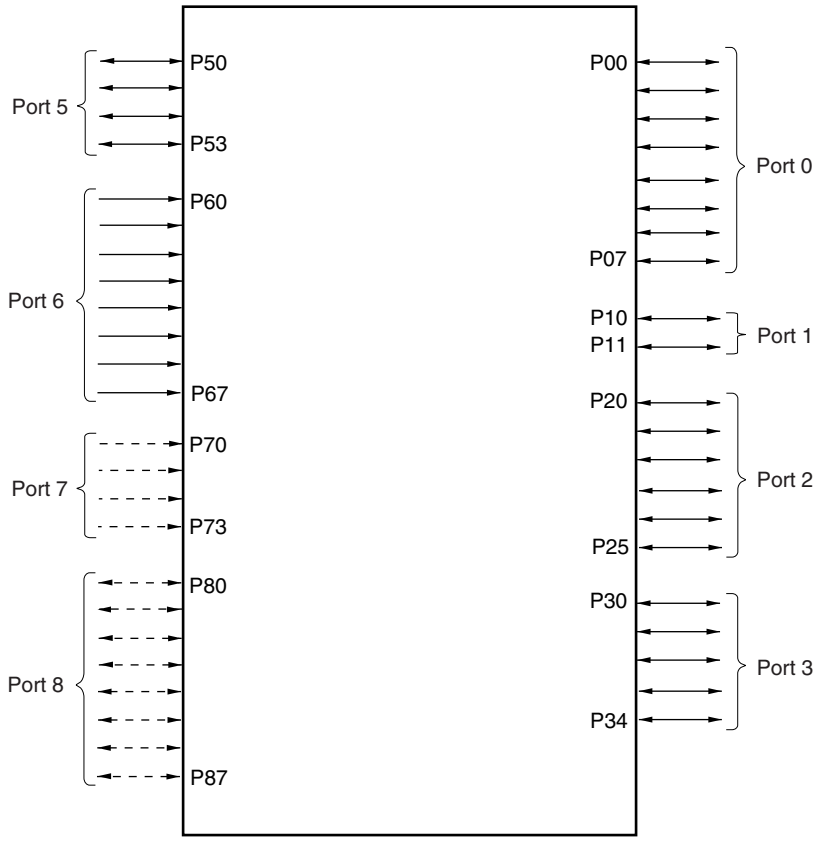
CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

The μPD789479 Subseries provides the ports shown in Figure 4-1, enabling various methods of control. The functions of each port are shown in Table 4-1.

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

Figure 4-1. Port Types



Remark Ports 7 and 8 are used when the port function is selected by a mask option or port function register.

Table 4-1. Port Functions

Port Name	Pin Name	Function
Port 0	P00 to P07	I/O port. Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be specified in 1-bit units by means of pull-up resistor option register B0 (PUB0).
Port 1	P10, P11	I/O port. Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be specified in 1-bit units by means of pull-up resistor option register B1 (PUB1).
Port 2	P20 to P25	I/O port. Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be specified in 1-bit units by means of pull-up resistor option register B2 (PUB2).
Port 3	P30 to P34	I/O port. Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be specified in 1-bit units by means of pull-up resistor option register B3 (PUB3).
Port 5	P50 to P53	N-ch open-drain I/O port. Input/output can be specified in 1-bit units. An on-chip pull-up resistor can be specified by mask option.
Port 6	P60 to P67	Input port
Port 7[Note 1]	P70 to P73	Input port (only when input port is selected by mask option or port function register)
Port 8[Note 2]	P80 to P87	I/O port (only when I/O port is selected by mask option)

Notes 1. Whether to use these pins as input port pins (P70 to P73) or segment outputs (S16 to S19) can be selected in 1-bit units by means of a mask option in the μPD789477, 789478, and 789479 or a port mode register in the μPD78F9478 and 78F9479 (refer to **4.3 (3) Port function registers** and **CHAPTER 20 MASK OPTIONS**).

2. Whether to use these pins as I/O port pins (P80 to P87) or segment outputs (S20 to S27) can be selected in 1-bit units by means of a mask option in the μPD789477, 789478, and 789479 or a port mode register in the μPD78F9478 and 78F9479 (refer to **4.3 (3) Port function registers** and **CHAPTER 20 MASK OPTIONS**).

4.2 Port Configuration

Ports have the following hardware configuration.

Table 4-2. Configuration of Port

Item	Configuration
Control registers	Port mode registers (PMm: m = 0 to 3, 5, 8) Pull-up resistor option registers (PUB0 to PUB3) Port function registers (PF7, PF8) (flash memory version only)
Ports	Total: 45 (CMOS I/O: 29, CMOS input: 12, N-ch open-drain I/O: 4)
Pull-up resistors	• Mask ROM version Total: 25 (software control: 21, mask option specification: 4) • Flash memory version Total: 21 (software control only)

4.2.1 Port 0

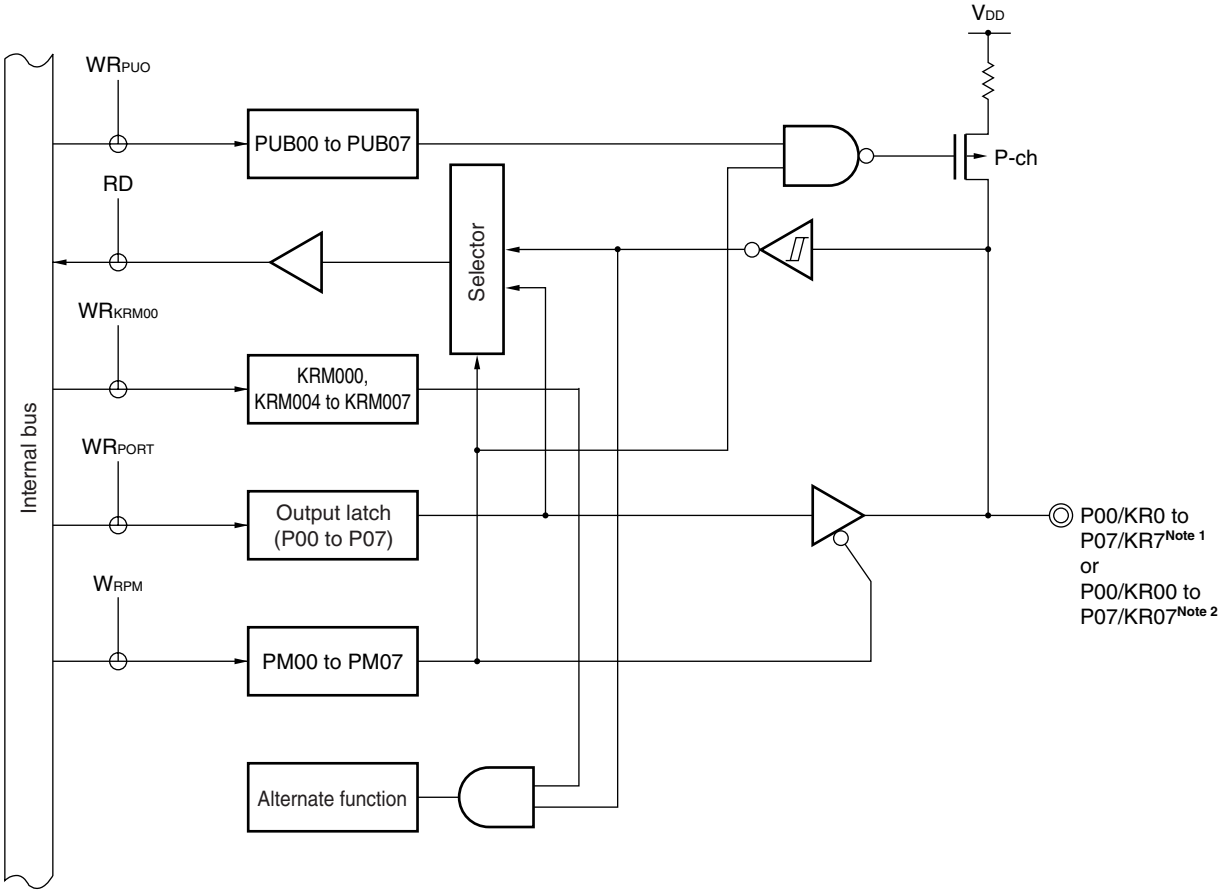
This is an 8-bit I/O port with an output latch. Port 0 can be specified in the input or output mode in 1-bit units by using port mode register 0 (PM0). When the P00 to P07 pins are used as input port pins, on-chip pull-up resistors can be connected in 1-bit units by using pull-up resistor option register B0 (PUB0).

This port is also used for key return signal input.

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

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

<R> **Figure 4-2. Block Diagram of P00 to P07**



KRM00: Key return mode register 00
PUB0: Pull-up resistor option register B0
PM: Port mode register
RD: Port 0 read signal
WR: Port 0 write signal

Notes 1. μPD789477, 789478, and 78F9478 only

2. μPD789479 and 78F9479 only

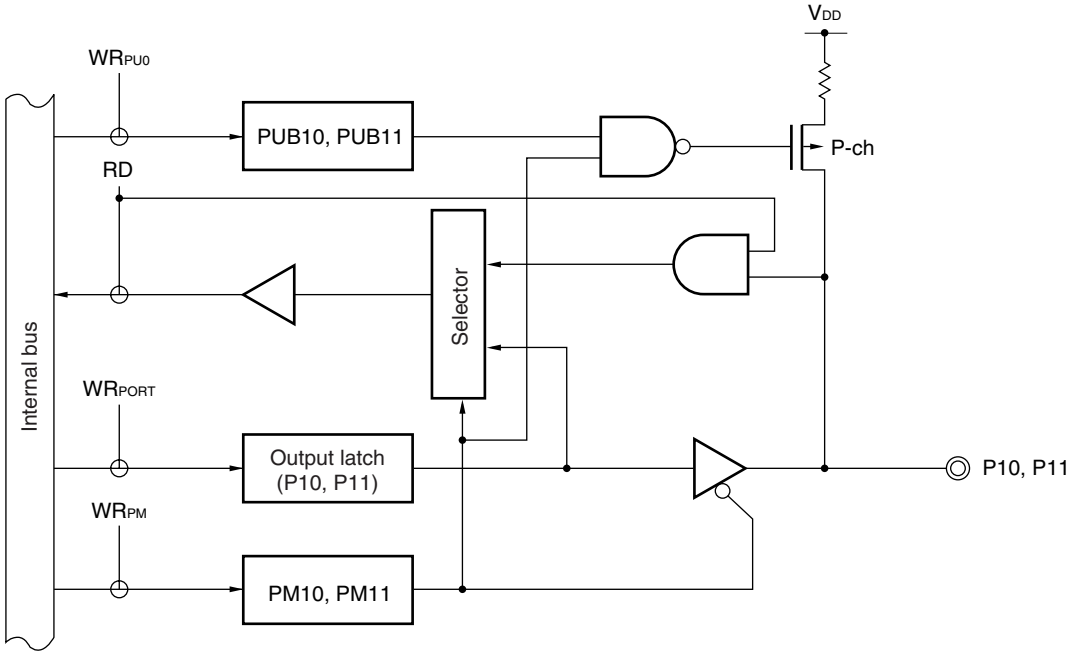
4.2.2 Port 1

This is a 2-bit I/O port with an output latch. Port 1 can be specified in the input or output mode in 1-bit units by using port mode register 1 (PM1). When using the P10 and P11 pins as input port pins, on-chip pull-up resistors can be connected in 1-bit units by using pull-up resistor option register B1 (PUB1).

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

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

Figure 4-3. Block Diagram of P10 and P11



PUB1: Pull-up resistor option register B1
PM: Port mode register
RD: Port 1 read signal
WR: Port 1 write signal

4.2.3 Port 2

This is a 6-bit I/O port with an output latch. Port 2 can be specified in the input or output mode in 1-bit units by using port mode register 2 (PM2). When using the P20 to P25 pins as input port pins, on-chip pull-up resistors can be connected in 1-bit units by using pull-up resistor option register B2 (PUB2).

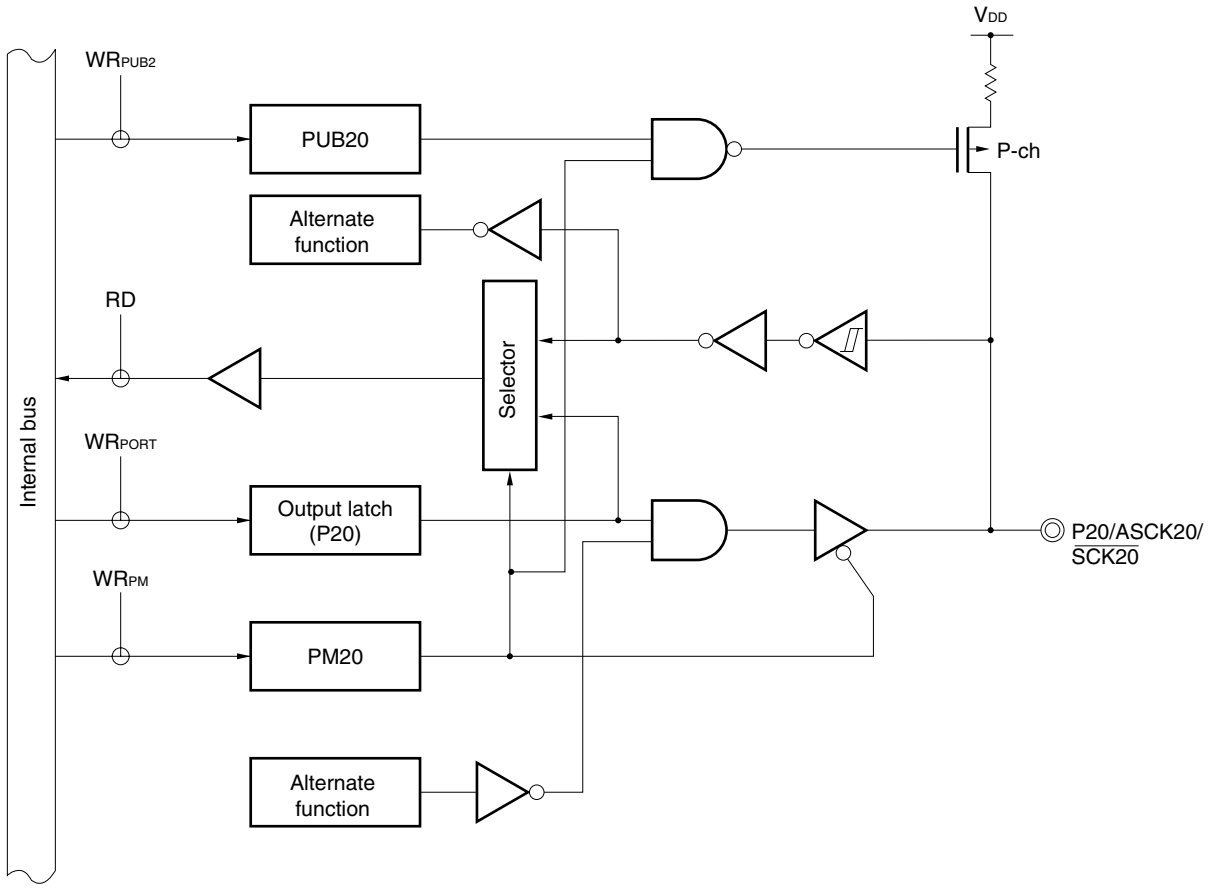
This port is also used for serial interface I/O.

$\overline{\text{RESET}}$ input set this port to input mode.

Figures 4-4 to 4-8 show block diagrams of port 2.

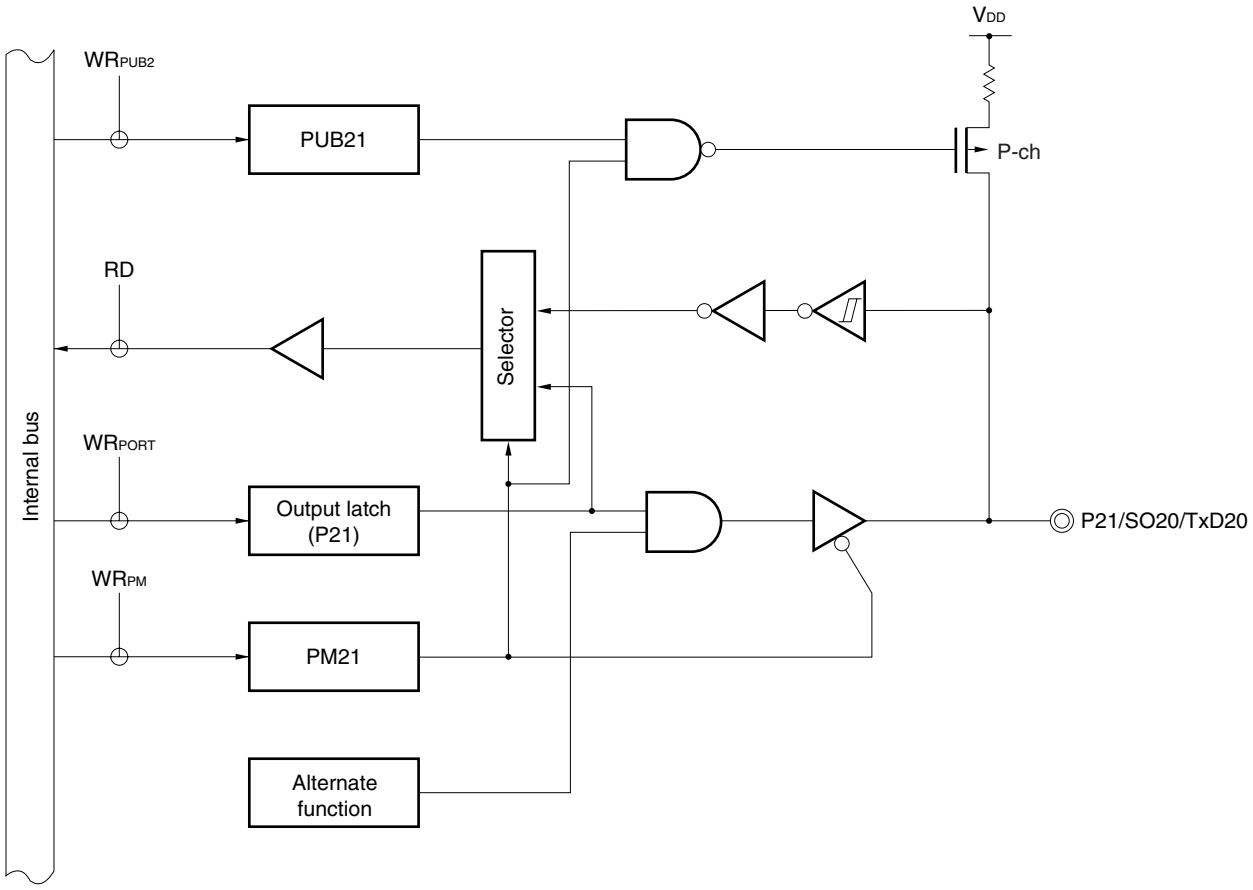
Caution When using the pins of port 2 as the serial interface, the I/O or output latch must be set according to the function to be used. For how to set the latches, see Table 11-2 Serial Interface 20 Operation Mode Settings and 12.3 (1) Serial operation mode register 1A0 (CSIM1A0).

Figure 4-4. Block Diagram of P20



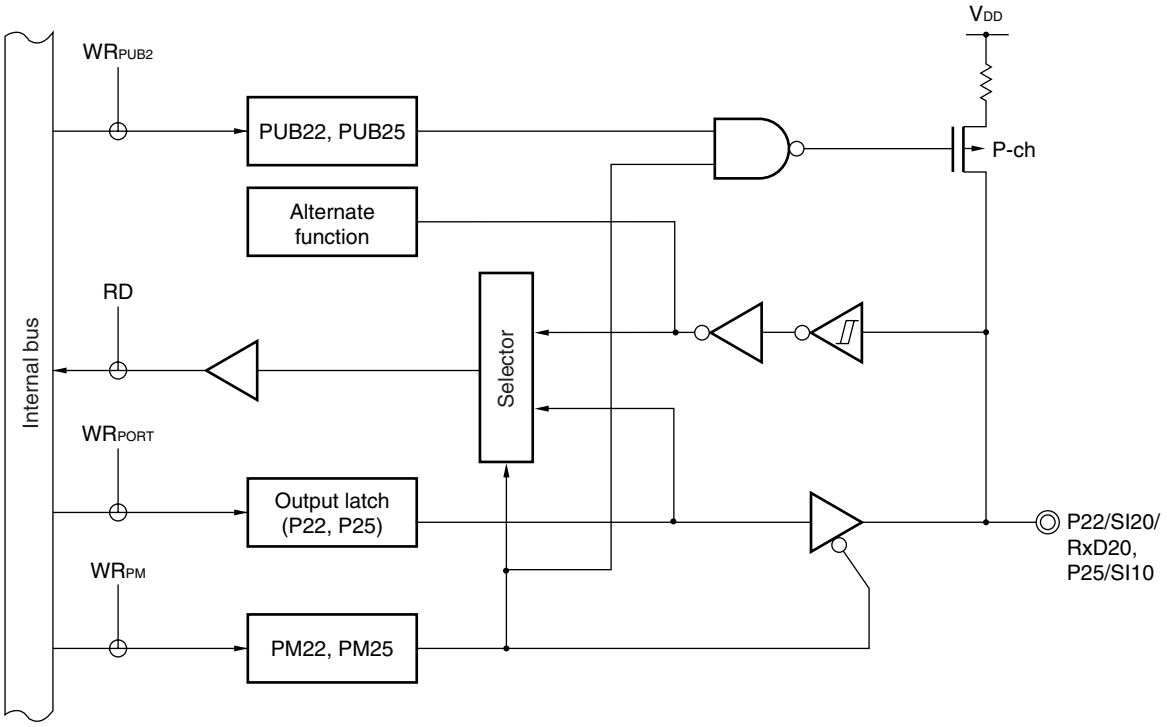
PUB2: Pull-up resistor option register B2
PM: Port mode register
RD: Port 2 read signal
WR: Port 2 write signal

Figure 4-5. Block Diagram of P21



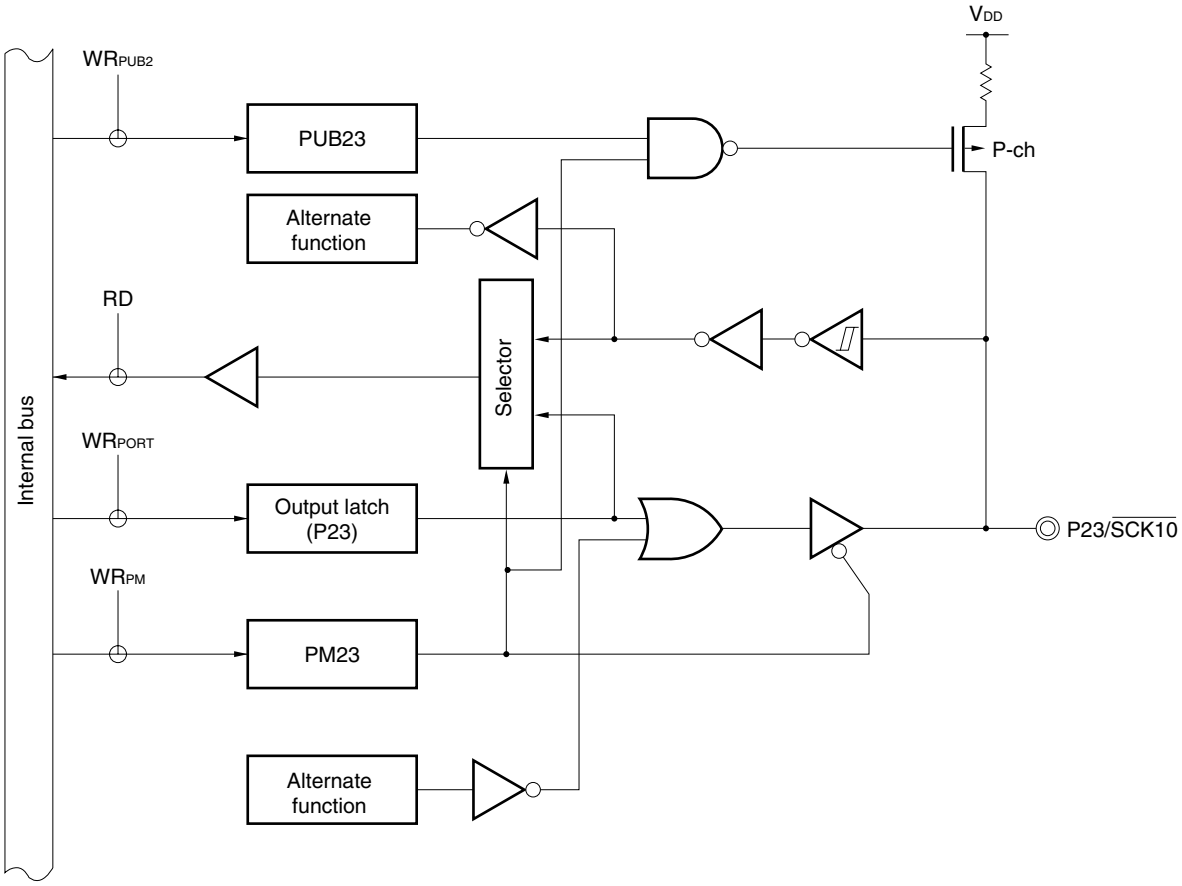
PUB2: Pull-up resistor option register B2
PM: Port mode register
RD: Port 2 read signal
WR: Port 2 write signal

Figure 4-6. Block Diagram of P22 and P25



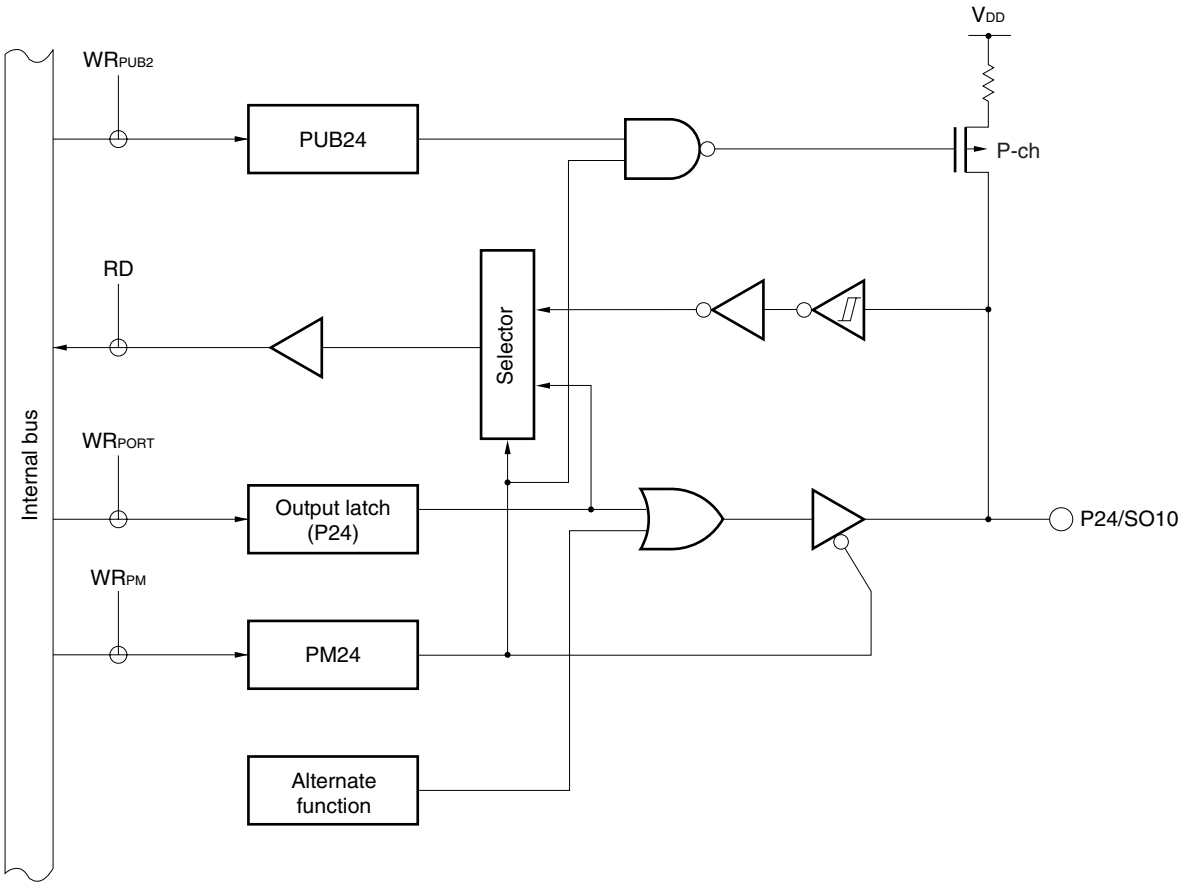
PUB2: Pull-up resistor option register B2
PM: Port mode register
RD: Port 2 read signal
WR: Port 2 write signal

Figure 4-7. Block Diagram of P23



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

Figure 4-8. Block Diagram of P24



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

4.2.4 Port 3

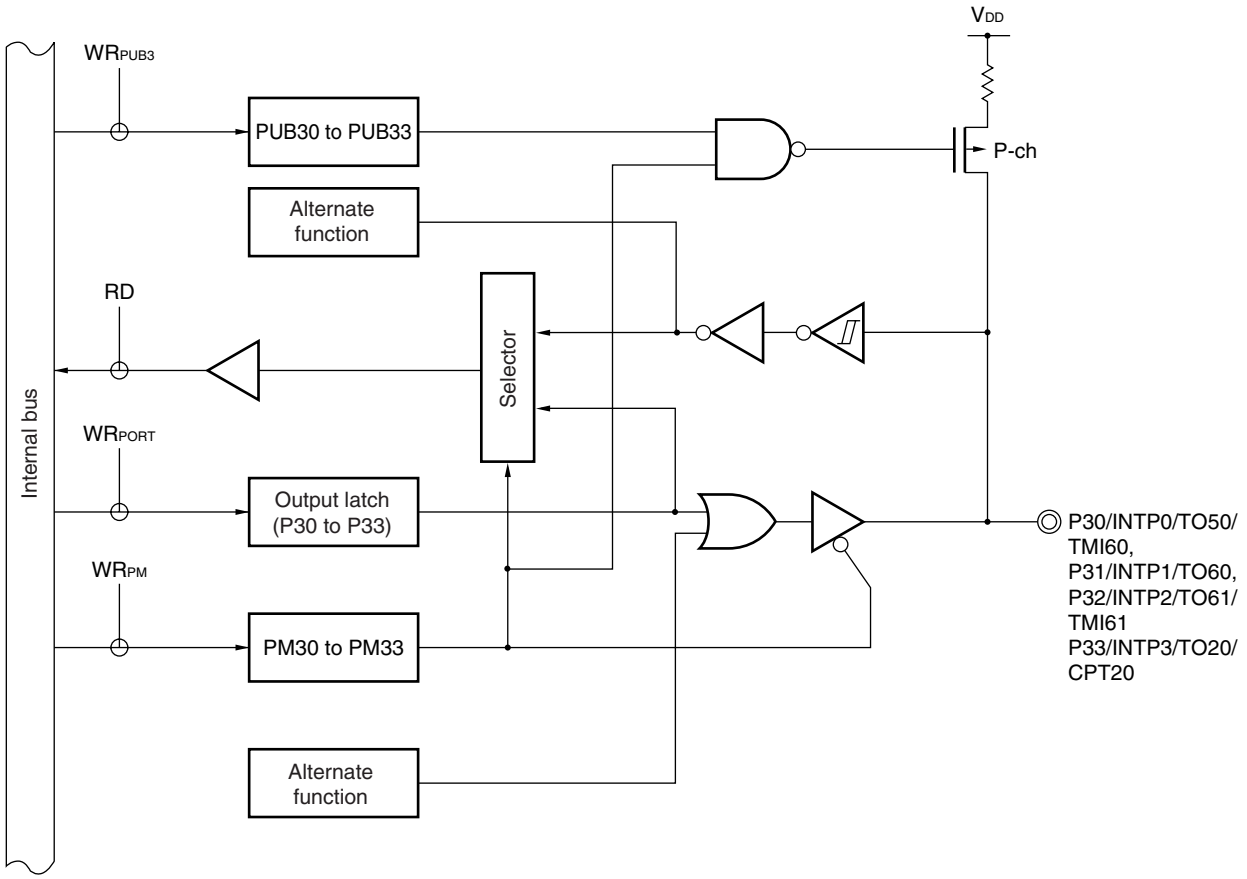
This is a 5-bit I/O port with an output latch. Port 3 can be specified in the input or output mode in 1-bit units by using port mode register 3 (PM3). When using the P30 to P34 pins as input port pins, on-chip pull-up resistors can be connected in 1-bit units by using pull-up resistor option register B3 (PUB3).

This port is also used as an external interrupt input, capture input, timer I/O, and remote control receive data input.

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

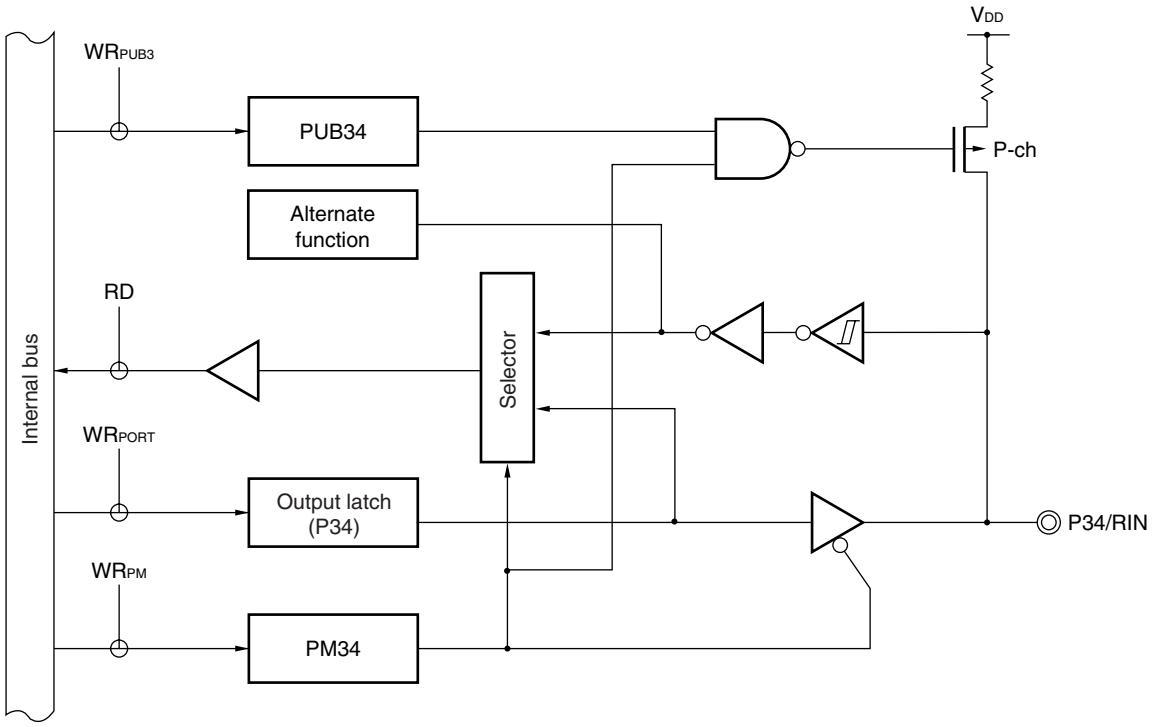
Figures 4-9 and 4-10 show block diagrams of port 3.

Figure 4-9. Block Diagram of P30 to P33



PUB3: Pull-up resistor option register B3
PM: Port mode register
RD: Port 3 read signal
WR: Port 3 write signal

Figure 4-10. Block Diagram of P34



PUB3: Pull-up resistor option register B3
PM: Port mode register
RD: Port 3 read signal
WR: Port 3 write signal

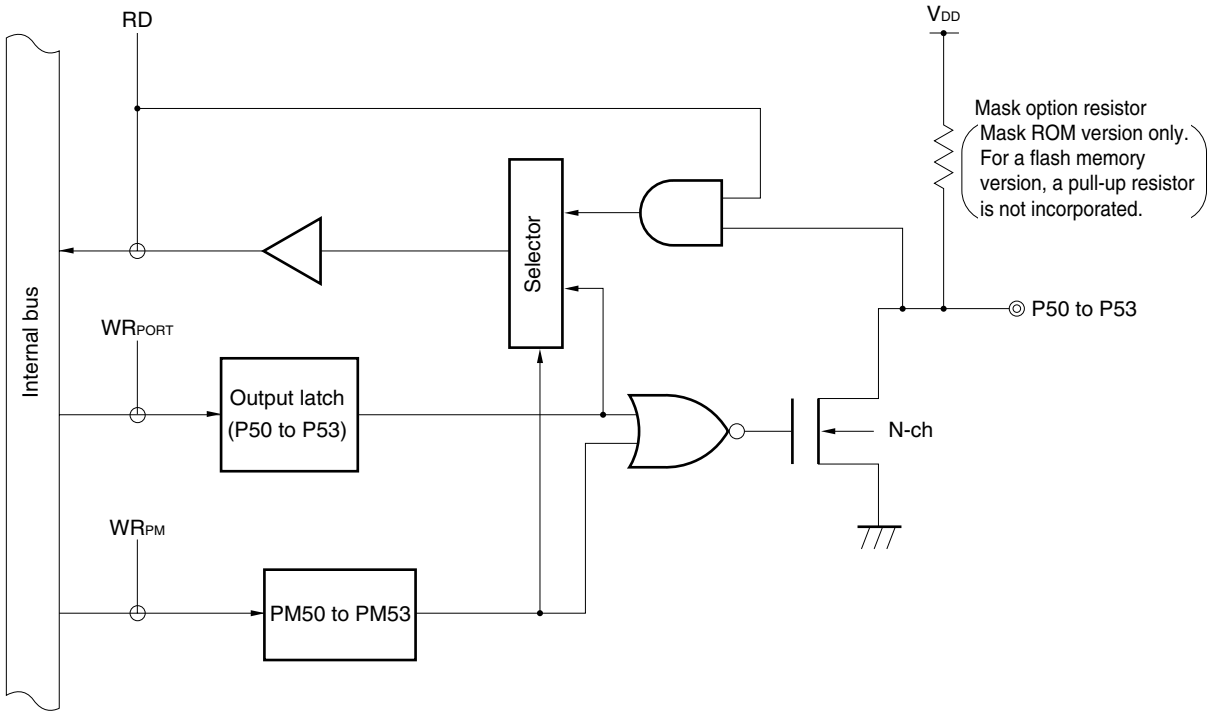
4.2.5 Port 5

This is a 4-bit N-ch open-drain I/O port with an output latch. Port 5 can be specified in the input or output mode in 1-bit units by using port mode register 5 (PM5). For a mask ROM version, use of an on-chip pull-up resistor can be specified by a mask option.

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

Figure 4-11 shows a block diagram of port 5.

Figure 4-11. Block Diagram of P50 to P53



PM: Port mode register

RD: Port 5 read signal

WR: Port 5 write signal

4.2.6 Port 6

This is an 8-bit input-only port.

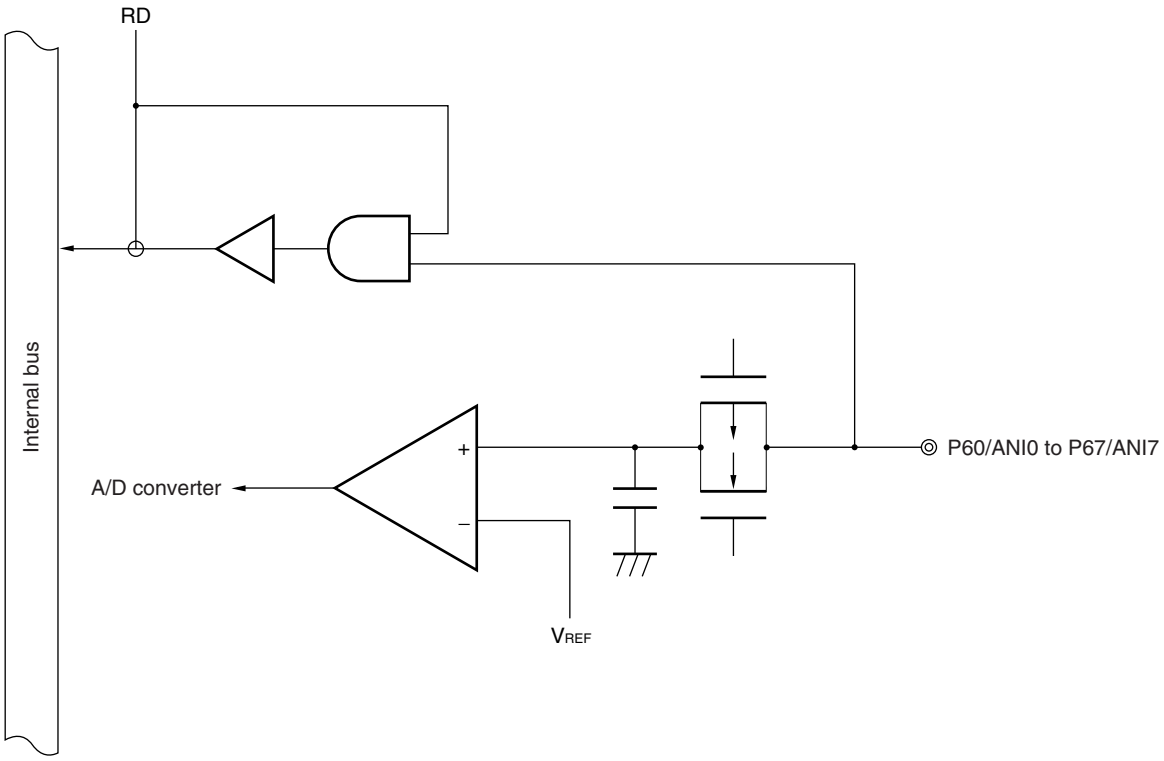
This port is also used for the analog input of an A/D converter and of key return signal input[Note].

Figure 4-12 shows a block diagram of port 6.

Note μPD789479 and 78F9479 only

Figure 4-12. Block Diagram of P60 to P67 (1/2)

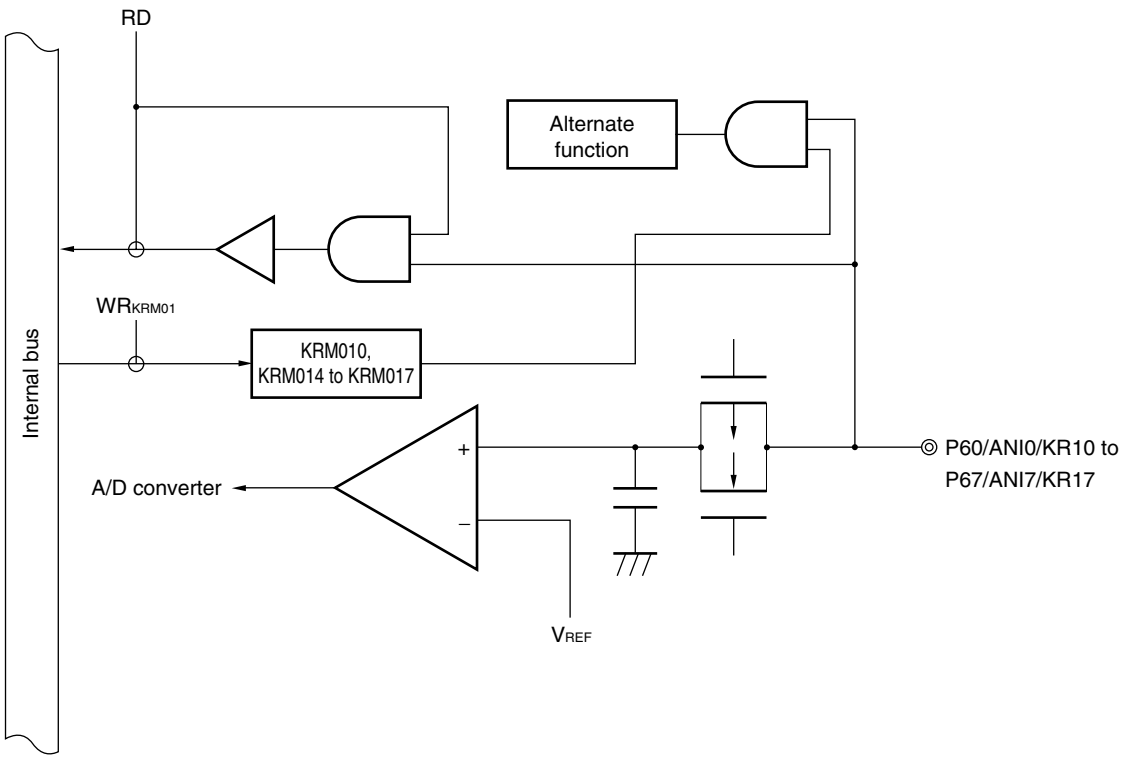
(a) μPD789477, 789478, and 78F9478



RD: Port 6 read signal

Figure 4-12. Block Diagram of P60 to P67 (2/2)

(b) μPD789479 and 78F9479



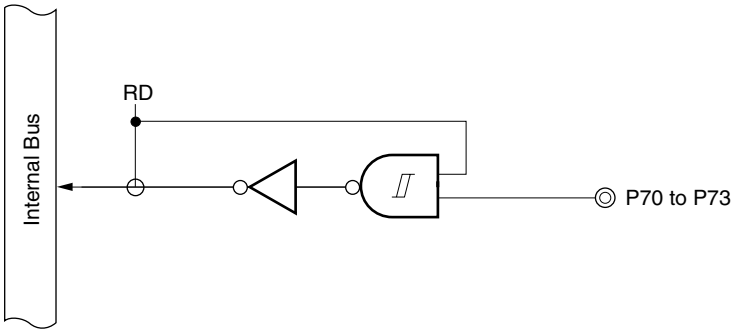
KRM01: Key return mode register 01
RD: Port 6 read signal

4.2.7 Port 7

This is a 4-bit input-only port. Only the bits for which the port function is selected can be used, by using a mask option in the μPD789477, 789478, and 789479 or port function register 7 (PF7) in the μPD78F9478 and 78F9479.

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

Figure 4-13. Block Diagram of P70 to P73



RD: Port 7 read signal

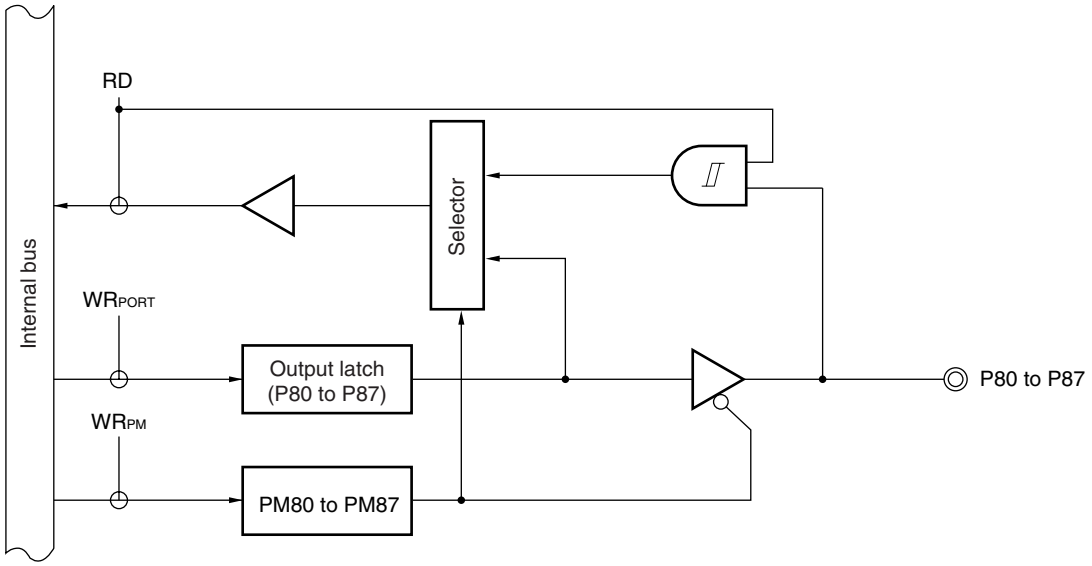
4.2.8 Port 8

This is an 8-bit I/O port with an output latch. Only the bits for which the port function is selected can be used, by using a mask option in the μPD789477, 789478, and 789479 or port function register 8 (PF8) in the μPD78F9478 and 78F9479. Port 8 can be specified in the input or output mode in 1-bit units by using port mode register 8 (PM8).

$\overline{\text{RESET}}$ input set this port to input mode.

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

Figure 4-14. Block Diagram of P80 to P87



PM: Port mode register
RD: Port 8 read signal
WR: Port 8 write signal

4.3 Registers Controlling Port Function

The ports are controlled by the following three types of registers.

- Port mode registers (PM0 to PM3, PM5, PM8)
- Pull-up resistor option registers (PUB0 to PUB3)
- Port function registers (PF7, PF8) (μPD78F9478 and 78F9479 only)

(1) Port mode registers (PM0 to PM3, PM5, PM8)

Input and output can be specified in 1-bit units.

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

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

When using the port pins as their alternate functions, set the port mode register and the output latch as shown in Table 4-3.

Caution Because P30 to P33 function alternately as external interrupt inputs, when the output level changes after the output mode of the port function is specified, the interrupt request flag will be inadvertently set. Therefore, be sure to preset the interrupt mask flag (PMK0 to PMK3) before using the port in output mode.

Figure 4-15. Port Mode Register Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00	FF20H	FFH	R/W
PM1	1	1	1	1	1	1	PM11	PM10	FF21H	FFH	R/W
PM2	1	1	PM25	PM24	PM23	PM22	PM21	PM20	FF22H	FFH	R/W
PM3	1	1	1	PM34	PM33	PM32	PM31	PM30	FF23H	FFH	R/W
PM5	1	1	1	1	PM53	PM52	PM51	PM50	FF25H	FFH	R/W
PM8	PM87	PM86	PM85	PM84	PM83	PM82	PM81	PM80	FF28H	FFH	R/W

PMmn	Pmn pin input/output mode selection (m = 0 to 3, 5, 8, n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Remark PM8 can only be used when one of pins P80 to P87 is selected as a port function pin by a mask option or port function register 8 (PF8).

Table 4-3. Port Mode Registers and Output Latch Settings When Using Alternate Functions

Pin Name	Alternate Function		PM××	P××
	Name	I/O		
P00 to P07	KR0 to KR7 or KR00 to KR07	Input	1	×
P30	INTP0	Input	1	×
	TO50	Output	0	0
	TMI60	Input	1	×
P31	INTP1	Input	1	×
	TO60	Output	0	0
P32	INTP2	Input	1	×
	TMI61	Input	1	×
	TO61	Output	0	0
P33	INTP3	Input	1	×
	CPT20	Input	1	×
	TO20	Output	0	0
P34	RIN	Input	1	×

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

Caution When port 2 is used for the serial interface, I/O and output latch settings must be made in accordance with the function used. For the setting method, refer to Table 11-2 Serial Interface 20 Operation Mode Settings and 12.3 (1) Serial operation mode register 1A0 (CSIM1A0).

(2) Pull-up resistor option registers (PUB0 to PUB3)

These registers set whether to use on-chip pull-up resistors for pins P00 to P07, P10, P11, P20 to P25, and P30 to P34. An on-chip pull-up resistor can be used only for those bits set to the input mode in a port for which the use of the on-chip pull-up resistor has been specified using PUB0 to PUB3.

For those bits set to the output mode, on-chip pull-up resistors cannot be used, regardless of the setting of PUB0 to PUB3. This also applies to alternate-function pins used as output pins.

PUB0 to PUB3 are set with a 1-bit or 8-bit memory manipulation instruction.

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

Figure 4-16. Format of Pull-Up Resistor Option Registers

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
PUB0	PUB07	PUB06	PUB05	PUB04	PUB03	PUB02	PUB01	PUB00	FF30H	00H	R/W
	7	6	5	4	3	2	<1>	<0>			
PUB1	0	0	0	0	0	0	PUB11	PUB10	FF31H	00H	R/W
	7	6	<5>	<4>	<3>	<2>	<1>	<0>			
PUB2	0	0	PUB25	PUB24	PUB23	PUB22	PUB21	PUB20	FF32H	00H	R/W
	7	6	5	<4>	<3>	<2>	<1>	<0>			
PUB3	0	0	0	PUB34	PUB33	PUB32	PUB31	PUB30	FF33H	00H	R/W

PUBmn	Pmn on-chip pull-up resistor selection (m = 0 to 3, n = 0 to 7)
0	An on-chip pull-up resistor is not connected.
1	An on-chip pull-up resistor is connected.

(3) Port function registers (PF7 and PF8) (μPD78F9478 and 78F9479 only)

These registers specify in 1-bit units whether to use P70 to P73 and P80 to P87 as ports or segment outputs.

PF7 and PF8 are set with 8-bit memory manipulation instruction.

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

Caution This register is valid only in the μPD78F9478 and 78F9479; however, writing to it in the μPD789477, 789478, and 789479 will simply make it invalid, causing no operational effect.

Figure 4-17. Port Function Register Format

Symbol	7	6	5	4	<3>	<2>	<1>	<0>	Address	After reset	R/W
PF7	0	0	0	0	PF73	PF72	PF71	PF70	FF57H	00H	W
	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>			
PF8	PF87	PF86	PF85	PF84	PF83	PF82	PF81	PF80	FF58H	00H	W

PFmn	Pmn port/segment output specification (m = 7 or 8, n = 0 to 7)
0	Pmn is used as a port pin.
1	Pmn is used as a segment output.

4.4 Port Function Operation

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

4.4.1 Writing to I/O port

(1) In output mode

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

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

(2) In input mode

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

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

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

4.4.2 Reading from I/O port

(1) In output mode

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

(2) In input mode

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

4.4.3 Arithmetic operation of I/O port

(1) In output mode

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

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

(2) In input mode

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

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

CHAPTER 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 used.

- **Main system clock oscillator**
  This circuit oscillates at 1.0 to 5.0 MHz. Oscillation can be stopped by executing the STOP instruction or setting the processor clock control register (PCC).
- **Subsystem clock oscillator**
  This circuit oscillates at 32.768 kHz. Oscillation can be stopped by the suboscillation mode register (SCKM). Also, a circuit to multiply the subsystem clock by 4 can be used by setting a mask option or the subclock selection register (SSCK).

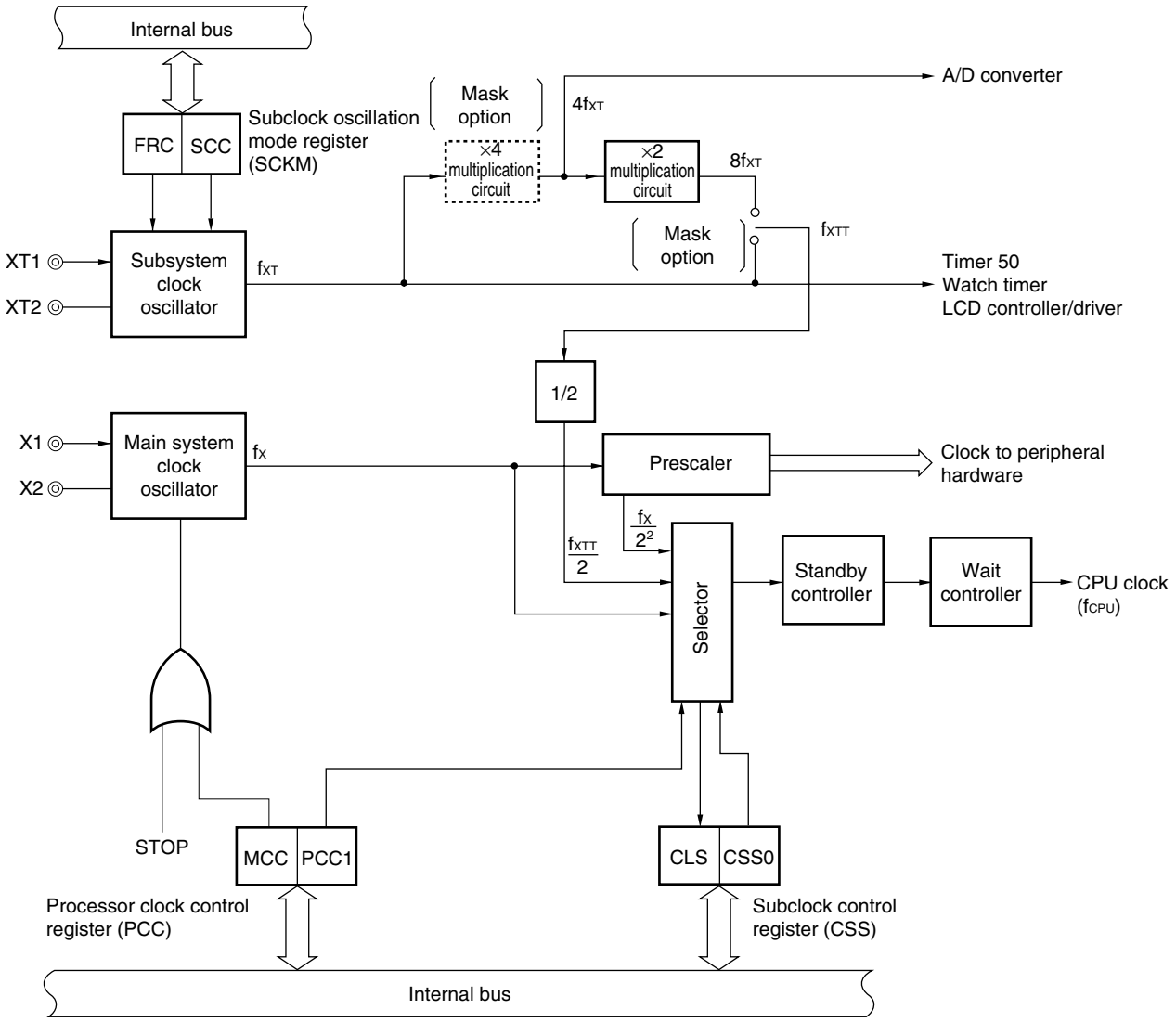
5.2 Clock Generator Configuration

The clock generator includes the following hardware.

Table 5-1. Configuration of Clock Generator

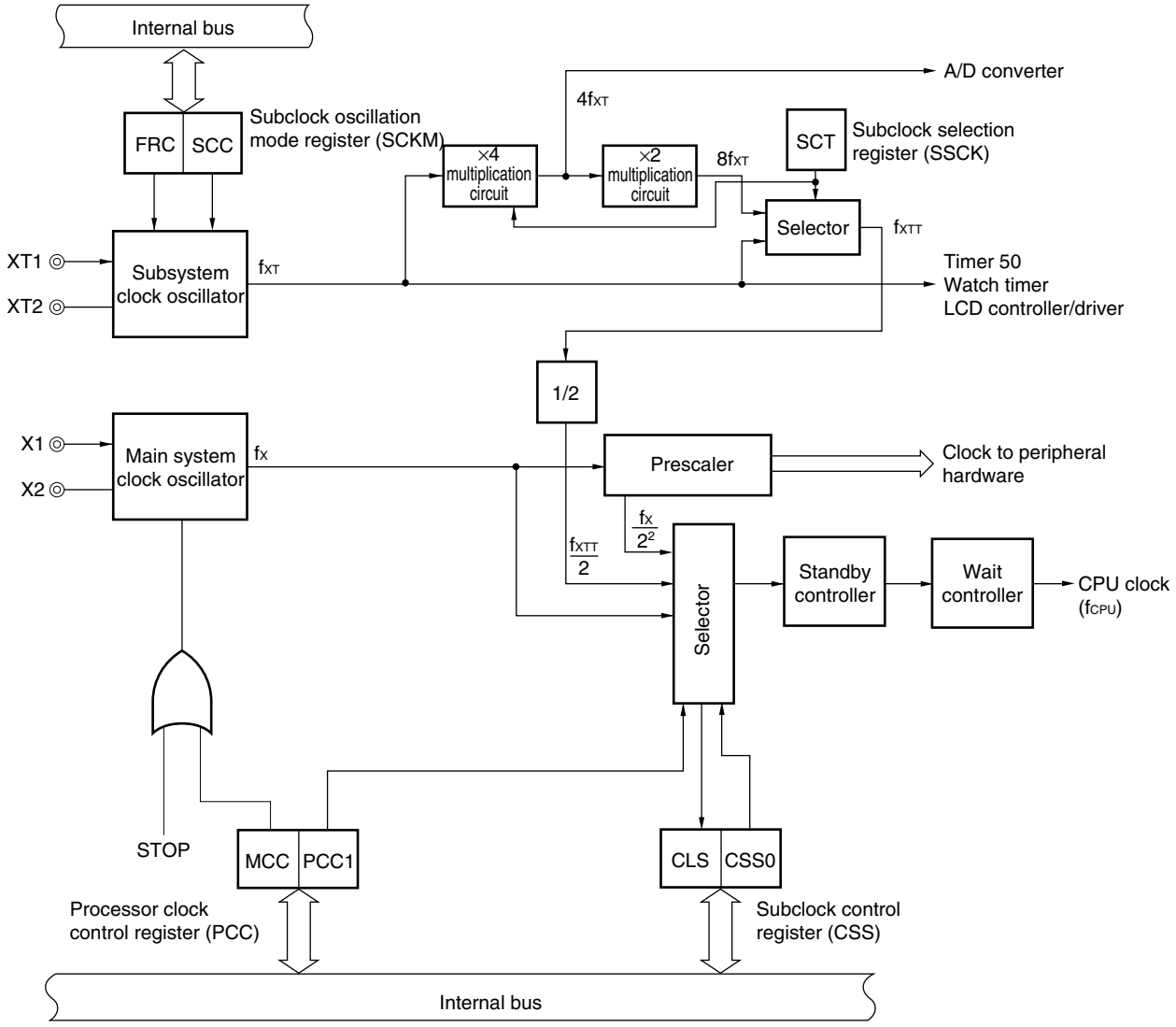
Item	Configuration
Control registers	Processor clock control register (PCC) Subclock oscillation mode register (SCKM) Subclock control register (CSS) Subclock selection register (SSCK) (μPD78F9478 and 78F9479 only)
Oscillators	Main system clock oscillator Subsystem clock oscillator

Figure 5-1. Clock Generator Block Diagram (μPD789477, 789478, and 789479)



Remark f_{XTT}: f_{XT} or $8f_{XT}$

Figure 5-2. Clock Generator Block Diagram (μPD78F9478, 78F9479)



Remark f_{XTT}: f_{XT} or $8f_{XT}$

5.3 Registers Controlling Clock Generator

The clock generator is controlled by the following four registers.

- Processor clock control register (PCC)
- Subclock oscillation mode register (SCKM)
- Subclock control register (CSS)
- Subclock selection register (SSCK) (μPD78F9478 and 78F9479 only)

(1) Processor clock control register (PCC)

This register is used to select the CPU clock and set the frequency division ratio.

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

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

Figure 5-3. Format of Processor Clock Control Register

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

MCC	Main system clock oscillator operation control
0	Operation enabled
1	Operation stopped

CSS0	PCC1	CPU clock (f_{CPU}) selection[Note]	Minimum instruction execution time: $2/f_{CPU}$
			f_X = 5.0 MHz or f_{XT} = 32.768 kHz
0	0	f_X	0.4 μs
0	1	$f_X/2^2$	1.6 μs
1	×	$f_{XT}/2$ $4f_{XT}$ (when ×4 multiplication circuit is used)	122 μs 15.26 μs (when ×4 multiplication circuit is used)

Note The CPU clock is selected by a combination of flag settings in the PCC and CSS registers. (Refer to **5.3 (3) Subclock control register (CSS)**.)

Cautions 1. Always set bits 0 and 2 to 6 to 0.

2. MCC can be set only when the subsystem clock is selected as the CPU clock. Setting MCC to 1 while the main system clock is operating is invalid.

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

(2) Subclock oscillation mode register (SCKM)

SCKM selects a feedback resistor for the subsystem clock, and controls the oscillation of the clock.

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

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

Figure 5-4. Format of Subclock Oscillation Mode Register

Symbol	7	6	5	4	3	2	1	<0>	Address	After reset	R/W
SCKM	0	0	0	0	0	0	FRC	SCC	FFF0H	00H	R/W

FRC	Feedback resistor selection[Note]
0	On-chip feedback resistor used
1	On-chip feedback resistor not used

SCC	Control of subsystem clock oscillator operation
0	Operation enabled
1	Operation disabled

Note The feedback resistor is necessary to adjust the bias point of the oscillation waveform to close to the mid point of the supply voltage. When only the subclock is not used, the power consumption in STOP mode can be further reduced by setting FRC = 1.

Caution Bits 2 to 7 must be set to 0.

(3) Subclock control register (CSS)

CSS specifies whether the main system or subsystem clock oscillator is to be selected. It also specifies the CPU clock operation status.

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

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

Figure 5-5. Format of Subclock Control Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
CSS	0	0	CLS	CSS0	0	0	0	0	FFF2H	00H	R/W[Note]

CLS	CPU clock operation status
0	Operation based on the output of the (divided) main system clock
1	Operation based on the subsystem clock

CSS0	Selection of the main system or subsystem clock oscillator
0	(Divided) output from the main system clock oscillator
1	Output from the subsystem clock oscillator

Note Bit 5 is read only.

Caution Bits 0 to 3, 6, and 7 must be set to 0.

(4) Subclock selection register (SSCK) (μPD78F9478 and 78F9479 only)

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

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

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

Caution This register is valid only in the μPD78F9478 and 78F9479; however, writing to it in the μPD789477, 789478, and 789479 will simply make it invalid, causing no operational effect.

Figure 5-6. Format of Subclock Selection Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
SSCK	0	0	0	0	0	0	0	SCT	FF46H	Retained[Note]	R/W

SCT	Control of ×4 subsystem clock multiplication circuit
0	Operation disabled (subsystem clock source (32.768 kHz) supplied to the CPU)
1	Operation enabled (clock that is the subsystem clock multiplied by 8 (262 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 CSS0 flag to 1 following the release of reset. Write operations following the first operation are invalid (input the $\overline{\text{RESET}}$ signal to rewrite).

5.4 System Clock Oscillators

5.4.1 Main system clock oscillator

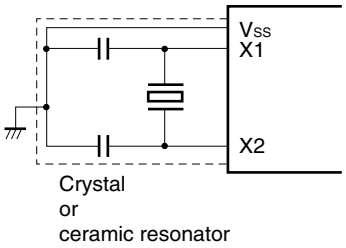
The main system clock oscillator is oscillated by the crystal or ceramic resonator (5.0 MHz TYP.) connected across the X1 and X2 pins.

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

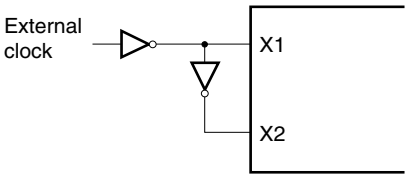
Figure 5-7 shows the external circuit of the main system clock oscillator.

Figure 5-7. External Circuit of Main System Clock Oscillator

(a) Crystal or ceramic oscillation



(b) External clock



Caution When using the main system clock oscillator, wire as follows in the area enclosed by the broken lines in Figure 5-7 to avoid an adverse effect from wiring capacitance.

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

5.4.2 Subsystem clock oscillator

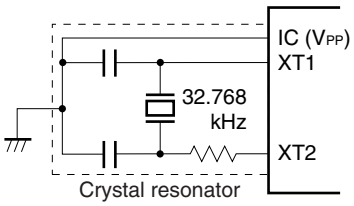
The subsystem clock oscillator is oscillated by the crystal resonator (32.768 kHz TYP.) connected across the XT1 and XT2 pins.

An external clock can also be input to the circuit. In this case, input the clock signal to the XT1 pin, and input the inverted signal to the XT2 pin.

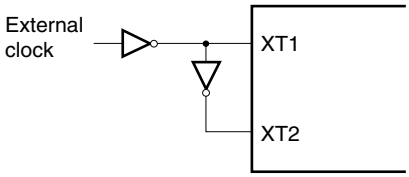
Figure 5-8 shows the external circuit of the subsystem clock oscillator.

Figure 5-8. External Circuit of Subsystem Clock Oscillator

(a) Crystal oscillation



(b) External clock



Caution When using the main system or subsystem clock oscillator, wire as follows in the area enclosed by the broken lines in Figures 5-7 and 5-8 to avoid an adverse effect from wiring capacitance.

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

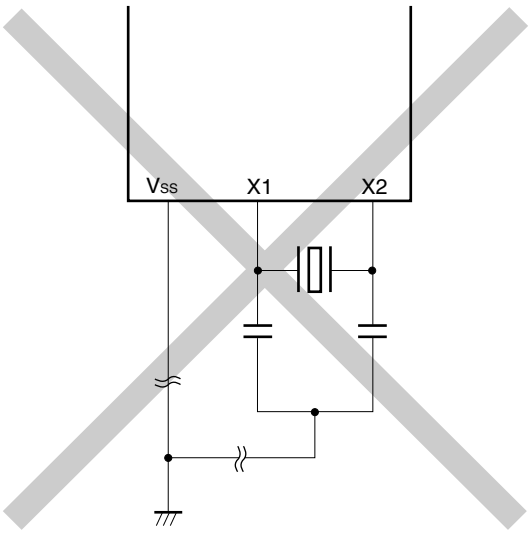
When using the subsystem clock, particular care is required because the subsystem clock oscillator is designed as a low-amplitude circuit for reducing power consumption.

5.4.3 Example of incorrect resonator connection

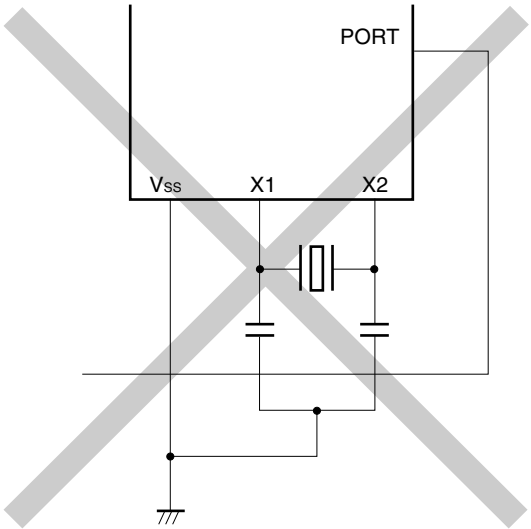
Figure 5-9 shows examples of incorrect resonator connection.

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

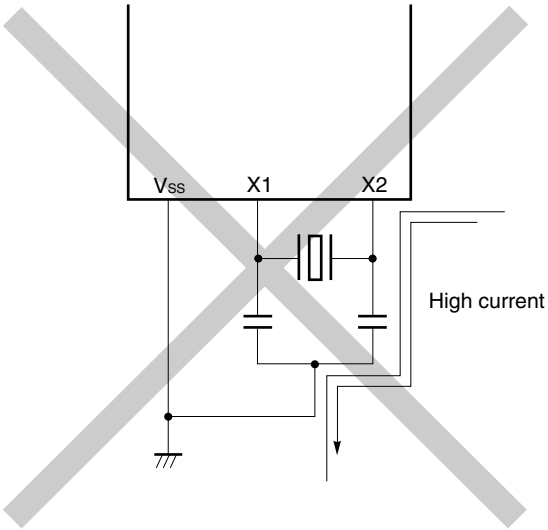
(a) Too long wiring



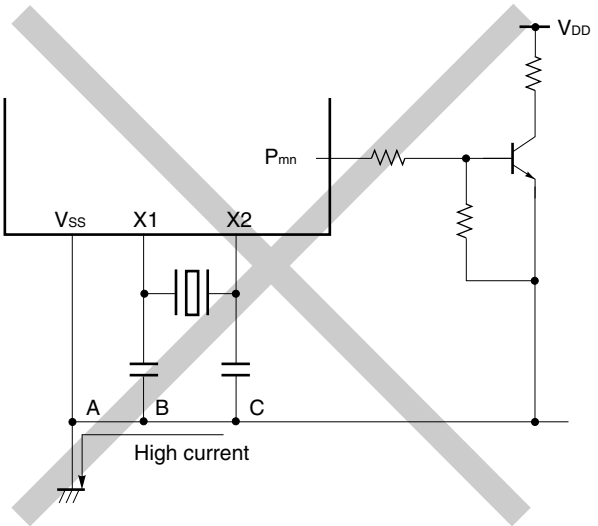
(b) Crossed signal line



(c) Wiring near high fluctuating current



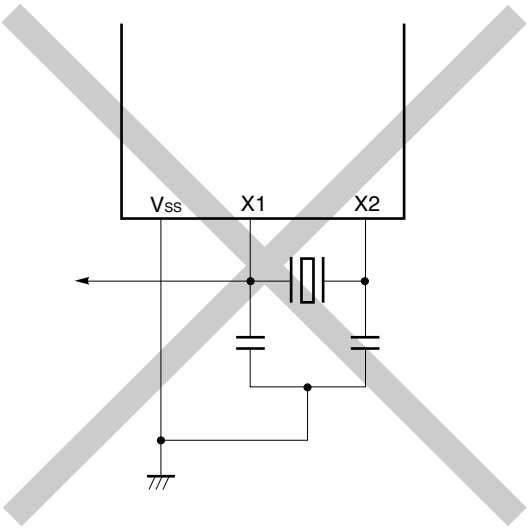
(d) Current flowing through ground line of oscillator (potential at points A, B, and C fluctuates)



Remark When using the subsystem clock, read X1 and X2 as XT1 and XT2, respectively, and connect a resistor to XT2 in series.

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

(e) Signal is fetched



Remark When using the subsystem clock, read X1 and X2 as XT1 and XT2, respectively, and connect a resistor to XT2 in series.

5.4.4 Divider circuit

The divider circuit divides the output of the main system clock oscillator (f_X) to generate various clocks.

5.4.5 When subsystem clock is not used

If the subsystem clock is not necessary, for example, for low-power consumption operation or clock operation, handle the XT1 and XT2 pins as follows.

XT1: Connect to V_{SS}
XT2: Leave open

In this case, however, a small current leaks via the on-chip feedback resistor in the subsystem clock oscillator when the main system clock is stopped. To avoid this, set bit 1 (FRC) of the subclock oscillation mode register (SCKM) so that the on-chip feedback resistor will not be used. Also in this case, handle the XT1 and XT2 pins as stated above.

5.4.6. Subsystem clock × 4 multiplication circuit

This circuit multiplies the subsystem clock by 4 and supplies it to the CPU. The circuit stops operations in the HALT mode (to reduce power consumption). When the circuit starts operating after the HALT mode is released, a one-clock wait of the original subsystem clock is inserted to eliminate noise.

5.5 Clock Generator Operation

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

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

The operation and function of the clock generator is determined by the processor clock control register (PCC), subclock oscillation mode register (SCKM), and subclock control register (CSS), as follows.

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

(b) Three types of minimum instruction execution time (0.4 μs and 1.6 μs: main system clock (at 5.0 MHz operation), 122 μs: subsystem clock (at 32.768 kHz operation)) can be selected by the PCC, SCKM, and CSS settings. Also, the subsystem clock can be changed to a clock that uses a circuit to multiply the subclock by 4 via a mask option in the μPD789477, 789478, and 789479 or the subclock selection register (SSCK) in the μPD78F9478 and 78F9479 (15.26 μs: a circuit to multiply the subsystem clock by 4 is used).

(c) Two standby modes, STOP and HALT, can be used with the main system clock selected. In a system where the subsystem clock is not used, setting bit 1 (FRC) of SCKM so that the on-chip feedback resistor cannot be used reduces power consumption in STOP mode. In a system where the subsystem clock is used, setting SCKM bit 0 to 1 can cause the subsystem clock to stop oscillation.

(d) CSS bit 4 (CSS0) can be used to select the subsystem clock so that low power consumption operation is used (122 μs: at 32.768 kHz operation).

(e) With the subsystem clock selected, it is possible to cause the main system clock to stop oscillating using bit 7 (MCC) of PCC. The HALT mode can be used, but the STOP mode cannot.

(f) The clock pulse for the peripheral hardware is generated by dividing the frequency of the main system clock, but the subsystem clock pulse is only supplied to 8-bit timer 50, the watch timer, and the LCD controller/driver. 8-bit timer 50, the watch timer, and the LCD controller/driver can therefore keep running even during standby. The other hardware stops when the main system clock stops because it runs based on the main system clock (except for external input clock operations). Because the subsystem clock pulse is supplied to the A/D converter via the $\times$4 multiplication circuit, the A/D converter cannot be used during standby.

5.6 Changing Setting of System Clock and CPU Clock

5.6.1 Time required for switching between system clock and CPU clock

The CPU clock can be selected by using bit 1 (PCC1) of the processor clock control register (PCC) and bit 4 (CSS0) of the subclock control register (CSS).

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

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

Set Value Before Switching		Set Value After Switching					
CSS0	PCC1	CSS0	PCC1	CSS0	PCC1	CSS0	PCC1
		0	0	0	1	1	×
0	0			4 clocks		$2f_X/f_{XT}$ clocks (306 clocks)	
	1	2 clocks				$f_X/2f_{XT}$ clocks (76 clocks)	
1	×	2 clocks		2 clocks			

Remarks 1. Two clocks are the minimum instruction execution time of the CPU clock before switching.

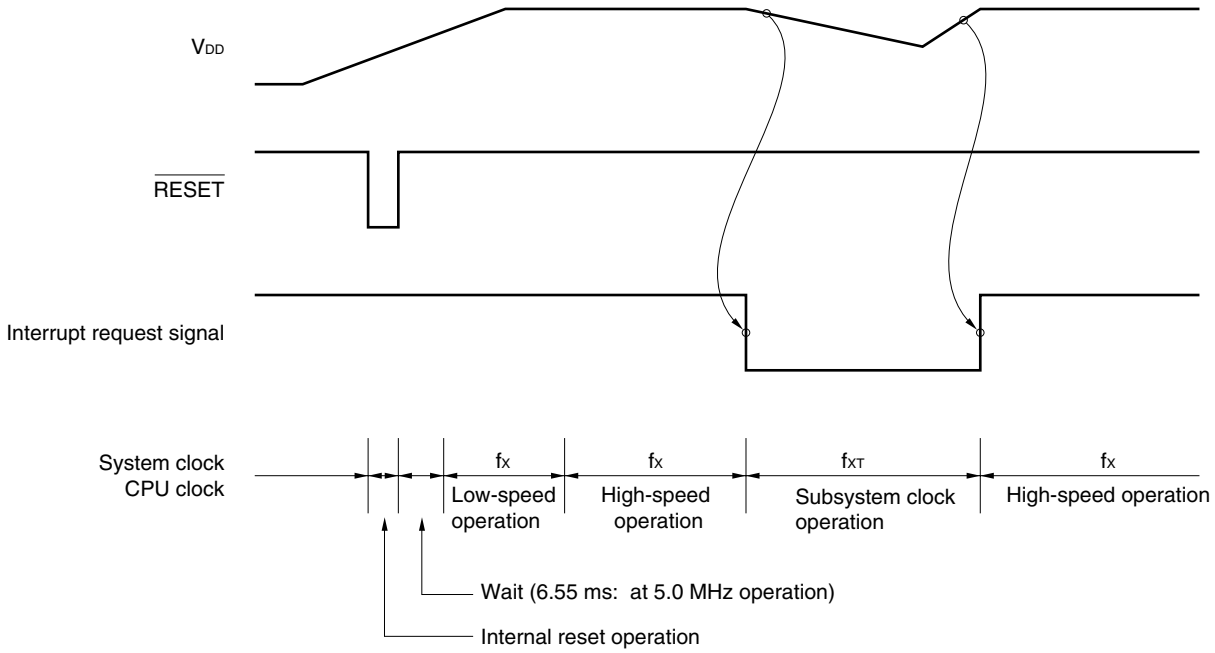
2. The parenthesized values apply to operation at f_X = 5.0 MHz or f_{XT} = 32.768 kHz.

3. ×: don't care

5.6.2 Switching between system clock and CPU clock

The following figure illustrates how the CPU clock and system clock switch.

Figure 5-10. Switching Between System Clock and CPU Clock



<1> The CPU is reset when the $\overline{\text{RESET}}$ pin is made low on power application. The effect of resetting is released when the $\overline{\text{RESET}}$ pin is later made high, and the main system clock starts oscillating. At this time, the oscillation stabilization time (2^{15}/fx) is automatically secured.
After that, the CPU starts instruction execution at the slow speed of the main system clock (1.6 μs: at 5.0 MHz operation).

<2> After the time required for the V_{DD} voltage to rise to the level at which the CPU can operate at high speed has elapsed, bit 1 (PCC1) of the processor clock control register (PCC) and bit 4 (CSS0) of the subclock control register (CSS) are rewritten so that high-speed operation can be selected.

<3> A drop of the V_{DD} voltage is detected with an interrupt request signal. The clock is switched to the subsystem clock (at this moment, the subsystem clock must be in the oscillation stabilization status).

<4> A recover of the V_{DD} voltage is detected with an interrupt request signal. Bit 7 (MCC) of PCC is set to 0, and then the main system clock starts oscillating. After the time required for the oscillation to stabilize has elapsed, PCC1 and CSS0 are rewritten so that high-speed operation can be selected again.

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 20

6.1 16-Bit Timer 20 Functions

16-bit timer 20 has the following functions.

- Timer interrupt
- Timer output
- Count value capture

(1) Timer interrupt

An interrupt is generated when a count value and compare value match.

(2) Timer output

Timer output can be controlled when a count value and compare value match.

(3) Count value capture

The count value of 16-bit timer counter 20 (TM20) is latched into a capture register in synchronization with the capture trigger and retained.

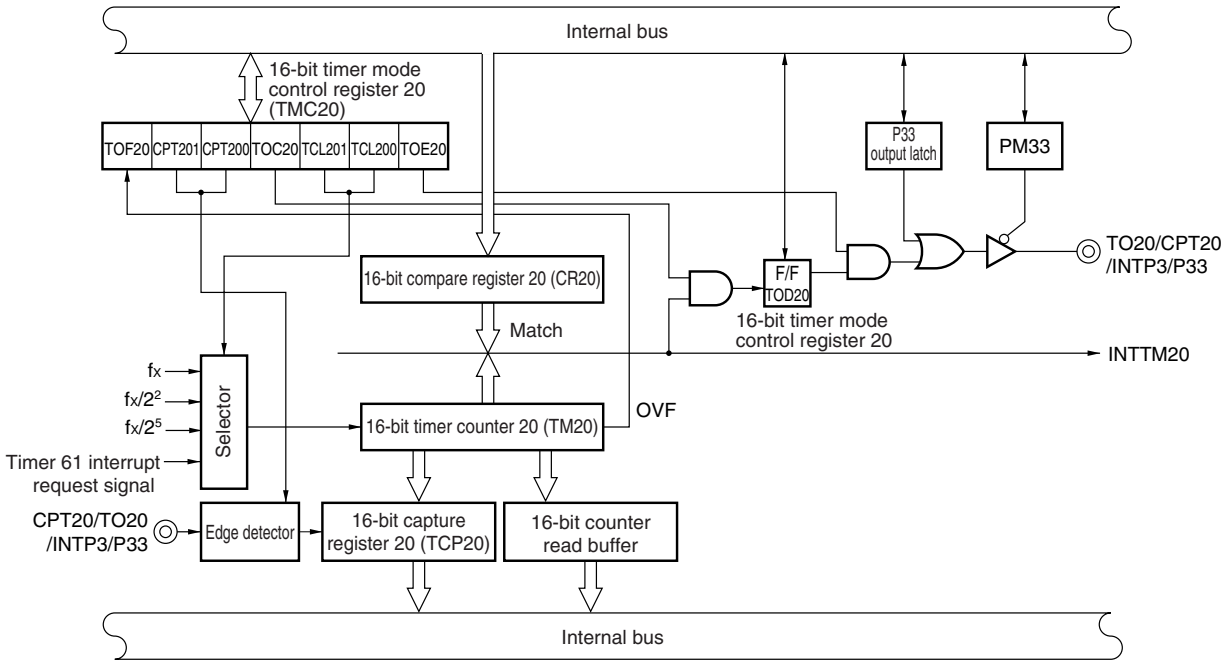
6.2 16-Bit Timer 20 Configuration

16-bit timer 20 includes the following hardware.

Table 6-1. 16-Bit Timer 20 Configuration

Item	Configuration
Timer counters	16 bits × 1 (TM20)
Registers	Compare register: 16 bits × 1 (CR20) Capture register: 16 bits × 1 (TCP20)
Timer outputs	1 (TO20)
Control registers	16-bit timer mode control register 20 (TMC20) Port mode register 3 (PM3) Port 3 (P3)

Figure 6-1. Block Diagram of 16-Bit Timer 20



(1) 16-bit compare register 20 (CR20)

This 16-bit register is used to continually compare the value set to CR20 with the count value in 16-bit timer counter 20 (TM20) and to issue an interrupt request (INTTM20) when a match occurs.

CR20 is set with a 16-bit memory manipulation instruction. Values from 0000H to FFFFH can be set.

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

Caution To rewrite CR20 during a count operation, first disable interrupts by setting interrupt mask flag register 0 (MK0). Also, set inversion inhibited for the timer output data in 16-bit timer mode control register 20 (TMC20). If the value in CR20 is rewritten in the interrupt-enabled state, an interrupt request may occur at the moment of rewrite.

(2) 16-bit timer counter 20 (TM20)

This is a 16-bit register that is used to count the count pulses.

TM20 can be read with a 16-bit memory manipulation instruction.

The counter is in free-running mode when the count clock is being input.

$\overline{\text{RESET}}$ input sets this counter to 0000H and restarts free-running mode.

Caution The count value after releasing STOP mode is undefined because the count operation occurred during the oscillation stabilization time.

(3) 16-bit capture register 20 (TCP20)

This is a 16-bit register used to capture the contents of 16-bit timer counter 20 (TM20).

TCP20 is set with a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes this register undefined.

(4) 16-bit counter read buffer 20

This buffer is used to latch and hold the count value for TM20.

6.3 Registers Controlling 16-Bit Timer 20

16-bit timer 20 is controlled by the following three registers.

- 16-bit timer mode control register 20 (TMC20)
- Port mode register 3 (PM3)
- Port 3 (P3)

(1) 16-bit timer mode control register 20 (TMC20)

16-bit timer mode control register 20 (TMC20) controls the setting of the count clock, capture edge, etc.

TMC20 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TMC20 to 00H.

Figure 6-2. Format of 16-Bit Timer Mode Control Register 20

Symbol	<7>	<6>	5	4	3	2	1	<0>	Address	After reset	R/W
TMC20	TOD20	TOF20	CPT201	CPT200	TOC20	TCL201	TCL200	TOE20	FF48H	00H	R/W Note 1

TOD20	Timer output data
0	Timer output is “0”
1	Timer output is “1”

TOF20	Set overflow flag
0	Reset and clear by software
1	Set by overflow of 16-bit timer

CPT201	CPT200	Selection of capture edge
0	0	Capture operation disabled
0	1	Rising edge of CPT20 pin
1	0	Falling edge of CPT20 pin
1	1	Both edges of CPT20 pin

TOC20	Timer output data inversion control
0	Inversion disabled
1	Inversion enabled

TCL201	TCL200	Selection of count clock for 16-bit timer counter 20
0	0	Timer 61 interrupt signal
0	1	f_X (5.0 MHz) Notes 2, 3
1	0	$f_X/2^2$ (1.25 MHz) Note 4
1	1	$f_X/2^5$ (156.25 kHz) Note 4

TOE20	Output control for 16-bit timer counter 20
0	Output disabled (port mode)
1	Output enabled

Notes 1. Bit 7 is read-only.

2. If f_X is selected for the count clock, the signal cannot be used as a capture signal.

3. In a read operation, set the CPU clock as the high-speed main clock (PCC1 = 0, CSS = 0).

4. In a read operation, set the CPU clock as the main clock (CSS = 0).

Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 5.0 MHz.

(2) Port mode register 3 (PM3)

This register is used to set the I/O mode of port 3 in 1-bit units.

When using the P33/INTP3/CPT20/TO20 pin as a capture input (CPT20), set PM33 to 1. When using the above pin as a timer output (TO20), set the PM33 and P33 output latches to 0.

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

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

Figure 6-3. Format of Port Mode Register 3

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM3	1	1	1	PM34	PM33	PM32	PM31	PM30	FF23H	FFH	R/W

PM33	Selection of P33 pin I/O mode
0	Output mode (output buffer is on)
1	Input mode (output buffer is off)

6.4 16-Bit Timer 20 Operation

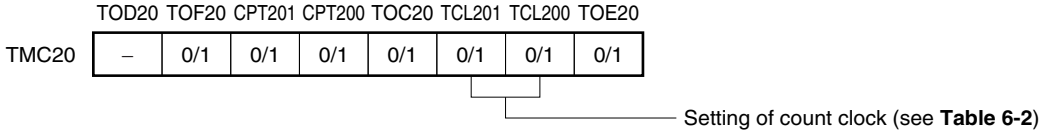
6.4.1 Operation as timer interrupt

16-bit timer 20 can generate interrupts repeatedly each time the free-running counter value reaches the value set to CR20. Since this counter is not cleared and holds the count even after an interrupt is generated, the interval time is equal to one cycle of the count clock set in TCL201 and TCL200.

To operate 16-bit timer 20 as a timer interrupt, the following settings are required.

- Set count values in CR20
- Set 16-bit timer mode control register 20 (TMC20) as shown in Figure 6-4.

Figure 6-4. Settings of 16-Bit Timer Mode Control Register 20 for Timer Interrupt Operation



Caution If both the CPT201 and CPT200 flags are set to 0, the capture edge operation is prohibited.

When the count value of 16-bit timer counter 20 (TM20) matches the value set in CR20, counting of TM20 continues and an interrupt request signal (INTTM20) is generated.

Table 6-2 shows interval time, and Figure 6-5 shows timing of timer interrupt operation.

Caution When rewriting the value in CR20 during a count operation, be sure to execute the following processing.

<1> Disable interrupts (set TMMK20 (bit 2 of interrupt mask flag register 1 (MK1)) to 1).
<2> Disable inversion control of timer output data (set TOC20 to 0)

If the value in CR20 is rewritten in the interrupt-enabled state, an interrupt request may occur at the moment of rewrite.

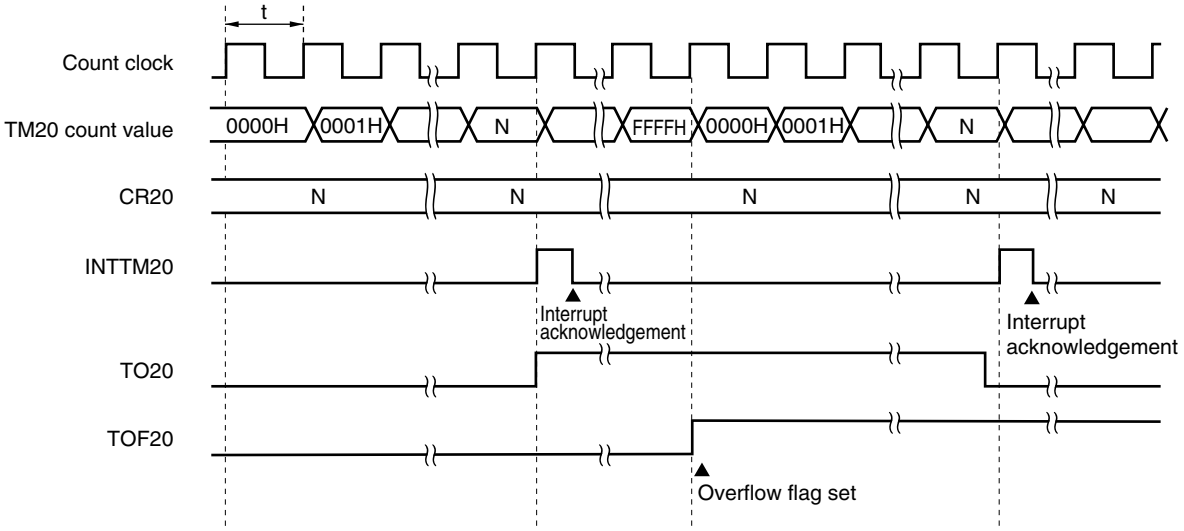
Table 6-2. Interval Time of 16-Bit Timer 20

TCL201	TCL200	Count Clock	Interval Time
0	0	Timer 61 interrupt signal	Cycle of timer 61 interrupt signal $\times 2^{16}$
0	1	$1/f_X$ (0.2 μs)	$2^{16}/f_X$ (13.1 ms)
1	0	$2^2/f_X$ (0.8 μs)	$2^{18}/f_X$ (52.4 ms)
1	1	$2^5/f_X$ (6.4 μs)	$2^{21}/f_{XT}$ (419 ms)

Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 5.0 MHz.

Figure 6-5. Timing of Timer Interrupt Operation



Remark N = 0000H to FFFFH

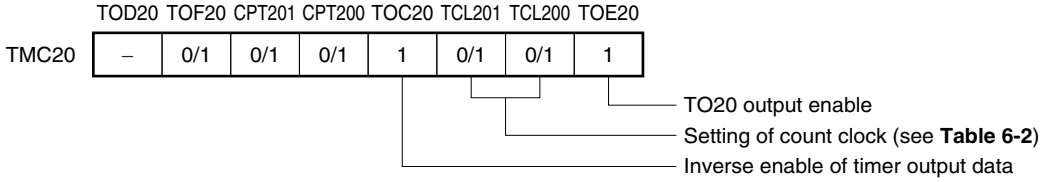
6.4.2 Operation as timer output

16-bit timer 20 can invert the timer output repeatedly each time the free-running counter value reaches the value set to CR20. Since this counter is not cleared and holds the count even after the timer output is inverted, the interval time is equal to one cycle of the count clock set in TCL201 and TCL200.

To operate 16-bit timer 20 as a timer output, the following settings are required.

- Set P33 to output mode (PM33 = 0).
- Reset the output latch of P33 to 0.
- Set the count value in CR20.
- Set 16-bit timer mode control register 20 (TMC20) as shown in Figure 6-6.

Figure 6-6. Settings of 16-Bit Timer Mode Control Register 20 for Timer Output Operation

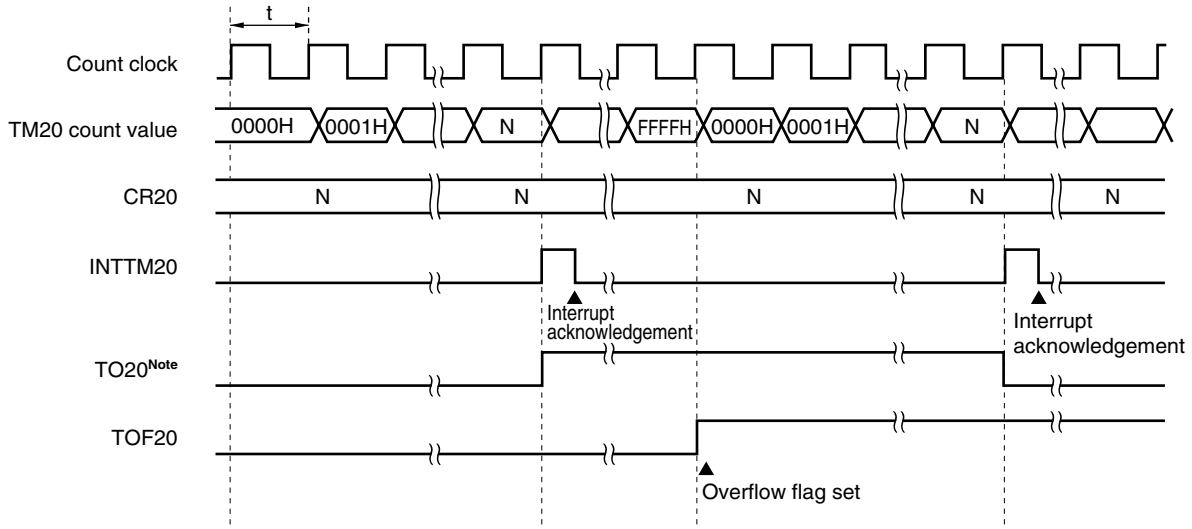


Caution If both the CPT201 flag and CPT200 flag are set to 0, the capture edge operation is prohibited.

When the count value of 16-bit timer counter 20 (TM20) matches the value set in CR20, the output status of the TO20 pin is inverted. This enables timer output. At that time, TM20 continues counting and an interrupt request signal (INTTM20) is generated.

Figure 6-7 shows the timing of timer output (see Table 6-2 for the interval time of 16-bit timer 20).

Figure 6-7. Timer Output Timing



Note The initial value of TO20 becomes low level when output is enabled (TOE20 = 1).

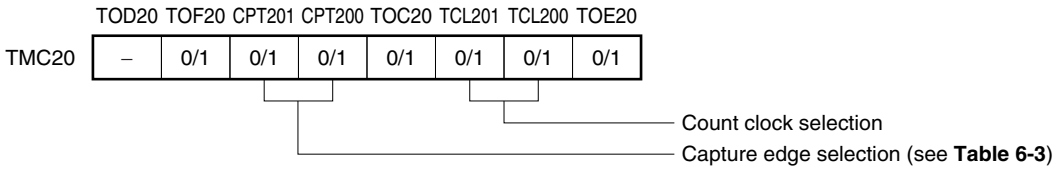
Remark N = 0000H to FFFFH

6.4.3 Capture operation

The capture operation consists of latching the count value of 16-bit timer counter 20 (TM20) into a capture register in synchronization with a capture trigger, and retaining the count value.

Set TMC20 as shown in Figure 6-8 to allow the 16-bit timer to start the capture operation.

Figure 6-8. Settings of 16-Bit Timer Mode Control Register 20 for Capture Operation



16-bit capture register 20 (TCP20) starts a capture operation after a CPT20 capture trigger edge is detected, and latches and retains the count value of 16-bit timer 20. TCP20 fetches the count value within 2 clocks and retains the count value until the next capture edge detection.

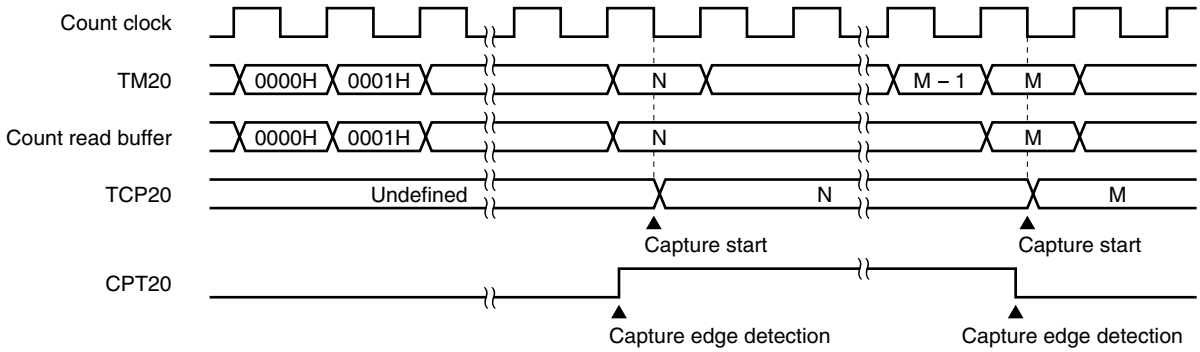
Table 6-3 and Figure 6-9 show the settings of the capture edge and the capture operation timing, respectively.

Table 6-3. Settings of Capture Edge

CPT201	CPT200	Capture Edge Selection
0	0	Capture operation prohibited
0	1	CPT20 pin rising edge
1	0	CPT20 pin falling edge
1	1	CPT20 pin both edges

Caution Because TCP20 is rewritten when a capture trigger edge is detected during TCP20 read, disable capture trigger edge detection during TCP20 read.

Figure 6-9. Capture Operation Timing (with Both Edges of CPT20 Pin Specified)



Remark N, M = 0000H to FFFFH

6.4.4 16-bit timer counter 20 readout

The count value of 16-bit timer counter 20 (TM20) is read out using a 16-bit manipulation instruction.

TM20 readout is performed via the counter read buffer. The counter read buffer latches the TM20 count value, the buffer operation is held pending at the CPU clock falling edge after the read signal of the TM20 lower byte rises, and the count value is retained. The retained counter read buffer value can be read out as the count value.

Cancellation of the pending state is performed at the CPU clock falling edge after the read signal of the TM20 higher byte falls.

$\overline{\text{RESET}}$ input sets TM20 to 0000H and TM20 starts free running.

Figure 6-10 shows the timing of 16-bit timer counter 20 readout.

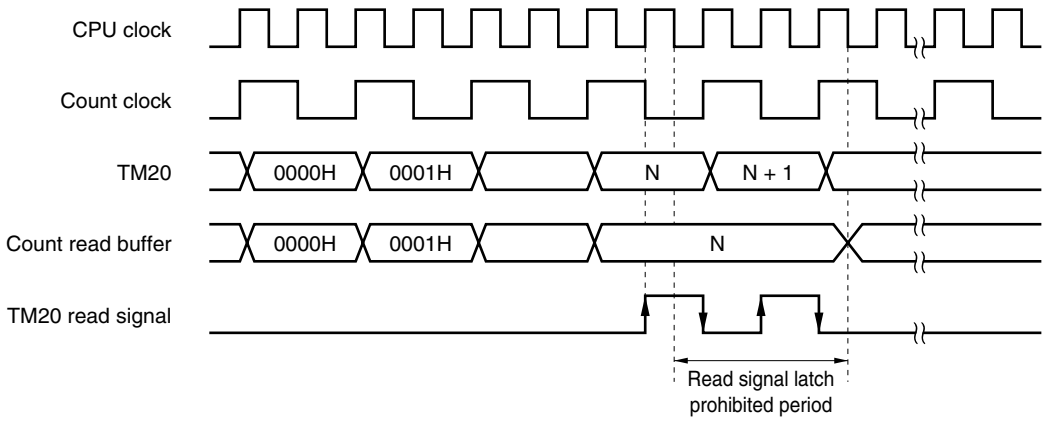
Cautions 1. The count value after releasing stop becomes undefined because the count operation is executed during the oscillation stabilization time.

2. Though TM20 is designed for a 16-bit transfer instruction, an 8-bit transfer instruction can also be used.

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

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

Figure 6-10. 16-Bit Timer Counter 20 Readout Timing



Remark N = 0000H to FFFFH

6.5 Cautions on Using 16-Bit Timer 20

6.5.1 Restrictions when rewriting 16-bit compare register 20

(1) Disable interrupts (TMMK20 = 1) and inversion control of timer output (TOC20 = 0) before rewriting the compare register (CR20).
If the value in CR20 is rewritten in the interrupt-enabled state, an interrupt request may occur at the moment of rewrite.

(2) Depending on the timing of rewriting the compare register (CR20), the interval time may become twice as long as the intended time. Similarly, a shorter waveform or twice-longer waveform than the intended timer output waveform may be output.
To avoid this problem, rewrite the compare register using either of the following procedures.

<Countermeasure A> When rewriting using 8-bit access

<1> Disable interrupts (TMMK20 = 1) and inversion control of timer output (TOC20 = 0).
<2> First rewrite the higher byte of CR20 (16 bits).
<3> Then rewrite the lower byte of CR20 (16 bits).
<4> Clear the interrupt request flag (TMIF20).
<5> Enable timer interrupts/timer output inversion after half a cycle or more of the count clock has elapsed from the start of the interrupt.

<Program example A> (count clock = 32/fx, CPU clock = fx)

```
TM20_VCT:  SET1 TMMK20      ; Disable timer interrupts (6 clocks)
           CLR1 TMC20.3     ; Disable timer output inversion (6 clocks)
           MOV A,#xxH       ; Set the rewrite value of the higher byte (6 clocks)
           MOV !0FF17H,A    ; Rewrite the CR20 higher byte (8 clocks)
           MOV A,#yyH       ; Set the rewrite value of the lower byte (6 clocks)
           MOV !0FF16H,A    ; Rewrite the CR20 lower byte (8 clocks)
           CLR1 TMIF20      ; Clear the interrupt request flag (6 clocks)
           CLR1 TMMK20      ; Enable timer interrupts (6 clocks)
           SET1 TMC20.3     ; Enable timer output inversion
```

Total: 16 clocks or more[Note]

Note Because the INTTM20 signal becomes high level for half a cycle of the count clock after an interrupt is generated, the output is inverted if TOC20 is set to 1 during this period.

<Countermeasure B> When rewriting using 16-bit access

<1> Disable interrupts (TMMK20 = 1) and inversion control of timer output (TOC20 = 0).

<2> Rewrite CR20 (16 bits).

<3> Wait for one cycle or more of the count clock.

<4> Clear the interrupt request flag (TMIF20).

<5> Enable timer interrupts/timer output inversion.

<Program example B> (count clock = 32/fx, CPU clock = fx)

```
TM20_VCT   SET1   TMMK20       ; Disable timer interrupts
           CLR1   TMC20.3      ; Disable timer output inversion
           MOVW   AX,#xxyyH    ; Set the rewrite value of CR20
           MOVW   CR20,AX      ; Rewrite CR20
           NOP                 
           NOP                 
            :                  ; 16 NOP instructions (wait for 32/fx) Note
           NOP                 
           NOP                 
           CLR1   TMIF20       ; Clear the interrupt request flag
           CLR1   TMMK20       ; Enable timer interrupts
           SET1   TMC20.3      ; Enable timer output inversion
```

Note Clear the interrupt request flag (TMIF20) after waiting for one cycle or more of the count clock from the instruction that rewrites CR20 (MOVW CR20, AX).

CHAPTER 7 8-BIT TIMERS 50, 60, AND 61

7.1 Functions of 8-Bit Timers 50, 60, and 61

One 8-bit timer channel (timer 50) and two 8-bit timer/event counter channels (timer 60 and 61) are incorporated in the μPD789479 Subseries. The operation modes listed in the following table can be set via mode register settings.

Table 7-1. Operation Modes

Mode \ Channel	Timer 50	Timer 60	Timer 61
8-bit timer counter mode (stand-alone mode)	Available	Available	Available
16-bit timer counter mode (cascade connection mode)	Available		Not available
Carrier generator mode	Available		Not available
PWM output mode	Available	Not available	Not available
PPG output mode	Not available	Available	Available
24-bit event counter mode (connect with 16-bit timer 20)	Not available	Not available	Available

(1) Mode to use 8-bit timer/event counter as discrete unit (stand-alone mode)

The following functions can be used in this mode.

<Timer 50>

- Interval timer with 8-bit resolution
- Square wave output with 8-bit resolution

<Timer 60 and 61>

- Interval timer with 8-bit resolution
- External event counter with 8-bit resolution
- Square wave output with 8-bit resolution

(2) Mode to use timer 50 and timer 60 connected in cascade (16-bit resolution: cascade connection)

Operation as a 16-bit timer/event counter is enabled in 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 60 is output in the cycle set by timer 50.

(4) PWM output mode (PWM: Pulse Width Modulator)

Pulses are output using any duty ratio (pulse width). The cycle (overflow cycle of the timer) becomes constant (free running).

(5) PPG output mode (PPG: Programmable Pulse Generator)

Pulses are output using any cycle or duty ratio (pulse width) set (both the cycle and pulse width are programmable).

(6) 24-bit event counter mode

Operation as an external event counter with 24-bit resolution is enabled using 16-bit timer 20 and timer 61. However, this mode operates only as a counter read function.

There is no compare, match, or clear function.

<Setting method>

<1> Select the timer 61 interrupt signal for the count clock of 16-bit timer 20 (TCL201 = 0, TCL200 = 0)

<2> Set timer 61 in stand-alone mode (TMD611 = 0)

Select the external clock input from pin TMI61 for the count clock of timer 61

((TCL612 = 0, TCL611 = 1) or (TCL612 = 1, TCL611 = 0))

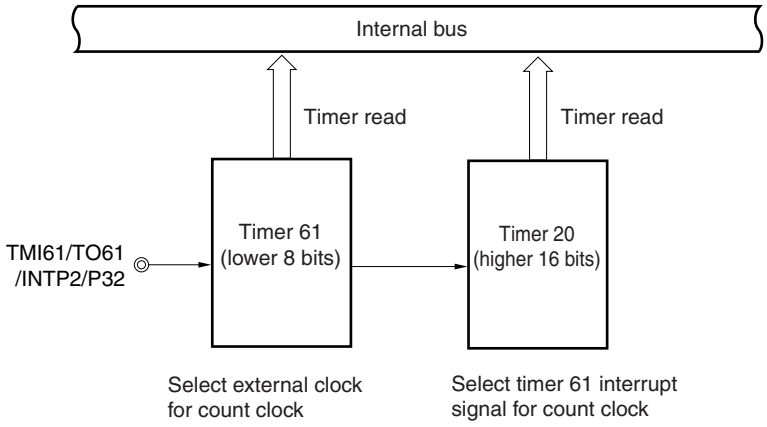
<3> Set CR61 to FFH

<4> Read the current count value of 16-bit timer 20

(16-bit timer 20 does not have a count clear function and is counting constantly)

<5> Enable timer 61 count operation (TCE61 = 1)

Figure 7-1. Block Diagram of 24-Bit Event Counter



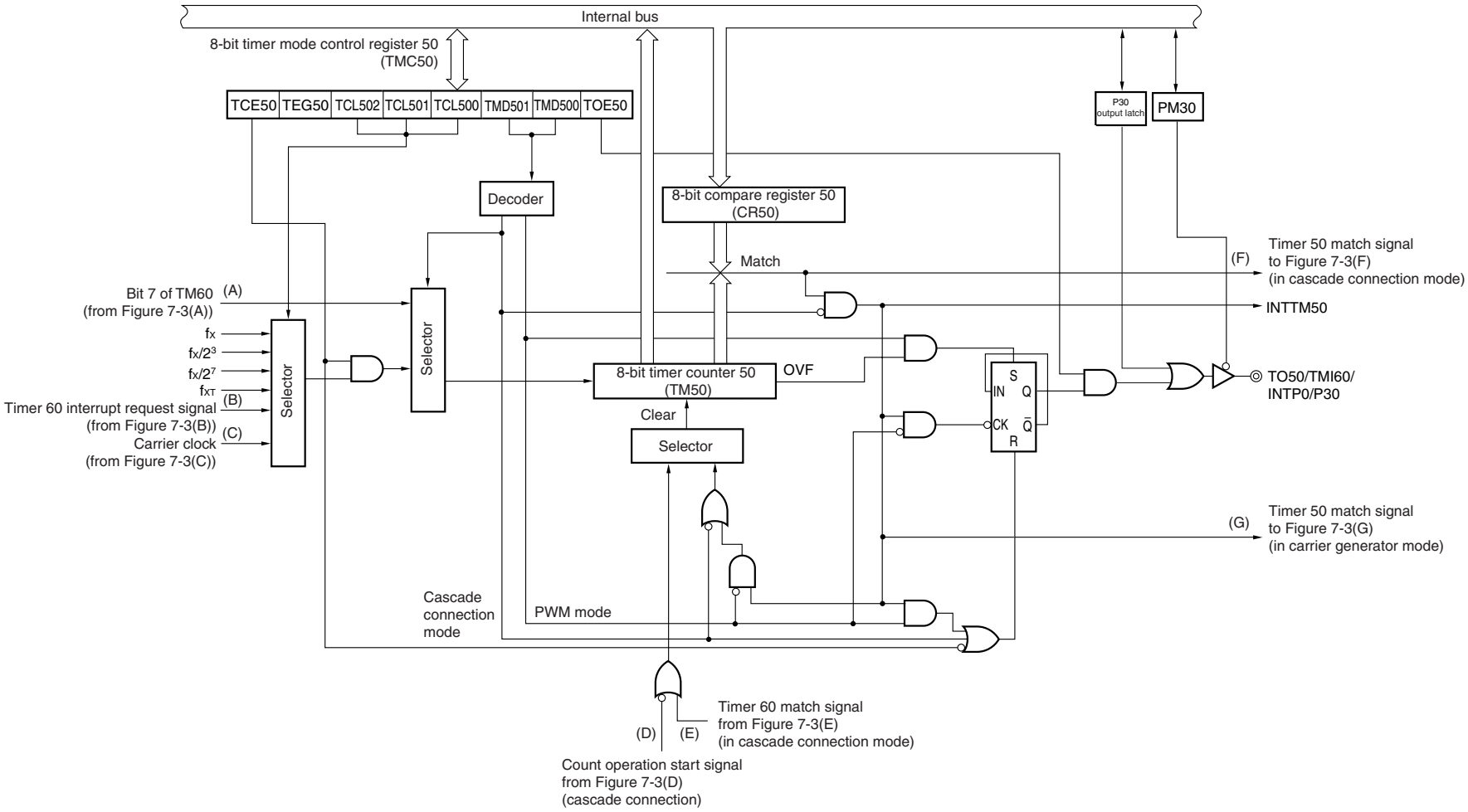
7.2 Configuration of 8-Bit Timers 50, 60, and 61

8-bit timers 50, 60, and 61 include the following hardware.

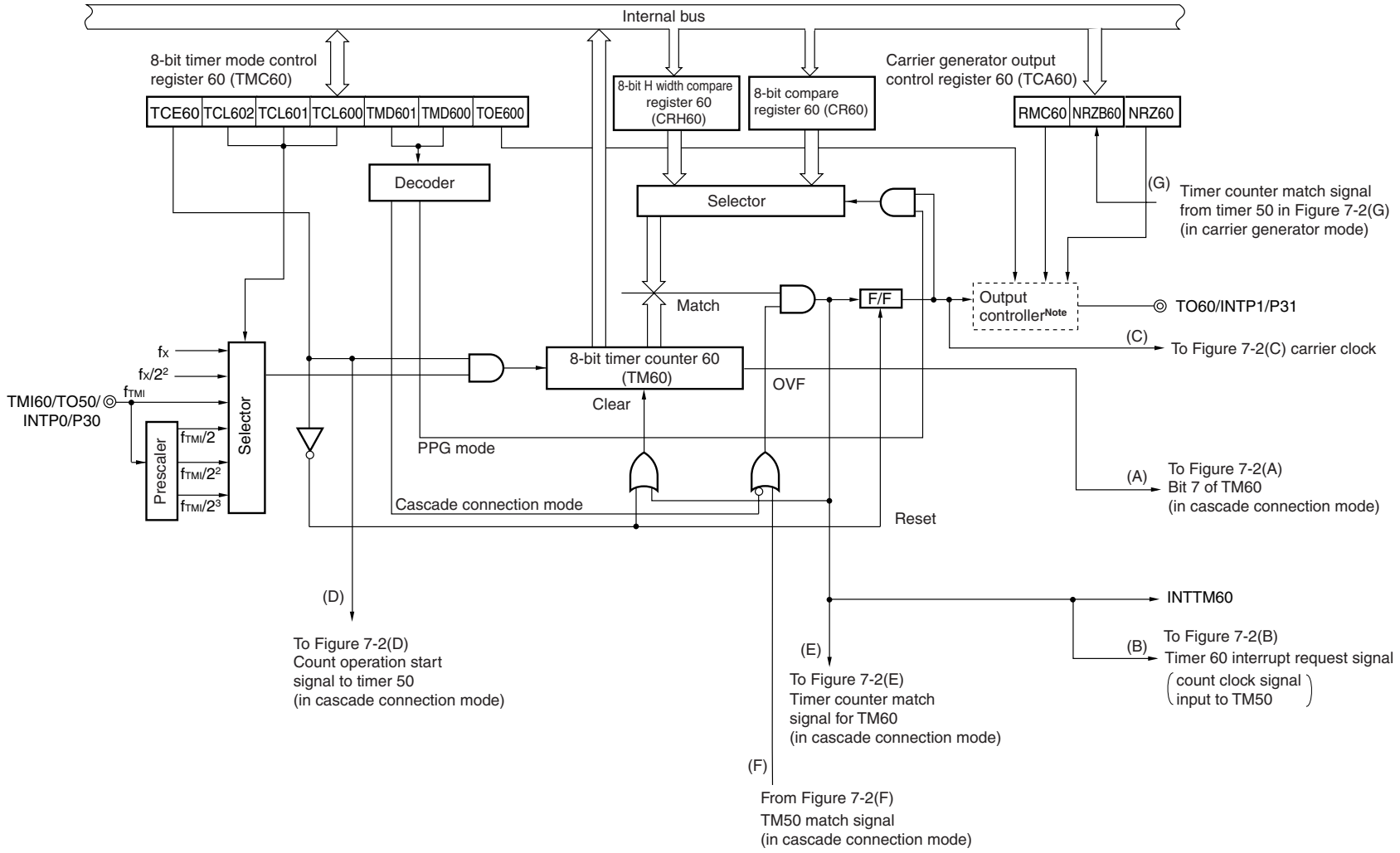
Table 7-2. Configuration of 8-Bit Timers 50, 60, and 61

Item	Configuration
Timer counter	8 bits × 3 (TM50, TM60, TM61)
Registers	Compare registers: 8 bits × 5 (CR50, CR60, CRH60, CR61, CRH61)
Timer outputs	3 (TO50, TO60, TO61)
Control registers	8-bit timer mode control register 50 (TMC50) 8-bit timer mode control register 60 (TMC60) Carrier generator output control register 60 (TCA60) 8-bit timer mode control register 61 (TMC61) Port mode register 3 (PM3) Port 3 (P3)

<R> **Figure 7-2. Block Diagram of Timer 50**

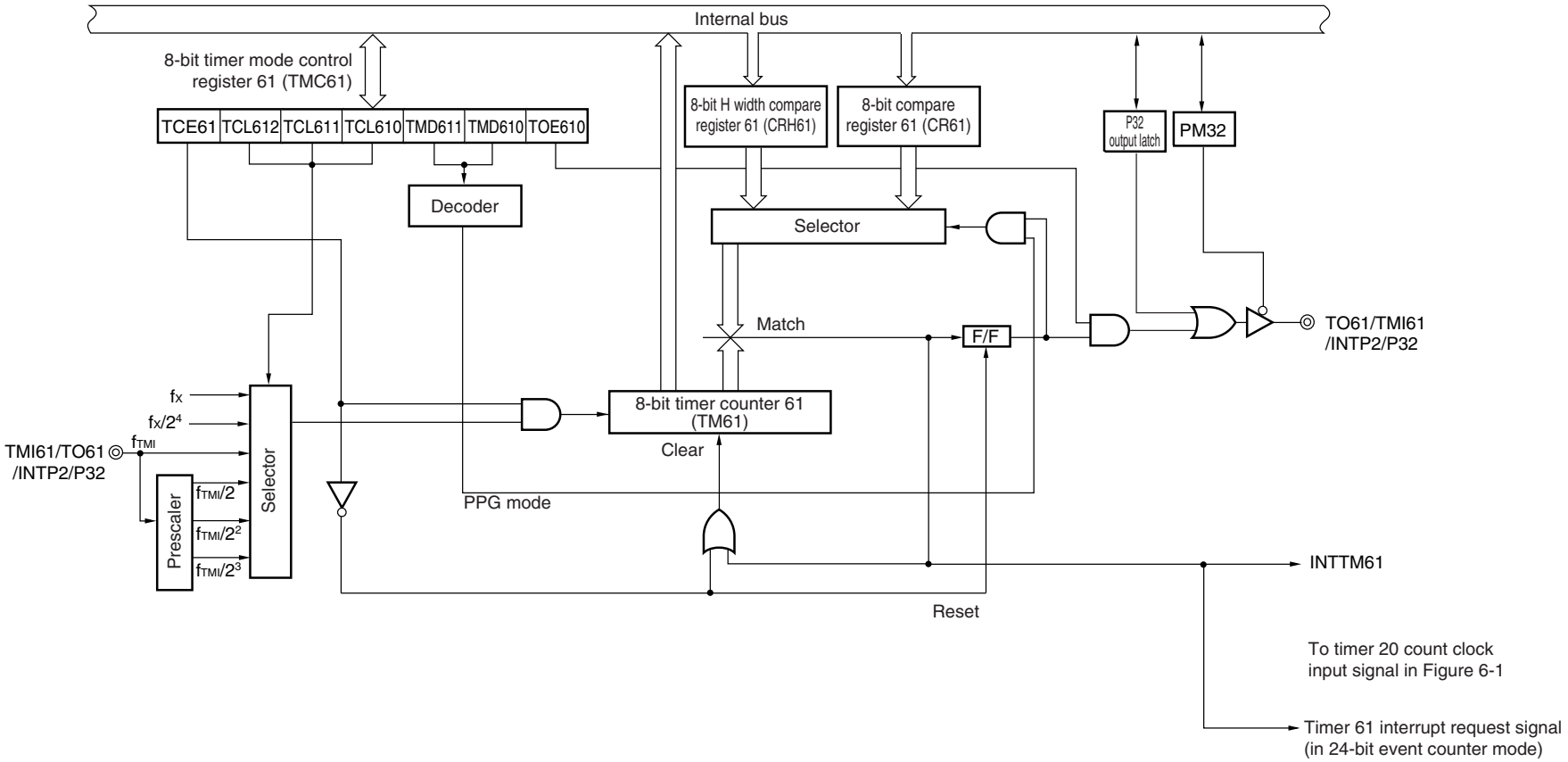


<R> **Figure 7-3. Block Diagram of Timer 60**

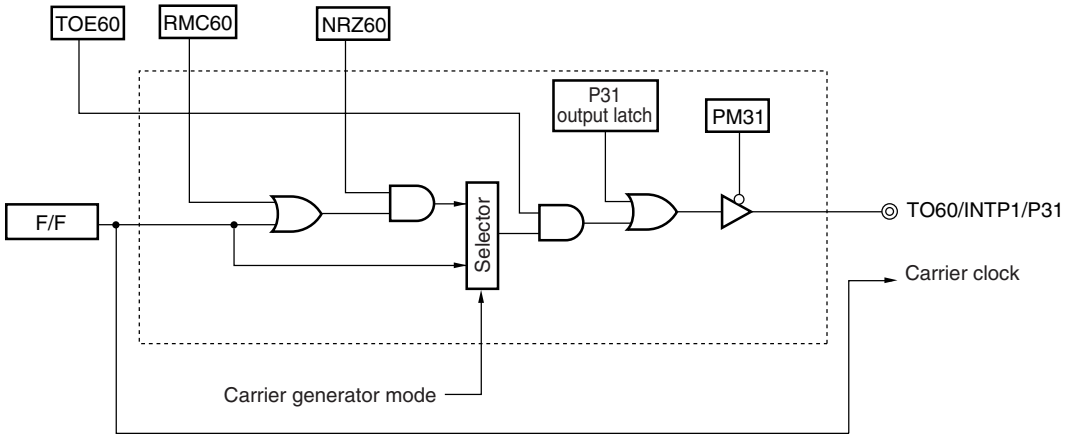


Note For details, see **Figure 7-5**.

Figure 7-4. Block Diagram of Timer 61



<R> **Figure 7-5. Block Diagram of Output Controller (Timer 60)**



(1) 8-bit compare register 50 (CR50)

This 8-bit register is used to continually compare the value set to CR50 with the count value in 8-bit timer counter 50 (TM50) and to issue an interrupt request (INTTM50) when a match occurs. In PWM mode, this register is used for high-level width setting.

CR50 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes this register undefined.

Cautions 1. In PWM output mode (TMD501 = 1, TMD500 = 0), if CR50 is rewritten while the timer is operating, a high level may be output for one clock cycle immediately after this rewrite operation. If this waveform may cause problems in the application, either <1> stop the timer when rewriting CR50, or <2> rewrite CR50 after TOE50 has been cleared.

2. If both edges have been selected as the valid edge of the count clock in PWM output mode (TEG50 = 1), do not set CR50 to 00H, 01H, or FFH. Also, if the rising edge has been selected as the valid edge (TEG50 = 0), do not set CR50 to 00H.

(2) 8-bit compare register 60 (CR60)

This 8-bit register is used to continually compare the value set to CR60 with the count value in 8-bit timer counter 60 (TM60) and issue an interrupt request (INTTM60) when a match occurs. When connected to TM50 via a cascade connection and using as a 16-bit timer/event counter, the interrupt request (INTTM60) occurs only when matches occur simultaneously between CR50 and TM50 and between CR60 and TM60 (INTTM50 is not generated).

<R> In carrier generator mode and PPG output mode, the high-level width of timer output is set by writing a value to CRH6n.

CR60 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes this register undefined.

(3) 8-bit compare register 61 (CR61)

This 8-bit register is used to continually compare the value set to CR61 with the count value in 8-bit timer counter 61 (TM61) and issue an interrupt request (INTTM61) when a match occurs.

In PPG output mode, this registered used for low-level width setting.

CR61 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes this register undefined.

(4) 8-bit H width compare registers 60 and 61 (CRH60, CRH61)

<R> This 8-bit register is used to continually compare the value set to CRH6n with the count value in TM6n and to issue an interrupt request (INTTM6n) when a match occurs.

In carrier generator mode and PPG output mode, the high-level width of timer output is set by writing a value to CRH6n.

CRH6n is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes this register undefined.

Remark n = 0, 1

(5) 8-bit timer counters 50, 60, and 61(TM50, TM60, TM61)

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

TM50, TM60, and TM61 are read with an 8-bit memory manipulation instruction.

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

TM50, TM60, and TM61 are cleared to 00H under the following conditions.

(a) Stand-alone mode

- After reset
- When TCEmn (bit 7 of 8-bit timer mode control register mn (TMCmn)) is cleared to 0
- When a match occurs between TMmn and CRmn
- When the TMmn count value overflows

Remark mn = 50, 60, 61

(b) Cascade connection mode (TM50 and TM60 are simultaneously cleared to 00H)

- After reset
- When the TCE60 flag is cleared to 0
- When matches occur simultaneously between TM50 and CR50 and between TM60 and CR60
- When the TM50 and TM60 count values overflow simultaneously

(c) Carrier generator (TM60) and PPG output mode (TM60 and TM61)

- After reset
- When the TCE6n flag is cleared to 0
- When a match occurs between TM6n and CR6n
- When a match occurs between TM6n and CRH6n
- When the TM6n count value overflows

Remark n = 0, 1

(d) PWM output mode (TM50)

- After reset
- When the TCE50 flag is cleared to 0
- When the TM50 count value overflows

7.3 Control Registers for 8-Bit Timers 50, 60, and 61

8-bit timers 50, 60, and 61 are controlled by the following six registers.

- 8-bit timer mode control register 50 (TMC50)
- 8-bit timer mode control register 60 (TMC60)
- Carrier generator output control register 60 (TCA60)
- 8-bit timer mode control register 61 (TMC61)
- Port mode register 3 (PM3)
- Port 3 (P3)

(1) 8-bit timer mode control register 50 (TMC50)

8-bit timer mode control register 50 (TMC50) is used to control the timer 50 count clock setting and the operation mode setting.

TMC50 is set with a 1-bit or 8-bit memory manipulation instruction.

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

Figure 7-6. Format of 8-Bit Timer Mode Control Register 50 (1/2)

Symbol	<7>	<6>	5	4	3	2	1	<0>	Address	After reset	R/W
TMC50	TCE50	TEG50	TCL502	TCL501	TCL500	TMD501	TMD500	TOE50	FF4DH	00H	R/W

TCE50	Control of TM50 count operation[Note 1]
0	Clear TM50 count value and stop operation
1	Start count operation

TEG50	Selection of valid edge of TM50 count clock
0	Count at the rising edge of the count clock
1	Count at both edges of the count clock[Note 2]

TCL502	TCL501	TCL500	Selection of timer 50 count clock
0	0	0	f_X (5.0 MHz)
0	0	1	$f_X/2^3$ (625 kHz)
0	1	0	$f_X/2^7$ (39.1 kHz)
0	1	1	f_{XT} (32.768 kHz)
1	0	0	Timer 60 match signal (INTTM60)
1	0	1	Carrier clock (in carrier generator mode) or timer 60 output signal (in other than carrier generator mode)
Other than above			Setting prohibited

TMD501	TMD500	TMD601	TMD600	Selection of operation mode for timer 50[Note 3]
0	0	×	0	Stand-alone mode (8-bit counter mode)
0	1	0	1	16-bit counter mode (cascade connection mode)
0	0	1	1	Carrier generator mode
1	0	×	0	PWM output mode
Other than above				Setting prohibited

Figure 7-6. Format of 8-Bit Timer Mode Control Register 50 (2/2)

Symbol	<7>	<6>	5	4	3	2	1	<0>	Address	After reset	R/W
TMC50	TCE50	TEG50	TCL502	TCL501	TCL500	TMD501	TMD500	TOE50	FF4DH	00H	R/W

TOE50	Control of timer output[Note 4]
0	Output disabled
1	Output enabled

Notes 1. Since the count operation is controlled by TCE60 (bit 7 of TMC60) in cascade connection mode, any setting for TCE50 is ignored.

2. Selection of both edges is valid only in PWM mode. In 8-bit counter mode or cascade connection mode, even if TEG50 is set to 1, counting occurs at the rising edge.

3. The operation mode selection is set by a combination of the TMC50 and TMC60 registers.

4. Since timer 50 output is disabled in cascade connection mode, set TOE50 to 0.

Cautions 1. In cascade connection mode, the timer 60 output signal is forcibly selected for the count clock.

2. To manipulate TMC50, follow the setting procedure below.

<1> Set the TM50 count operation to stop.

<2> Set the operation mode and count clock.

<3> The count operation starts.

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

3. The parenthesized values apply to operation at f_X = 5.0 MHz or f_{XT} = 32.768 kHz.

4. ×: don't care

(2) 8-bit timer mode control register 60 (TMC60)

8-bit timer mode control register 60 (TMC60) is used to control the timer 60 count clock setting and the operation mode setting.

TMC60 is set with a 1-bit or 8-bit memory manipulation instruction.

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

Figure 7-7. Format of 8-Bit Timer Mode Control Register 60

Symbol	<7>	6	5	4	3	2	1	<0>	Address	After reset	R/W
TMC60	TCE60	0	TCL602	TCL601	TCL600	TMD601	TMD600	TOE600	FF4EH	00H	R/W

TCE60	Control of TM60 count operation[Note 1]
0	Clear TM60 count value and stops operation (the count value is also cleared for TM50 in cascade connection mode)
1	Start count operation (the count operation is also started for TM50 in cascade connection mode)

TCL602	TCL601	TCL600	Selection of timer 60 count clock
0	0	0	f_X (5.0 MHz)
0	0	1	$f_X/2^2$ (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

TMD501	TMD500	TMD601	TMD600	Selection of operation mode for timer 60[Note 2]
×	0	0	0	Stand-alone mode (8-bit counter mode)
0	1	0	1	16-bit counter mode (cascade connection mode)
0	0	1	1	Carrier generator mode
×	0	1	0	PPG output mode
Other than above				Setting prohibited

TOE600	Control of timer output
0	Output disabled
1	Output enabled

Notes 1. Since the count operation is controlled by TCE60 (bit 7 of TMC60) in cascade connection mode, any setting for TCE50 is ignored.

2. The operation mode selection is set by a combination of the TMC50 and TMC60 registers.

Caution To manipulate TMC60, follow the setting procedure below.

<1> Set the TM60 count operation to stop.

<2> Set the operation mode and count clock.

<3> The count operation starts.

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{TMI}: External input clock frequency

3. The parenthesized values apply to operation at f_X = 5.0 MHz.

4. ×: don't care

(3) Carrier generator output control register 60 (TCA60)

This register is used to set the timer output data in carrier generator mode.

TCA60 is set with a 1-bit or 8-bit memory manipulation instruction.

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

Figure 7-8. Format of Carrier Generator Output Control Register 60

Symbol	7	6	5	4	3	<2>	<1>	<0>	Address	After reset	R/W
TCA60	0	0	0	0	0	RMC60	NRZB60	NRZ60	FF4FH	00H	R/W[Note]

RMC60	Control of remote control output
0	When NRZ60 = 1, a carrier pulse is output to TO60/INTP1/P31 pin (when NRZ60 = 0, a low level is output to TO60/INTP1/P31 pin)
1	When NRZ60 = 1, high-level signal is output to TO60/INTP1/P31 pin (when NRZ60 = 0, a low level is output to TO60/INTP1/P31 pin)

NRZB60	This is the bit that stores the next data to be output to NRZ60. When a match signal occurs (for a match with timer 50), the data is output to NRZ60.

NRZ60	No return zero data
0	Output low-level signal (carrier clock is stopped)
1	Output carrier pulse or high-level signal

Note Bit 0 is write-only

Cautions 1. At the count start, input the values of the data reloaded from NRZB60 to NRZ60. For NRZB60, input the data required by the program in advance.

2. When timer 60 output is disabled (TOE600 = 0), use of a 1-bit memory manipulation instruction for TCA60 output is disabled (only an 8-bit memory manipulation instruction can be used).

3. When timer 60 output is enabled (TOE600 = 1), a write operation to NRZ is invalid. However, while the timer 50 interrupt signal (INTTM50) is high level, the NRZB60 value is immediately transferred to NRZ60 if TCA60 is rewritten. Rewrite TCA60 after waiting for half a clock of the TM50 count clock during INTTM50 interrupt servicing.

(4) 8-bit timer mode control register 61 (TMC61)

8-bit timer mode control register 61 (TMC61) is used to control the timer 61 count clock setting and the operation mode setting.

TMC61 is set with a 1-bit or 8-bit memory manipulation instruction.

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

Figure 7-9. Format of 8-Bit Timer Mode Control Register 61

Symbol	<7>	6	5	4	3	2	1	<0>	Address	After reset	R/W
TMC61	TCE61	0	TCL612	TCL611	TCL610	TMD611	TMD610	TOE610	FF41H	00H	R/W

TCE61	Control of TM61 count operation
0	Clear TM61 count value and stop operation
1	Start count operation

TCL612	TCL611	TCL610	Selection of timer 61 count clock[Note]
0	0	0	f_X (5.0 MHz)
0	0	1	$f_X/2^4$ (313 kHz)
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

TMD611	TMD610	Selection of operation mode for timer 61[Note]
0	0	Stand-alone mode (8-bit counter mode)
1	0	PPG output mode
Other than above		Setting prohibited

TOE610	Control of timer output
0	Output disabled
1	Output enabled

Note To set the register in 24-bit event counter mode, the external input clock and stand-alone mode need to be selected.

Caution To manipulate TMC61, follow the setting procedure below.
<1> Set the TM61 count operation to stop.
<2> Set the operation mode and count clock.
<3> The count operation starts.

Remarks 1. f_X: Main system clock oscillation frequency
2. f_{TMI}: External input clock frequency
3. The parenthesized values apply to operation at f_X = 5.0 MHz.

(5) Port mode register 3 (PM3)

This register is used to set the I/O mode of port 3 in 1-bit units.

When using the P30/INTP0/TO50/TMI60 pin as a timer output (TO50), set PM30 and the P30 output latch to 0. When used as a timer input (TMI60), set PM30 to 1.

When using the P31/INTP1/TO60 pin as a timer output (TO60), set PM31 and the P31 output latch to 0.

When using the P32/INTP2/TO61/TMI61 pin as a timer input (TMI61), set PM32 to 1. When used as a timer output (TO61), set PM32 and the P32 output latch to 0.

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

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

Figure 7-10. Format of Port Mode Register 3

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM3	1	1	1	PM34	PM33	PM32	PM31	PM30	FF23H	FFH	R/W

PM3n	I/O mode of P3n pin (n = 0 to 2)
0	Output mode (output buffer is on)
1	Input mode (output buffer is off)

7.4 Operation of 8-Bit Timers 50, 60, and 61

7.4.1 Operation as 8-bit timer counter

Timer 50, timer 60, and timer 61 can be independently used as 8-bit timer counters.
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 60 and timer 61 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 nm (CRnm).
To operate 8-bit timer nm as an interval timer, settings must be made in the following sequence.

<1> Disable operation of 8-bit timer counter nm (TMnm) (TCEnm = 0).
<2> For timer 50, disable timer output of TO50 (TOE50 = 0).
For timer 60, disable timer output of TO60 (TOE600 = 0).
For timer 61, disable timer output of TO61 (TOE610 = 0).
<3> Set a count value in CRnm.
<4> Set the operation mode of timer nm to 8-bit timer counter mode
(see Figures 7-6, 7-7, and 7-9).
<5> Set the count clock for timer nm (see Figures 7-6, 7-7, and 7-9).
<6> Enable the operation of TMnm (TCEnm = 1).

When the count value of 8-bit timer counter nm (TMnm) matches the value set in CRnm, TMnm is cleared to 00H and continues counting. At the same time, an interrupt request signal (INTTMnm) is generated.
Tables 7-3 to 7-5 show the interval time, and Figures 7-11 to 7-16 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 nm = 50, 60, 61

Table 7-3. Interval Time of Timer 50

TCL502	TCL501	TCL500	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	0	$1/f_X$ (0.2 μs)	$2^8/f_X$ (51.2 μs)	$1/f_X$ (0.2 μs)
0	0	1	$2^3/f_X$ (1.6 μs)	$2^{11}/f_X$ (409.6 μs)	$2^3/f_X$ (1.6 μs)
0	1	0	$2^7/f_X$ (25.6 μs)	$2^{15}/f_X$ (6.55 ms)	$2^7/f_X$ (25.6 μs)
0	1	1	$1/f_{XT}$ (30.5 μs)	$2^8/f_{XT}$ (7.81 ms)	$1/f_{XT}$ (30.5 μs)
1	0	0	Input cycle of timer 60 match signal	Input cycle of timer 60 match signal $\times 2^8$	Input cycle of timer 60 match signal
1	0	1	Input cycle of timer 60 output	Input cycle of timer 60 output $\times 2^8$	Input cycle of timer 60 output

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

3. The parenthesized values apply to operation at f_X = 5.0 MHz or f_{XT} = 32.768 kHz.

Table 7-4. Interval Time of Timer 60

TCL602	TCL601	TCL600	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	0	$1/f_X$ (0.2 μs)	$2^8/f_X$ (51.2 μs)	$1/f_X$ (0.2 μs)
0	0	1	$2^2/f_X$ (0.8 μs)	$2^{10}/f_X$ (204 μs)	$2^2/f_X$ (0.8 μs)
0	1	0	f_{TMI} input cycle	f_{TMI} input cycle $\times 2^8$	f_{TMI} input cycle
0	1	1	$f_{TMI}/2$ input cycle	$f_{TMI}/2$ input cycle $\times 2^8$	$f_{TMI}/2$ input cycle
1	0	0	$f_{TMI}/2^2$ input cycle	$f_{TMI}/2^2$ input cycle $\times 2^8$	$f_{TMI}/2^2$ input cycle
1	0	1	$f_{TMI}/2^3$ input cycle	$f_{TMI}/2^3$ input cycle $\times 2^8$	$f_{TMI}/2^3$ input cycle

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{TMI}: External input clock frequency

3. The parenthesized values apply to operation at f_X = 5.0 MHz.

Table 7-5. Interval Time of Timer 61

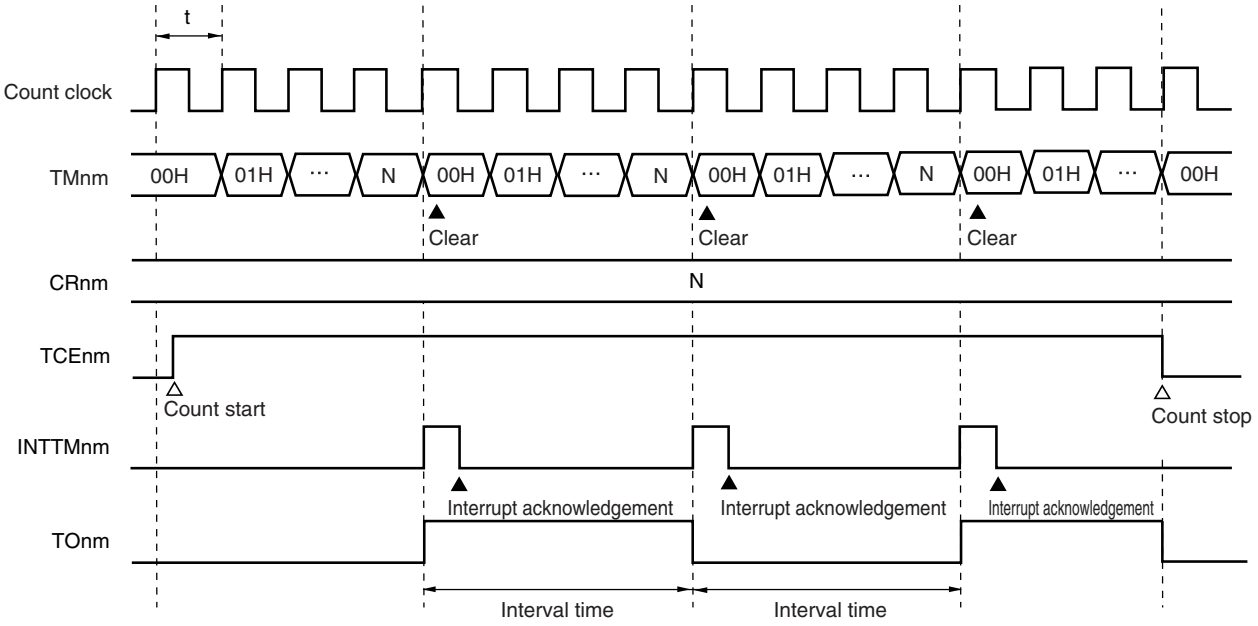
TCL612	TCL611	TCL610	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	0	$1/f_X$ (0.2 μs)	$2^8/f_X$ (51.2 μs)	$1/f_X$ (0.2 μs)
0	0	1	$2^4/f_X$ (3.2 μs)	$2^{12}/f_X$ (819 μs)	$2^4/f_X$ (3.2 μs)
0	1	0	f_{TMI} input cycle	f_{TMI} input cycle $\times 2^8$	f_{TMI} input cycle
0	1	1	$f_{TMI}/2$ input cycle	$f_{TMI}/2$ input cycle $\times 2^8$	$f_{TMI}/2$ input cycle
1	0	0	$f_{TMI}/2^2$ input cycle	$f_{TMI}/2^2$ input cycle $\times 2^8$	$f_{TMI}/2^2$ input cycle
1	0	1	$f_{TMI}/2^3$ input cycle	$f_{TMI}/2^3$ input cycle $\times 2^8$	$f_{TMI}/2^3$ input cycle

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{TMI}: External input clock frequency

3. The parenthesized values apply to operation at f_X = 5.0 MHz.

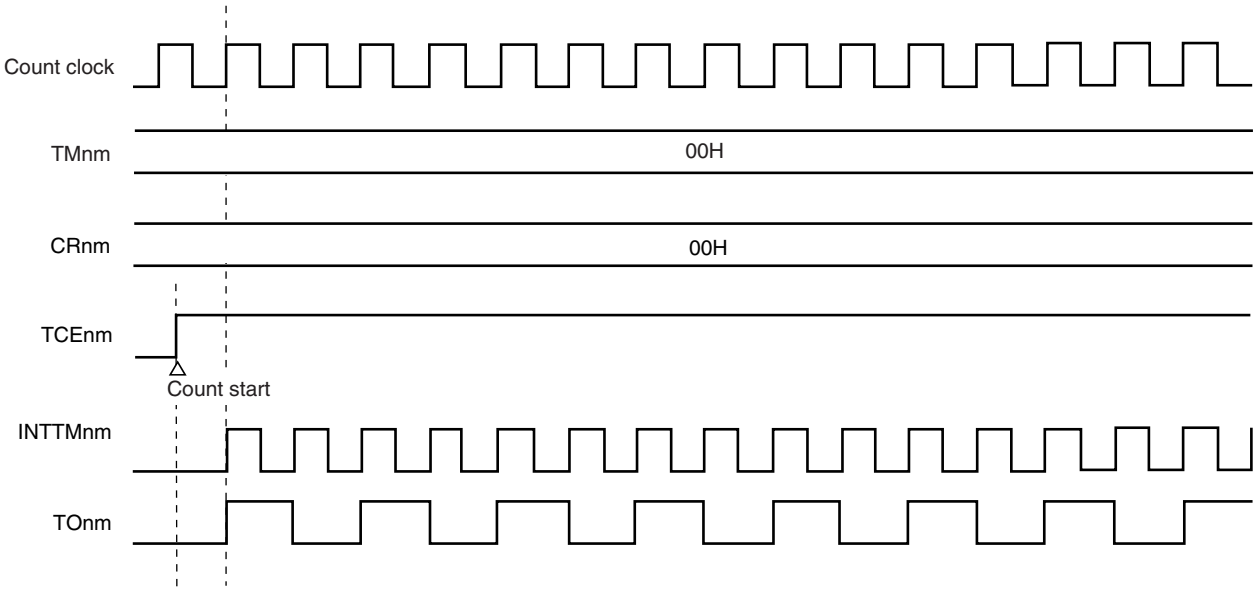
<R> **Figure 7-11. Timing of Interval Timer Operation with 8-Bit Resolution (Basic Operation)**



Remarks 1. Interval time = (N + 1) × t: N = 00H to FFH

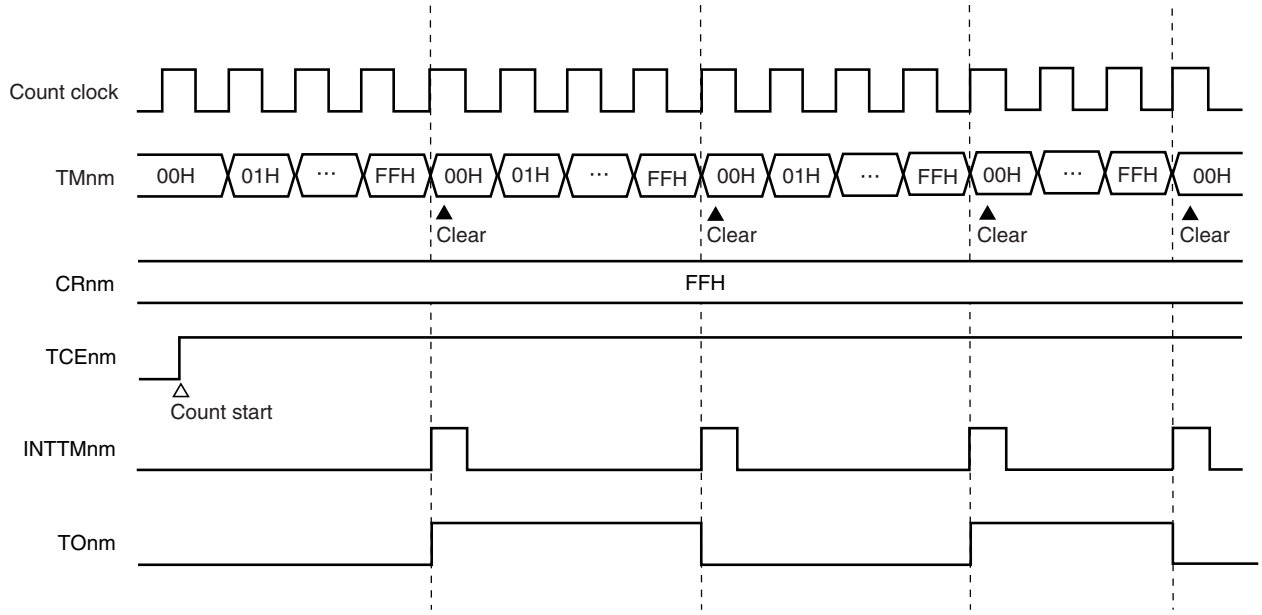
2. nm = 50, 60, 61

Figure 7-12. Timing of Interval Timer Operation with 8-Bit Resolution (When CRnm Is Set to 00H)



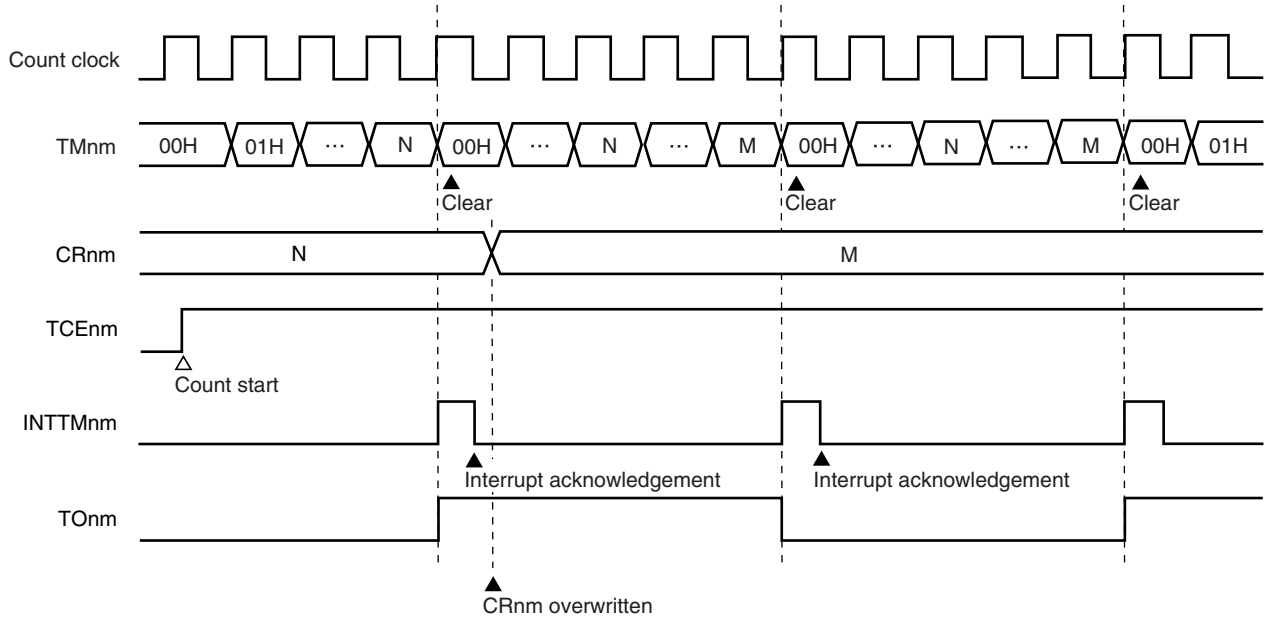
Remark nm = 50, 60, 61

<R> **Figure 7-13. Timing of Interval Timer Operation with 8-Bit Resolution (When CRnm Is Set to FFH)**



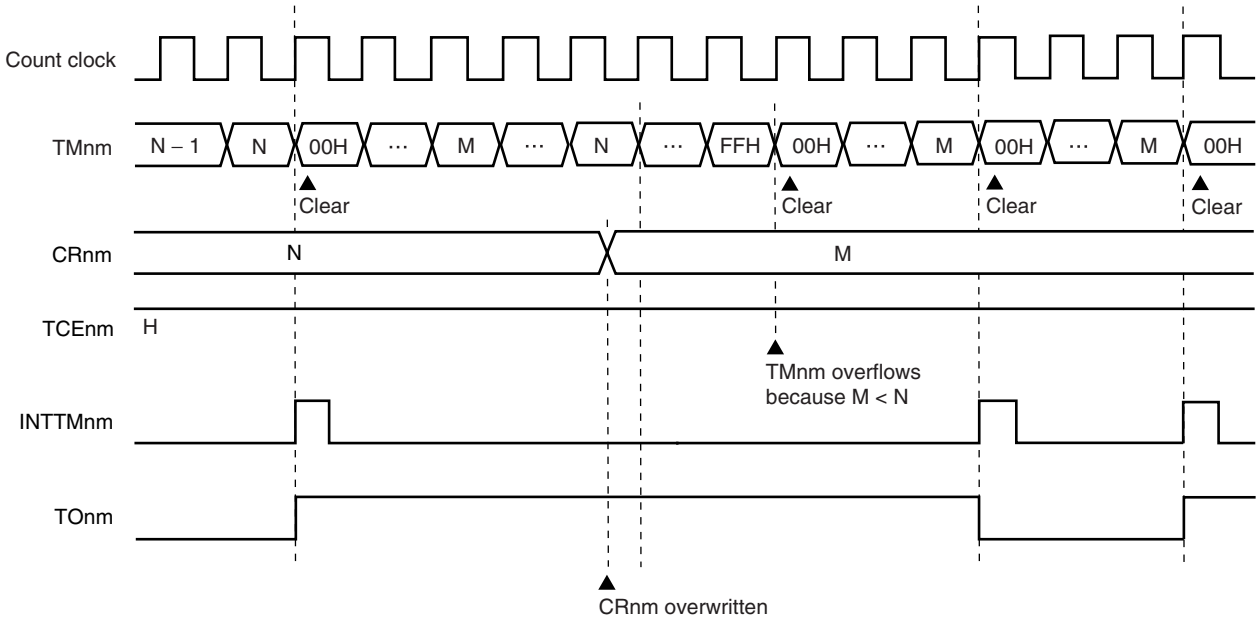
Remark nm = 50, 60, 61

Figure 7-14. Timing of Interval Timer Operation with 8-Bit Resolution (When CRnm Changes from N to M (N < M))



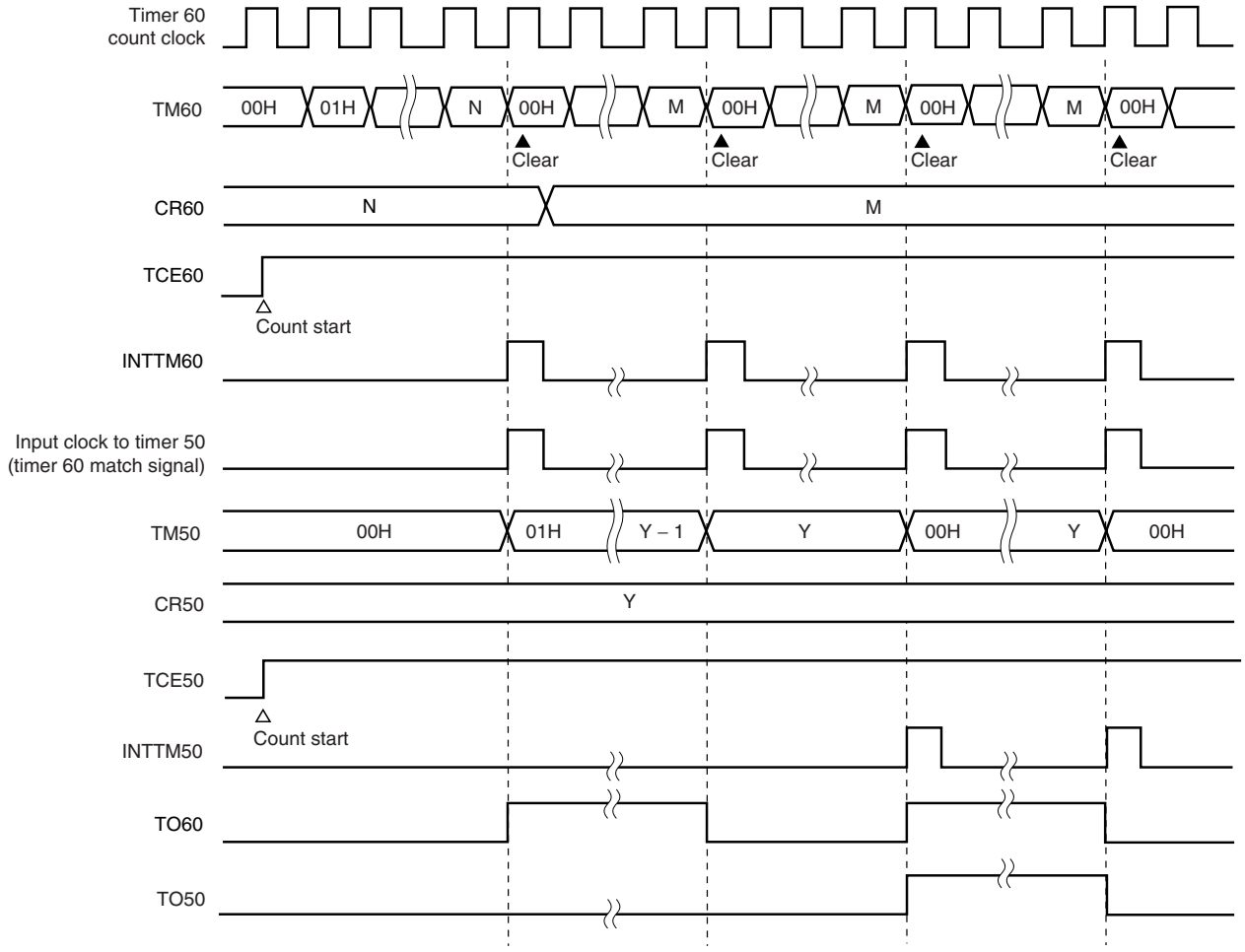
Remark 00H ≤ N < M ≤ FFH
nm = 50, 60, 61

Figure 7-15. Timing of Interval Timer Operation with 8-Bit Resolution (When CRnm Changes from N to M (N > M))



Remark 00H ≤ M < N ≤ FFH
nm = 50, 60, 61

Figure 7-16. Timing of Interval Timer Operation with 8-Bit Resolution (When Timer 60 Match Signal Is Selected for Timer 50 Count Clock)



Remark 00H ≤ N < M ≤ FFH
Y = 00H to FFH

(2) Operation as external event counter with 8-bit resolution (timer 60 and timer 61 only)

The external event counter counts the number of external clock pulses input to the TMI6m pin by using 8-bit timer counter 6m (TM6m).

To operate timer 6m as an external event counter, settings must be made in the following sequence.

<1> Disable operation of 8-bit timer counter 6m (TM6m) (TCE6m = 0).
<2> Disable timer output of TO6m (TOE6m0 = 0).
<3> When using timer 60, set P30 to input mode (PM30 = 1).
When using timer 61, set P32 to input mode (PM32 = 1).
<4> Select the external input clock for timer 6m (see Figures 7-7 and 7-9).
<5> Set the operation mode of timer 6m to 8-bit timer counter mode (see Figures 7-7 and 7-9).
<6> Set a count value in CR6m.
<7> Enable the operation of TM6m (TCE6m = 1).

Each time the valid edge is input, the value of TM6m is incremented.

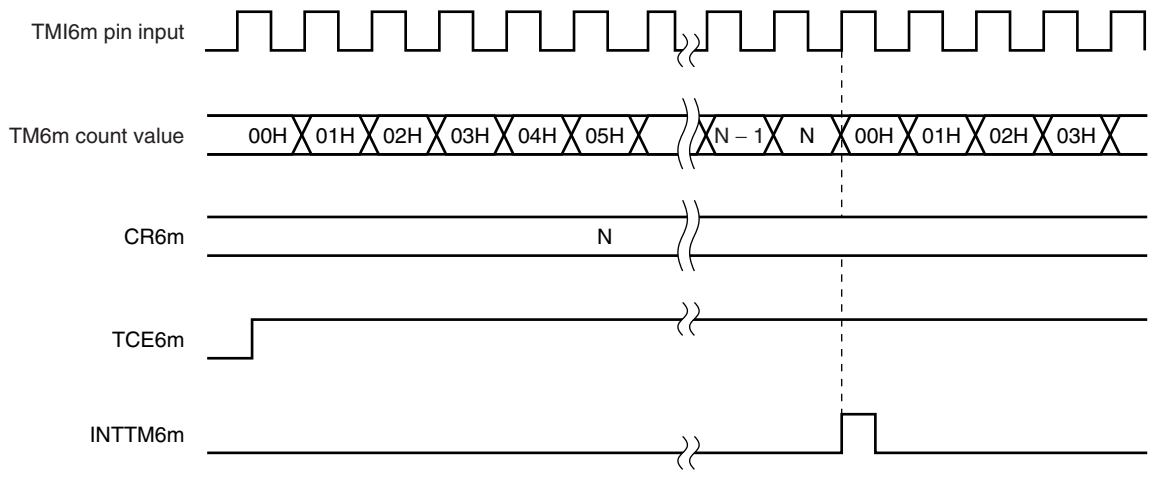
When the count value of TM6m matches the value set in CR6m, TM6m is cleared to 00H and continues counting. At the same time, an interrupt request signal (INTTM6m) is generated.

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.

Remark m = 0, 1

<R> **Figure 7-17. 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 nm (CRnm).

To operate timer nm for square-wave output, settings must be made in the following sequence.

<1> When using timer 50, set P30 to output mode (PM30 = 0) and the P30 output latch to 0, respectively.
When using timer 60, set P31 to output mode (PM31 = 0) and the P31 output latch to 0, respectively.
When using timer 61, set P32 to output mode (PM32 = 0) and the P32 output latch to 0, respectively.

<2> Disable operation of timer counter nm (TMnm) (TCEnm = 0).

<3> Set a count clock for timer nm (see Figures 7-6, 7-7 and 7-9)

<R> <4> For timer 50, enable timer output of TO50 (TOE50 = 1).
For timer 60, enable timer output of TO60 (TOE600 = 1).
For timer 61, enable timer output of TO61 (TOE610 = 1).

<5> Set a count value in CRnm.

<6> Enable the operation of TMnm (TCEnm0 = 1).

When the count value of TMnm matches the value set in CRnm, the TOnm pin output will be inverted. Through application of this mechanism, square waves of any frequency can be output. As soon as a match occurs, TMnm is cleared to 00H and continues counting. At the same time, an interrupt request signal (INTTMnm) is generated.

The square-wave output is cleared to 0 by setting TCEnm to 0.

Tables 7-6 to 7-8 show the square-wave output range, and Figure 7-18 shows the timing of square-wave output.

Caution Be sure to stop the timer operation before overwriting the count clock with different data.

Remark nm = 50, 60, 61

Table 7-6. Square-Wave Output Range of Timer 50

TCL502	TCL501	TCL500	Minimum Pulse Width	Maximum Pulse Width	Resolution
0	0	0	$1/f_X$ (0.2 μs)	$2^8/f_X$ (51.2 μs)	$1/f_X$ (0.2 μs)
0	0	1	$2^3/f_X$ (1.6 μs)	$2^{11}/f_X$ (409.6 μs)	$2^3/f_X$ (1.6 μs)
0	1	0	$2^7/f_X$ (25.6 μs)	$2^{15}/f_X$ (6.55 ms)	$2^7/f_X$ (25.6 μs)
0	1	1	$1/f_{XT}$ (30.5 μs)	$2^9/f_{XT}$ (7.81 ms)	$1/f_{XT}$ (30.5 μs)
1	0	0	Input cycle of timer 60 match signal	Input cycle of timer 60 match signal $\times 2^8$	Input cycle of timer 60 match signal
1	0	1	Input cycle of timer 60 output	Input cycle of timer 60 output $\times 2^8$	Input cycle of timer 60 output

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

3. The parenthesized values apply to operation at f_X = 5.0 MHz or f_{XT} = 32.768 kHz.

Table 7-7. Square-Wave Output Range of Timer 60

TCL602	TCL601	TCL600	Minimum Pulse Width	Maximum Pulse Width	Resolution
0	0	0	$1/f_X$ (0.2 μs)	$2^8/f_X$ (51.2 μs)	$1/f_X$ (0.2 μs)
0	0	1	$2^2/f_X$ (0.8 μs)	$2^{10}/f_X$ (204 μs)	$2^2/f_X$ (0.8 μs)
0	1	0	f_{TMI} input cycle	f_{TMI} input cycle $\times 2^8$	f_{TMI} input cycle
0	1	1	$f_{TMI}/2$ input cycle	$f_{TMI}/2$ input cycle $\times 2^8$	$f_{TMI}/2$ input cycle
1	0	0	$f_{TMI}/2^2$ input cycle	$f_{TMI}/2^2$ input cycle $\times 2^8$	$f_{TMI}/2^2$ input cycle
1	0	1	$f_{TMI}/2^3$ input cycle	$f_{TMI}/2^3$ input cycle $\times 2^8$	$f_{TMI}/2^3$ input cycle

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{TMI}: External input clock frequency

3. The parenthesized values apply to operation at f_X = 5.0 MHz.

Table 7-8. Square-Wave Output Range of Timer 61

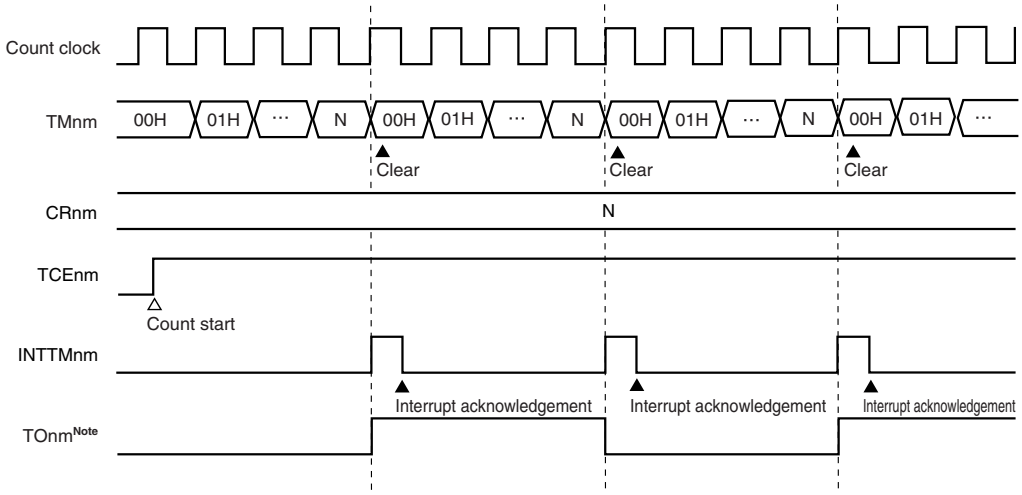
TCL612	TCL611	TCL610	Minimum Pulse Width	Maximum Pulse Width	Resolution
0	0	0	$1/f_X$ (0.2 μs)	$2^8/f_X$ (51.2 μs)	$1/f_X$ (0.2 μs)
0	0	1	$2^4/f_X$ (3.2 μs)	$2^{12}/f_X$ (819 μs)	$2^4/f_X$ (3.2 μs)
0	1	0	f_{TMI} input cycle	f_{TMI} input cycle $\times 2^8$	f_{TMI} input cycle
0	1	1	$f_{TMI}/2$ input cycle	$f_{TMI}/2$ input cycle $\times 2^8$	$f_{TMI}/2$ input cycle
1	0	0	$f_{TMI}/2^2$ input cycle	$f_{TMI}/2^2$ input cycle $\times 2^8$	$f_{TMI}/2^2$ input cycle
1	0	1	$f_{TMI}/2^3$ input cycle	$f_{TMI}/2^3$ input cycle $\times 2^8$	$f_{TMI}/2^3$ input cycle

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{TMI}: External input clock frequency

3. The parenthesized values apply to operation at f_X = 5.0 MHz.

Figure 7-18. Timing of Square-Wave Output with 8-Bit Resolution



Note The initial value of TOnm is low level when output is enabled.

Remark N = 00H to FFH

nm = 50, 60, 61

7.4.2 Operation as 16-bit timer counter

Timer 50 and timer 60 can be used as a 16-bit timer counter using cascade connection. In this case, 8-bit timer counter 50 (TM50) is the higher 8 bits and 8-bit timer counter 60 (TM60) is the lower 8 bits. 8-bit timer 60 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 50 (CR50) and 8-bit compare register 60 (CR60).

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 50 (TM50) and 8-bit timer counter 60 (TM60) (TCE50 = 0, TCE60 = 0).
<2> Disable timer output of TO60 (TOE600 = 0).
<3> Set the count clock for timer 60 (see Figure 7-7).
<4> Set the operation mode of timer 50 and timer 60 to 16-bit timer counter mode (see Figures 7-6 and 7-7).
<5> Set a count value in CR50 and CR60.
<6> Enable the operation of TM50 and TM60 (TCE60 = 1[Note]).

Note Start and clear of the timer in the 16-bit timer counter mode are controlled by TCE60 (the value of TCE50 is invalid).

When the count values of TM50 and TM60 match the values set in CR50 and CR60 respectively, both TM50 and TM60 are simultaneously cleared to 00H and continues counting. At the same time, an interrupt request signal (INTTM60) is generated (INTTM50 is not generated).

Table 7-9 shows interval time, and Figure 7-19 shows the timing of the interval timer operation.

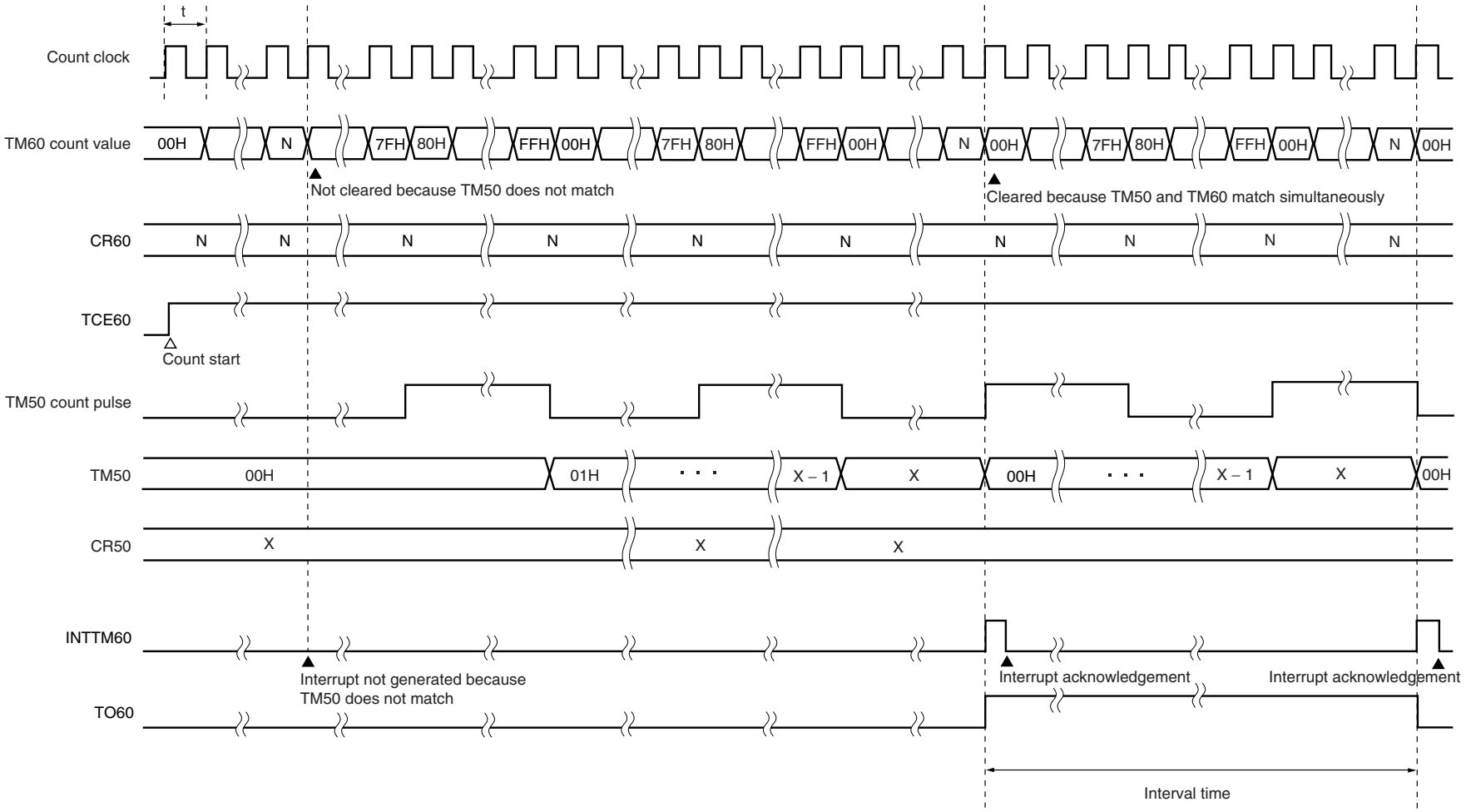
Cautions 1. Be sure to stop the timer operation before overwriting the count clock with different data.
2. In the 16-bit timer counter mode, TO50 cannot be used. Be sure to set TOE50 = 0 to disable TO50 output.

Table 7-9. Interval Time with 16-Bit Resolution

TCL602	TCL601	TCL600	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	0	$1/f_X$ (0.2 μs)	$2^{16}/f_X$ (13.1 ms)	$1/f_X$ (0.2 μs)
0	0	1	$2^2/f_X$ (0.8 μs)	$2^{18}/f_X$ (52.4 ms)	$2^2/f_X$ (0.8 μs)
0	1	0	f_{TMI} input cycle	f_{TMI} input cycle $\times 2^{16}$	f_{TMI} input cycle
0	1	1	$f_{TMI}/2$ input cycle	$f_{TMI}/2$ input cycle $\times 2^{16}$	$f_{TMI}/2$ input cycle
1	0	0	$f_{TMI}/2^2$ input cycle	$f_{TMI}/2^2$ input cycle $\times 2^{16}$	$f_{TMI}/2^2$ input cycle
1	0	1	$f_{TMI}/2^3$ input cycle	$f_{TMI}/2^3$ input cycle $\times 2^{16}$	$f_{TMI}/2^3$ input cycle

Remarks 1. f_X: Main system clock oscillation frequency
2. f_{TMI}: External input clock frequency
3. The parenthesized values apply to operation at f_X = 5.0 MHz.

Figure 7-19. 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 TMI60 pin by TM50 and TM60.

To operate as an external event counter with 16-bit resolution, settings must be made in the following sequence.

<1> Disable operation of TM50 and TM60 (TCE50 = 0, TCE60 = 0).
<2> Disable timer output of TO60 (TOE600 = 0).
<3> Set P31 to input mode (PM31 = 1).
<4> Select the external input clock for timer 60 (see Figure 7-7).
<5> Set the operation mode of timer 50 and timer 60 to 16-bit timer counter mode (see Figures 7-6 and 7-7).
<6> Set a count value in CR50 and CR60.
<7> Enable the operation of TM50 and TM60 (TCE60 = 1[Note]).

Note Start and clear of the timer in the 16-bit timer counter mode are controlled by TCE60 (the value of TCE50 is invalid).

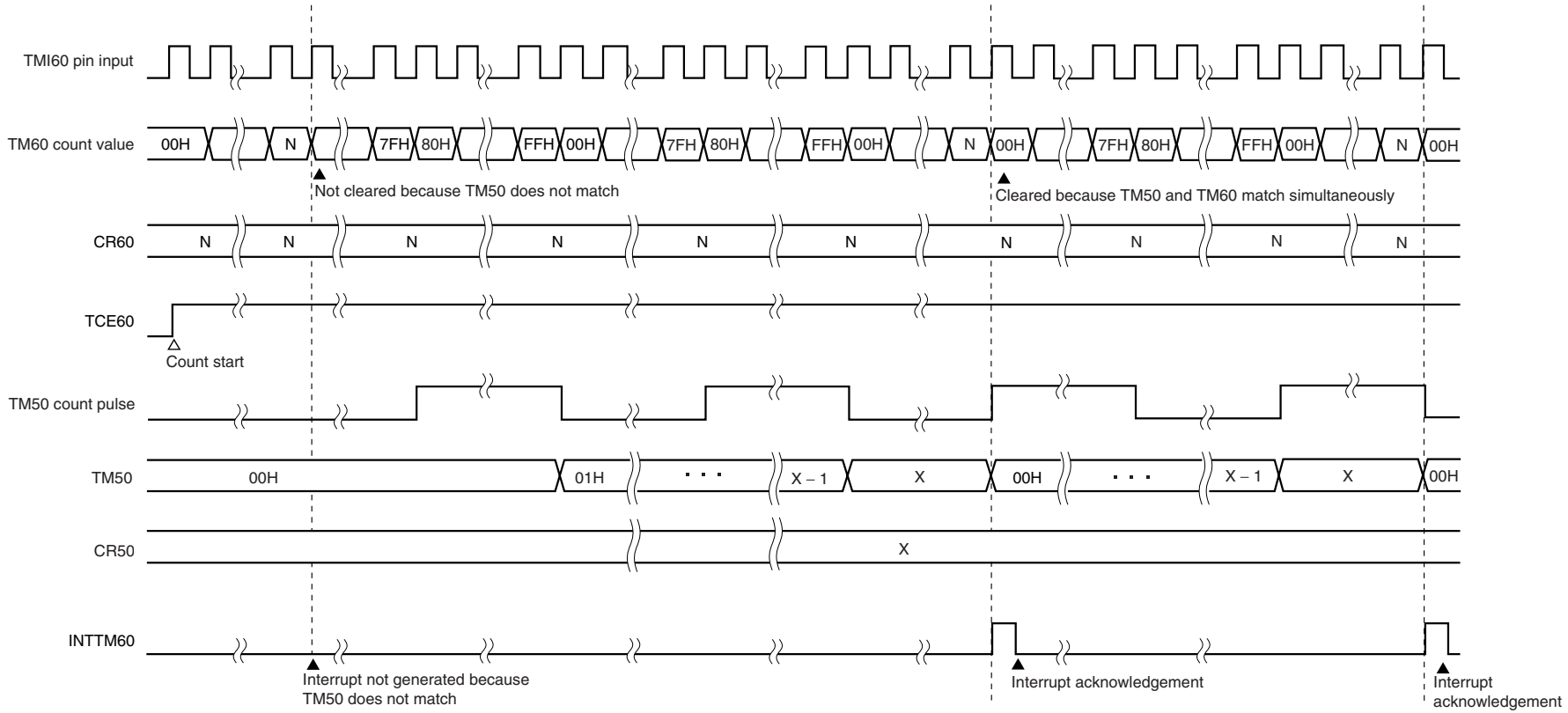
Each time the valid edge is input, the values of TM50 and TM60 are incremented.

When the count values of TM50 and TM60 simultaneously match the values set in CR50 and CR60 respectively, both TM50 and TM60 are cleared to 00H and continues counting. At the same time, an interrupt request signal (INTTM60) is generated (INTTM50 is not generated).

Figure 7-20 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 7-20. Timing of External Event Counter Operation with 16-Bit Resolution



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 CR50 and CR60.

To operate as a square-wave output with 16-bit resolution, settings must be made in the following sequence.

<1> Disable operation of TM50 and TM60 (TCE50 = 0, TCE60 = 0).
<2> Disable output of TO50 and TO60 (TOE50 = 0, TOE600 = 0).
<3> Set a count clock for timer 60. (see Figure 7-7)
<4> Set P31 to the output mode (PM31 = 0), set the P31 output latch to 0, and set TO60 to output enable (TOE600 = 1). (Use of TO50 is prohibited.)
<5> Set the operation mode of timer 50 and timer 60 to 16-bit timer counter mode (see Figures 7-6 and 7-7).
<6> Set count values in CR50 and CR60.
<7> Enable the operation of TM60 (TCE60 = 1[Note]).

Note Start and clear of the timer in the 16-bit timer counter mode are controlled by TCE60 (the value of TCE50 is invalid).

When the count values of TM50 and TM60 simultaneously match the values set in CR50 and CR60 respectively, the TO60 pin output will be inverted. Through application of this mechanism, square waves of any frequency can be output. As soon as a match occurs, TM50 and TM60 are cleared to 00H and continue counting. At the same time, an interrupt request signal (INTTM60) is generated (INTTM50 is not generated).

The square-wave output is cleared to 0 by setting TCE60 to 0.

Table 7-10 shows the square-wave output range, and Figure 7-21 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, TO50 cannot be used. Be sure to set TOE50 = 0 to disable TO50 output.

Table 7-10. Square-Wave Output Range with 16-Bit Resolution

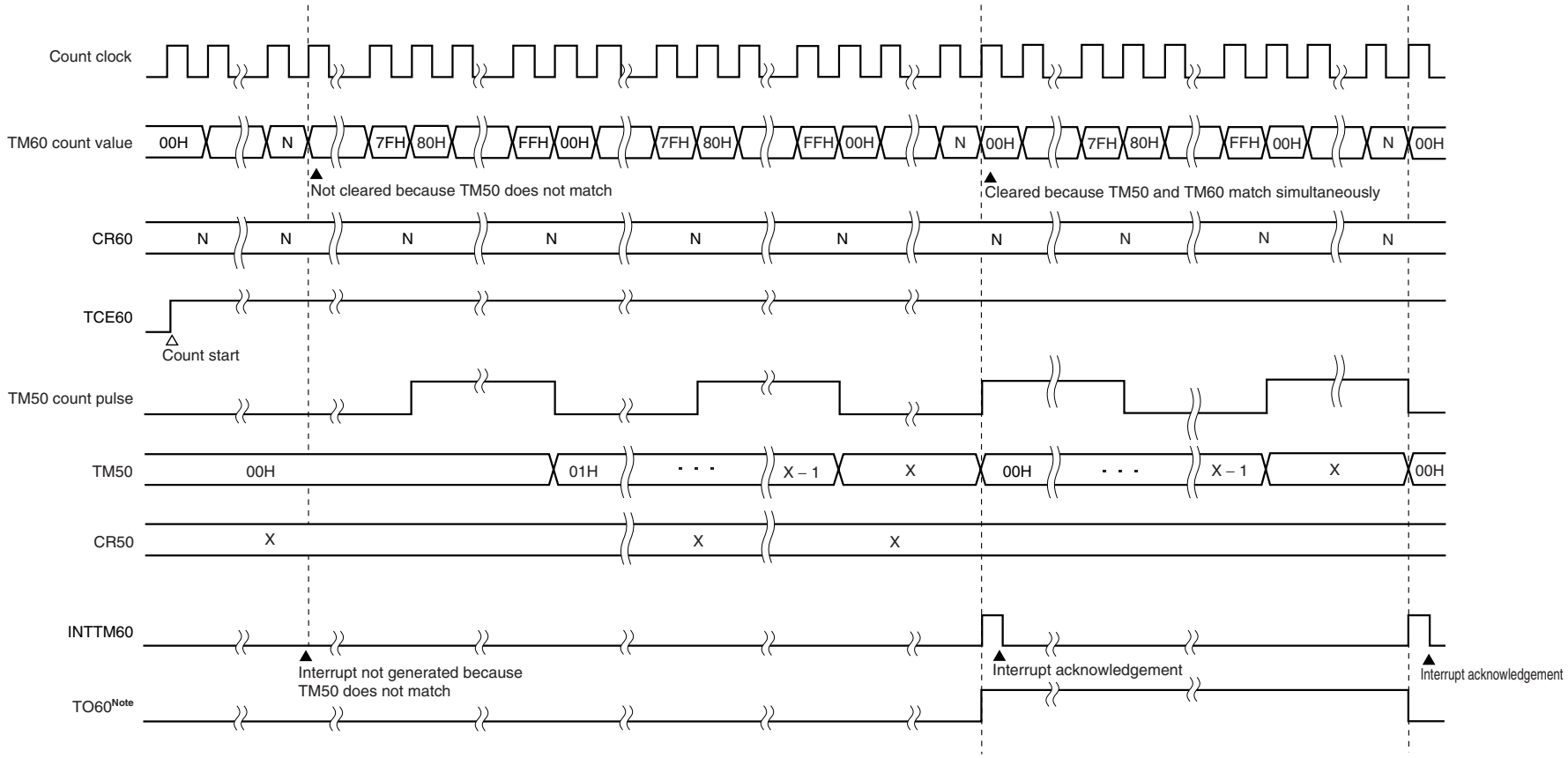
TCL602	TCL601	TCL600	Minimum Pulse Width	Maximum Pulse Width	Resolution
0	0	0	$1/f_X$ (0.2 μs)	$2^{16}/f_X$ (13.1 ms)	$1/f_X$ (0.2 μs)
0	0	1	$2^2/f_X$ (0.8 μs)	$2^{18}/f_X$ (52.4 ms)	$2^2/f_X$ (0.8 μs)
0	1	0	f_{TMI} input cycle	f_{TMI} input cycle $\times 2^{16}$	f_{TMI} input cycle
0	1	1	$f_{TMI}/2$ input cycle	$f_{TMI}/2$ input cycle $\times 2^{16}$	$f_{TMI}/2$ input cycle
1	0	0	$f_{TMI}/2^2$ input cycle	$f_{TMI}/2^2$ input cycle $\times 2^{16}$	$f_{TMI}/2^2$ input cycle
1	0	1	$f_{TMI}/2^3$ input cycle	$f_{TMI}/2^3$ input cycle $\times 2^{16}$	$f_{TMI}/2^3$ input cycle

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{TMI}: External input clock frequency

3. The parenthesized values apply to operation at f_X = 5.0 MHz.

Figure 7-21. Timing of Square-Wave Output with 16-Bit Resolution



Note The initial value of TO60 is low level when output is enabled.

Remark X = 00H to FFH, N = 00H to FFH

7.4.3 Operation as carrier generator

An arbitrary carrier clock generated by TM60 can be output in the cycle set in TM50.

To operate timer 50 and timer 60 as carrier generators, settings must be made in the following sequence.

<1> Disable operation of TM50 and TM60 (TCE50 = 0, TCE60 = 0).

<2> Disable timer output of TO50 and TO60 (TOE50 = 0, TOE600 = 0).

<3> Set count values in CR50, CR60, and CRH60.

<4> Set the operation mode of timer 50 and timer 60 to carrier generator mode (see Figures 7-6 and 7-7).

<5> Set the count clock for timer 50 and timer 60.

<6> Set remote control output to carrier pulse (RMC60 (bit 2 of carrier generator output control register 60 (TCA60)) = 0).
Input the required value to NRZB60 (bit 1 of TCA60) by program.
Input a value to NRZ60 (bit 0 of TCA60) before it is reloaded from NRZB60.

<7> Set P31 to the output mode (PM31 = 0), set the P31 output latch to 0, and set TO60 to output enable (TOE600 = 1).

<8> Enable the operation of TM50 and TM60 (TCE50 = 1, TCE60 = 1).

<R> <9> When the value of NRZB60 is transferred to NRZ60, input the value to be transferred to NRZ60 next time to NRZB60 after INTTM50 falling.

<R> <10> Generate the desired carrier signal by repeating <9>.

The operation of the carrier generator is as follows.

<1> When the count value of TM60 matches the value set in CR60, an interrupt request signal (INTTM60) is generated and output of timer 60 is inverted, which makes the compare register switch from CR60 to CRH60.

<2> After that, when the count value of TM60 matches the value set in CRH60, an interrupt request signal (INTTM60) is generated and output of timer 60 is inverted again, which makes the compare register switch from CRH60 to CR60.

<3> The carrier clock is generated by repeating <1> and <2> above.

<4> When the count value of TM50 matches the value set in CR50, an interrupt request signal (INTTM50) is generated. The rising edge of INTTM50 is the data reload signal of NRZB60 and is transferred to NRZ60.

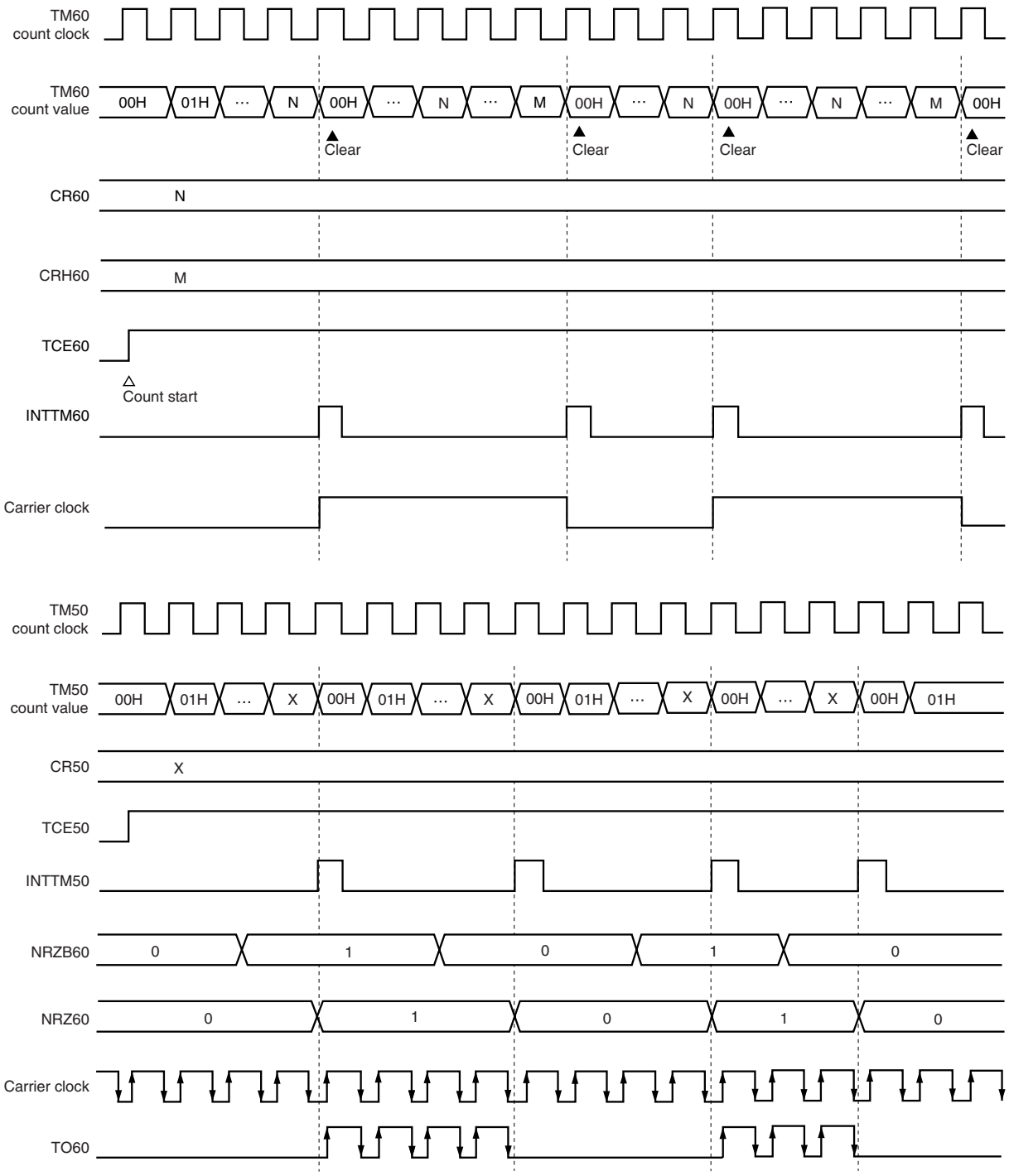
<5> When NRZ60 is 1, a carrier clock is output from the TO60 pin.

Cautions

1. **While timer 60 output is disabled (TOE600 = 0), TCA60 cannot be set with 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 NRZB60 because the previous value is not retained. In this case also a 1-bit memory manipulation instruction cannot be used while timer 60 output is disabled (TOE600 = 0). Be sure to use an 8-bit memory manipulation instruction.**
3. **When timer 60 output is enabled (TOE600 = 1), a write operation to NRZ is invalid. However, while the timer 50 interrupt signal (INTTM50) is high level, the NRZB60 value is immediately transferred to NRZ60 if TCA60 is rewritten. Rewrite TCA60 after waiting for half a clock of the TM50 count clock during INTTM50 interrupt servicing.**

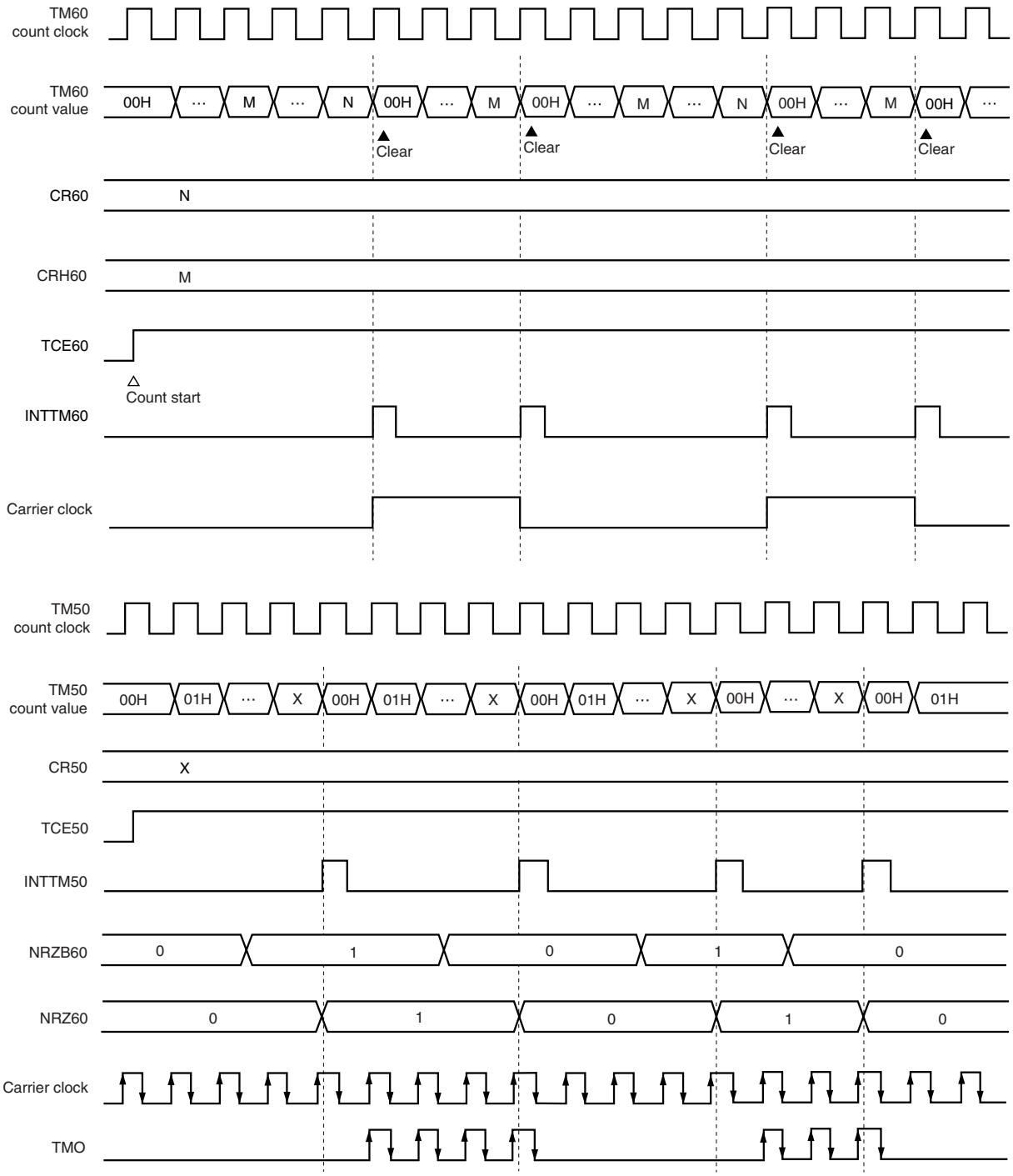
Figures 7-22 to 7-24 show the operation timing of the carrier generator.

<R> **Figure 7-22. Timing of Carrier Generator Operation (When CR60 = N, CRH60 = M (M > N))**



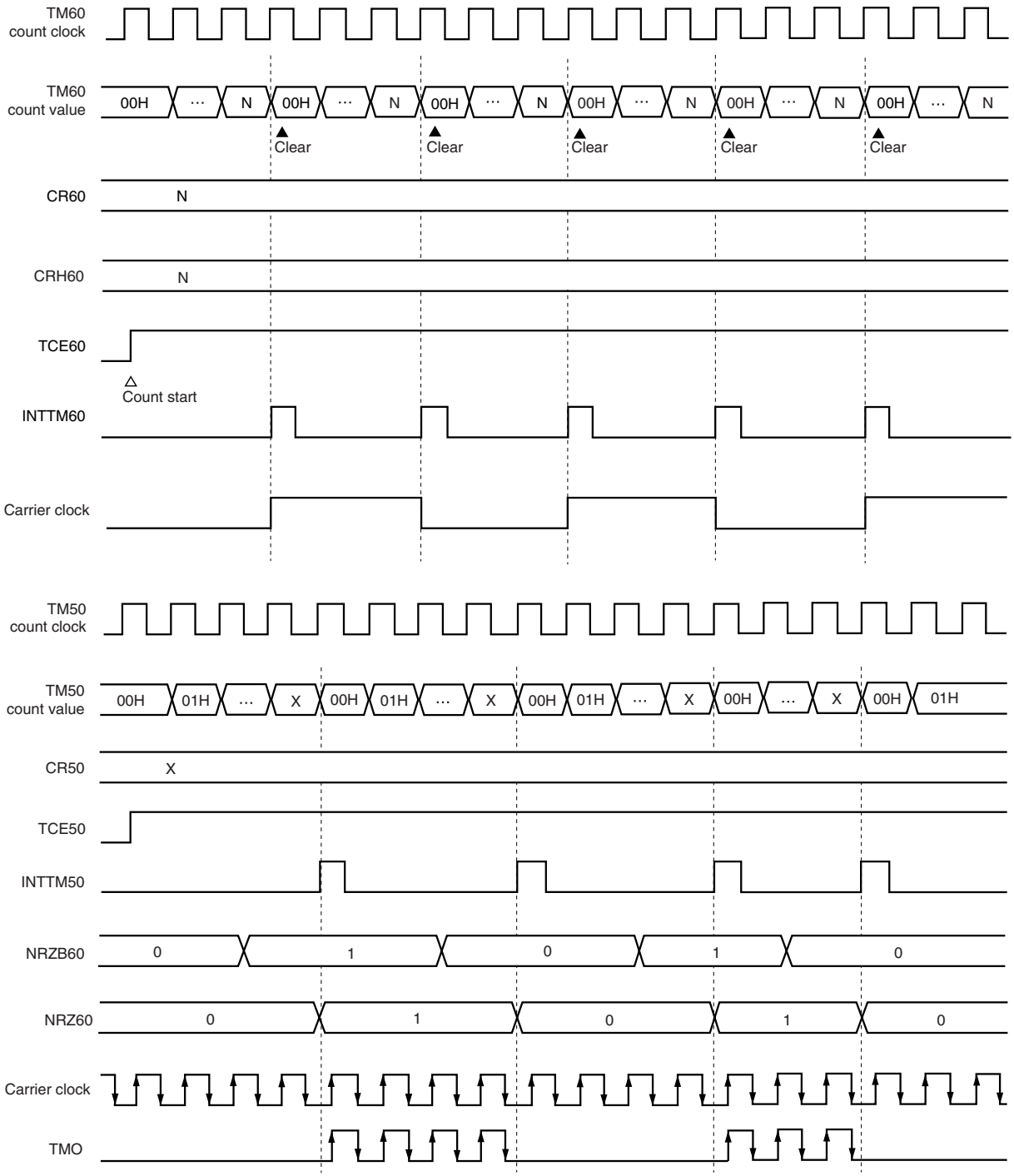
Remark 00H ≤ N < M ≤ FFH, X = 00H to FFH

<R> **Figure 7-23. Timing of Carrier Generator Operation (When CR60 = N, CRH60 = M (M < N))**



Remark 00H ≤ M < N ≤ FFH, X = 00H to FFH

Figure 7-24. Timing of Carrier Generator Operation (When CR60 = CRH60 = N)



Remark N = 00H to FFH, X = 00H to FFH

<R> **7.4.4 PWM output mode operation (timer 50)**

In the PWM output mode, TO50 becomes high level when TM50 overflows, and TO50 becomes low level when CR50 and TM50 match. It is thus possible to output a pulse with any duty ratio (free-running).

To operate timer 50 in the PWM output mode, settings must be made in the following sequence.

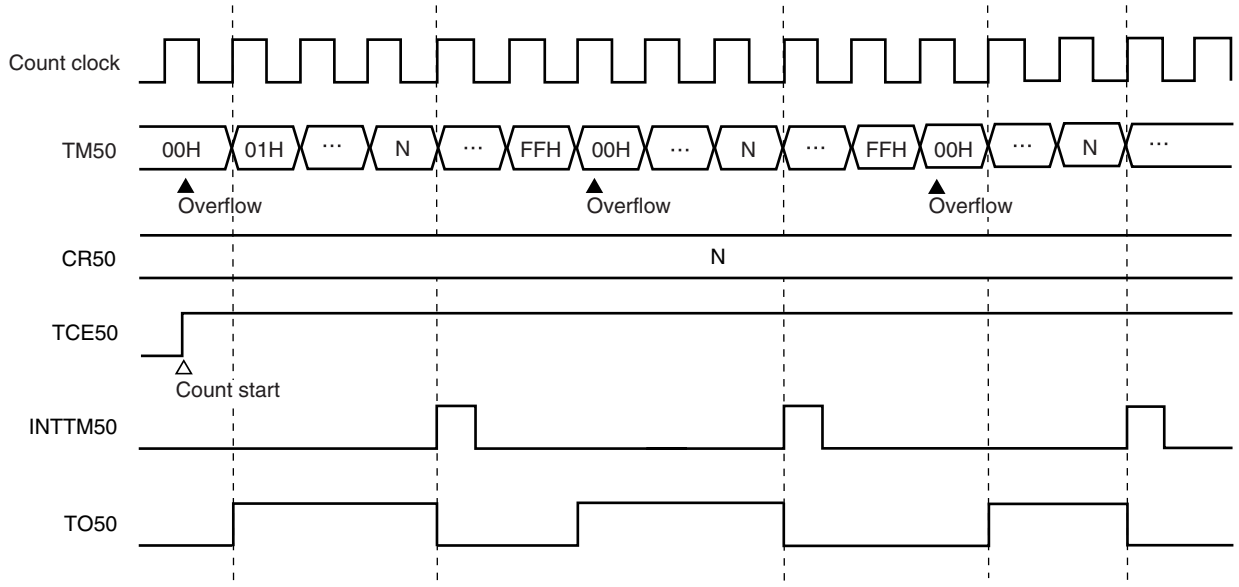
<1> Disable operation of TM50 (TCE50 = 0).
<2> Disable timer output of TO50 (TOE50 = 0).
<3> Set a count value to CR50.
<4> Set the operation mode of timer 50 to the PWM output mode (see Figure 7-6).
<5> Set the count clock for timer 50.
<6> Set P30 to the output mode (PM30 = 0) and the P30 output latch to 0 and enable timer output of TO50 (TOE50 = 1).
<7> Enable the operation of TM50 (TCE50 = 1).

The operation in the PWM output mode is as follows.

<1> When the count value of TM50 matches the value set in CR50, an interrupt request signal (INTTM50) is generated and a low level is output by the TO50. The TM50 continues counting without being cleared.
<2> TO50 outputs a high level when the TM50 overflows.

A pulse of any duty is output by repeating the above procedure. Figures 7-25 to 7-28 show the operation timing in the PWM output mode.

Figure 7-25. Operation Timing in PWM Output Mode (When Rising Edge Is Selected)

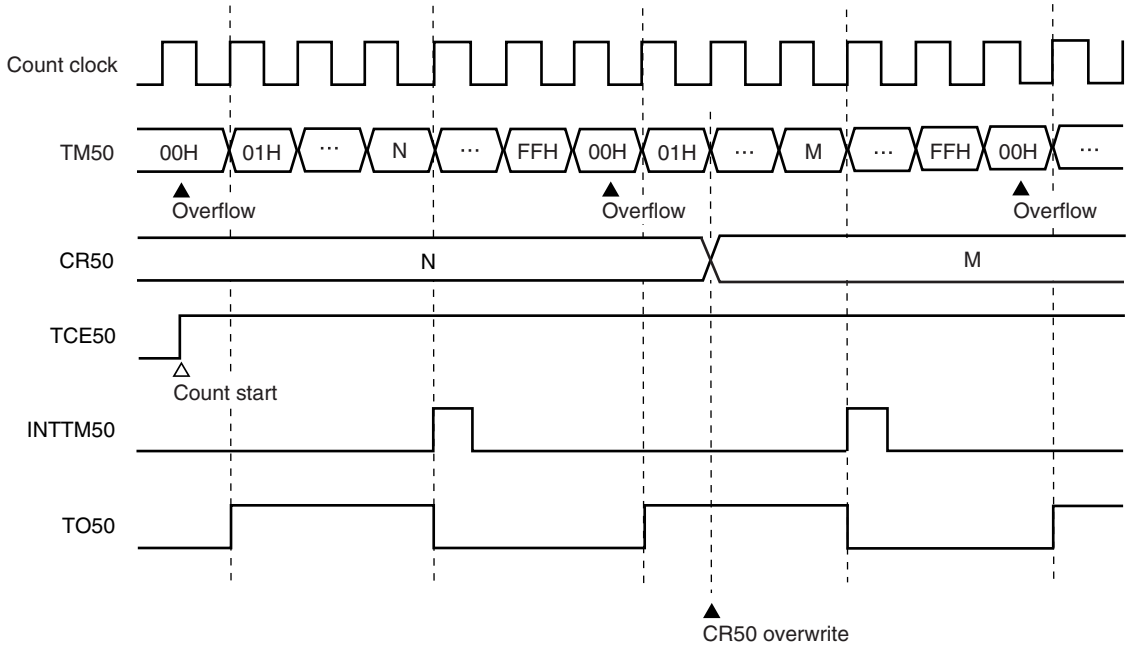


Caution When the rising edge is selected, do not set CR50 to 00H. If CR50 is set to 00H, PWM output may not be performed normally.

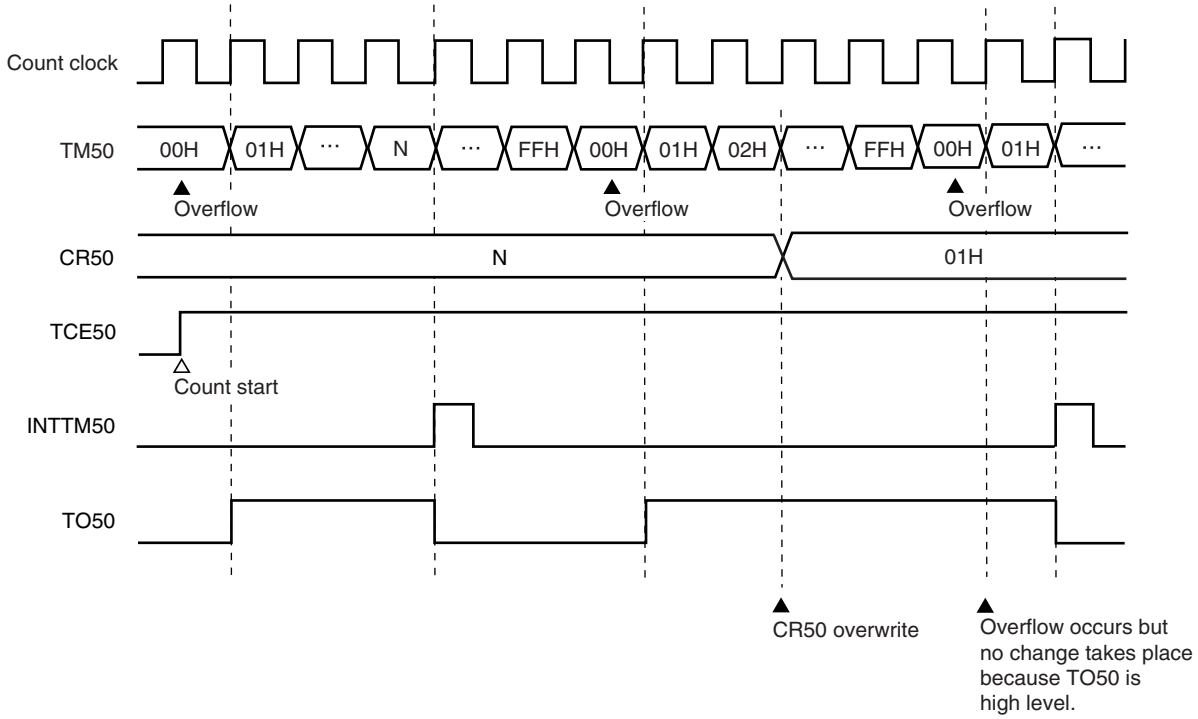
Remark N = 00H to FFH

Figure 7-26. Operation Timing When Overwriting CR50 (When Rising Edge Is Selected)

(1) When setting CR50 > TM50 after overflow



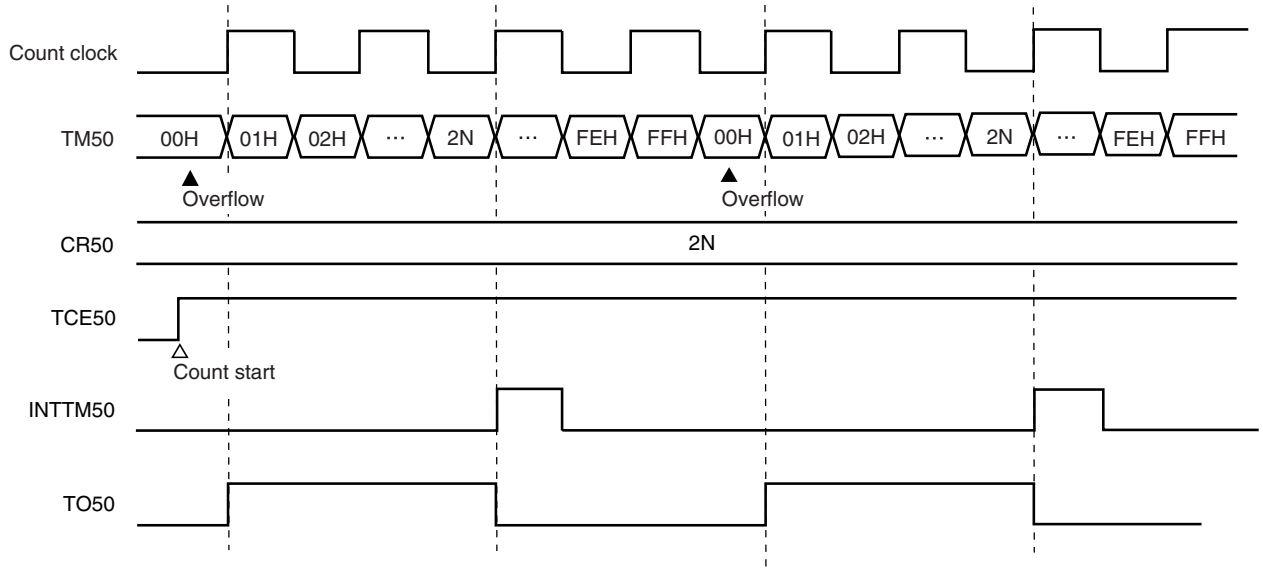
(2) When setting CR50 < TM50 after overflow



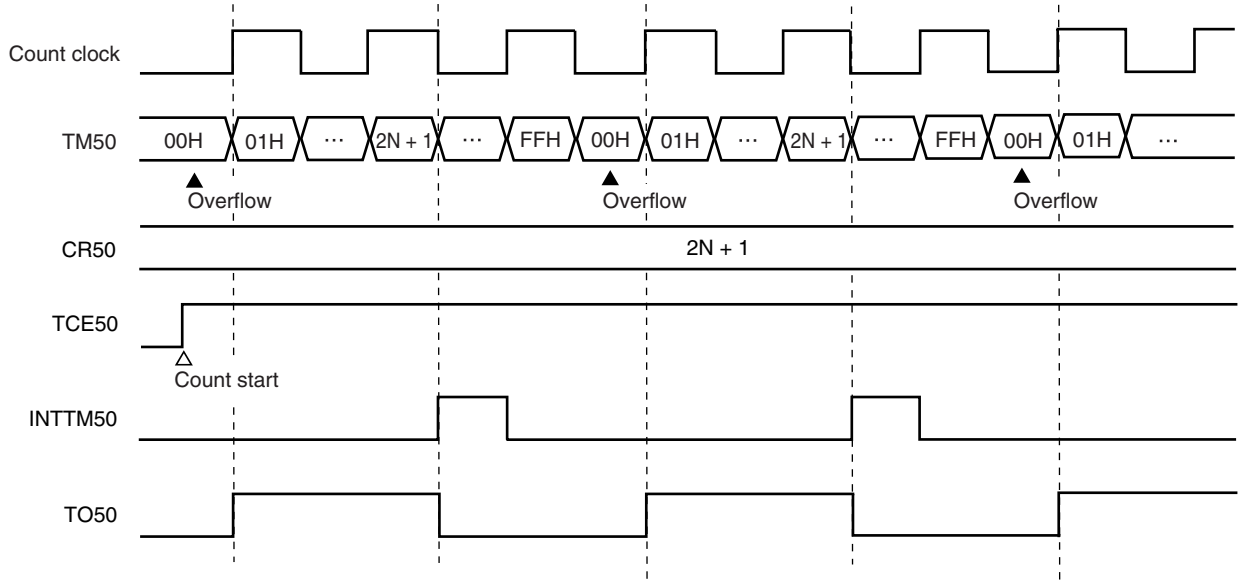
Remark N, M = 00H to FFH

Figure 7-27. Operation Timing in PWM Output Mode (When Both Edges Are Selected)

(1) CR50 = Even number



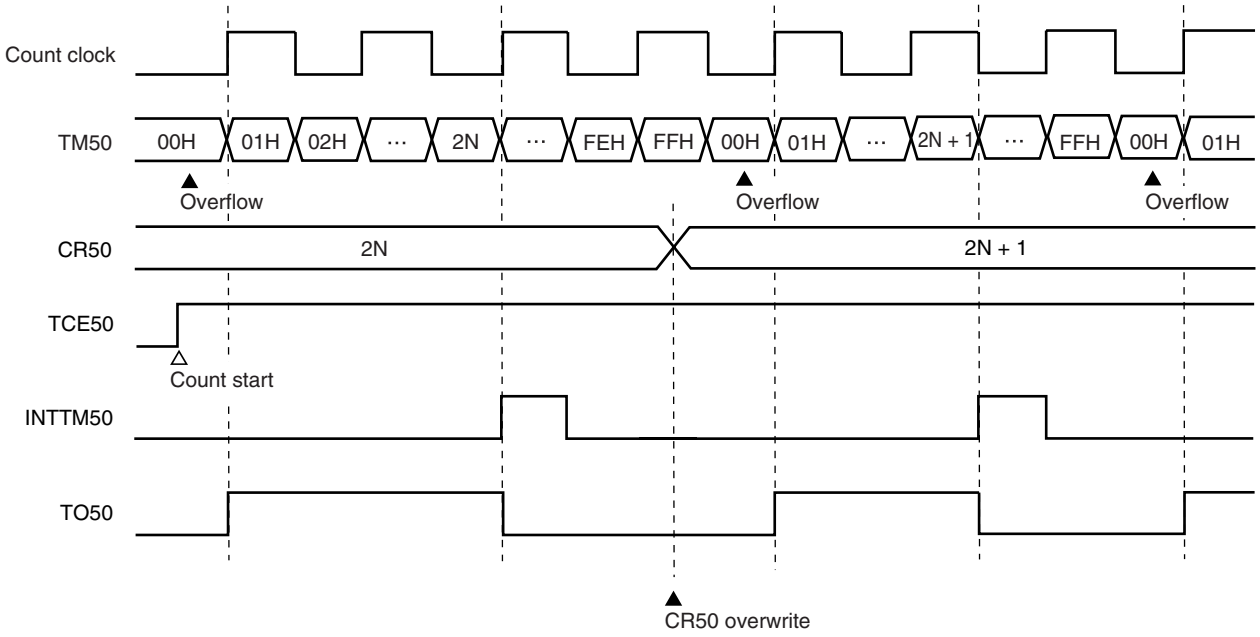
(2) When CR50 = Odd number



Caution When both edges are selected, do not set CR50 to 00H, 01H, and FFH. If CR50 is set to these values, PWM output may not be performed normally.

Remark N = 00H to FFH

Figure 7-28. Operation Timing in PWM Output Mode (When Both Edges Are Selected) (When CR50 Is Overwritten)



Remark N = 00H to FFH

<R> **7.4.5 Operation as PPG output mode (timer 60 and timer 61)**

In the PPG output mode, a pulse of any duty ratio can be output by setting a low-level width using CR6m and a high-level width using CRH6m.

To operate timer 6m in PPG output mode, settings must be made in the following sequence.

<1> Disable operation of TM6m (TCE6m = 0).
<2> Disable timer output of TO6m (TOE6m0 = 0).
<3> Set count values in CR6m and CRH6m.
<4> Set the operation mode of timer 6m to the PPG output mode (see Figures 7-7 and 7-9).
<5> Set the count clock for timer 6m.
<6> For timer 60, set P31 to the output mode (PM31 = 0) and the P31 output latch to 0.
For timer 61, set P32 to the output mode (PM32 = 0) and the P32 output latch to 0.
<7> Enable timer output of TO6m (TOE6m0 = 1).
<8> Enable the operation of TM6m (TCE6m = 1).

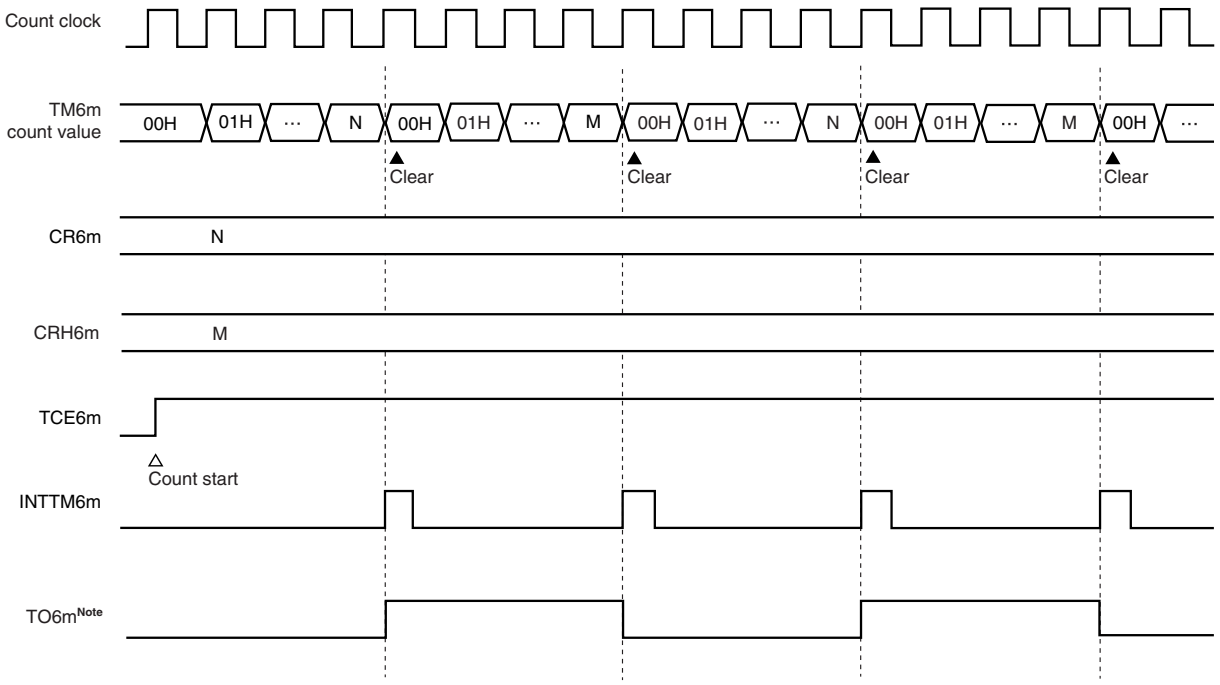
The operation in the PPG output mode is as follows.

<1> When the count value of TM6m matches the value set in CR6m, an interrupt request signal (INTTM6m) is generated and output of timer 6m is inverted, which makes the compare register switch from CR6m to CRH6m.
<2> A match between TM6m and CR6m clears the TM6m value to 00H and then counting starts again.
<3> After that, when the count value of TM6m matches the value set in CRH6m, an interrupt request signal (INTTM6m) is generated and output of timer 6m is inverted again, which makes the compare register switch from CRH6m to CR6m.
<4> A match between TM6m and CRH6m clears the TM6m value to 00H and then counting starts again.

A pulse of any duty ratio is output by repeating <1> to <4> above. Figures 7-29 and 7-30 show the operation timing in the PPG output mode.

Remark m = 0, 1

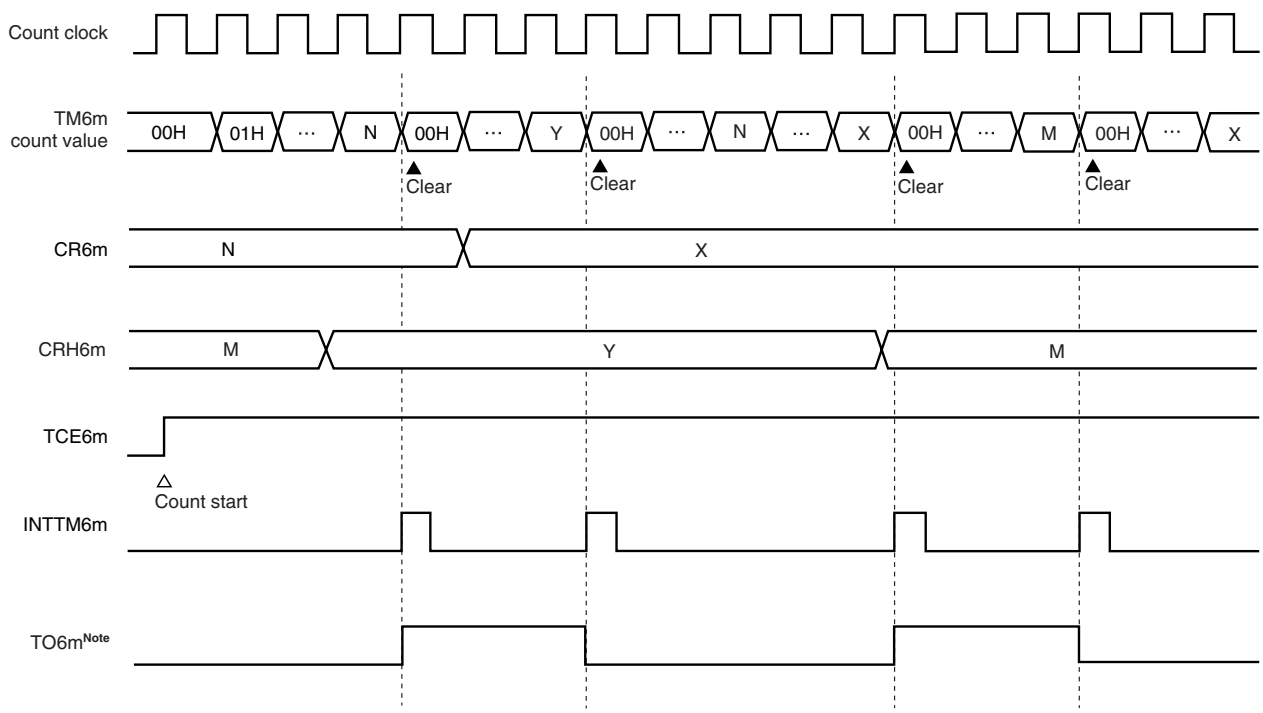
Figure 7-29. PPG Output Mode Timing (Basic Operation)



Note The initial value of TO6m is low level when output is enabled (TOE6m0 = 1).

Remark N, M = 00H to FFH
m = 0, 1

Figure 7-30. PPG Output Mode Timing (When CR6m and CRH6m Are Overwritten)



Note The initial value of TO6m is low level when output is enabled (TOE6m0 = 1).

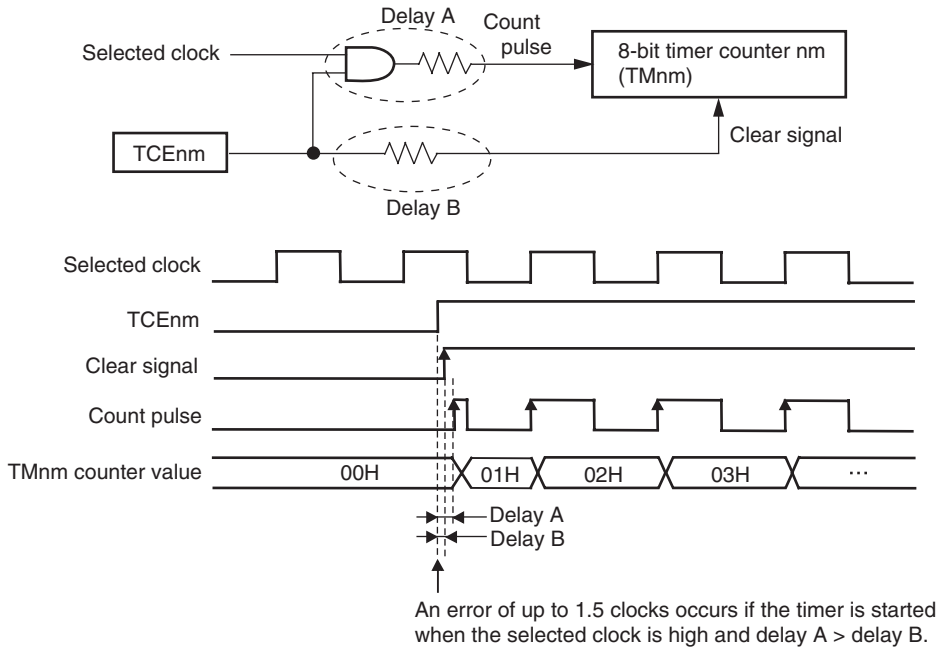
Remark N, M, X, Y = 00H to FFH
m = 0, 1

7.5 Cautions on Using 8-Bit Timers 50, 60, and 61

<R> **(1) Error on starting timer**

An error of up to 1.5 clocks is included in the time between the timer being started and a match signal being generated. This is because the rising edge is detected and the counter is incremented if the timer is started while the count clock is high (see **Figure 7-31**).

Figure 7-31. Case in Which Error of 1.5 Clocks (Max.) Occurs



Remark nm = 50, 60, 61

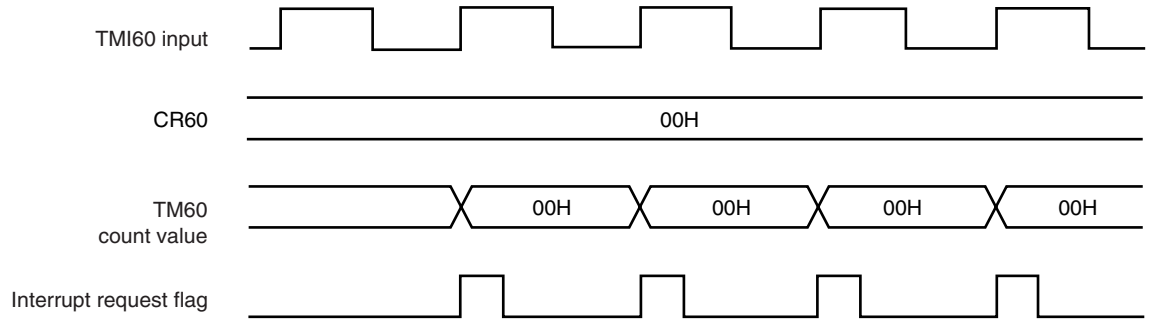
(2) Setting of 8-bit compare register nm

8-bit compare register nm (CRnm) can be set to 00H.

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

Remark nm= 50, 60, 61

Figure 7-32. Timing of Operation as External Event Counter (8-Bit Resolution)



CHAPTER 8 WATCH TIMER

8.1 Watch Timer Functions

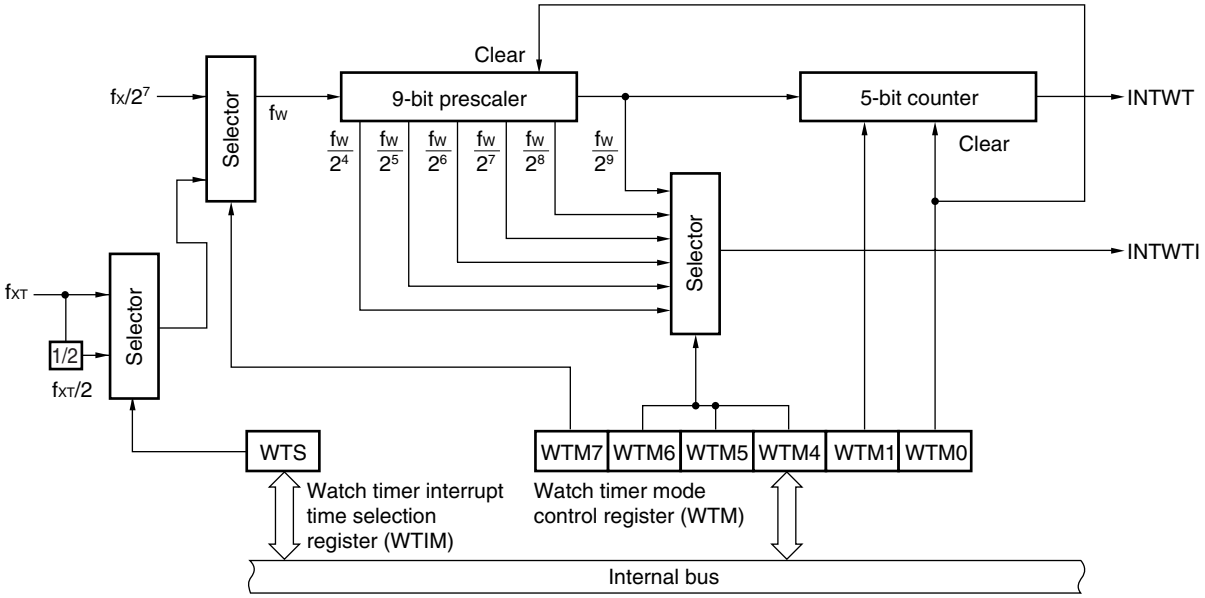
The watch timer has the following functions.

- Watch timer
- Interval timer

The watch and interval timers can be used at the same time.

Figure 8-1 shows a block diagram of the watch timer.

Figure 8-1. Block Diagram of Watch Timer



(1) Watch timer

An interrupt request (INTWT) occurs at an interval of 0.5 second when using either the 4.19 MHz main system clock or the 32.768 kHz subsystem clock.

Also, an interrupt request (INTWT) occurs at an interval of 1.0 seconds when using the 32.768 kHz subsystem clock via a setting in the watch timer interrupt time selection register (WTIM).

Caution An interval of 0.5 second cannot be created when using the 5.0 MHz main system clock. Instead, switch to the 32.768 kHz subsystem clock, and then create the 0.5-second interval.

(2) Interval timer

An interrupt request (INTWTI) occurs at preset intervals.

Table 8-1. Interval Time of Interval Timer

Interval Time	At f_X = 5.0 MHz	At f_X = 4.19 MHz	At f_{XT} = 32.768 kHz	At $f_{XT}/2$ = 16.384 kHz
$2^4 \times 1/f_W$	409.6 μs	488 μs	488 μs	976 μs
$2^5 \times 1/f_W$	819.2 μs	977 μs	977 μs	1.95 ms
$2^6 \times 1/f_W$	1.64 ms	1.95 ms	1.95 ms	3.90 ms
$2^7 \times 1/f_W$	3.28 ms	3.91 ms	3.91 ms	7.82 ms
$2^8 \times 1/f_W$	6.55 ms	7.81 ms	7.81 ms	15.6 ms
$2^9 \times 1/f_W$	13.1 ms	15.6 ms	15.6 ms	31.2 ms

Remarks 1. f_W: Watch timer clock frequency ($f_X/2^7$, f_{XT}, or $f_{XT}/2$)

2. f_X: Main system clock oscillation frequency

3. f_{XT}: Subsystem clock oscillation frequency

8.2 Watch Timer Configuration

The watch timer includes the following hardware.

Table 8-2. Configuration of Watch Timer

Item	Configuration
Counter	5 bits × 1
Prescaler	9 bits × 1
Control registers	Watch timer mode control register (WTM) Watch timer interrupt time selection register (WTIM)

8.3 Control Registers for Watch Timer

The watch timer is controlled by the following registers.

- Watch timer mode control register (WTM)
- Watch timer interrupt time selection register (WTIM)

(1) Watch timer mode control register (WTM)

This register is used to control the watch timer count clock, operation enable/disable status, prescaler interval time, and the 5-bit counter operation.

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

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

Figure 8-2. Format of Watch Timer Mode Control Register

Symbol	7	6	5	4	3	2	<1>	<0>	Address	After reset	R/W
WTM	WTM7	WTM6	WTM5	WTM4	0	0	WTM1	WTM0	FF4AH	00H	R/W

WTM7	Selection of watch timer count clock (f_W)
0	$f_X/2^7$ (39.1 kHz)
1	f_{XT} (32.768 kHz) or $f_{XT}/2$ (16.384 kHz)[Note]

WTM6	WTM5	WTM4	Selection of prescaler interval time
0	0	0	$2^4/f_W$
0	0	1	$2^5/f_W$
0	1	0	$2^6/f_W$
0	1	1	$2^7/f_W$
1	0	0	$2^8/f_W$
1	0	1	$2^9/f_W$
Other than above			Setting prohibited

WTM1	Control of 5-bit counter operation
0	Cleared after stopping operation
1	Start

WTM0	Watch timer operation enable/disable
0	Operation stopped (prescaler and timer are both cleared)
1	Operation enabled

Note This is the frequency (f_{XT} or $f_{XT}/2$) set via the watch timer interrupt time selection register (WTIM).

Remarks 1. f_W: Watch timer clock frequency ($f_X/2^7$, f_{XT}, or $f_{XT}/2$)

2. f_X: Main system clock oscillation frequency

3. f_{XT}: Subsystem clock oscillation frequency

4. The parenthesized values apply to operation at f_X = 5.0 MHz or f_{XT} = 32.768 kHz.

(2) Watch timer interrupt time selection 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 watch timer.

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

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

Figure 8-3. Format of Watch Timer Interrupt Time Selection Register

Symbol	7	6	5	4	3	2	1	<0>	Address	After reset	R/W
WTIM	0	0	0	0	0	0	0	WTS	FF4BH	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 (WTM7) of the watch timer mode control register (WTM) is 1.

Remark f_{XT}: Subsystem clock oscillation frequency

8.4 Watch Timer Operation

8.4.1 Operation as watch timer

The main system clock (4.19 MHz) or subsystem clock (32.768 kHz) is used to enable the watch timer to operate at 0.5-second intervals.

Also, an interrupt request (INTWT) occurs at an interval of 1.0 seconds when using the 32.768 kHz subsystem clock via a setting in the watch timer interrupt time selection register (WTIM).

The watch timer is used to generate an interrupt request at specified intervals.

By setting bits 0 and 1 (WTM0 and WTM1) of the watch timer mode control register (WTM) to 1, the watch timer starts counting. By setting them to 0, the 5-bit counter is cleared and the watch timer stops counting.

It is possible to start the watch timer from zero seconds by clearing WTM1 to 0 when the interval timer and watch timer operate at the same time. In this case, however, an error of up to $2^9 \times 1/f_W$ seconds may occur in the overflow (INTWT) after the zero-second start of the watch timer because the 9-bit prescaler is not cleared to 0.

8.4.2 Operation as interval timer

The interval timer is used to repeatedly generate an interrupt request at the interval specified by a preset count value.

The interval can be selected by bits 4 to 6 (WTM4 to WTM6) of the watch timer mode control register (WTM).

Table 8-3. Interval Time of Interval Timer

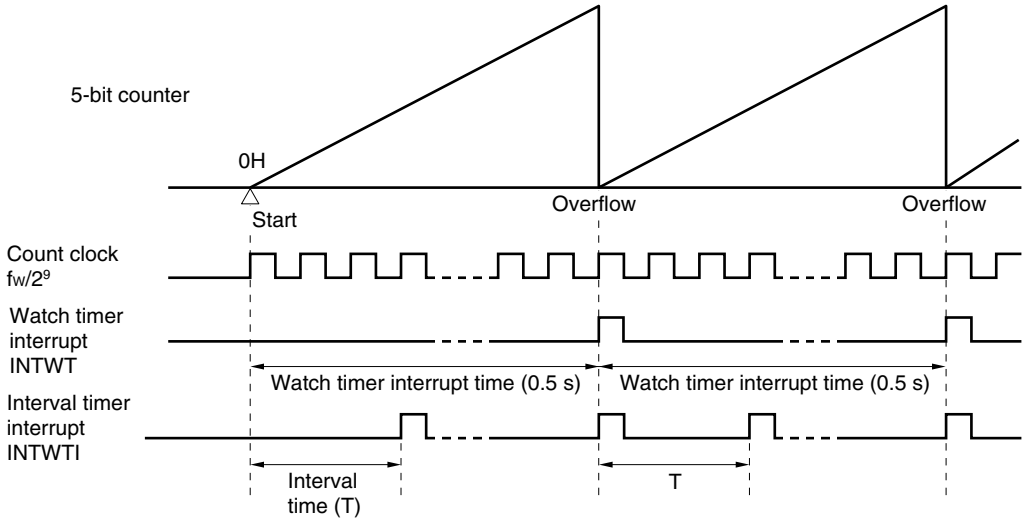
WTM6	WTM5	WTM4	Interval Time	At f_X = 5.0 MHz	At f_X = 4.19 MHz	At f_{XT} = 32.768 kHz	At f_{XT} = 16.384 kHz
0	0	0	$2^4 \times 1/f_W$	409.6 μs	488 μs	488 μs	976 μs
0	0	1	$2^5 \times 1/f_W$	819.2 μs	977 μs	977 μs	1.95 ms
0	1	0	$2^6 \times 1/f_W$	1.64 ms	1.95 ms	1.95 ms	3.90 ms
0	1	1	$2^7 \times 1/f_W$	3.28 ms	3.91 ms	3.91 ms	7.82 ms
1	0	0	$2^8 \times 1/f_W$	6.55 ms	7.81 ms	7.81 ms	15.6 ms
1	0	1	$2^9 \times 1/f_W$	13.1 ms	15.6 ms	15.6 ms	31.2 ms
Other than above			Setting prohibited				

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

3. f_W: Watch timer clock frequency

Figure 8-4. Watch Timer/Interval Timer Operation Timing



Caution When operation of the watch timer and 5-bit counter operation is enabled by setting bit 0 (WTM0) of the watch timer mode control register (WTM) to 1, the interval until the first interrupt request (INTWT) is generated after the register is set does not exactly match the watch timer interrupt time (0.5 s). This is because there is a delay of one 9-bit prescaler output cycle until the 5-bit counter starts counting. Subsequently, however, the INTWT signal is generated at the specified intervals.

Remarks 1. fw: Watch timer clock frequency

2. The parenthesized values apply to operation at fw = 32.768 kHz.

CHAPTER 9 WATCHDOG TIMER

9.1 Watchdog Timer Functions

The watchdog timer has the following functions.

- Watchdog timer
- Interval timer

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

(1) Watchdog timer

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

Table 9-1. Watchdog Timer Program Loop Detection Time

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

fx: Main system clock oscillation frequency

(2) Interval timer

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

Table 9-2. Interval Time

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

fx: Main system clock oscillation frequency

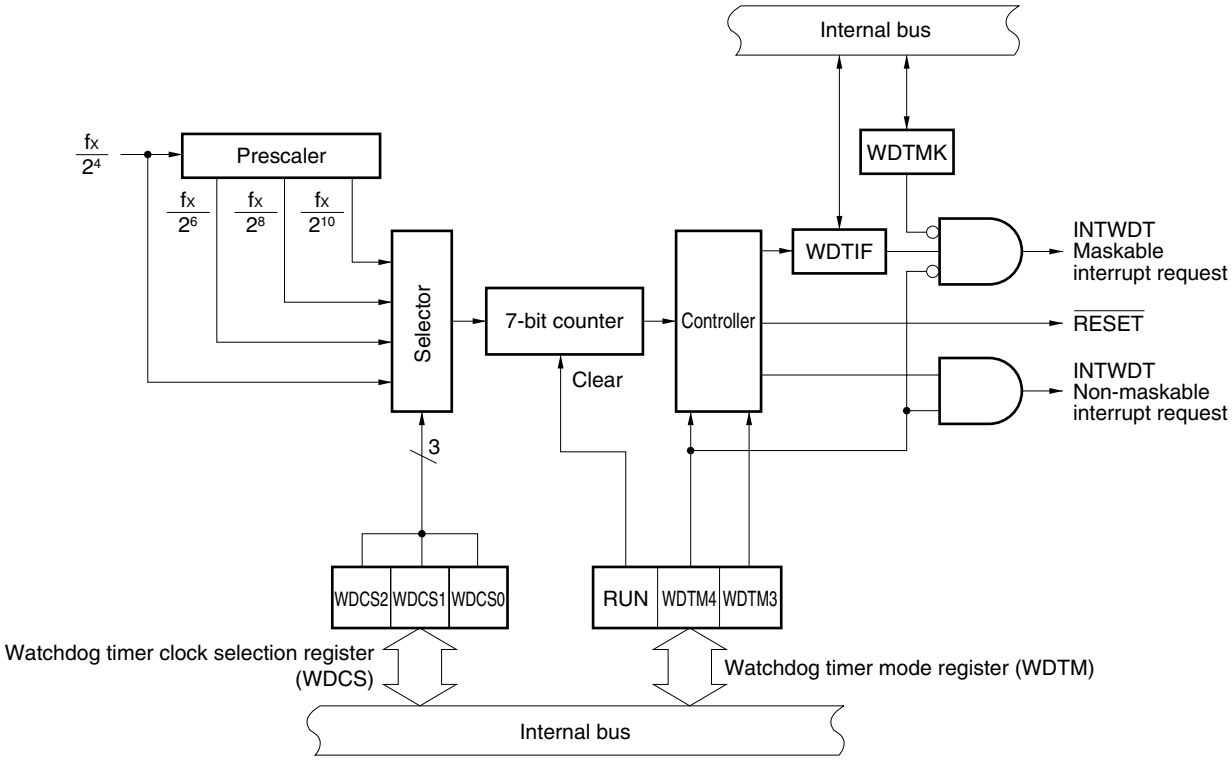
9.2 Watchdog Timer Configuration

The watchdog timer includes the following hardware.

Table 9-3. Configuration of Watchdog Timer

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

Figure 9-1. Block Diagram of Watchdog Timer



9.3 Watchdog Timer Control Registers

The watchdog timer is controlled by the following two registers.

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

(1) Watchdog timer clock selection register (WDCS)

This register sets the watchdog timer count clock.

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

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

Figure 9-2. Format of Watchdog Timer Clock Selection Register

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

WDCS2	WDCS1	WDCS0	Watchdog timer count clock selection	Interval
0	0	0	$f_X/2^4$ (312.5 kHz)	$2^{11}/f_X$ (410 μs)
0	1	0	$f_X/2^6$ (78.1 kHz)	$2^{13}/f_X$ (1.64 ms)
1	0	0	$f_X/2^8$ (19.5 kHz)	$2^{15}/f_X$ (6.55 ms)
1	1	0	$f_X/2^{10}$ (4.88 kHz)	$2^{17}/f_X$ (26.2 ms)
Other than above			Setting prohibited	

Remarks 1. f_X : Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 5.0 MHz.

(2) Watchdog timer mode register (WDTM)

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

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

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

Figure 9-3. Format of Watchdog Timer Mode Register

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

RUN	Watchdog timer operation selection[Note 1]
0	Stop counting.
1	Clear counter and start counting.

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

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

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

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

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

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

9.4 Watchdog Timer Operation

9.4.1 Operation as watchdog timer

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

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

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

Cautions 1. The actual program loop detection time may be up to 0.8% shorter than the set time.

2. When the subsystem clock is selected as the CPU clock, the watchdog timer count operation is stopped. Even when the main system clock continues oscillating in this case, watchdog timer count operation is stopped.

Table 9-4. Watchdog Timer Program Loop Detection Time

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

f_X: Main system clock oscillation frequency

9.4.2 Operation as interval timer

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

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

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

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

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

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

Table 9-5. Interval Time of Interval Timer

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

fx: Main system clock oscillation frequency

CHAPTER 10 8-BIT A/D CONVERTER

10.1 8-Bit A/D Converter Functions

The 8-bit A/D converter is a 8-bit resolution converter used to convert analog inputs into digital signals. This converter can control eight channels (ANI0 to ANI7) of analog inputs.

A/D conversion can only be started by software.

One of analog inputs ANI0 to ANI7 is selected for A/D conversion. A/D conversion is performed repeatedly, with an interrupt request (INTAD0) being issued each time A/D conversion is complete.

A conversion operation is also possible using the subsystem clock multiplied by 4 (131 kHz).

Caution A/D conversion is stopped in the HALT and STOP modes.

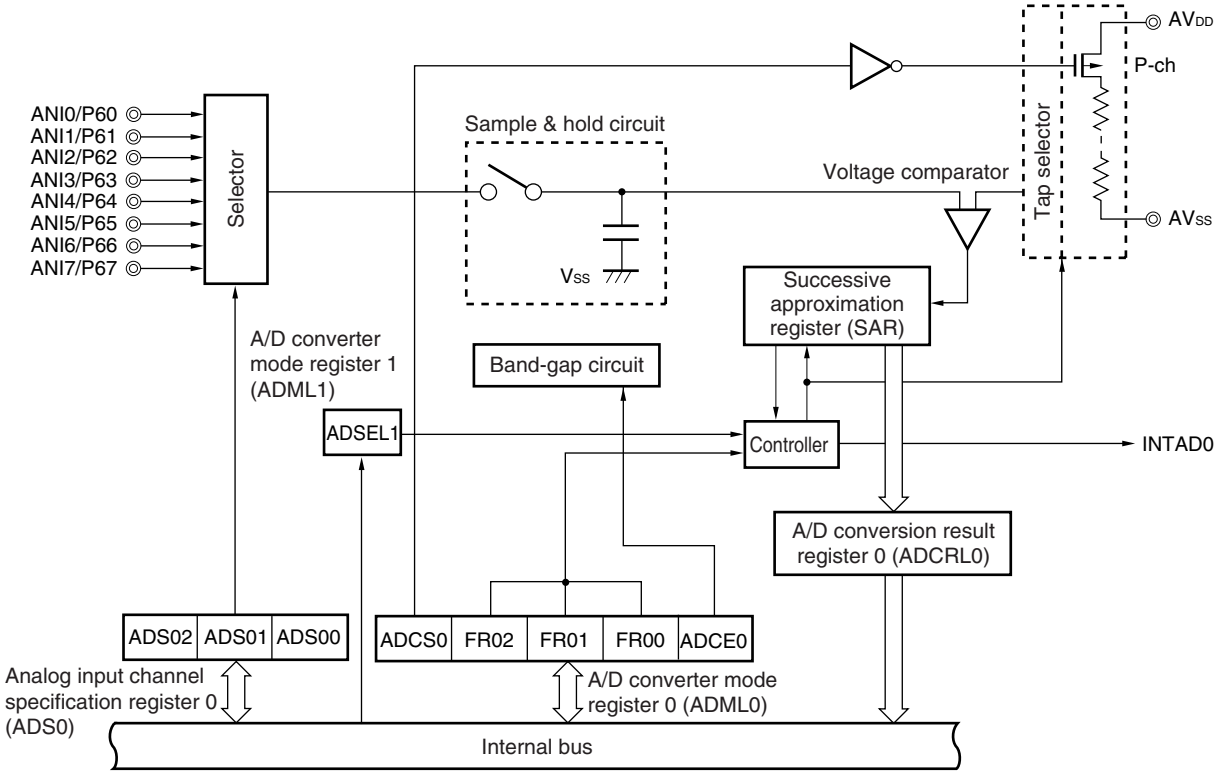
10.2 8-Bit A/D Converter Configuration

The 8-bit A/D converter includes the following hardware.

Table 10-1. Configuration of 8-Bit A/D Converter

Item	Configuration
Analog inputs	8 channels (ANI0 to ANI7)
Registers	Successive approximation register (SAR) A/D conversion result register 0 (ADCRL0)
Control registers	A/D converter mode register 0 (ADML0) A/D converter mode register 1 (ADML1) Analog input channel specification register 0 (ADS0)

<R> **Figure 10-1. Block Diagram of 8-Bit A/D Converter**



(1) Successive approximation register (SAR)

The SAR receives the result of comparing an analog input voltage and a voltage at a voltage tap (comparison voltage), received from the series resistor string, starting from the most significant bit (MSB).

Upon receiving all the bits, down to the least significant bit (LSB), that is, upon the completion of A/D conversion, the SAR sends its contents to A/D conversion result register 0 (ADCRL0).

(2) A/D conversion result register 0 (ADCRL0)

ADCRL0 is an 8-bit register that holds the result of A/D conversion. Each time A/D conversion ends, the conversion result in the successive approximation register is loaded into ADCRL0. The results are stored in ADCRL0 from the highest bit.

ADCRL0 can be read with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ADCRL0 to 00H.

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
ADCRL0									FF15H	00H	R

(3) Sample & hold circuit

The sample & hold circuit samples consecutive analog inputs from the input circuit, one by one, and sends them to the voltage comparator. The sampled analog input voltage is held during A/D conversion.

(4) Voltage comparator

The voltage comparator compares an analog input with the voltage output by the series resistor string.

(5) Series resistor string

The series resistor string is configured between AV_{DD} and AV_{SS}. It generates the reference voltages against which analog inputs are compared.

(6) ANI0 to ANI7

The ANI0 to ANI7 pins are the 8-channel analog input pins for the A/D converter. They are used to receive the analog signals for A/D conversion.

Caution Do not supply the ANI0 to ANI7 pins with voltages that fall outside the rated range. If a voltage greater than or equal to AV_{DD} or less than or equal to AV_{SS} (even if within the absolute maximum ratings) is applied to any of these pins, the conversion value for the corresponding channel will be undefined. Furthermore, the conversion values for the other channels may also be affected.

(7) AV_{SS} pin

The AV_{SS} pin is the ground potential pin for the A/D converter. This pin must be held at the same potential as the V_{SS} pin, even while the A/D converter is not being used.

(8) AV_{DD} pin

The AV_{DD} pin is the analog power supply pin for the A/D converter. This pin must be held at the same potential as the V_{DD} pin, even while the A/D converter is not being used.

(9) Band-gap circuit

The band-gap circuit activates the reference voltage inside the comparator prior to A/D conversion. Start conversion after 14 μs have elapsed following the activation of the band-gap circuit.

10.3 8-Bit A/D Converter Control Registers

The 8-bit A/D converter is controlled by the following three registers.

- A/D converter mode register 0 (ADML0)
- A/D converter mode register 1 (ADML1)
- Analog input channel specification register 0 (ADS0)

(1) A/D converter mode register 0 (ADML0)

ADML0 specifies the A/D conversion time for analog inputs. It also specifies whether to enable conversion.

ADML0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ADML0 to 00H.

Figure 10-2. Format of A/D Converter Mode Register 0

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
ADML0	ADCS0	0	FR02	FR01	FR00	0	0	ADCE0	FF80H	00H	R/W

ADCS0	A/D conversion control
0	Conversion disabled
1	Conversion enabled

ADSEL1	FR02	FR01	FR00	A/D conversion time selection[Note 1]
0	0	0	0	144/fx (28.8 μs)
0	0	0	1	120/fx (24.0 μs)
0	0	1	0	96/fx (19.2 μs)
0	1	0	0	72/fx (14.4 μs)
0	1	0	1	60/fx (Setting prohibited[Note 2])
0	1	1	0	48/fx (Setting prohibited[Note 2])
1	×	×	×	A/D conversion using subsystem clock multiplied by 4[Note 3] (conversion time is undefined)
Other than above				Setting prohibited

ADCE0	Control of band-gap circuit
0	Band-gap circuit stopped
1	Band-gap circuit operating

Notes 1. The selection of the A/D conversion time is set using a combination of both the ADML0 and ADML1 registers. Be sure to set these bits so that the A/D conversion time is at least 14 μs.

2. These bit combinations must not be set, as the A/D conversion time will fall below 14 μs at fx = 5.0 MHz.

3. When using the subsystem clock multiplied by 4, enable the ×4 multiplication circuit using a mask option or the subclock selection register.

Cautions 1. Start conversion (ADCS0 = 1) after 14 μs have elapsed following the setting of ADCE0. If ADCE0 is not used, the conversion result immediately after the setting of ADCS0 is undefined.

2. The conversion result may be undefined after ADCS0 has been cleared to 0. To read the conversion result, perform the read operation during A/D conversion. If the conversion result needs to be read after A/D conversion has been stopped, stop the A/D conversion operation before the end of the next A/D conversion.

3. Always set bits 1, 2, and 6 to 0.

Remarks 1. fx: Main system clock oscillation frequency

2. The parenthesized values apply to operation at fx = 5.0 MHz.

(2) A/D converter mode register 1 (ADML1)

This register is used to perform A/D conversion using the subsystem clock multiplied by 4.

ADML1 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ADML1 to 00H.

Figure 10-3. Format of A/D Converter Mode Register 1

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
ADML1	ADSEL1	0	0	0	0	0	0	0	FF81H	00H	R/W

ADSEL1	FR02	FR01	FR00	A/D conversion time selection[Note 1]
0	0	0	0	144/fx (28.8 μs)
0	0	0	1	120/fx (24.0 μs)
0	0	1	0	96/fx (19.2 μs)
0	1	0	0	72/fx (14.4 μs)
0	1	0	1	60/fx (Setting prohibited[Note 2])
0	1	1	0	48/fx (Setting prohibited[Note 2])
1	×	×	×	A/D conversion using subsystem clock multiplied by 4[Note 3] (conversion time is undefined)
Other than above				Setting prohibited

Notes 1. The selection of the A/D conversion time is set using a combination of both the ADML0 and ADML1 registers. Be sure to set these bits so that the A/D conversion time is at least 14 μs.

2. These bit combinations must not be set, as the A/D conversion time will fall below 14 μs at fx = 5.0 MHz.

3. When using the subsystem clock multiplied by 4, enable the ×4 multiplication circuit using a mask option or the subclock selection register.

Remarks 1. fx: Main system clock oscillation frequency

2. The parenthesized values apply to operation at fx = 5.0 MHz.

(3) Analog input channel specification register 0 (ADS0)

ADS0 specifies the port used to input the analog voltage to be converted to a digital signal.

ADS0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ADS0 to 00H.

Figure 10-4. Format of Analog Input Channel Specification Register 0

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
ADS0	0	0	0	0	0	ADS02	ADS01	ADS00	FF84H	00H	R/W

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

Caution Bits 3 to 7 must be set to 0.

10.4 8-Bit A/D Converter Operation

10.4.1 Basic operation of 8-bit A/D converter

<1> Bit 0 of A/D converter mode register 0 (ADML0) is set (ADCE0 = 1).

<2> Select a channel for A/D conversion, using analog input channel specification register 0 (ADS0).

<3> When 14 μs or more have elapsed after ADCE0 was set, set bit 7 of ADML0 (ADCS0 = 1). The voltage supplied to the selected analog input channel is sampled using the sample & hold circuit.

<4> After sampling continues for a certain period of time, the sample & hold circuit is put on hold to keep the input analog voltage until A/D conversion is completed.

<5> Bit 7 of the successive approximation register (SAR) is set. The series resistor string tap voltage at the tap selector is set to half of AV_{DD}.

<6> The series resistor string tap voltage is compared with the analog input voltage using the voltage comparator. If the analog input voltage is higher than half of AV_{DD}, the MSB of SAR is left set. If it is lower than half of AV_{DD}, the MSB is reset.

<7> Bit 6 of SAR is set automatically, and comparison shifts to the next stage. The next tap voltage of the series resistor string is selected according to bit 7, which reflects the previous comparison result, as follows.

- Bit 7 = 1: Three quarters of AV_{DD}
- Bit 7 = 0: One quarter of AV_{DD}

The tap voltage is compared with the analog input voltage. Bit 6 is set or reset according to the result of comparison.

- Analog input voltage ≥ tap voltage: Bit 6 = 1
- Analog input voltage < tap voltage: Bit 6 = 0

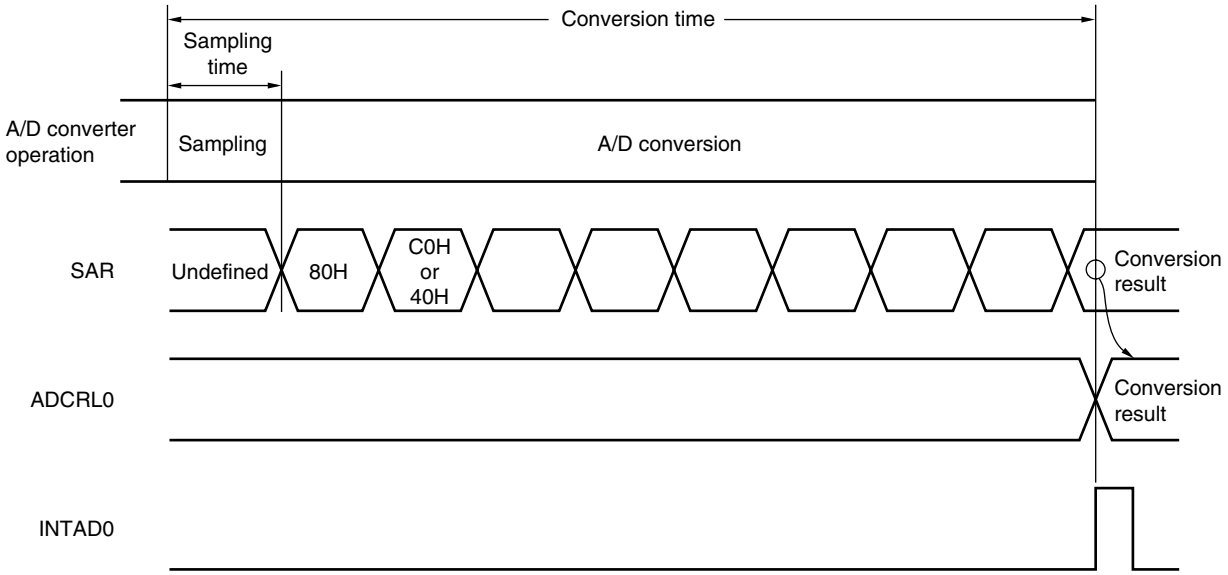
<8> Comparison is repeated until bit 0 of SAR is reached.

<9> When comparison is completed for all of the 8 bits, a significant digital result is left in SAR. This value is sent to and latched in A/D conversion result register 0 (ADCRL0). At the same time, it is possible to generate an A/D conversion end interrupt request (INTAD0).

Cautions 1. Start conversion (ADCS0 = 1) after 14 μs have elapsed following the setting of ADCE0. If ADCE0 is not used, the conversion result immediately after the setting of ADCS0 is undefined.

2. In standby mode, A/D converter operation is stopped.

Figure 10-5. Basic Operation of 8-Bit A/D Converter



A/D conversion continues until bit 7 (ADCS0) of A/D converter mode register 0 (ADML0) is reset (0) by software.

If an attempt is made to write to ADML0 or analog input channel specification register 0 (ADS0) during A/D conversion, the A/D conversion in progress is canceled. In this case, A/D conversion is restarted from the beginning, if ADCS0 is set (1).

$\overline{\text{RESET}}$ input clears A/D conversion result register 0 (ADCRL0) to 00H.

10.4.2 Input voltage and conversion result

The relationship between the analog input voltage at the analog input pins (ANI0 to ANI7) and the A/D conversion result (A/D conversion result register 0 (ADCRL0)) is represented by:

$$ADCRL0 = INT\left(\frac{V_{IN}}{AV_{DD}} \times 256 + 0.5\right)$$

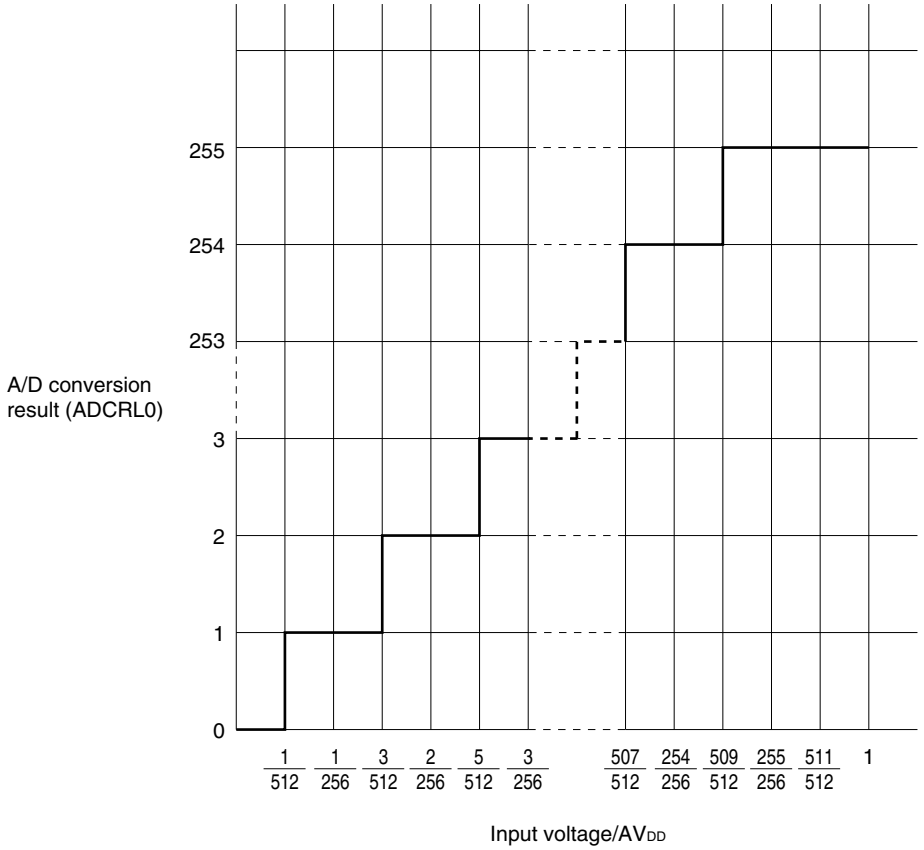
or

$$(ADCRL0 - 0.5) \times \frac{AV_{DD}}{256} \leq V_{IN} < (ADCRL0 + 0.5) \times \frac{AV_{DD}}{256}$$

INT( ): Function that returns the integer part of a parenthesized value
V_{IN}: Analog input voltage
AV_{DD}: Supply voltage for the A/D converter
ADCRL0: Value in A/D conversion result register 0 (ADCRL0)

Figure 10-6 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 10-6. Relationship Between Analog Input Voltage and A/D Conversion Result



10.4.3 Operation mode of 8-bit A/D converter

The A/D converter is initially in select mode. In this mode, analog input channel specification register 0 (ADS0) is used to select an analog input channel from ANI0 to ANI7 for A/D conversion.

A/D conversion can be started only by software, that is, by setting A/D converter mode register 0 (ADML0).

The A/D conversion result is saved to A/D conversion result register 0 (ADCRL0). At the same time, an interrupt request signal (INTAD0) is generated.

- **Software-started A/D conversion**

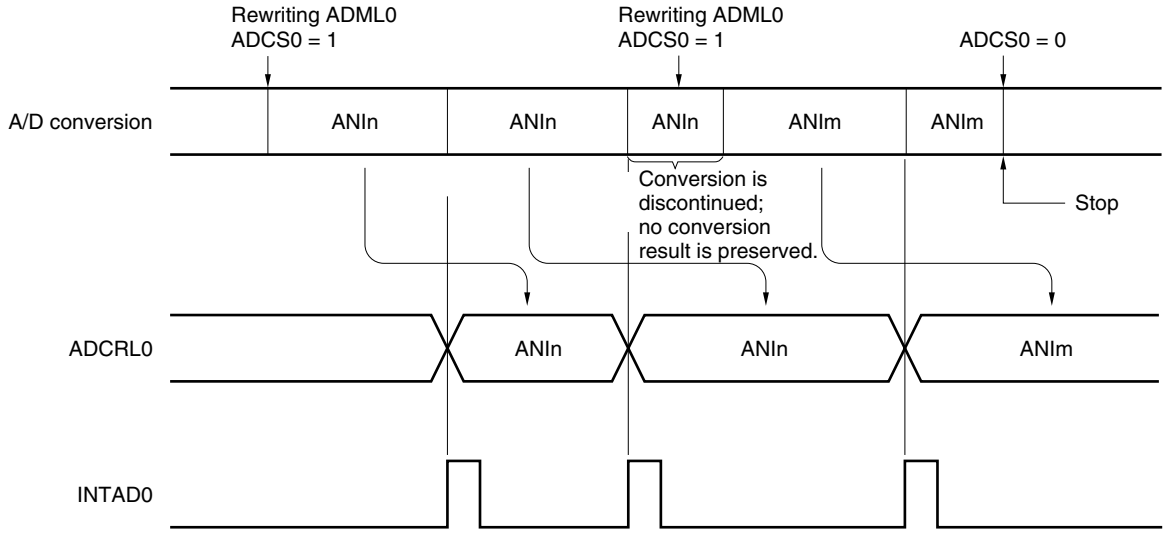
  Setting bit 7 (ADCS0) of A/D converter mode register 0 (ADML0) to 1 triggers A/D conversion for the voltage applied to the analog input pin specified by analog input channel specification register 0 (ADS0).

  Upon completion of A/D conversion, the conversion result is saved to A/D conversion result register 0 (ADCRL0). At the same time, an interrupt request signal (INTAD0) is generated. Once A/D conversion is activated and completed, another session of A/D conversion is started. A/D conversion is repeated until new data is written to ADML0.

  If data where ADCS0 is 1 is written to ADML0 again during A/D conversion, the A/D conversion in progress is discontinued, and a new session of A/D conversion begins for the new data.

  If data where ADCS0 is 0 is written to ADML0 again during A/D conversion, A/D conversion is stopped immediately.

Figure 10-7. Software-Started A/D Conversion



Remarks 1. n = 0 to 7

2. m = 0 to 7

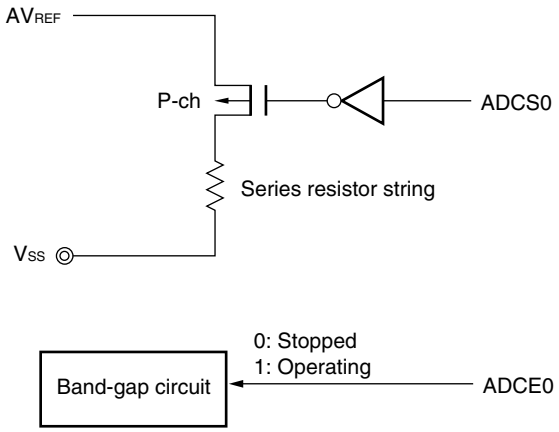
10.5 Cautions Related to 8-Bit A/D Converter

<R> **(1) Power consumption in standby mode**

In standby mode, the A/D converter stops operation. Clearing bit 7 (ADCS0) and bit 0 (ADCE0) of A/D converter mode register 0 (ADML0) to 0 can reduce the power consumption.

Figure 10-8 shows how to reduce the power consumption in standby mode.

Figure 10-8. How to Reduce Power Consumption in Standby Mode



(2) Input range for pins ANI0 to ANI7

Be sure to keep the input voltage at ANI0 to ANI7 within the rating. If a voltage greater than or equal to AV_{DD} or less than or equal to AV_{SS} (even within the absolute maximum ratings) is input into a conversion channel, the conversion output of the channel becomes undefined, which may affect the conversion output of the other channels.

(3) Conflict

<1> Conflict between writing to A/D conversion result register 0 (ADCRL0) at the end of conversion and reading from ADCRL0 using instruction

Reading from ADCRL0 takes precedence. After reading, the new conversion result is written to ADCRL0.

<2> Conflict between writing to ADCRL0 at the end of conversion and writing to A/D converter mode register 0 (ADML0) or analog input channel specification register 0 (ADS0)

Writing to ADML0 or ADS0 takes precedence. ADCRL0 is not written to. No A/D conversion end interrupt request signal (INTAD0) is generated.

(4) Conversion result immediately after start of A/D conversion

If the band-gap circuit is not used (ADCE0 = 0) or conversion is started before 14 μs have elapsed following the setting of ADCE, only the first A/D conversion value immediately after A/D conversion has been started is undefined. Poll the A/D conversion end interrupt request (INTAD0), drop the first conversion result and use the second and subsequent conversion results. When 14 μs have elapsed following the activation of the band-gap circuit (ADCE0 = 1), the first conversion value is normal.

(5) Timing of undefined A/D conversion result

The A/D conversion value may become undefined if the timing of the completion of A/D conversion and that to stop the A/D conversion operation conflict. Therefore, read the A/D conversion result while the A/D conversion operation is in progress. To read the A/D conversion result after the A/D conversion operation has been stopped, stop the A/D conversion operation before the next conversion operation is completed. Figures 10-9 and 10-10 show the timing at which the conversion result is read.

Figure 10-9. Conversion Result Read Timing (if Conversion Result Is Undefined)

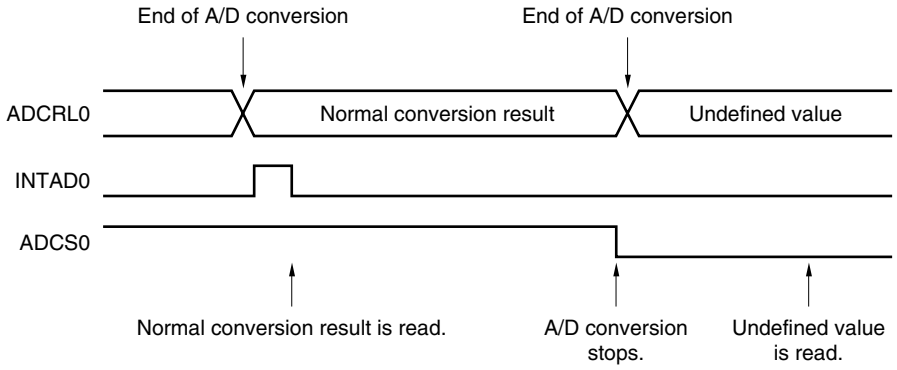
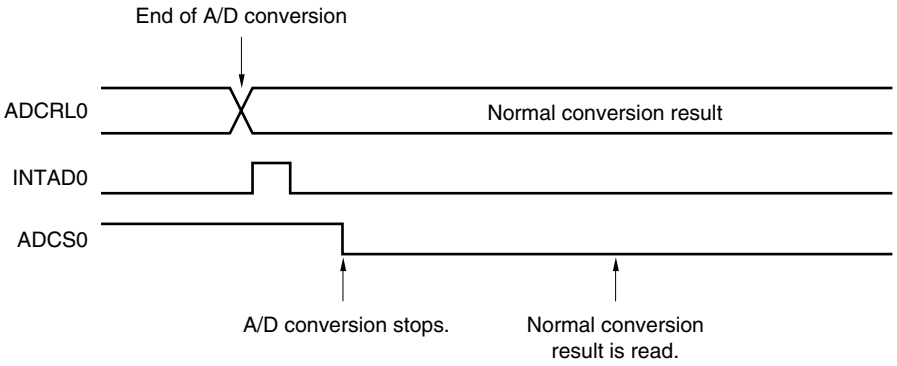


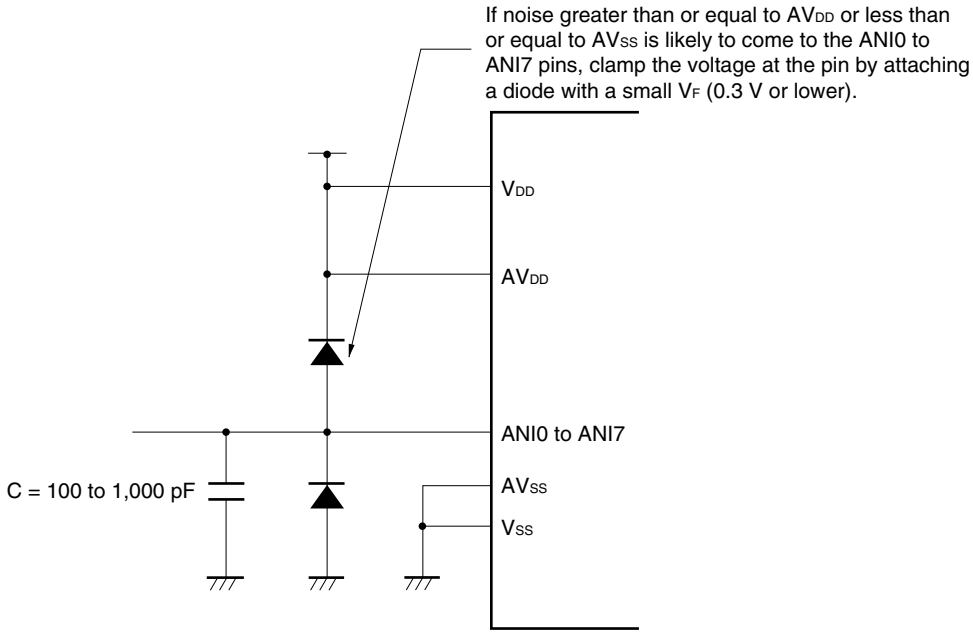
Figure 10-10. Conversion Result Read Timing (if Conversion Result Is Normal)



(6) Noise prevention

To maintain a resolution of 8 bits, watch for noise at the AV_{DD} and ANI0 to ANI7 pins. The higher the output impedance of the analog input source, the larger the effect by noise. To reduce noise, attach an external capacitor to the relevant pins as shown in Figure 10-11.

Figure 10-11. Analog Input Pin Treatment



(7) ANI0 to ANI7

The analog input pins (ANI0 to ANI7) are alternate-function pins. They are also used as port pins (P60 to P67).

If any of ANI0 to ANI7 has been selected for A/D conversion, do not execute input instructions for the ports; otherwise the conversion resolution may be reduced.

If a digital pulse is applied to a pin adjacent to the analog input pins during A/D conversion, coupling noise may occur that prevents an A/D conversion result from being obtained as expected. Avoid applying a digital pulse to pins adjacent to the analog input pins during A/D conversion.

(8) Input impedance of ANI0 to ANI7 pins

This A/D converter charges the internal sampling capacitor for about 1/10 of the conversion time, and performs sampling.

Therefore at times other than sampling, only the leak current flows. During sampling, the current for charging the capacitor also flows, so the input impedance fluctuates and has no meaning.

However, to ensure adequate sampling, it is recommended that the output impedance of the analog input source be set to 10 kΩ or lower, or a capacitor of about 100 pF be connected to the ANI0 to ANI7 pins (see **Figure 10-11**).

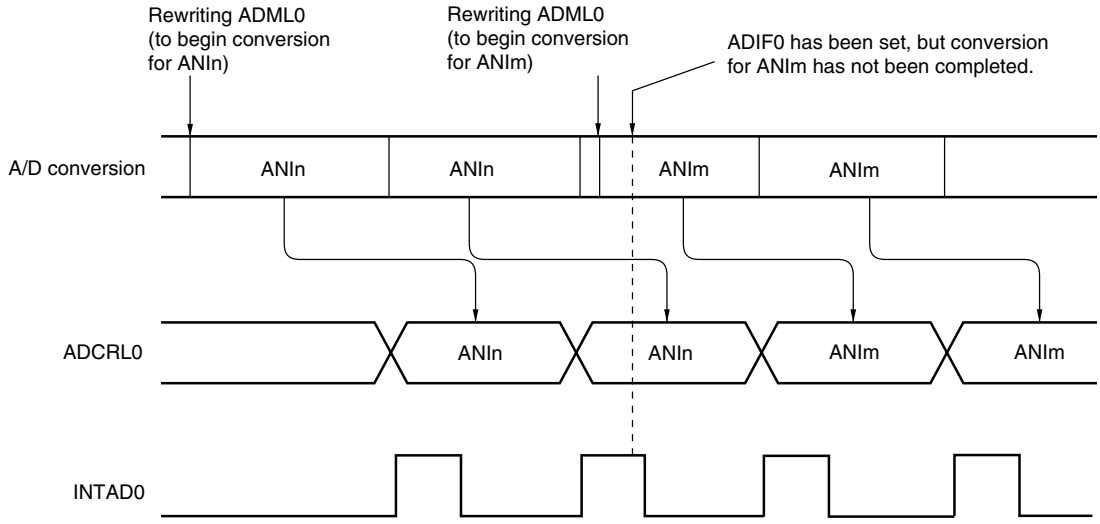
(9) Interrupt request flag (ADIF0)

Changing the contents of A/D converter mode register 0 (ADML0) does not clear the interrupt request flag (ADIF0).

If the analog input pins are changed during A/D conversion, therefore, the A/D conversion result and the conversion end interrupt request flag may be set for the previous analog input immediately before rewriting ADML0. In this case, ADIF0 may already be set if it is read-accessed immediately after ADML0 is rewritten, even when A/D conversion has not been completed for the new analog input.

In addition, when A/D conversion is restarted, ADIF0 must be cleared beforehand.

Figure 10-12. A/D Conversion End Interrupt Request Generation Timing



Remarks 1. n = 0 to 7

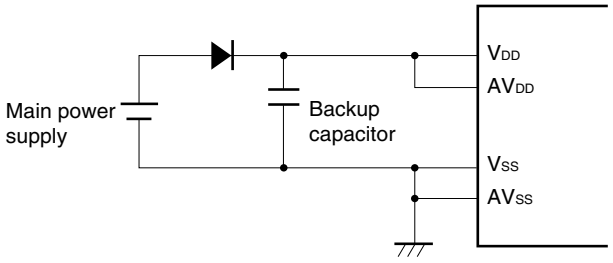
2. m = 0 to 7

(10) AV_{DD} pin

The AV_{DD} pin is used to supply power to the analog circuit. It is also used to supply power to the ANI0 to ANI7 input circuit.

If your application is designed to be changed to backup power, the AV_{DD} pin must be supplied with the same voltage level as the V_{DD} pin, as shown in Figure 10-13.

Figure 10-13. AV_{DD} Pin Handling



(11) AV_{DD} pin input impedance

A series resistor string of several tens of kΩ is connected between the AV_{DD} and AV_{SS} pins. Consequently, if the output impedance of the reference voltage supply is high, the reference voltage supply will form a series connection with the series resistor string, creating a large reference voltage differential.

CHAPTER 11 SERIAL INTERFACE 20

11.1 Serial Interface 20 Functions

Serial interface 20 has the following three modes.

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

(1) Operation stop mode

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

(2) Asynchronous serial interface (UART) mode

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

Serial interface 20 contains a UART-dedicated baud rate generator, enabling communication over a wide range of baud rates. It is also possible to define baud rates by dividing the frequency of the clock input to the ASCK20 pin.

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

This mode is used to transmit 8-bit data, using three lines: a serial clock line ($\overline{\text{SCK20}}$) and two serial data lines (SI20 and SO20).

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

Because, in 3-wire serial I/O mode, it is possible to select whether 8-bit data transmission begins with the MSB or LSB, serial interface 20 can be connected to any device regardless of whether that device is designed for MSB-first or LSB-first transmission.

3-wire serial I/O mode is useful for connecting peripheral I/O circuits and display controllers having conventional synchronous serial interfaces, such as those of the 75XL, 78K, and 17K microcontrollers.

11.2 Serial Interface 20 Configuration

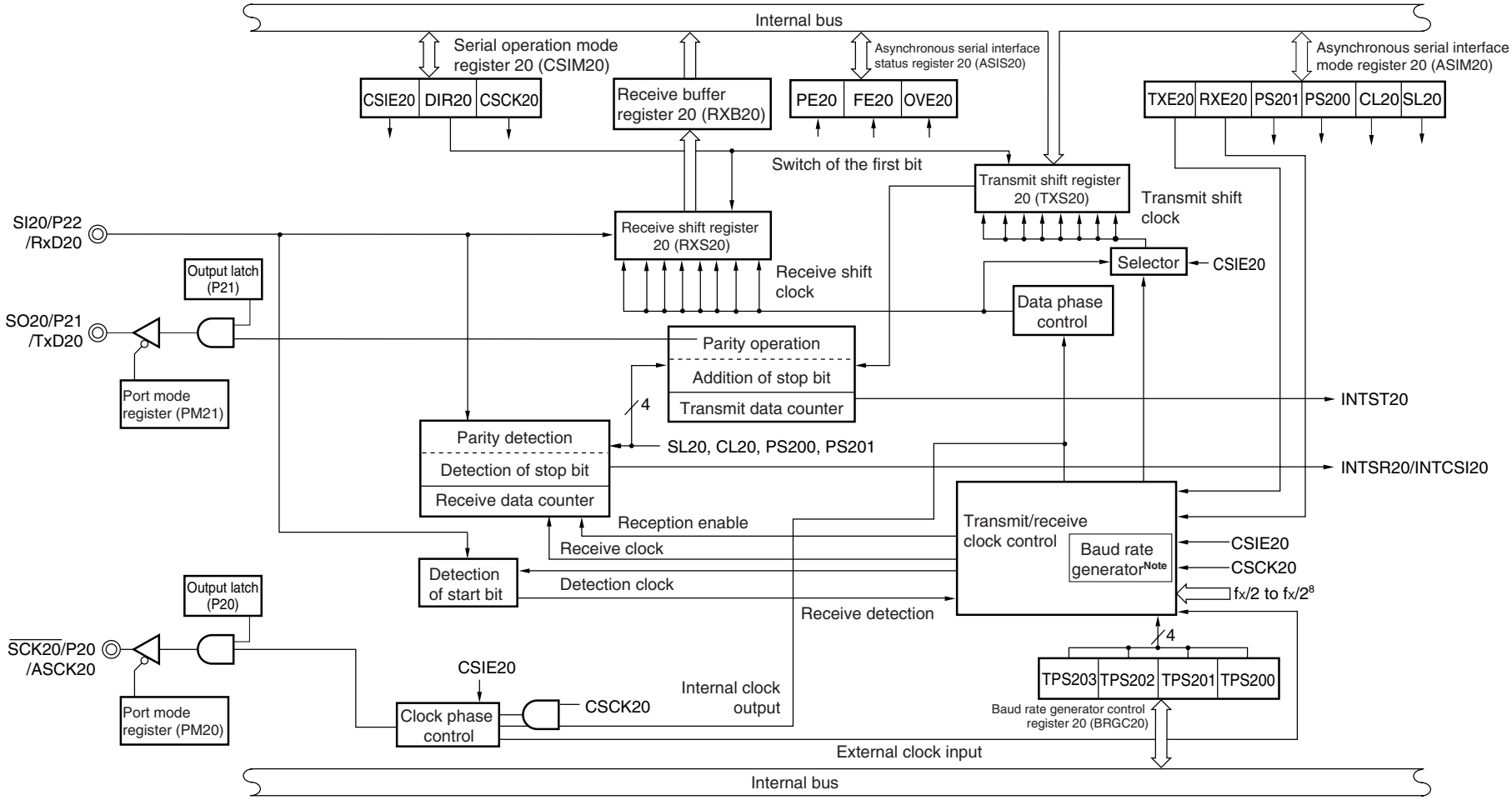
Serial interface 20 includes the following hardware.

Table 11-1. Configuration of Serial Interface 20

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

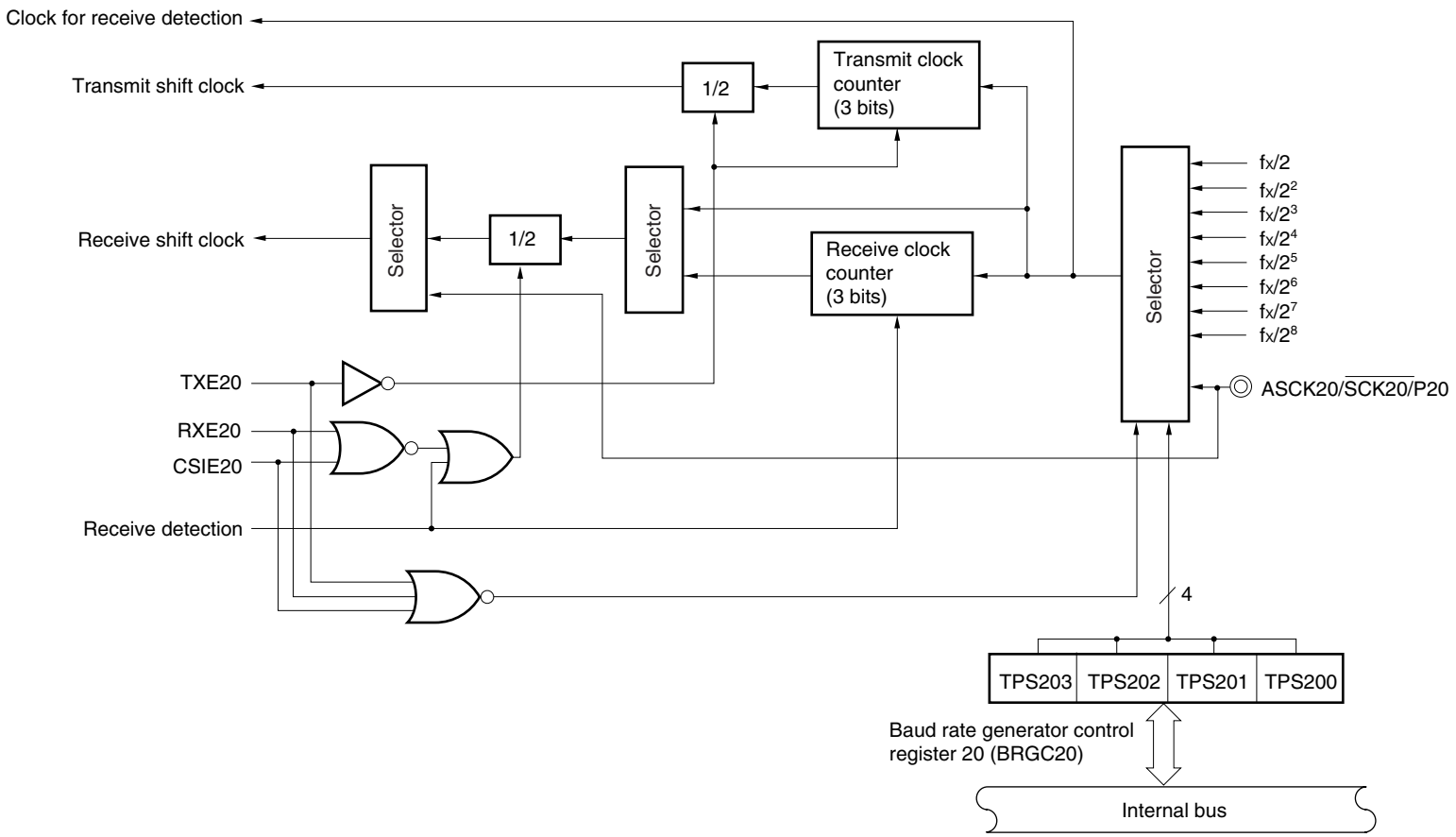
<R>

Figure 11-1. Block Diagram of Serial Interface 20



Note See **Figure11-2** for the configuration of the baud rate generator.

Figure 11-2. Block Diagram of Baud Rate Generator 20



(1) Transmit shift register 20 (TXS20)

TXS20 is a register in which transmit data is prepared. The transmit data is output from TXS20 bit-serially.

When the data length is seven bits, bits 0 to 6 of the data in TXS20 will be transmit data. Writing data to TXS20 triggers transmission.

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

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

Caution Do not write to TXS20 during transmission.

TXS20 and receive buffer register 20 (RXB20) are mapped at the same address, so any attempt to read from TXS20 results in a value being read from RXB20.

(2) Receive shift register 20 (RXS20)

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

RXS20 cannot be manipulated directly by a program.

(3) Receive buffer register 20 (RXB20)

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

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

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

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

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

(4) Transmit controller

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

(5) Receive controller

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

11.3 Serial Interface 20 Control Registers

Serial interface 20 is controlled by the following six registers.

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

(1) Serial operation mode register 20 (CSIM20)

CSIM20 is used to make the settings related to 3-wire serial I/O mode.

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

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

Figure 11-3. Format of Serial Operation Mode Register 20

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

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

DIR20	First-bit specification
0	MSB
1	LSB

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

Cautions 1. Bits 0 and 3 to 6 must be set to 0.

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

<R> **3. When the external input clock is selected in 3-wire serial I/O mode, set input mode by setting bit 0 of port mode register 2 (PM2) to 1.**

<R> **4. Switch operating modes after halting the serial transmit/receive operation.**

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

ASIM20 is used to make the settings related to asynchronous serial interface mode.

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

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

Figure 11-4. Format of Asynchronous Serial Interface Mode Register 20

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

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

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

PS201	PS200	Parity bit specification
0	0	No parity
0	1	Always add 0 parity at transmission. Parity check is not performed at reception (no parity error is generated).
1	0	Odd parity
1	1	Even parity

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

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

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

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

3. Switch operation modes after halting the serial transmission/reception operation.

Table 11-2. Serial Interface 20 Operation Mode Settings

(1) Operation stop mode

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

(2) 3-wire serial I/O mode

ASIM20		CSIM20			PM22	P22	PM21	P21	PM20	P20	First Bit	Shift Clock	P22/SI20/ RxD20 Pin Function	P21/SO20/ TxD20 Pin Function	P20/$\overline{\text{SCK20}}$/ ASCK20 Pin Function
TXE20	RXE20	CSIE20	DIR20	CSCK20											
0	0	1	0	0	1Note 2	×Note 2	0	1	1	×	MSB	External clock	SI20Note 2	SO20 (CMOS output)	$\overline{\text{SCK20}}$ input
				1					0	1		Internal clock			$\overline{\text{SCK20}}$ output
		1	1	0					1	×	LSB	External clock			$\overline{\text{SCK20}}$ input
				1					0	1		Internal clock			$\overline{\text{SCK20}}$ output
Other than above											Setting prohibited				

(3) Asynchronous serial interface mode

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

Notes 1. These pins can be used for port functions.

2. When only transmission is used, this pin can be used as P22 (CMOS I/O).

Remark ×: don't care

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

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

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

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

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

Figure 11-5. Format of Asynchronous Serial Interface Status Register 20

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

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

FE20	Framing error flag
0	No framing error occurred.
1	A framing error occurred (when stop bit was not detected).[Note 1]

OVE20	Overrun error flag
0	No overrun error occurred.
1	An overrun error occurred[Note 2] (when the next receive operation was completed before the data was read from receive buffer register 20).

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

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

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

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

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

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

Figure 11-6. Format of Baud Rate Generator Control Register 20

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

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

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

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

<R> **2. Be sure not to select n = 1 in UART mode when f_X > 2.5 MHz because the baud rate will exceed the rated range.**

<R> **3. When the external input clock is selected, set input mode by setting bit 0 of port mode register 2 (PM2) to 1.**

Remarks 1. f_X: Main system clock oscillation frequency

2. n: Values determined by the settings of TPS200 to TPS203 ($1 \leq n \leq 8$)

3. The parenthesized values apply to operation at f_X = 5.0 MHz.

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

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

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

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

f_X: Main system clock oscillation frequency

n: Values in Figure 11-6, determined by the values of TPS200 to TPS203 ($2 \leq n \leq 8$)

Table 11-3. Example of Relationship Between System Clock and Baud Rate

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

<R> **Caution Do not select n = 1 when f_X > 2.5 MHz because the resulting baud rate exceeds the rated range.**

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

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

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

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

Table 11-4. Relationship Between ASCK20 Pin Input Frequency and Baud Rate (When BRGC20 Is Set to 80H)

Baud Rate (bps)	ASCK20 Pin Input Frequency (kHz)
75	1.2
150	2.4
300	4.8
600	9.6
1,200	19.2
2,400	38.4
4,800	76.8
9,600	153.6
19,200	307.2
31,250	500.0
38,400	614.4

(c) Generation of serial clock from system clock in 3-wire serial I/O mode

The serial clock is generated by dividing the system clock. The frequency of the serial clock can be obtained by the following expression. If the serial clock is externally input to the $\overline{\text{SCK20}}$ pin, setting BRGC20 is not necessary.

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

f_X: Main system clock oscillation frequency

n: Values in Figure 11-6 determined by the settings of TPS200 to TPS203 ($1 \leq n \leq 8$)

11.4 Serial Interface 20 Operation

Serial interface 20 provides the following three modes.

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

11.4.1 Operation stop mode

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

(1) Register setting

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

(a) Serial operation mode register 20 (CSIM20)

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

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

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

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

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

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

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

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

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

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

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

Caution Bits 0 and 1 must be set to 0.

11.4.2 Asynchronous serial interface (UART) mode

In this mode, the one-byte data following the start bit is transmitted/received, enabling full-duplex communication.

This device incorporates a UART-dedicated baud rate generator that enables communications at the desired baud rate. In addition, the baud rate can also be defined by dividing the clock input to the ASCK20 pin.

The UART-dedicated baud rate generator also can output the 31.25 kbps baud rate that complies with the MIDI standard.

(1) Register setting

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

(a) Serial operation mode register 20 (CSIM20)

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

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

Set CSIM20 to 00H when UART mode is selected.

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

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

DIR20	First-bit specification
0	MSB
1	LSB

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

Cautions 1. Bits 0 and 3 to 6 must be set to 0.

<R> **2. Switch operation modes after halting the serial transmission/reception operation.**

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

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

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

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

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

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

PS201	PS200	Parity bit specification
0	0	No parity
0	1	Always add 0 parity at transmission. Parity check is not performed at reception (no parity error is generated).
1	0	Odd parity
1	1	Even parity

CL20	Character length specification
0	7 bits
1	8 bits

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

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

2. Switch operation modes after halting the serial transmission/reception operation.

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

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

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

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

PE20	Parity error flag
0	No parity error occured
1	A parity error occured (when the parity of transmit data did not match)

FE20	Framing error flag
0	No framing error occured
1	A framing error occured (when stop bit was not detected)Note 1

OVE20	Overrun error flag
0	No overrun error occured
1	An overrun error occuredNote 2 (when the next receive operation was completed before data was read from reception buffer register 20)

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

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

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

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

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

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

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

Note Can only be used in the UART mode.

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

<R> **2. Be sure not to select n = 1 in UART mode when f_X > 2.5 MHz because the baud rate will exceed the rated range.**

<R> **3. When the external input clock is selected, set input mode by setting bit 0 of port mode register 2 (PM2) to 1.**

Remarks 1. f_X: Main system clock oscillation frequency

2. n: Values determined by the settings of TPS200 to TPS203 ($1 \leq n \leq 8$)

3. The parenthesized values apply to operation at f_X = 5.0 MHz.

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

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

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

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

f_X: Main system clock oscillation frequency

n: Values in the above table determined by the settings of TPS200 to TPS203 ($2 \leq n \leq 8$)

Table 11-5. Example of Relationship Between System Clock and Baud Rate

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

<R> **Caution Do not select n = 1 when f_X > 2.5 MHz because the resulting baud rate exceeds the rated range.**

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

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

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

f_{ASCK}: Frequency of clock input to ASCK20 pin

Table 11-6. Relationship Between ASCK20 Pin Input Frequency and Baud Rate (When BRGC20 Is Set to 80H)

Baud Rate (bps)	ASCK20 Pin Input Frequency (kHz)
75	1.2
150	2.4
300	4.8
600	9.6
1,200	19.2
2,400	38.4
4,800	76.8
9,600	153.6
19,200	307.2
31,250	500.0
38,400	614.4

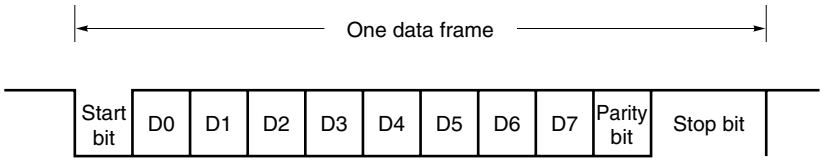
(2) Communication operation

(a) Data format

The transmit/receive data format is as shown in Figure 11-7. One data frame consists of a start bit, character bits, parity bit, and stop bit(s).

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

Figure 11-7. Format of Asynchronous Serial Interface Transmit/Receive Data



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

When 7 bits are selected as the number of character bits, only the lower 7 bits (bits 0 to 6) are valid; in transmission the most significant bit (bit 7) is ignored, and in reception the most significant bit (bit 7) is always “0”.

The serial transfer rate is selected by baud rate generator control register 20 (BRGC20).

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

(b) Parity types and operation

The parity bit is used to detect a bit error in the communication data. Normally, the same kind of parity bit is used on the transmitting side and the receiving side. With even parity and odd parity, a one-bit (odd number) error can be detected. With 0 parity and no parity, an error cannot be detected.

(i) Even parity

- **At transmission**

  The parity bit is determined so that the number of bits with a value of “1” in the transmit data including the parity bit is even. The parity bit value should be as follows.

  The number of bits with a value of “1” is an odd number in transmit data: 1
  The number of bits with a value of “1” is an even number in transmit data: 0

- **At reception**

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

(ii) Odd parity

- **At transmission**

  Opposite to even parity, the parity bit is determined so that the number of bits with a value of “1” in the transmit data including parity bit is odd. The parity bit value should be as follows.

  The number of bits with a value of “1” is an odd number in transmit data: 0
  The number of bits with a value of “1” is an even number in transmit data: 1

- **At reception**

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

(iii) 0 parity

When transmitting, the parity bit is set to “0” irrespective of the transmit data.

At reception, a parity bit check is not performed. Therefore, a parity error does not occur, irrespective of whether the parity bit is set to “0” or “1”.

(iv) No parity

A parity bit is not added to the transmit data. At reception, data is received assuming that there is no parity bit. Since there is no parity bit, a parity error does not occur.

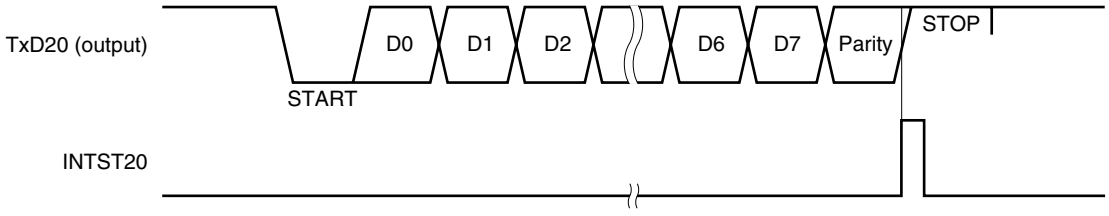
(c) Transmission

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

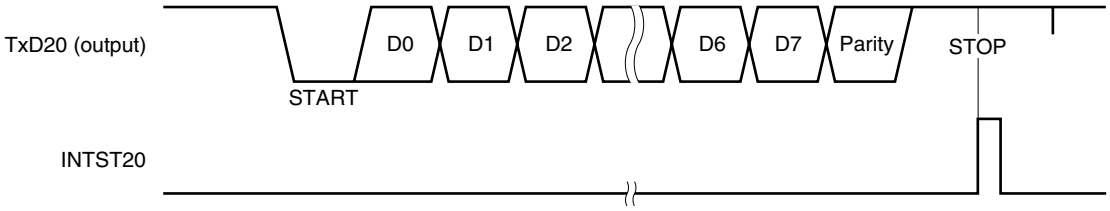
When the transmit operation starts, the data in TXS20 is shifted out, and when TXS20 is empty, a transmission completion interrupt (INTST20) is generated.

Figure 11-8. Asynchronous Serial Interface Transmission Completion Interrupt Timing

(a) Stop bit length: 1



(b) Stop bit length: 2



Caution Do not rewrite asynchronous serial interface mode register 20 (ASIM20) during a transmit operation. If the ASIM20 register is rewritten during transmission, subsequent transmission may not be able to be performed (the normal state is restored by $\overline{\text{RESET}}$ input).

It is possible to determine whether transmission is in progress by software by using a transmission completion interrupt (INTST20) or the interrupt request flag (STIF20) set by INTST20.

(d) Reception

When bit 6 (RXE20) of asynchronous serial interface mode register 20 (ASIM20) is set (1), a receive operation is enabled and sampling of the RxD20 pin input is performed.

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

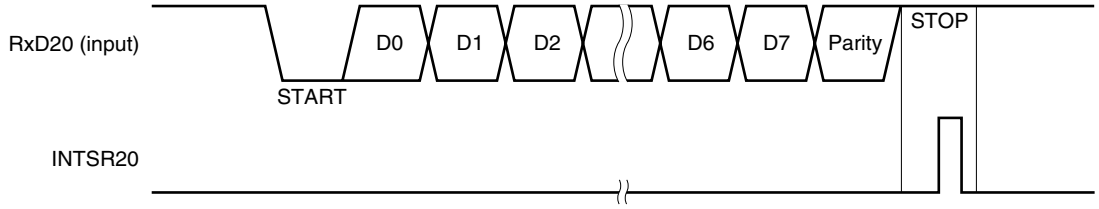
When the RxD20 pin input becomes low, the 3-bit counter starts counting, and when half the time determined by the specified baud rate has passed, the data sampling start timing signal is output. If the RxD20 pin input sampled again as a result of this start timing signal is low, it is identified as a start bit, the 3-bit counter is initialized and starts counting, and data sampling is performed. When character data, a parity bit, and one stop bit are detected after the start bit, reception of one frame of data ends.

When one frame of data has been received, the receive data in the shift register is transferred to receive buffer register 20 (RXB20), and a reception completion interrupt (INTSR20) is generated.

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

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

Figure 11-9. Asynchronous Serial Interface Reception Completion Interrupt Timing



Caution Be sure to read receive buffer register 20 (RXB20) even if a receive error occurs. If RXB20 is not read, an overrun error will occur when the next data is received, and the receive error state will continue indefinitely.

(e) Receive errors

The following three errors may occur during a receive operation: a parity error, framing error, and overrun error. After data reception, an error flag is set in asynchronous serial interface status register 20 (ASIS20). Receive error causes are shown in Table 11-7.

It is possible to determine what kind of error occurred during reception by reading the contents of ASIS20 in the reception error interrupt servicing (see **Table 11-7** and **Figure 11-10**).

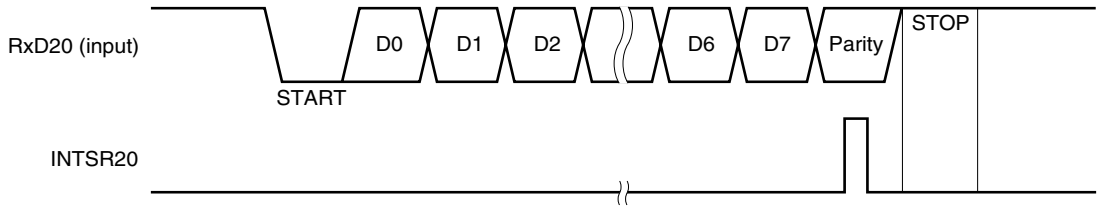
The contents of ASIS20 are reset (0) by reading receive buffer register 20 (RXB20) or receiving the next data (if there is an error in the next data, the corresponding error flag is set).

Table 11-7. Receive Error Causes

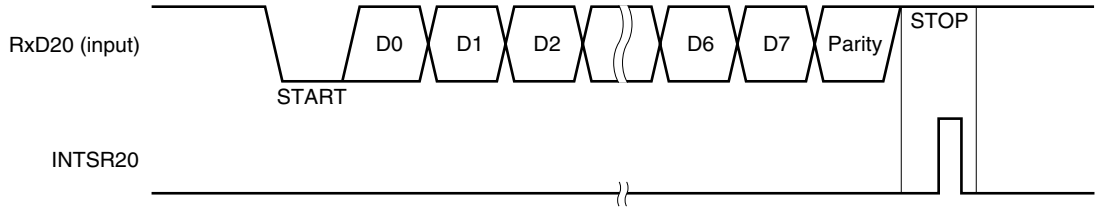
Receive Errors	Cause	ASIS20 Value
Parity error	Parity at transmission and reception do not match	04H
Framing error	Stop bit not detected	02H
Overrun error	Reception of next data is completed before data is read from receive buffer register	01H

Figure 11-10. Receive Error Timing

(a) Parity error occurrence



(b) Framing error or overrun error occurrence



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

2. Be sure to read receive buffer register 20 (RXB20) also when a receive error occurs. If RXB20 is not read, an overrun error will occur when the next data is received, and the receive error state will continue indefinitely.

(f) Reading receive data

When the reception completion interrupt (INTSR20) occurs, receive data can be read by reading the value of receive buffer register 20 (RXB20).

To read the receive data stored in receive buffer register 20 (RXB20), read while reception is enabled (RXE20 = 1).

Remark However, if it is necessary to read receive data after reception has stopped (RXE20 = 0), read using either of the following methods.

(a) Read after setting RXE20 = 0 after waiting for one cycle or more of the source clock selected by BRGC20.
(b) Read after bit 2 (DIR20) of serial operation mode register 20 (CSIM20) is set (1).

Program example of (a) (BRGC20 = 00H (source clock = fx/2))

```
INTREX:                 ;<Reception completion interrupt routine>
       NOP              ;2 clocks
       CLR1 RXE20       ;Reception stopped
       MOV  A, RXB20    ;Read receive data
```

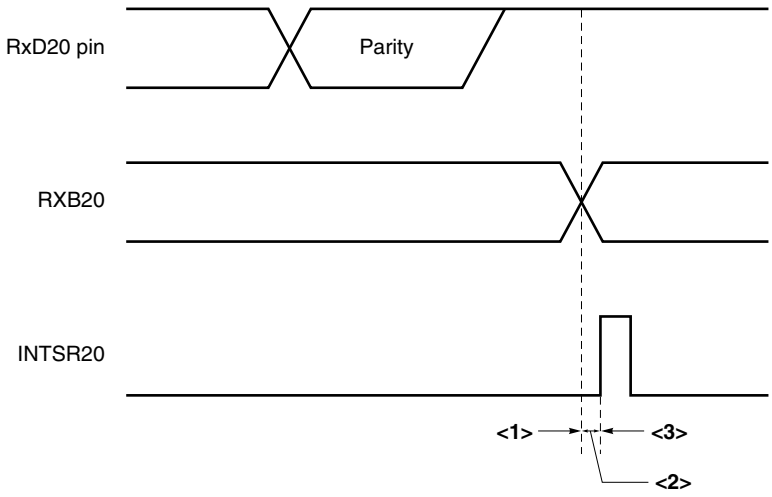
Program example of (b)

```
INTRXE:                 ;<Reception completion interrupt routine>
       SET1 CSIM20.2    ;DIR20 flag is set to LSB first
       CLR1 RXE20       ;Reception stopped
       MOV  A, RXB20    ;Read receive data
```

(3) Cautions related to UART mode

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

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



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

11.4.3 3-wire serial I/O mode

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

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

(1) Register setting

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

(a) Serial operation mode register 20 (CSIM20)

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

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

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

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

DIR20	First-bit specification
0	MSB
1	LSB

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

Cautions 1. Bits 0 and 3 to 6 must be set to 0.

<R> **2. When the external input clock is selected, set input mode by setting bit 0 of port mode register 2 (PM2) to 1.**

<R> **3. Switch operation modes after halting the serial transmission/reception operation.**

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

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

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

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

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

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

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

PS201	PS200	Parity bit specification
0	0	No parity
0	1	Always add 0 parity at transmission. Parity check is not performed at reception (no parity error occurs).
1	0	Odd parity
1	1	Even parity

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

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

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

2. Switch operation modes after halting the serial transmission/reception operation.

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

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

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

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

TPS203	TPS202	TPS201	TPS200	Selection of baud rate generator source clock	n
0	0	0	0	$f_X/2$ (2.5 MHz)	1
0	0	0	1	$f_X/2^2$ (1.25 MHz)	2
0	0	1	0	$f_X/2^3$ (625 kHz)	3
0	0	1	1	$f_X/2^4$ (313 kHz)	4
0	1	0	0	$f_X/2^5$ (156 kHz)	5
0	1	0	1	$f_X/2^6$ (78.1 kHz)	6
0	1	1	0	$f_X/2^7$ (39.1 kHz)	7
0	1	1	1	$f_X/2^8$ (19.5 kHz)	8
Other than above				Setting prohibited	

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

Remarks 1. f_X: Main system clock oscillation frequency

2. n: Values determined by the settings of TPS200 to TPS203 ($1 \leq n \leq 8$)

3. The parenthesized values apply to operation at f_X = 5.0 MHz.

If the internal clock is used as the serial clock for 3-wire serial I/O mode, set bits TPS200 to TPS203 to set the frequency of the serial clock. To obtain the frequency to be set, use the following expression. When an external clock is input to the $\overline{\text{SCK20}}$ pin, setting BRGC20 is not necessary.

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

f_X: Main system clock oscillation frequency

n: Values in the above table determined by the settings of TPS200 to TPS203 ($1 \leq n \leq 8$)

(2) Communication operation

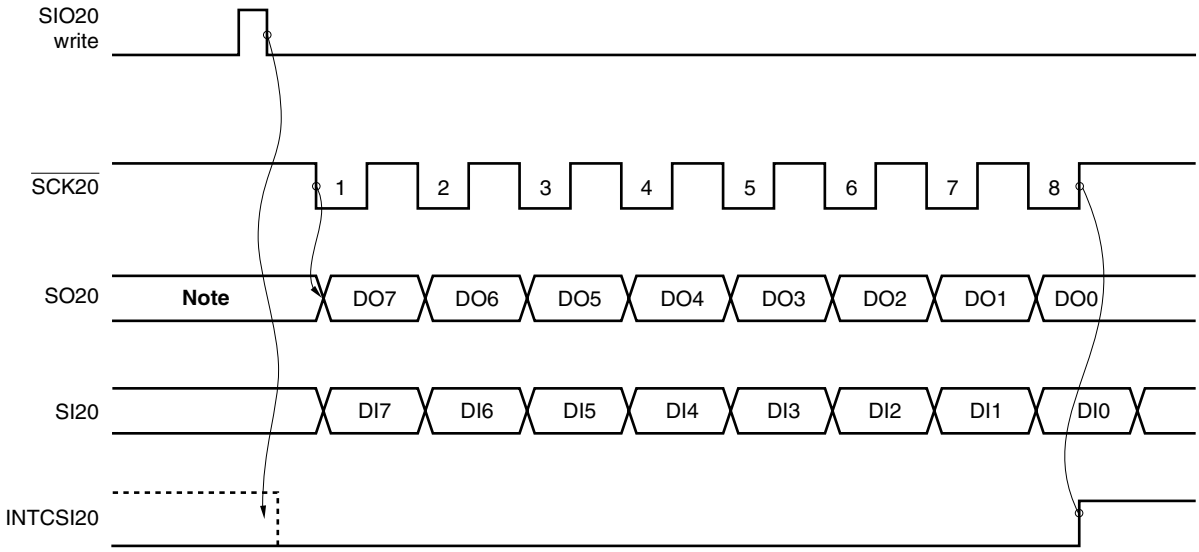
In 3-wire serial I/O mode, data transmission/reception is performed in 8-bit units. Data is transmitted/received bit by bit in synchronization with the serial clock.

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

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

Figure 11-11. 3-Wire Serial I/O Mode Timing (1/2)

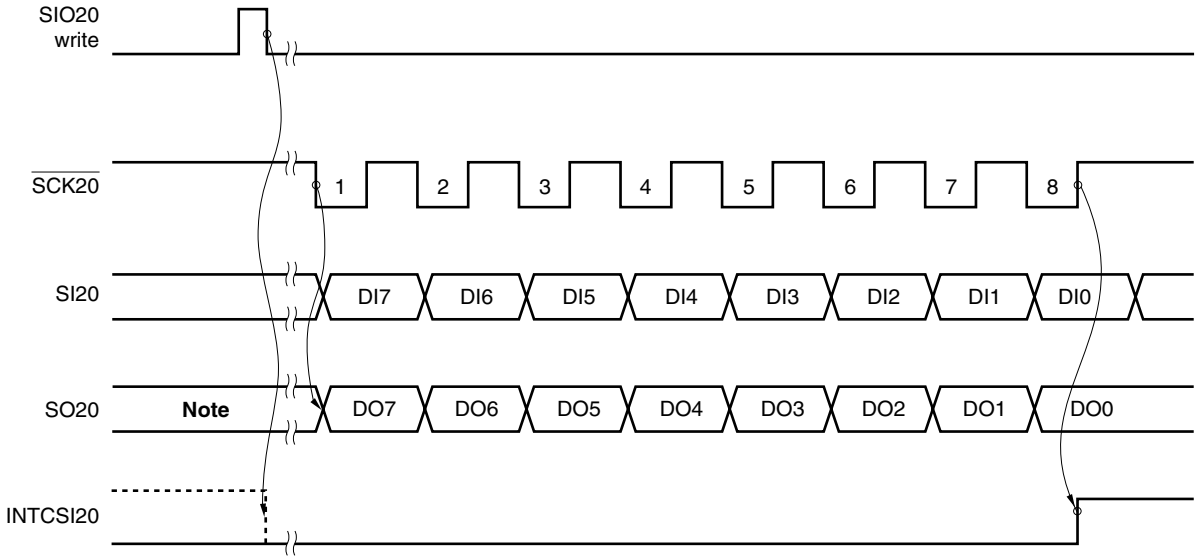
(i) Master operation timing (CSCK20=0)



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

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

(ii) Slave operation timing (CSCK20=1)



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

(3) Transfer start

Serial transfer is started by setting transfer data to transmit shift register 20 (TXS20/SIO20) when the following two conditions are satisfied.

- Bit 7 (CSIE20) of serial operation mode register 20 (CSIM20) = 1
- Internal serial clock is stopped or $\overline{\text{SCK20}}$ is high after 8-bit serial transfer.

Caution If CSIE20 is set to "1" after data is written to TXS20/SIO20, transfer does not start.

Termination of 8-bit transfer stops the serial transfer automatically and generates the interrupt request signal (INTCSI20).

CHAPTER 12 SERIAL INTERFACE 1A0

12.1 Function of Serial Interface 1A0

Serial interface 1A0 has the following three modes.

- Operation stop mode
- 3-wire serial I/O mode
- 3-wire serial I/O mode with automatic transmit/receive function

(1) Operation stop mode

This mode is used when serial transfer will not be performed. It enables a reduction in power consumption.

(2) 3-wire serial I/O mode (MSB/LSB-first switchable)

This mode is used to transfer 8-bit data using three lines: a serial clock line ($\overline{\text{SCK10}}$) and two serial data lines (SI10 and SO10).

Because this mode supports simultaneous transmission and reception, 3-wire serial I/O mode requires less processing time for data transfer.

Also, when using 3-wire serial I/O mode, it is possible to select whether 8-bit data transfer will start with the MSB or LSB, so any device can be connected regardless of whether that device is designed for MSB-first or LSB-first transfers.

3-wire serial I/O mode is useful for connecting peripheral I/O circuits and display controllers with conventional clocked serial interfaces, such as those found in the 75XL, 78K, and 17K microcontrollers.

(3) 3-wire serial mode with automatic transmit/receive function (MSB/LSB-first switchable)

This mode has an automatic transmit/receive function in addition to the functions in (2) above.

The automatic transmit/receive function is used to transmit/receive data with a maximum of 16 bytes. This function enables the hardware to transmit/receive data to/from the OSD (On Screen Display) device and a device with an on-chip display controller/driver independently of the CPU, thus alleviating the software load.

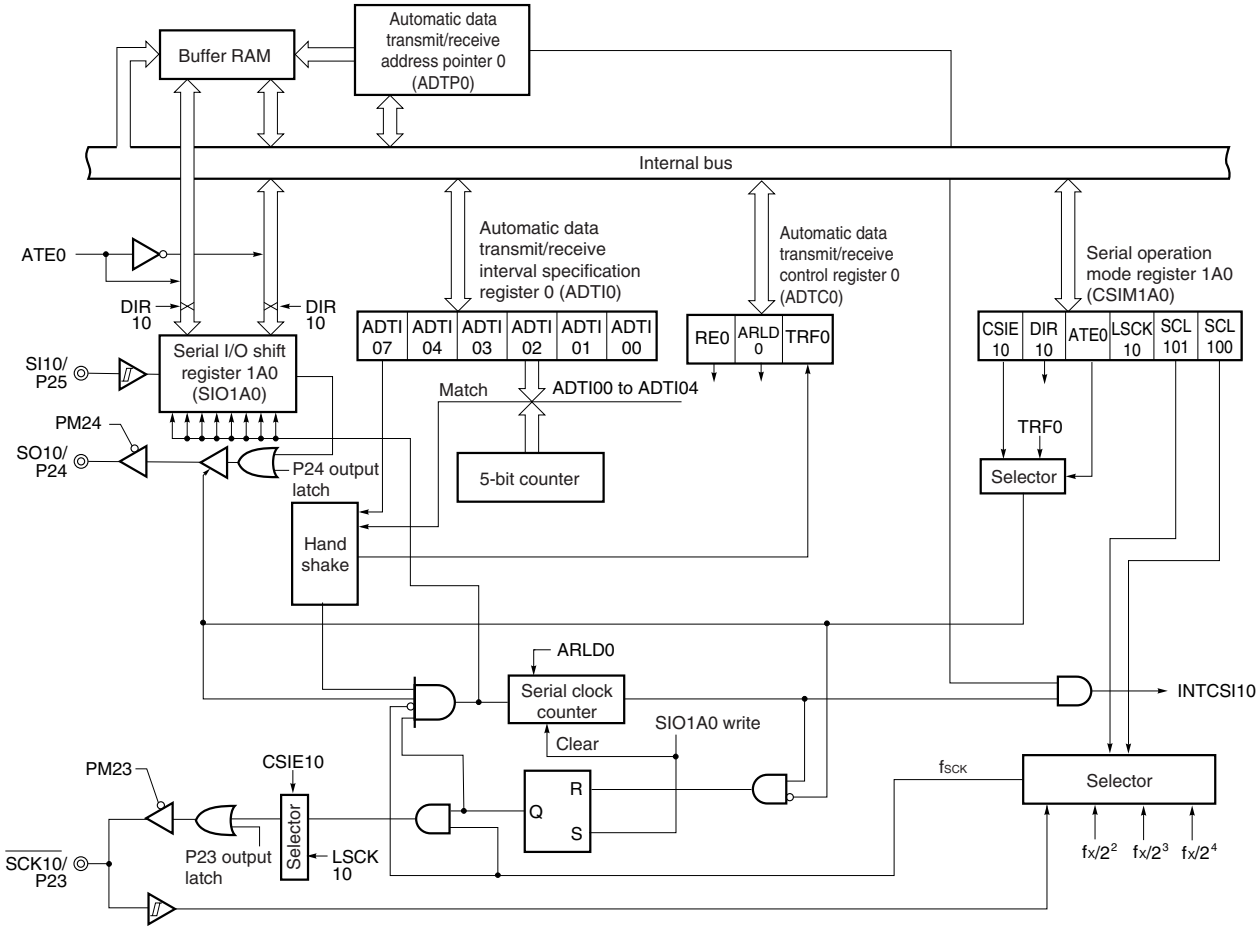
12.2 Configuration of Serial Interface 1A0

Serial interface 1A0 includes the following hardware.

Table 12-1. Configuration of Serial Interface 1A0

Item	Configuration
Registers	Serial I/O shift register 1A0 (SIO1A0) Automatic data transmit/receive address pointer 0 (ADTP0)
Control registers	Serial operation mode register 1A0 (CSIM1A0) Automatic data transmit/receive control register 0 (ADTC0) Automatic data transmit/receive interval specification register 0 (ADTI0) Port mode register 2 (PM2) Port 2 (P2)

Figure 12-1. Block Diagram of Serial Interface 1A0



(1) Serial I/O shift register 1A0 (SIO1A0)

This is an 8-bit register used to carry out parallel/serial conversion and to carry out serial transmission/reception in synchronization with the serial clock.

SIO1A0 is set with an 8-bit memory manipulation instruction.

When the value in bit 7 (CSIE10) of serial operation mode register 1A0 (CSIM1A0) is 1, writing data to SIO1A0 starts a serial operation.

During transmission, data written to SIO1A0 is output to the serial output (SO10). During reception, data is read from the serial input (SI10) to SIO1A0.

$\overline{\text{RESET}}$ input sets SIO1A0 to 00H.

Caution Do not write data to SIO1A0 while the automatic transmit/receive function is activated.

(2) Automatic data transmit/receive address pointer 0 (ADTP0)

This register stores value of (transmit data byte – 1) while the automatic transmit/receive function is activated. As data is transferred/received, it is automatically decremented.

ADTP0 is set with an 8-bit memory manipulation instruction. The higher 4 bits must be set to 0.

$\overline{\text{RESET}}$ input makes ADTP0 undefined.

Caution Do not write data to ADTP0 while the automatic transmit/receive function is activated.

(3) Serial clock counter

This counter counts the serial clocks to be output and input during transmission/reception to check whether 8-bit data has been transmitted/received.

12.3 Control Registers for Serial Interface 1A0

Serial interface 1A0 is controlled by the following five registers.

- Serial operation mode register 1A0 (CSIM1A0)
- Automatic data transmit/receive control register 0 (ADTC0)
- Automatic data transmit/receive interval specification register 0 (ADTI0)
- Port mode register 2 (PM2)
- Port 2 (P2)

(1) Serial operation mode register 1A0 (CSIM1A0)

This register sets serial interface 1A0 serial clock, operation mode, operation enable/disable, and automatic transmission/reception operation enable/disable.

CSIM1A0 is set with a 1-bit or 8-bit memory manipulation instruction.

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

Caution Set the port mode register 2 (PM2) in the 3-wire serial I/O mode or 3-wire serial mode with automatic transmit/receive function as follows. Set the output latch to 0.

- **In the case of serial clock output (master transmission or master reception)**
  Set the $\overline{\text{SCK10}}$/P23 pin to output mode (PM23 = 0) and the P23 output latch to 0.
- **In the case of serial clock input (slave transmission or slave reception)**
  Set the $\overline{\text{SCK10}}$/P23 pin to input mode (PM23 = 1).
- **In transmission or transmission/reception mode**
  Set the SO10/P24 pin to output mode (PM24 = 0) and the P24 output latch to 0.
  Set the SI10/P25 pin to input mode (PM25 = 1).
- **In reception mode**
  Set the SI10/P25 pin to input mode (PM25 = 1).

Figure 12-2. Format of Serial Operation Mode Register 1A0

Symbol	<7>	6	<5>	<4>	3	2	1	0	Address	After reset	R/W
CSIM1A0	CSIE10	DIR10	ATE0	LSCK10	0	0	SCL101	SCL100	FF78H	00H	R/W

CSIE10	Specification of operation enable/disable		
	Shift register operation	Serial counter	Port[Note]
0	Operation stopped	Cleared	Port function
1	Operation enabled	Count operation enabled	Serial function + port function

DIR10	Specification of first bit of serial transfer data
0	MSB
1	LSB

ATE0	Selection of operation mode
0	3-wire serial I/O mode
1	3-wire serial I/O mode with automatic transmit/receive function

LSCK10	Chip enable control of $\overline{\text{SCK10}}$ pin
0	$\overline{\text{SCK10}}$ is used as port (P23) when CSIE10 = 0. $\overline{\text{SCK10}}$ is used for clock output when CSIE10 = 1.
1	$\overline{\text{SCK10}}$ is fixed to high-level output when CSIE10 = 0. $\overline{\text{SCK10}}$ is used for clock output when CSIE10 = 1.

SCL101	SCL100	Selection of serial clock (f_{SCK})
0	0	External clock input to $\overline{\text{SCK10}}$ pin
0	1	$f_X/2^2$ (1.25 MHz)
1	0	$f_X/2^3$ (625 kHz)
1	1	$f_X/2^4$ (313 kHz)

Note When CSIE10 = 0 (SIO1A0 operation stop status), the $\overline{\text{SCK10}}$/P23, SO10/P24, and SI10/P25 pins can freely be used as port pins. Also, when CSIE10 is used for transmission only, the SI10/P25 pin can be used as P25 (CMOS I/O) (set bit 7 (RE0) of ADTC0 to 0).

Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 5.0 MHz.

(2) Automatic data transmit/receive control register 0 (ADTC0)

This register sets automatic reception enable/disable, the operation mode, and displays the state of automatic transmit/receive control.

ADTC0 is set with a 1-bit or 8-bit memory manipulation instruction.

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

Figure 12-3. Format of Automatic Data Transmit/Receive Control Register 0

Symbol	<7>	<6>	5	4	<3>	2	1	0	Address	After reset	R/W
ADTC0	RE0	ARLD0	0	0	TRF0	0	0	0	FF79H	00H	R/W[Note 1]

RE0	Control of reception of automatic transmit/receive function
0	Reception disabled[Note 2]
1	Reception enabled

ARLD0	Selection of operation mode for automatic transmit/receive function
0	One-shot mode
1	Repeat mode

TRF0	Status of automatic transmission/reception function[Note 3]
0	Detection of termination of automatic transmission/reception (this bit is set to 0 upon suspension of automatic transmission/reception or when ARLD0 = 0)
1	Automatic transmission/reception in progress (this bit is set to 1 when data is written to SIO1A0)

Notes 1. Bit 3 (TRF0) is read-only.

2. When RE0 is reset to 0, P25 (CMOS I/O) is used even when bit 7 (CSIE10) of serial operation mode register 1A0 (CSIM1A0) is set to 1.

3. Use TRF0, instead of CSIIF10 (interrupt request flag), to identify the completion of automatic transmission/reception.

(3) Automatic data transmit/receive interval specification register 0 (ADTI0)

This register sets the automatic data transmit/receive function data transfer interval.

ADTI0 is set with a 1-bit or 8-bit memory manipulation instruction.

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

Figure 12-4. Format of Automatic Data Transmit/Receive Interval Specification Register 0 (1/2)

Symbol	<7>	6	5	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
ADTI0	ADTI07	0	0	ADTI04	ADTI03	ADTI02	ADTI01	ADTI00	FF7BH	00H	R/W

ADTI07	Data transfer interval control
0	No control of interval by ADTI00 to ADTI04[Note 1]
1	Control of interval by ADTI00 to ADTI04

ADTI04	ADTI03	ADTI02	ADTI01	ADTI00	Data transfer interval specification (f_X = 5.0 MHz, f_{SCK} = 1.25 MHz)[Note 2]	n
0	0	0	0	0	1.60 μs + 0.5/f_{SCK}	0
0	0	0	0	1		1
0	0	0	1	0	2.40 μs + 0.5/f_{SCK}	2
0	0	0	1	1	3.20 μs + 0.5/f_{SCK}	3
0	0	1	0	0	4.00 μs + 0.5/f_{SCK}	4
0	0	1	0	1	4.80 μs + 0.5/f_{SCK}	5
0	0	1	1	0	5.60 μs + 0.5/f_{SCK}	6
0	0	1	1	1	6.40 μs + 0.5/f_{SCK}	7
0	1	0	0	0	7.20 μs + 0.5/f_{SCK}	8
0	1	0	0	1	8.00 μs + 0.5/f_{SCK}	9
0	1	0	1	0	8.80 μs + 0.5/f_{SCK}	10
0	1	0	1	1	9.60 μs + 0.5/f_{SCK}	11
0	1	1	0	0	10.4 μs + 0.5/f_{SCK}	12
0	1	1	0	1	11.2 μs + 0.5/f_{SCK}	13
0	1	1	1	0	12.0 μs + 0.5/f_{SCK}	14
0	1	1	1	1	12.8 μs + 0.5/f_{SCK}	15

Figure 12-4. Format of Automatic Data Transmit/Receive Interval Specification Register 0 (2/2)

Symbol	<7>	6	5	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
ADTI0	ADTI07	0	0	ADTI04	ADTI03	ADTI02	ADTI01	ADTI00	FF7BH	00H	R/W

ADTI04	ADTI03	ADTI02	ADTI01	ADTI00	Data transfer interval specification (f_X = 5.0 MHz, f_{SCK} = 1.25 MHz)[Note 2]	n
1	0	0	0	0	13.6 μs + 0.5/f_{SCK}	16
1	0	0	0	1	14.4 μs + 0.5/f_{SCK}	17
1	0	0	1	0	15.2 μs + 0.5/f_{SCK}	18
1	0	0	1	1	16.0 μs + 0.5/f_{SCK}	19
1	0	1	0	0	16.8 μs + 0.5/f_{SCK}	20
1	0	1	0	1	17.6 μs + 0.5/f_{SCK}	21
1	0	1	1	0	18.4 μs + 0.5/f_{SCK}	22
1	0	1	1	1	19.2 μs + 0.5/f_{SCK}	23
1	1	0	0	0	20.0 μs + 0.5/f_{SCK}	24
1	1	0	0	1	20.8 μs + 0.5/f_{SCK}	25
1	1	0	1	0	21.6 μs + 0.5/f_{SCK}	26
1	1	0	1	1	22.4 μs + 0.5/f_{SCK}	27
1	1	1	0	0	23.2 μs + 0.5/f_{SCK}	28
1	1	1	0	1	24.0 μs + 0.5/f_{SCK}	29
1	1	1	1	0	24.8 μs + 0.5/f_{SCK}	30
1	1	1	1	1	25.6 μs + 0.5/f_{SCK}	31

Notes 1. The interval time depends only on the CPU processing.

2. The data transfer interval time is found from the following expressions (n: Value set to ADTI00 to ADTI04).

<1> n = 0

$$\text{Interval time} = \frac{2}{f_{SCK}} + \frac{0.5}{f_{SCK}}$$

<2> n = 1 to 31

$$\text{Interval time} = \frac{n+1}{f_{SCK}} + \frac{0.5}{f_{SCK}}$$

Cautions 1. Do not write to ADTI0 during operation of the automatic transmit/receive function.

2. Be sure to set bits 5 and 6 to 0.

Remark f_X: Main system clock oscillation frequency

f_{SCK}: Serial clock frequency

12.4 Serial Interface 1A0 Operation

Serial interface 1A0 provides the following three modes.

- Operation stop mode
- 3-wire serial I/O mode
- 3-wire serial I/O mode with automatic transmit/receive function

12.4.1 Operation stop mode

In operation stop mode, serial transfer is not executed, thereby reducing the power consumption. The P23/$\overline{\text{SCK10}}$, P24/SO10, and P25/SI10 pins can be used as normal I/O ports.

(1) Register setting

Operation stop mode is set by serial operation mode register 1A0 (CSIM1A0).

(a) Serial operation mode register 1A0 (CSIM1A0)

CSIM1A0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM1A0 to 00H.

Symbol	<7>	6	<5>	<4>	3	2	1	0	Address	After reset	R/W
CSIM1A0	CSIE10	DIR10	ATE0	LSCK10	0	0	SCL101	SCL100	FF78H	00H	R/W

CSIE10	Specification of operation enable/disable		
	Shift register operation	Serial counter	Port[Note]
0	Operation stopped	Cleared	Port function
1	Operation enabled	Count operation enabled	Serial function + port function

Note When CSIE10 = 0 (SIO1A0 operation stop status), the $\overline{\text{SCK10}}$/P23, SO10/P24, and SI10/P25 pins can freely be used as port pins.

12.4.2 3-wire serial I/O mode

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

Communication is performed using three lines: a serial clock ($\overline{SCK10}$), serial output (SO10), and serial input (SI10).

(1) Register setting

3-wire serial I/O mode settings are performed using serial operation mode register 1A0 (CSIM1A0).

(a) Serial operation mode register 1A0 (CSIM1A0)

CSIM1A0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{RESET}$ input sets CSIM1A0 to 00H.

Caution Set the port mode register 2 (PM2) in the 3-wire serial I/O mode as follows.

- **In the case of serial clock output (master transmission or master reception)**
  Set the $\overline{SCK10}$/P23 pin to output mode (PM23 = 0) and the P23 output latch to 0.
- **In the case of serial clock input (slave transmission or slave reception)**
  Set the $\overline{SCK10}$/P23 pin to input mode (PM23 = 1).
- **In transmission or transmission/reception mode**
  Set the SO10/P24 pin to output mode (PM24 = 0) and the P24 output latch to 0.
  Set the SI10/P25 pin to input mode (PM25 = 1).
- **In reception mode**
  Set the SI10/P25 pin to input mode (PM25 = 1).

Symbol	<7>	6	<5>	<4>	3	2	1	0	Address	After reset	R/W
CSIM1A0	CSIE10	DIR10	ATE0	LSCK10	0	0	SCL101	SCL100	FF78H	00H	R/W

CSIE10	Specification of operation enable/disable		
	Shift register operation	Serial counter	Port[Note]
0	Operation stopped	Cleared	Port function
1	Operation enabled	Count operation enabled	Serial function + port function

DIR10	Specification of first bit of serial transfer data
0	MSB
1	LSB

ATE0	Selection of operation mode
0	3-wire serial I/O mode
1	3-wire serial I/O mode with automatic transmit/receive function

LSCK10	Chip enable control of $\overline{\text{SCK10}}$ pin
0	$\overline{\text{SCK10}}$ is used as port (P23) when CSIE10 = 0. $\overline{\text{SCK10}}$ is used for clock output when CSIE10 = 1.
1	$\overline{\text{SCK10}}$ is fixed to high-level output when CSIE10 = 0. $\overline{\text{SCK10}}$ is used for clock output when CSIE10 = 1.

SCL101	SCL100	Selection of serial clock
0	0	External clock input to $\overline{\text{SCK10}}$ pin
0	1	$f_X/2^2$ (1.25 MHz)
1	0	$f_X/2^3$ (625 kHz)
1	1	$f_X/2^4$ (313 kHz)

Note When CSIE10 = 0 (SIO1A0 operation stop status), the $\overline{\text{SCK10}}$/P23, SO10/P24, and SI10/P25 pins can freely be used as port pins. Also, when CSIE10 is used for transmission only, the SI10/P25 pin can be used as P25 (CMOS I/O) (set bit 7 (RE0) of ADTC0 to 0).

Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 5.0 MHz.

(2) Communication operation

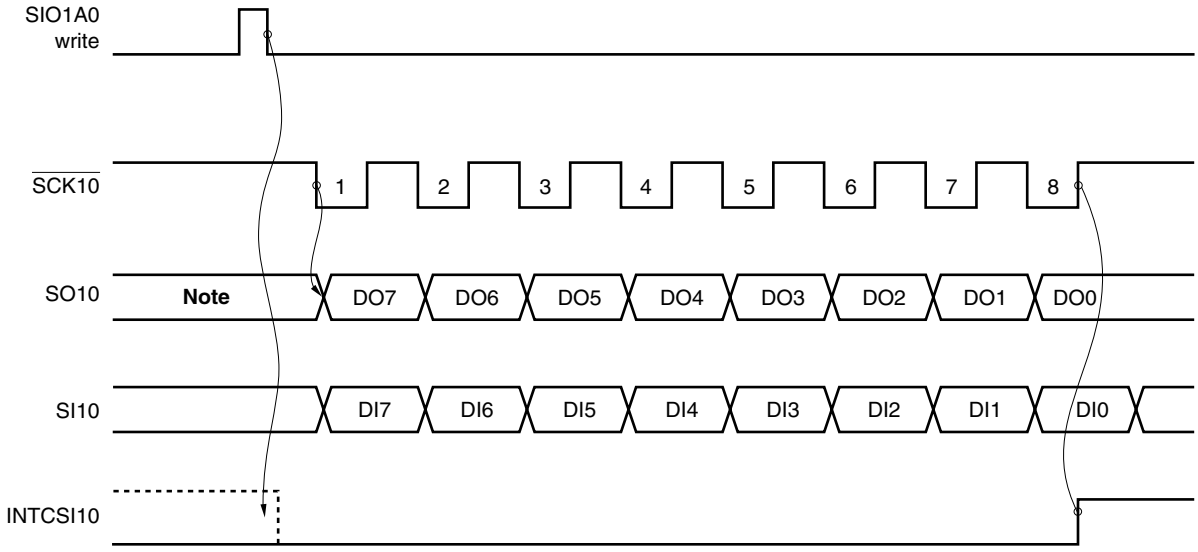
In 3-wire serial I/O mode, data transmission/reception is performed in 8-bit units. Data is transmitted/received bit by bit in synchronization with the serial clock.

Serial I/O shift register 1A0 (SIO1A0) shift operations are performed in synchronization with the fall of the serial clock ($\overline{SCK10}$). Then transmit data is held in the SO10 latch and output from the SO10 pin. Also, receive data input to the SI10 pin is latched in the SIO1A0 on the rise of $\overline{SCK10}$.

At the end of an 8-bit transfer, the operation of SIO1A0 stops automatically, and the interrupt request signal (INTCSI10) is generated.

Figure 12-5. 3-Wire Serial I/O Mode Timing (1/2)

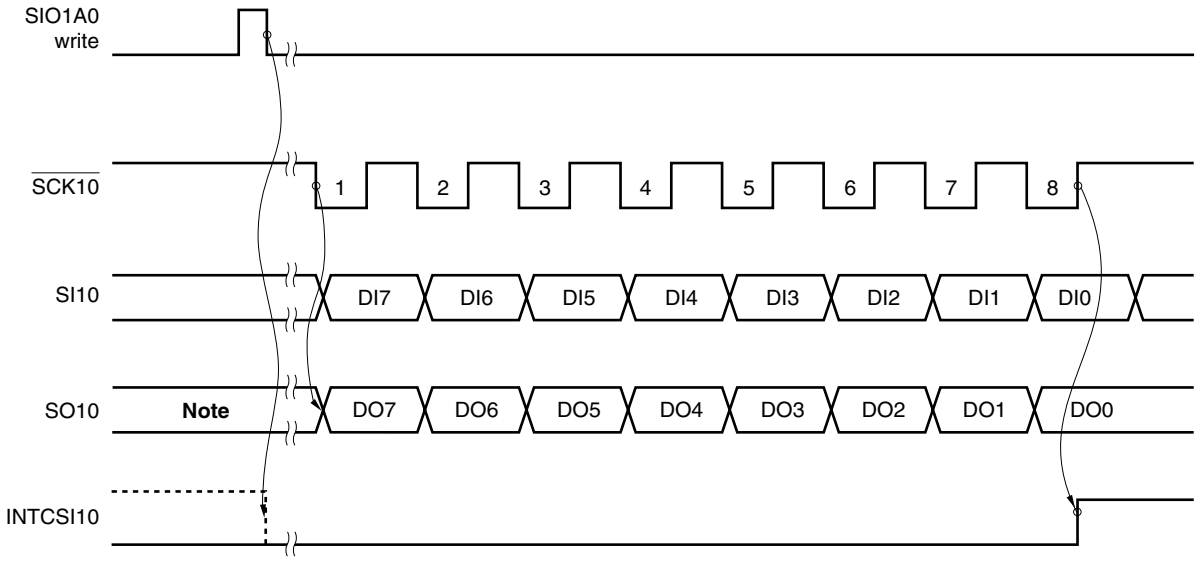
(i) Master operation timing



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

Figure 12-5. 3-Wire Serial I/O Mode Timing (2/2)

(ii) Slave operation timing



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

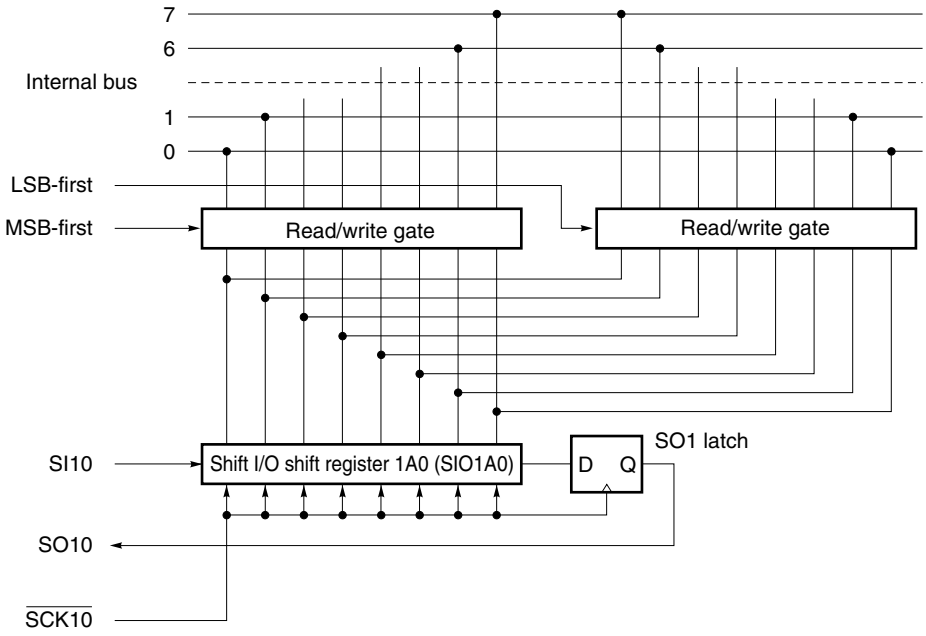
(3) MSB/LSB switching as the start bit

In the 3-wire serial I/O mode, transfer can be selected to start from the MSB or LSB.

Figure 12-6 shows the configuration of serial I/O shift register 1A0 (SIO1A0) and the internal bus. As shown in the figure, MSB/LSB can be read/written in reverse form.

MSB/LSB switching as the start bit can be specified with bit 6 (DIR10) of serial operation mode register 1A0 (CSIM1A0).

Figure 12-6. Circuit of Switching in Transfer Bit Order



Start bit switching is realized by switching the bit order for data write to SIO1A0. The SIO1A0 shift order remains unchanged.

Thus, switching between MSB-first and LSB-first must be performed before writing data to the shift register.

(4) Transfer start

Serial transfer is started by setting transfer data to serial I/O shift register 1A0 (SIO1A0) when the following two conditions are satisfied.

- Bit 7 (CSIE10) of serial operation mode register 1A0 (CSIM1A0) = 1
- Internal serial clock is stopped or $\overline{\text{SCK10}}$ is high after 8-bit serial transfer.

Caution If CSIE10 is set to "1" after data is written to SIO1A0, transfer does not start.

Termination of 8-bit transfer stops the serial transfer automatically and generates the interrupt request signal (INTCSI10).

12.4.3 3-wire serial I/O mode with automatic transmit/receive function

This 3-wire serial I/O mode is used for transmission/reception of a maximum of 16-byte data without the use of software. Once transfer is started, the set number of bytes of the data prestored in the RAM can be transmitted, and the set number of bytes of data can be received and stored in the RAM.

(1) Register setting

The 3-wire serial I/O mode with automatic transmit/receive function is set with serial operation mode register 1A0 (CSIM1A0), automatic data transmit/receive control register 0 (ADTC0) and automatic data transmit/receive interval specification register 0 (ADTI0).

(a) Serial operation mode register 1A0 (CSIM1A0)

CSIM1A0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CSIM1A0 to 00H.

Caution Set the port mode register 2 (PM2) in the 3-wire serial I/O mode with automatic transmit/receive function as follows.

- **In the case of serial clock output (master transmission or master reception)**
  Set the $\overline{\text{SCK10}}$/P23 pin to output mode (PM23 = 0) and the P23 output latch to 0.
- **In the case of serial clock input (slave transmission or slave reception)**
  Set the $\overline{\text{SCK10}}$/P23 pin to input mode (PM23 = 1).
- **In transmission or transmission/reception mode**
  Set the SO10/P24 pin to output mode (PM24 = 0) and the P24 output latch to 0.
  Set the SI10/P25 pin to input mode (PM25 = 1).
- **In reception mode**
  Set the SI10/P25 pin to input mode (PM25 = 1).

Symbol	<7>	6	<5>	<4>	3	2	1	0	Address	After reset	R/W
CSIM1A0	CSIE10	DIR10	ATE0	LSCK10	0	0	SCL101	SCL100	FF78H	00H	R/W

CSIE10	Specification of operation enable/disable		
	Shift register operation	Serial counter	Port[Note]
0	Operation stopped	Cleared	Port function
1	Operation enabled	Count operation enabled	Serial function + port function

DIR10	Specification of first bit of serial transfer data
0	MSB
1	LSB

ATE0	Selection of operation mode
0	3-wire serial I/O mode
1	3-wire serial I/O mode with automatic transmit/receive function

LSCK10	Chip enable control of $\overline{\text{SCK10}}$ pin
0	$\overline{\text{SCK10}}$ is used as port (P23) when CSIE10 = 0. $\overline{\text{SCK10}}$ is used for clock output when CSIE10 = 1.
1	$\overline{\text{SCK10}}$ is fixed to high-level output when CSIE10 = 0. $\overline{\text{SCK10}}$ is used for clock output when CSIE10 = 1.

SCL101	SCL100	Selection of serial clock
0	0	External clock input to $\overline{\text{SCK10}}$ pin
0	1	$f_X/2^2$ (1.25 MHz)
1	0	$f_X/2^3$ (625 kHz)
1	1	$f_X/2^4$ (313 kHz)

Note When CSIE10 = 0 (SIO1A0 operation stop status), the $\overline{\text{SCK10}}$/P23, SO10/P24, and SI10/P25 pins can freely be used as port pins. Also, when CSIE10 is used for transmission only, the SI10/P25 pin can be used as P25 (CMOS I/O) (set bit 7 (RE0) of ADTC0 to 0).

Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 5.0 MHz.

(b) Automatic data transmit/receive control register 0 (ADTC0)

ADTC0 is set with a 1-bit or 8-bit memory manipulation instruction.

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

Symbol	<7>	<6>	5	4	<3>	2	1	0	Address	After reset	R/W
ADTC0	RE0	ARLD0	0	0	TRF0	0	0	0	FF79H	00H	R/W[Note 1]

RE0	Control of reception of automatic transmit/receive function
0	Reception disabled[Note 2]
1	Reception enabled

ARLD0	Selection of operation mode for automatic transmit/receive function
0	One-shot mode
1	Repeat mode

TRF0	Status of automatic transmit/receive function[Note 3]
0	Detection of termination of automatic transmission/reception (this bit is set to 0 upon suspension of automatic transmission/reception or when ARLD0 = 0)
1	Automatic transmission/reception in progress (this bit is set to 1 when data is written to SIO1A0)

Notes 1. Bit 3 (TRF0) is read-only.

2. When RE0 is reset to 0, P25 (CMOS I/O) is used even when bit 7 (CSIE10) of serial operation mode register 1A0 (CSIM1A0) is set to 1.

3. Use TRF0, instead of CSIIF10 (interrupt request flag), to identify the completion of automatic transmission/reception.

(c) Automatic data transmit/receive interval specification register 0 (ADTI0)

ADTI0 is set with a 1-bit or 8-bit memory manipulation instruction.

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

Symbol	<7>	6	5	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
ADTI0	ADTI07	0	0	ADTI04	ADTI03	ADTI02	ADTI01	ADTI00	FF7BH	00H	R/W

ADTI07	Data transfer interval control
0	No control of interval by ADTI00 to ADTI04[Note 1]
1	Control of interval by ADTI00 to ADTI04

ADTI04	ADTI03	ADTI02	ADTI01	ADTI00	Data transfer interval specification (f_X = 5.0 MHz, f_{SCK} = 1.25 MHz)[Note 2]	n
0	0	0	0	0	1.60 μs + 0.5/f_{SCK}	0
0	0	0	0	1		1
0	0	0	1	0	2.40 μs + 0.5/f_{SCK}	2
0	0	0	1	1	3.20 μs + 0.5/f_{SCK}	3
0	0	1	0	0	4.00 μs + 0.5/f_{SCK}	4
0	0	1	0	1	4.80 μs + 0.5/f_{SCK}	5
0	0	1	1	0	5.60 μs + 0.5/f_{SCK}	6
0	0	1	1	1	6.40 μs + 0.5/f_{SCK}	7
0	1	0	0	0	7.20 μs + 0.5/f_{SCK}	8
0	1	0	0	1	8.00 μs + 0.5/f_{SCK}	9
0	1	0	1	0	8.80 μs + 0.5/f_{SCK}	10
0	1	0	1	1	9.60 μs + 0.5/f_{SCK}	11
0	1	1	0	0	10.4 μs + 0.5/f_{SCK}	12
0	1	1	0	1	11.2 μs + 0.5/f_{SCK}	13
0	1	1	1	0	12.0 μs + 0.5/f_{SCK}	14
0	1	1	1	1	12.8 μs + 0.5/f_{SCK}	15

(Continued)

Symbol	<7>	6	5	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
ADTI0	ADTI07	0	0	ADTI04	ADTI03	ADTI02	ADTI01	ADTI00	FF7BH	00H	R/W

ADTI04	ADTI03	ADTI02	ADTI01	ADTI00	Data transfer interval specification (f_X = 5.0 MHz, f_{SCK} = 1.25 MHz)[Note 2]	n
1	0	0	0	0	13.6 μs + 0.5/f_{SCK}	16
1	0	0	0	1	14.4 μs + 0.5/f_{SCK}	17
1	0	0	1	0	15.2 μs + 0.5/f_{SCK}	18
1	0	0	1	1	16.0 μs + 0.5/f_{SCK}	19
1	0	1	0	0	16.8 μs + 0.5/f_{SCK}	20
1	0	1	0	1	17.6 μs + 0.5/f_{SCK}	21
1	0	1	1	0	18.4 μs + 0.5/f_{SCK}	22
1	0	1	1	1	19.2 μs + 0.5/f_{SCK}	23
1	1	0	0	0	20.0 μs + 0.5/f_{SCK}	24
1	1	0	0	1	20.8 μs + 0.5/f_{SCK}	25
1	1	0	1	0	21.6 μs + 0.5/f_{SCK}	26
1	1	0	1	1	22.4 μs + 0.5/f_{SCK}	27
1	1	1	0	0	23.2 μs + 0.5/f_{SCK}	28
1	1	1	0	1	24.0 μs + 0.5/f_{SCK}	29
1	1	1	1	0	24.8 μs + 0.5/f_{SCK}	30
1	1	1	1	1	25.6 μs + 0.5/f_{SCK}	31

Notes 1. The interval time depends only on the CPU processing.

2. The data transfer interval time is found from the following expressions (n: Value set to ADTI00 to ADTI04).

<1> n = 0

$$\text{Interval time} = \frac{2}{f_{SCK}} + \frac{0.5}{f_{SCK}}$$

<2> n = 1 to 31

$$\text{Interval time} = \frac{n+1}{f_{SCK}} + \frac{0.5}{f_{SCK}}$$

Cautions 1. Do not write to ADTI0 during operation of the automatic transmit/receive function.

2. Be sure to set bits 5 and 6 to 0.

Remark f_X: Main system clock oscillation frequency

f_{SCK}: Serial clock frequency

(2) Automatic transmit/receive data setting

(a) Transmit data setting

<1> Write transmit data from the least significant address FFA0H of buffer RAM (up to FFAFH). The transmit data should be in the order from higher address to lower address.
<2> Set the value obtained by subtracting 1 from the number of transmit data bytes to automatic data transmit/receive address pointer 0 (ADTP0).

(b) Automatic transmit/receive mode setting

<1> Set bit 7 (CSIE10) and bit 5 (ATE0) of serial operation mode register 1A0 (CSIM1A0) to 1.
<2> Set bit 7 (RE0) of automatic data transmit/receive control register 0 (ADTC0) to 1.
<3> Set the data transmit/receive interval in automatic data transmit/receive interval specification register 0 (ADTI0).
<4> Write any value to serial I/O shift register 1A0 (SIO1A0) (transfer start trigger).

Caution Writing any value to SIO1A0 orders the start of automatic transmission/reception operation; the written value has no meaning.

The following operations are automatically carried out when (a) and (b) are carried out.

- After the buffer RAM data specified by ADTP0 is transferred to SIO1A0, transmission is carried out (start of automatic transmission/reception).
- The received data is written to the buffer RAM address specified by ADTP0.
- ADTP0 is decremented and the next data transmission/reception is carried out. Data transmission/reception continues until the ADTP0 decremental output becomes 00H and address FFA0H data is output (end of automatic transmission/reception).
- When automatic transmission/reception is terminated, bit 3 (TRF0) of ADTC0 is cleared to 0.

(3) Communication operation

(a) Basic transmit/receive mode

This transmit/receive mode is the same as the 3-wire serial I/O mode in which the specified number of data are transmitted/received in 8-bit units.

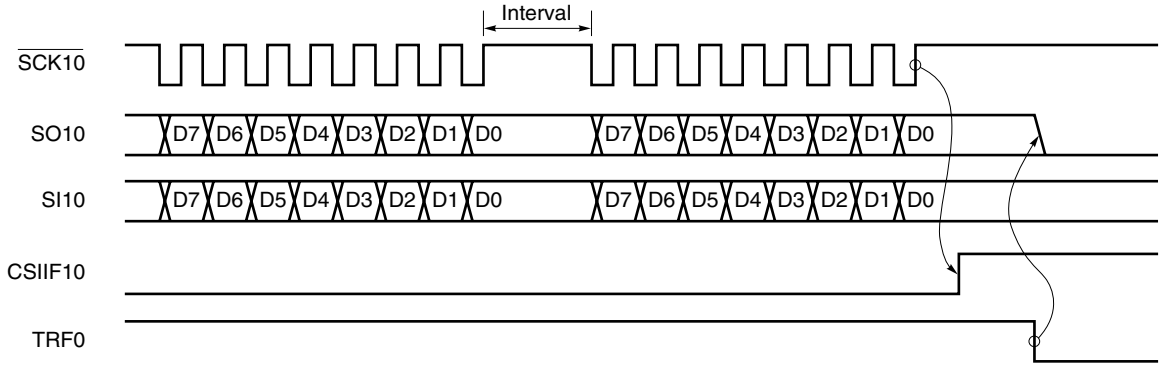
Serial transfer is started when any data is written to serial I/O shift register 1A0 (SIO1A0) while bit 7 (CSIE10) of serial operation mode register 1A0 (CSIM1A0) is set to 1.

Upon completion of transmission of the last byte, the interrupt request flag (CSIIF10) is set. The termination of automatic transmission/reception should be checked by using bit 3 (TRF0) of automatic data transmit/receive control register 0 (ADTC0), not by CSIIF10 because CSIIF10 of the interrupt request flag is cleared if an interrupt is acknowledged.

Figure 12-7 shows the basic transmit/receive mode operation timing, and Figure 12-8 shows the operation flowchart.

Figure 12-9 shows buffer RAM operation at 6-byte transmission.

Figure 12-7. Basic Transmit/Receive Mode Operation Timing

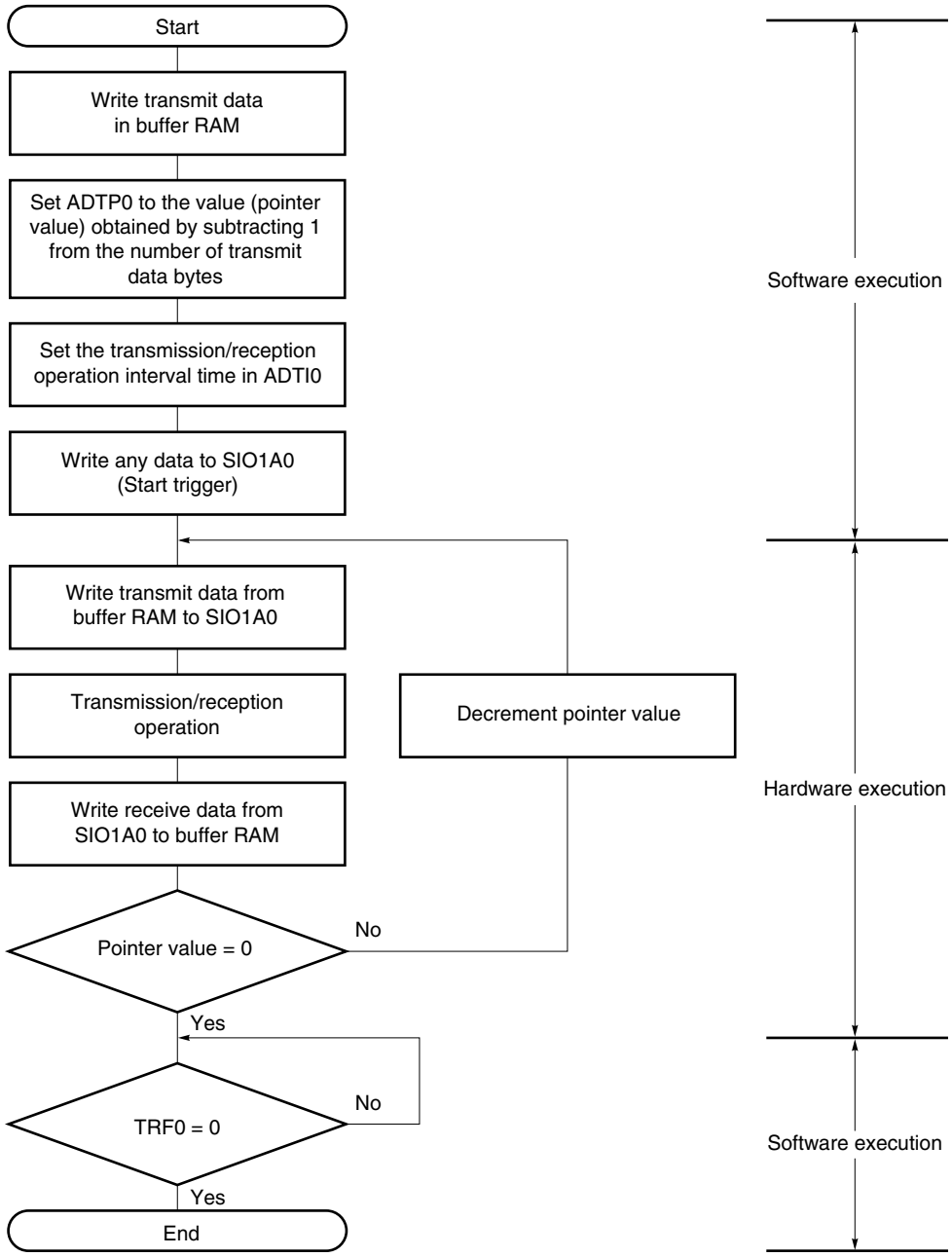


Cautions 1. Because, in the basic transmit/receive mode, the automatic transmit/receive function writes/reads data to/from the buffer RAM after 1-byte transmission/reception, an interval is inserted till the next transmission/reception. As the buffer RAM write/read is performed at the same time as CPU processing, the maximum interval is dependent upon CPU processing and the value of automatic data transmit/receive interval specification register 0 (ADTI0) (refer to 12.4.3 (5) Interval time of automatic transmission/reception).

2. When TRF0 is cleared, the SO10 pin becomes low level.

Remark CSIIF10: Interrupt request flag
TRF0: Bit 3 of automatic data transmit/receive control register 0 (ADTC0)

Figure 12-8. Basic Transmit/Receive Mode Flowchart



Remark ADTP0: Automatic data transmit/receive address pointer 0
ADTI0: Automatic data transmit/receive interval specification register 0
SIO1A0: Serial I/O shift register 1A0
TRF0: Bit 3 of automatic data transmit/receive control register 0 (ADTC0)

In 6-byte transmission/reception (bit 6 (ARLD0) and bit 7 (RE0) of automatic data transmit/receive control register 0 (ADTC0) are 0 and 1, respectively) in basic transmit/receive mode, buffer RAM operates as follows.

(i) Before transmission/reception (refer to Figure 12-9 (a))

After any data has been written to SIO1A0 (start trigger: this data is not transferred), transmit data 1 (T1) is transferred from the buffer RAM to SIO1A0. When transmission of the first byte is completed, receive data 1 (R1) is transferred from SIO1A0 to the buffer RAM, and automatic data transmit/receive address pointer 0 (ADTP0) is decremented. Then transmit data 2 (T2) is transferred from the buffer RAM to SIO1A0.

(ii) 4th byte transmit/receive point (refer to Figure 12-9 (b))

Transmission/reception of the third byte is completed, and transmit data 4 (T4) is transferred from the buffer RAM to SIO1A0. When transmission of the fourth byte is completed, receive data 4 (R4) is transferred from SIO1A0 to the buffer RAM, and ADTP0 is decremented.

(iii) Completion of transmission/reception (refer to Figure 12-9 (c))

When transmission of the sixth byte is completed, receive data 6 (R6) is transferred from SIO1A0 to the buffer RAM, and the interrupt request flag (CSIIF10) is set (INTCSI10 generation).

Figure 12-9. Buffer RAM Operation in 6-Byte Transmission/Reception (in Basic Transmit/Receive Mode) (1/2)

(a) Before transmission/reception

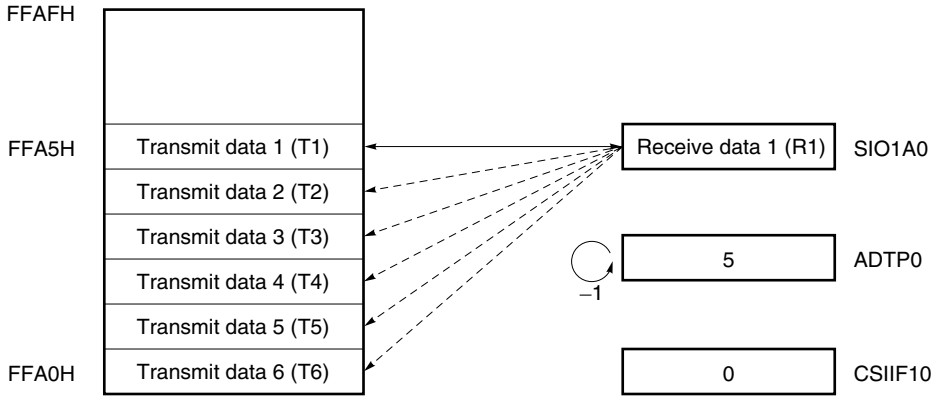
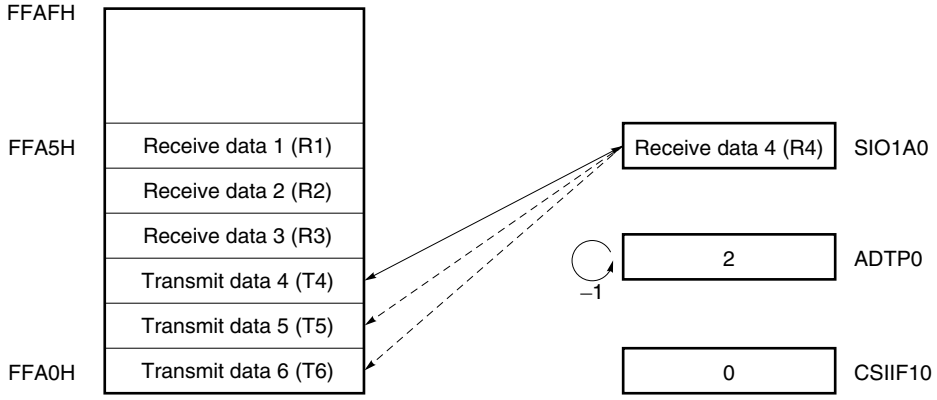
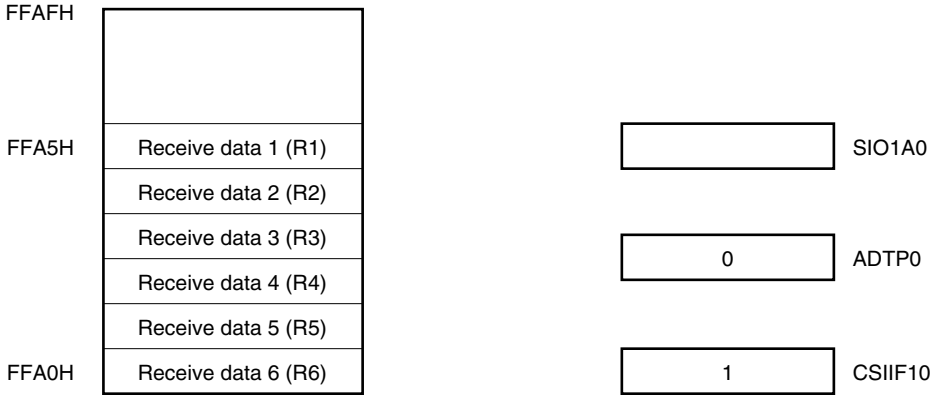


Figure 12-9. Buffer RAM Operation in 6-Byte Transmission/Reception (in Basic Transmit/Receive Mode) (2/2)

(b) 4th byte transmission/reception



(c) Completion of transmission/reception



(b) Basic transmit mode

In this mode, the specified number of 8-bit unit data are transmitted.

Serial transfer is started when any data is written to serial I/O shift register 1A0 (SIO1A0) while bit 7 (CSIE10) of serial operation mode register 1A0 (CSIM1A0) is set to 1, and bit 7 (RE0) of automatic data transmit/receive control register 0 (ADTC0) is set to 0.

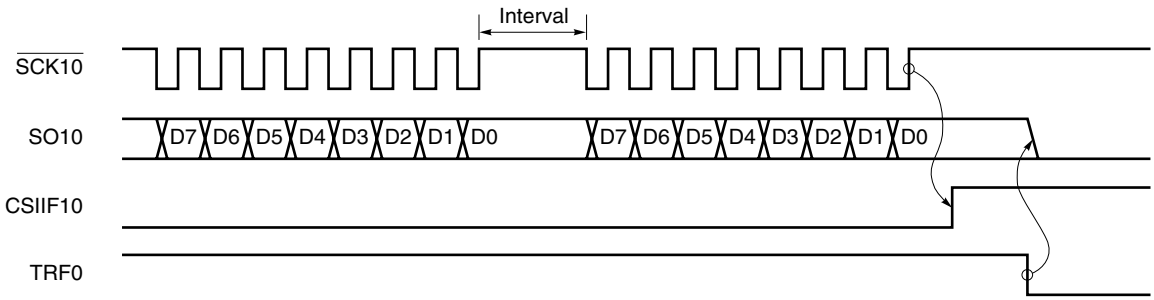
Upon completion of transmission of the last byte, the interrupt request flag (CSIIF10) is set. The termination of automatic transmission/reception should be checked by using bit 3 (TRF0) of automatic data transmit/receive control register 0 (ADTC0), not by CSIIF10.

If a receive operation is not executed, the P25/SI10 pin can be used as normal I/O port.

Figure 12-10 shows the basic transmit mode operation timing, and Figure 12-11 shows the operation flowchart.

Figure 12-12 shows buffer RAM operation when repeatedly transmit 6 bytes.

Figure 12-10. Basic Transmit Mode Operation Timing



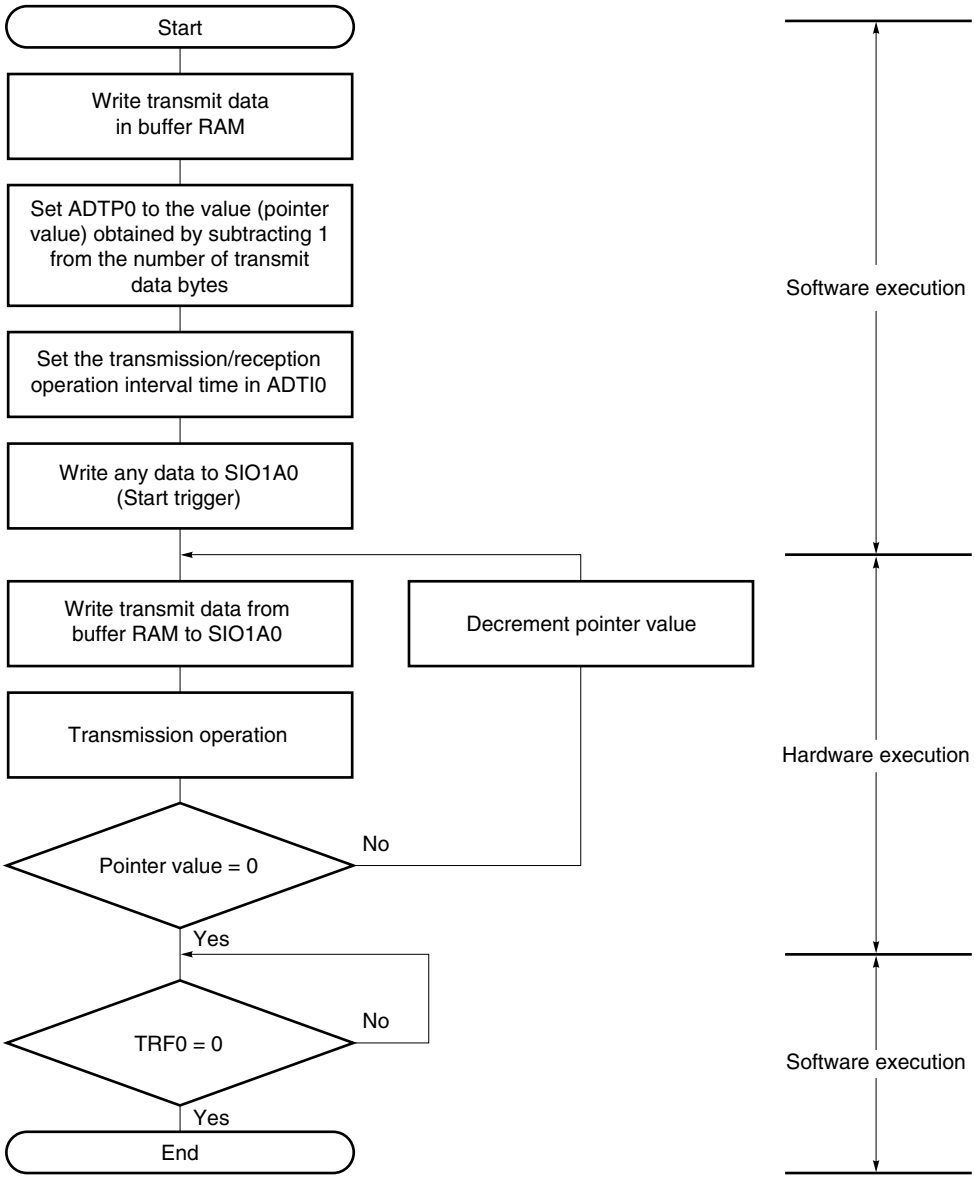
Cautions 1. Because, in the basic transmit mode, the automatic transmit/receive function reads data from the buffer RAM after 1-byte transmission, an interval is inserted until the next transmission. As the buffer RAM read is performed at the same time as CPU processing, the maximum interval is dependent upon CPU processing and the value of automatic data transmit/receive interval specification register 0 (ADTI0) (refer to 12.4.3 (5) Interval time of automatic transmission/reception).

2. When TRF0 is cleared, the SO10 pin becomes low level.

Remark CSIIF10: Interrupt request flag

TRF0: Bit 3 of automatic data transmit/receive control register 0 (ADTC0)

Figure 12-11. Basic Transmit Mode Flowchart



Remark ADTP0: Automatic data transmit/receive address pointer 0
ADTI0: Automatic data transmit/receive interval specification register 0
SIO1A0: Serial I/O shift register 1A0
TRF0: Bit 3 of automatic data transmit/receive control register 0 (ADTC0)

In 6-byte transmission (bit 6 (ARLD0) and bit 7 (RE0) of automatic data transmit/receive control register 0 (ADTC0) are 0) in basic transmit mode, buffer RAM operates as follows.

(i) Before transmission (refer to Figure 12-12 (a))

After any data has been written to SIO1A0 (start trigger: this data is not transferred), transmit data 1 (T1) is transferred from the buffer RAM to SIO1A0. When transmission of the first byte is completed, ADTP0 is decremented. Then transmit data 2 (T2) is transferred from the buffer RAM to SIO1A0.

(ii) 4th byte transmission point (refer to Figure 12-12 (b))

Transmission of the third byte is completed, and transmit data 4 (T4) is transferred from the buffer RAM to SIO1A0. When transmission of the fourth byte is completed, ADTP0 is decremented.

(iii) Completion of transmission/reception (refer to Figure 12-12 (c))

When transmission of the sixth byte is completed, the interrupt request flag (CSIIF10) is set (INTCSI10 generation).

Figure 12-12. Buffer RAM Operation in 6-Byte Transmission (in Basic Transmit Mode) (1/2)

(a) Before transmission

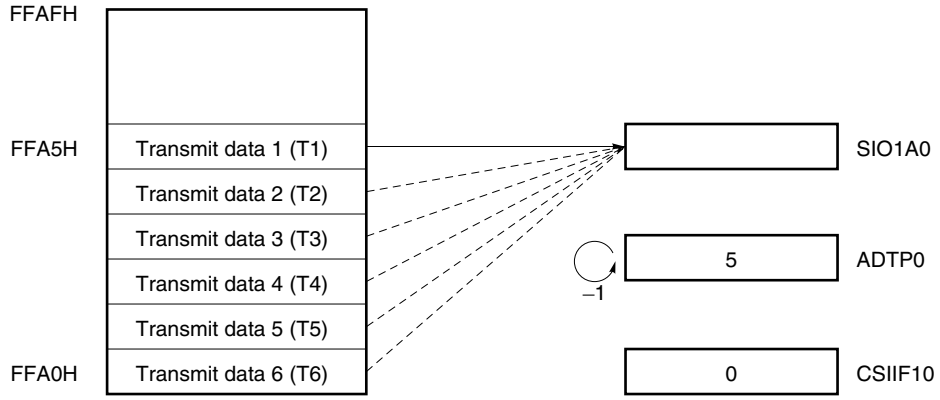
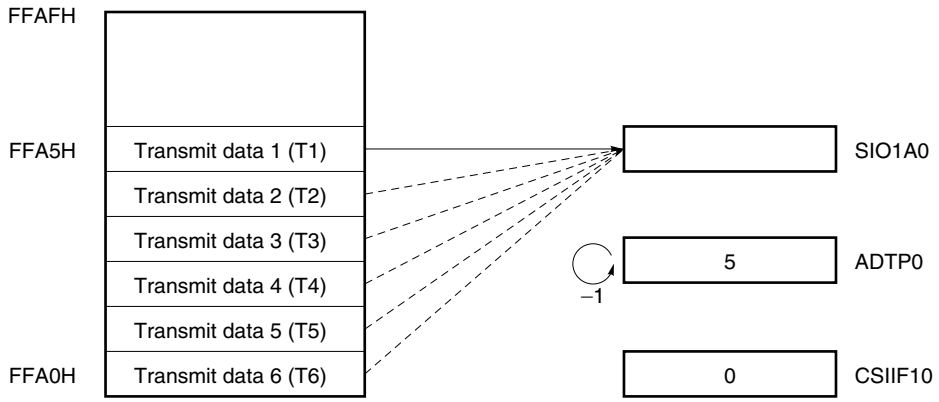
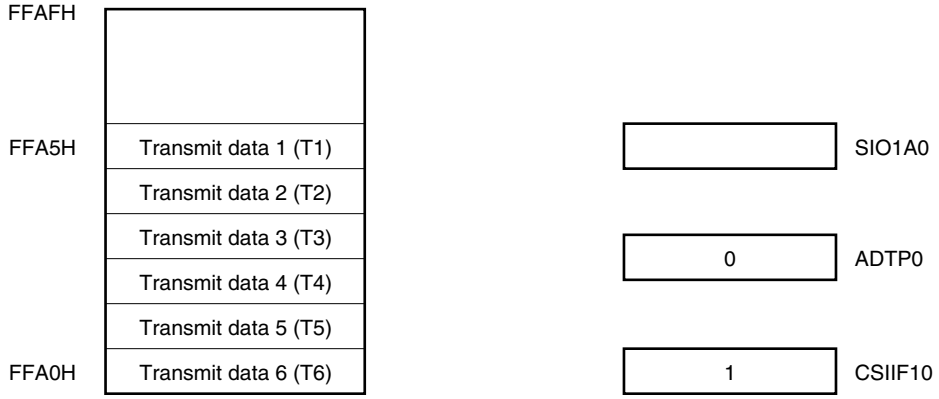


Figure 12-12. Buffer RAM Operation in 6-Byte Transmission (in Basic Transmit Mode) (2/2)

(b) 4th byte transmission point



(c) Completion of transmission/reception



(c) Repeat transmit mode

In this mode, data stored in the buffer RAM is transmitted repeatedly.

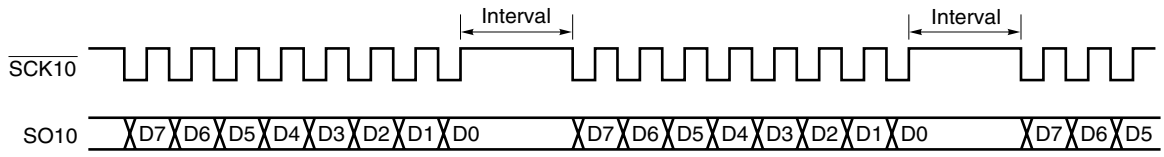
Serial transfer is started by writing any data to serial I/O shift register 1A0 (SIO1A0) when bit 7 (CSIE10) of serial operation mode register 1A0 (CSIM1A0) is set to 1, and bit 7 (RE0) of automatic data transmit/receive control register 0 (ADTC0) is set to 0.

Unlike the basic transmission mode, after the last byte (data in address FFA0H) has been transmitted, the interrupt request flag (CSIIF10) is not set, the value at the time when the transmission was started is set in automatic data transmit/receive address pointer 0 (ADTP0) again, and the buffer RAM contents are transmitted again.

When a reception operation is not performed, the P25/SI10 pin can be used as a normal I/O port.

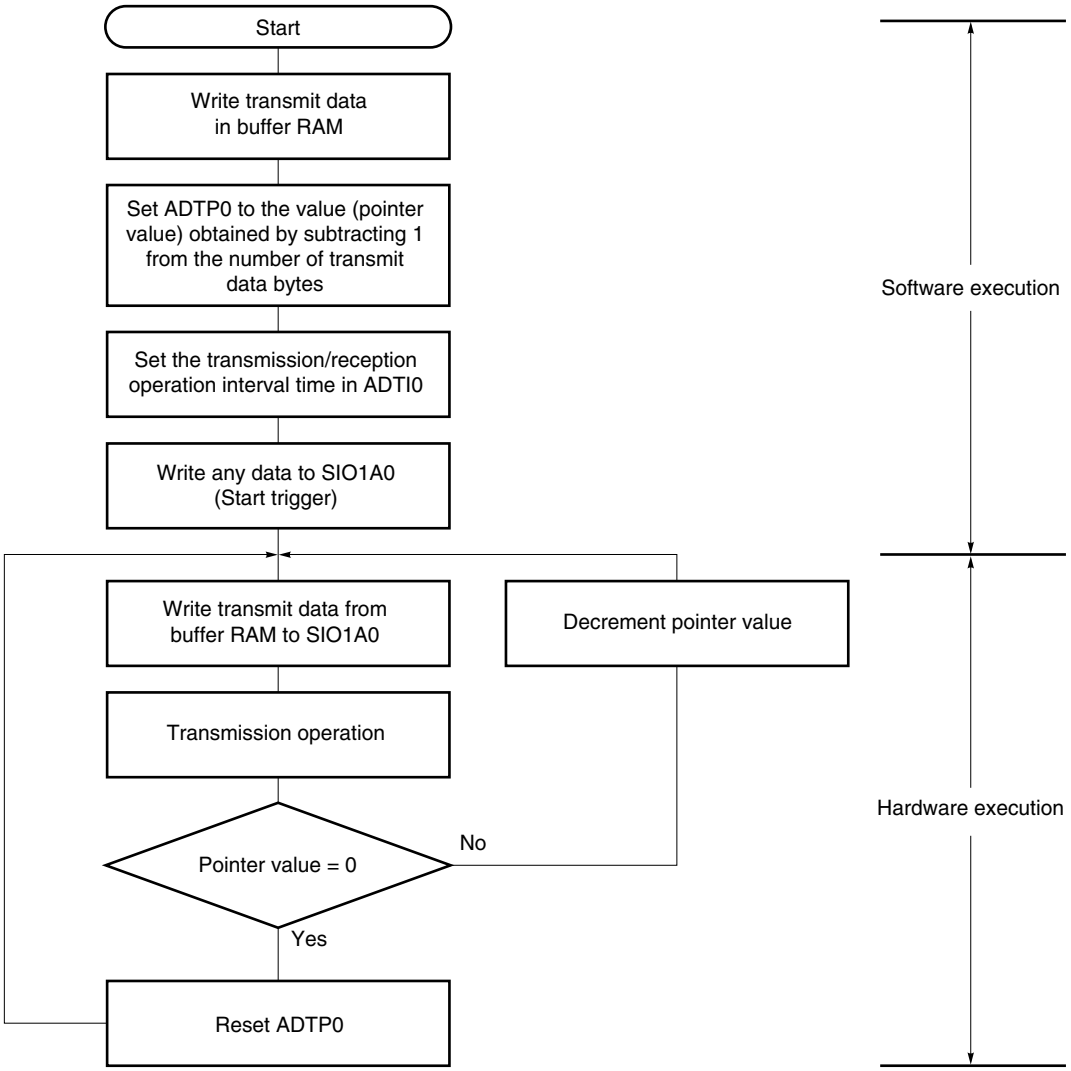
The repeat transmit mode operation timing is shown in Figure 12-13, and the operation flowchart in Figure 12-14.

Figure 12-13. Repeat Transmit Mode Operation Timing



Caution Because, in the repeat transmit mode, a read is performed on the buffer RAM after the transmission of one byte, the interval is included in the period up to the next transmission. As the buffer RAM read is performed at the same time as CPU processing, the maximum interval is dependent upon the CPU operation and the value of automatic data transmit/receive interval specification register 0 (ADTI0) (refer to 12.4.3 (5) Interval time of automatic transmission/reception).

Figure 12-14. Repeat Transmit Mode Flowchart



Remark ADTP0: Automatic data transmit/receive address pointer 0
ADTI0: Automatic data transmit/receive interval specification register 0
SIO1A0: Serial I/O shift register 1A0

In 6-byte transmission (bit 6 (ARLD0) and bit 7 (RE0) of automatic data transmit/receive control register 0 (ADTC0) are 1 and 0, respectively) in repeat transmit mode, buffer RAM operates as follows.

(i) Before transmission (refer to Figure 12-15 (a))

After any data has been written to SIO1A0 (start trigger: this data is not transferred), transmit data 1 (T1) is transferred from the buffer RAM to SIO1A0. When transmission of the first byte is completed, ADTP0 is decremented. Then transmit data 2 (T2) is transferred from the buffer RAM to SIO1A0.

(ii) Upon completion of transmission of 6 bytes (refer to Figure 12-15 (b))

When transmission of the sixth byte is completed, the interrupt request flag (CSIIF10) is not set. The previous pointer value is assigned to the ADTP0.

(iii) 7th byte transmission point (refer to Figure 12-15 (c))

Transmit data 1 (T1) is transferred from the buffer RAM to SIO1A0 again. When transmission of the first byte is completed, ADTP0 is decremented. Then transmit data 2 (T2) is transferred from the buffer RAM to SIO1A0.

Figure 12-15. Buffer RAM Operation in 6-Byte Transmission (in Repeat Transmit Mode) (1/2)

(a) Before transmission

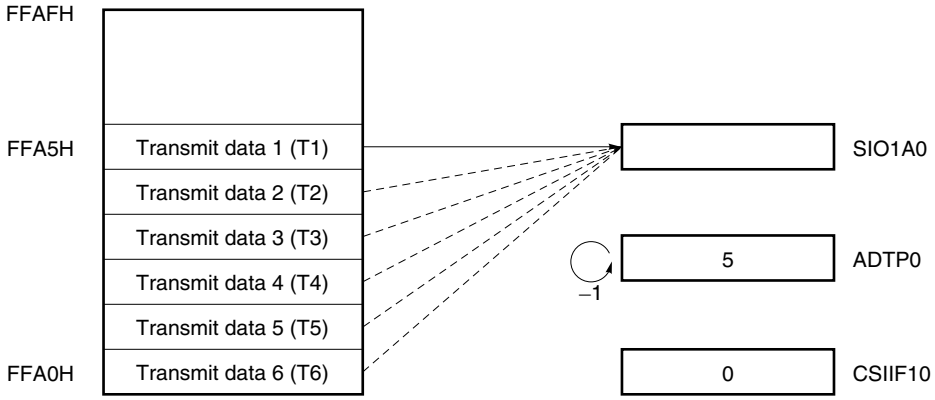
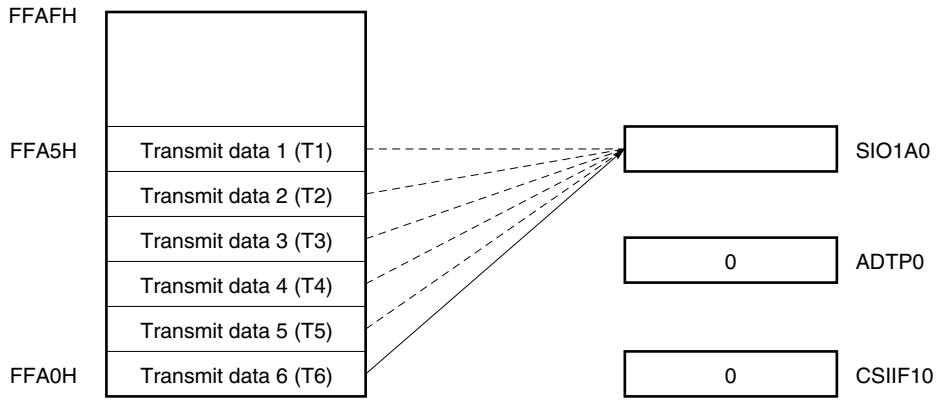
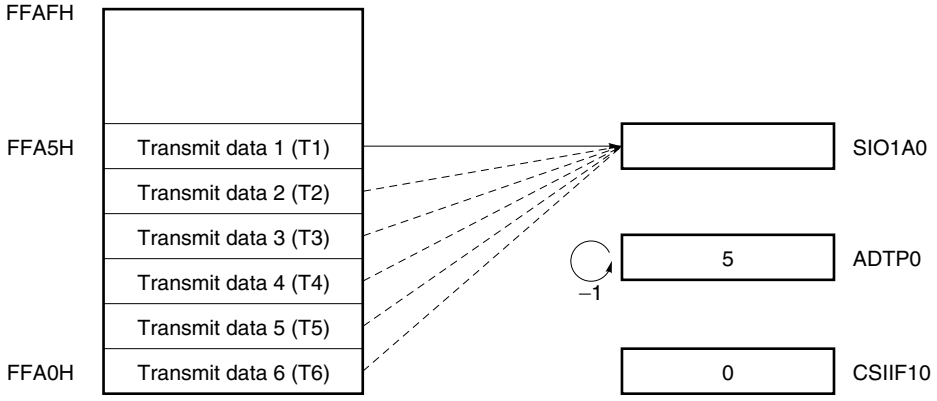


Figure 12-15. Buffer RAM Operation in 6-Byte Transmission (in Repeat Transmit Mode) (2/2)

(b) Upon completion of transmission of 6 bytes



(c) 7th byte transmission point



(d) Automatic transmission/reception suspension and restart

Automatic transmission/reception can be temporarily suspended by setting bit 7 (CSIE10) of serial operation mode register 1A0 (CSIM1A0) to 0.

During 8-bit data transfer, the transmission/reception is not suspended if bit 7 (CSIE10) is set to 0. It is suspended upon completion of 8-bit data transfer.

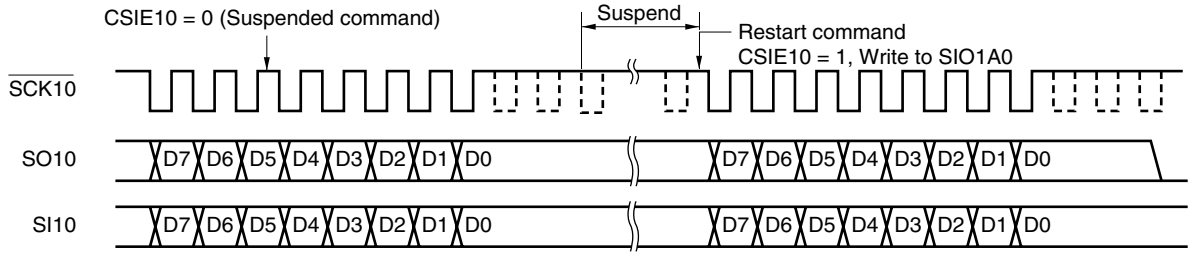
When suspended, bit 3 (TRF0) of automatic data transmit/receive control register 0 (ADTC0) is set to 0 after transfer of the 8th bit, and all the port pins used alternately as serial interface pins (P23/$\overline{\text{SCK10}}$, P24/SO10, P25/SI10) are set to the port mode.

During restart of transmission/reception, the remaining data can be transferred by setting CSIE10 to 1 and writing any data to serial I/O shift register 1A0 (SIO1A0).

Cautions 1. If the HALT instruction is executed during automatic transmission/reception, transfer is suspended and the HALT mode is set even if 8-bit data is being transferred.

2. When suspending automatic transmission/reception, do not change the operation mode to 3-wire serial mode while TRF0 = 1.

Figure 12-16. Automatic Transmission/Reception Suspension and Restart



CSIE10: Bit 7 of serial operation mode register 1A0 (CSIM1A0)

(4) Timing of interrupt request signal generation

The interrupt request signal is generated in synchronization with the timing shown in Table 12-2.

Table 12-2. Timing of Interrupt Request Signal Generation

Operation Mode		Timing of Interrupt Request Signal
Single mode	Master mode	10th serial clock at end of transfer
	Slave mode	8th serial clock at end of transfer
Repeat transmit mode		Not generated

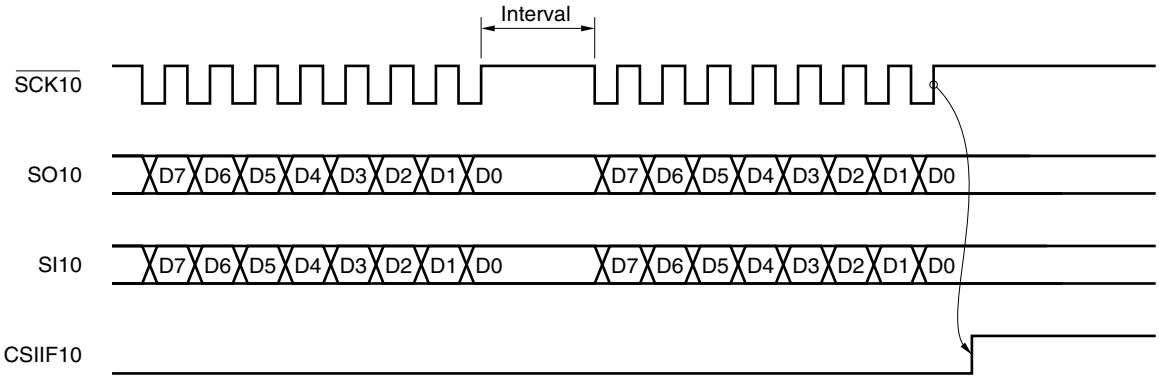
(5) Interval time of automatic transmission/reception

Because read/write to/from the buffer RAM using the automatic transmit/receive function is performed asynchronously to the CPU processing, the interval time is dependent on the CPU processing of the timing of the eighth rising of the serial clock and the set value of automatic data transmit/receive interval specification register 0 (ADTI0). Whether the interval time is dependent on ADTI0 is selected by setting bit 7 (ADTI07) of ADTI0. If ADTI07 is reset to 0, the interval time is 2/f_{SCK}. If ADTI07 is set to 1, whichever is greater of the interval time determined by the set contents of ADTI0 or the interval time (2/f_{SCK}) determined by the CPU processing is selected.

Figure 12-17 shows the interval time of automatic transmission/reception.

Remark f_{SCK}: Serial clock frequency

Figure 12-17. Interval Time of Automatic Transmission/Reception



The following expression must be satisfied to access the buffer RAM.

1 transfer cycle + Interval time ≥ Read access + Write access + CPU buffer RAM access (time)

In the case of a “high-speed CPU & low-speed SCK”, the interval time is not necessary. In the case of a “low-speed CPU & high-speed SCK”, the interval time is necessary.

In this case, make sure that a sufficient interval time elapses, by using automatic data transmit/receive interval specification register 0 (ADTI0), so that the above expression is satisfied.

CHAPTER 13 LCD CONTROLLER/DRIVER

13.1 LCD Controller/Driver Functions

The functions of the LCD controller/driver of the μPD789479 Subseries are as follows.

(1) Automatic output of segment and common signals based on automatic display data memory read
(2) Two different display modes:
- 1/3 duty (1/3 bias)
- 1/4 duty (1/3 bias)

(3) Four different frame frequencies, selectable in each display mode
(4) 16 to 28 segment signal outputs (S0 to S15, S16 to S27[Note]), 4 common signal outputs (COM0 to COM3)
(5) Operation with subsystem clock is possible

Note Usable by mask option or port function register

The maximum number of displayable pixels is shown in Table 13-1 below.

Table 13-1. Maximum Number of Display Pixels

Bias Method	Time Division	Common Signals Used	Maximum Number of Segments	Maximum Number of Display Pixels
1/3	3	COM0 to COM2	28	84 (28 segments × 3 commons)[Note 1]
	4	COM0 to COM3		112 (28 segments × 4 commons)[Note 2]

Notes 1. The LCD panel of the figure 8. consists of 9 rows with 3 segments per row.
2. The LCD panel of the figure 8. consists of 14 rows with 2 segments per row.

13.2 LCD Controller/Driver Configuration

The LCD controller/driver includes the following hardware.

Table 13-2. Configuration of LCD Controller/Driver

Item	Configuration
Display outputs	Segment signals: 16 to 28 Common signals: 4 (COM0 to COM3)
Control registers	LCD display mode register 0 (LCDM0) LCD clock control register 0 (LCDC0)

The correspondence with the LCD display RAM is shown in Figure 13-1 below.

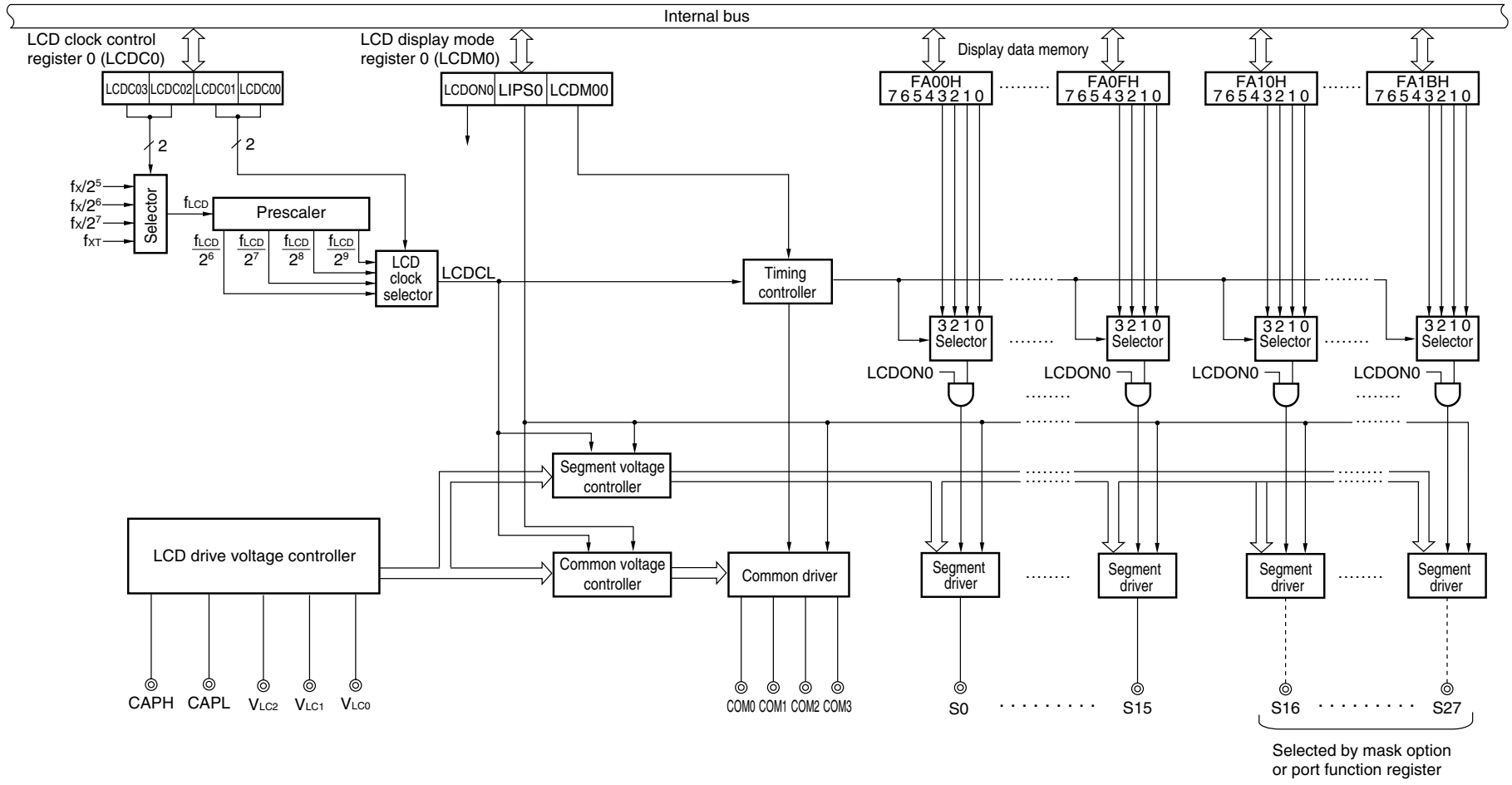
Figure 13-1. Correspondence with LCD Display RAM

Address	Bit								Segment
	7	6	5	4	3	2	1	0	
FA1BH	0	0	0	0					→ S27[Note]
FA1AH	0	0	0	0					→ S26[Note]
FA19H	0	0	0	0					→ S25[Note]
FA18H	0	0	0	0					→ S24[Note]
FA17H	0	0	0	0					→ S23[Note]
FA16H	0	0	0	0					→ S22[Note]
FA15H	0	0	0	0					→ S21[Note]
FA14H	0	0	0	0					→ S20[Note]
FA13H	0	0	0	0					→ S19[Note]
FA12H	0	0	0	0					→ S18[Note]
FA11H	0	0	0	0					→ S17[Note]
FA10H	0	0	0	0					→ S16[Note]
FA0FH	0	0	0	0					→ S15
FA0EH	0	0	0	0					→ S14
FA0DH	0	0	0	0					→ S13
FA0CH	0	0	0	0					→ S12
FA0BH	0	0	0	0					→ S11
FA0AH	0	0	0	0					→ S10
FA09H	0	0	0	0					→ S9
FA08H	0	0	0	0					→ S8
FA07H	0	0	0	0					→ S7
FA06H	0	0	0	0					→ S6
FA05H	0	0	0	0					→ S5
FA04H	0	0	0	0					→ S4
FA03H	0	0	0	0					→ S3
FA02H	0	0	0	0					→ S2
FA01H	0	0	0	0					→ S1
FA00H	0	0	0	0					→ S0
					↑	↑	↑	↑	
				Common	COM3	COM2	COM1	COM0	

Note Segments S16 to S27 are selected in 1-bit units via a mask option or port function register (segment output pin/port pin).

Remark Bits 4 to 7 are fixed to 0.

Figure 13-2. LCD Controller/Driver Block Diagram



13.3 Registers Controlling LCD Controller/Driver

The LCD controller/driver is controlled by the following two registers.

- LCD display mode register 0 (LCDM0)
- LCD clock control register 0 (LCDC0)

(1) LCD display mode register 0 (LCDM0)

LCDM0 specifies whether to enable display. It also specifies the segment/common pin output and display mode.

LCDM0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets LCDM0 to 00H.

Figure 13-3. Format of LCD Display Mode Register 0

Symbol	<7>	6	5	<4>	3	2	1	0	Address	After reset	R/W
LCDM0	LCDON0	0	0	LIPS0	0	0	0	LCDM00	FFB0H	00H	R/W

LCDON0	LCD display enable/disable
0	Display off (all segment outputs are deselect signal outputs)
1	Display on

LIPS0	Segment pin/common pin output control[Note]
0	Output ground level to segment/common pin
1	Output select level to segment pin and LCD waveform to common pin

LCDM00	LCD controller/driver display mode selection	
	Number of time slices	Bias mode
0	4	1/3
1	3	1/3

Note When the LCD display panel is not used, set LIPS0 to 0 to reduce power consumption.

Caution Bits 1 to 3, 5, and 6 must be set to 0.

(2) LCD clock control register 0 (LCDC0)

LCDC0 specifies the LCD source clock and LCD clock. The frame frequency is determined according to the LCD clock and number of time slices.

LCDC0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets LCDC0 to 00H.

Figure 13-4. Format of LCD Clock Control Register 0

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
LCDC0	0	0	0	0	LCDC03	LCDC02	LCDC01	LCDC00	FFB2H	00H	R/W

LCDC03	LCDC02	LCD source clock (f_{LCD}) selection[Note]
0	0	f_{XT} (32.768 kHz)
0	1	$f_X/2^5$ (156.3 kHz)
1	0	$f_X/2^6$ (78.1 kHz)
1	1	$f_X/2^7$ (39.1 kHz)

LCDC01	LCDC00	LCD clock (LCDCL) selection
0	0	$f_{LCD}/2^6$
0	1	$f_{LCD}/2^7$
1	0	$f_{LCD}/2^8$
1	1	$f_{LCD}/2^9$

Note Specify an LCD source clock (f_{LCD}) frequency of at least 32 kHz.

Cautions 1. Bits 4 to 7 must be set to 0.

2. Before changing the LCDC0 setting, be sure to disable the display (LCDON0 = 0).

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

3. The parenthesized values apply to operation at f_X = 5.0 MHz or f_{XT} = 32.768 kHz.

As an example, Table 13-3 lists the frame frequencies used when f_{XT} (32.768 kHz) is supplied as the LCD source clock (f_{LCD}).

Table 13-3. Frame Frequencies (Hz)

LCD Clock (LCDCL) Frequency / Number of Time Slices	$f_{XT}/2^9$ (64 Hz)	$f_{XT}/2^8$ (128 Hz)	$f_{XT}/2^7$ (256 Hz)	$f_{XT}/2^6$ (512 Hz)
3	21	43	85	171
4	16	32	64	128

13.4 Setting LCD Controller/Driver

Set the LCD controller/driver using the following procedure.

<1> Set the LCD clock using LCD clock control register 0 (LCDC0).
<2> Set the time slice using LCDM00 (bit 0 of LCD display mode register 0 (LCDM0)).
<3> Set LIPS0 (bit 4 of LCDM0) (LIPS0 = 1) and output the deselect potential.
<4> Start output corresponding to each data memory by setting LCDON0 (bit 7 of LCDM0) (LCDON0 = 1).

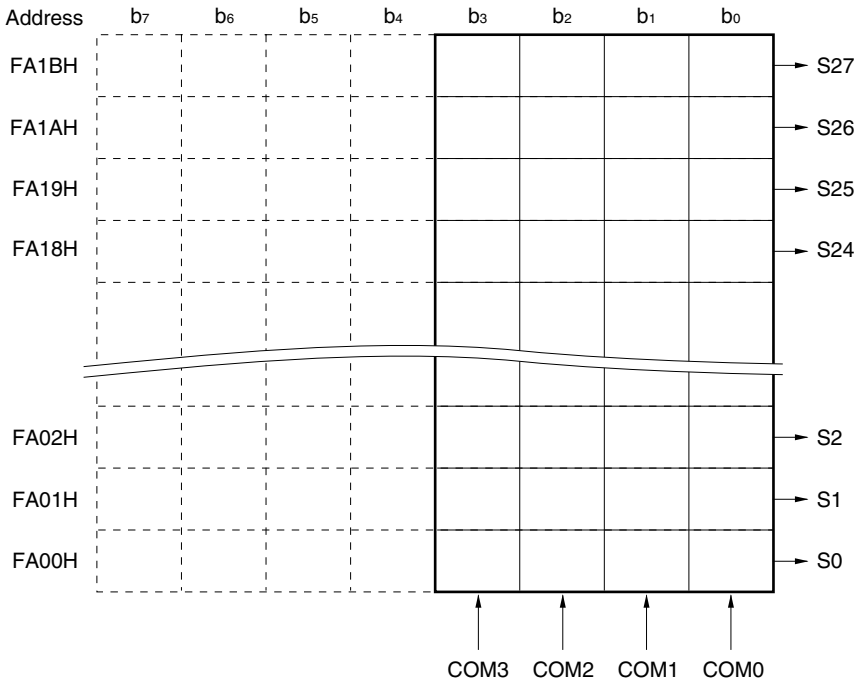
13.5 LCD Display Data Memory

The LCD display data memory is mapped at addresses FA00H to FA1BH. Data in the LCD display data memory can be displayed on the LCD panel using the LCD controller/driver.

Figure 13-5 shows the relationship between the contents of the LCD display data memory and the segment/common outputs.

That part of the display data memory which is not used for display can be used as ordinary RAM.

Figure 13-5. Relationship Between LCD Display Data Memory Contents and Segment/Common Outputs (When Using S16 to S27)



Caution No memory has been installed as the higher 4 bits of the LCD display data memory. Be sure to set them to 0.

13.6 Common and Segment Signals

Each pixel of the LCD panel turns on when the potential difference between the corresponding common and segment signals becomes higher than a specific voltage (LCD drive voltage, V_{LCD}). It turns off when the potential difference becomes lower than V_{LCD}.

Applying DC voltage to the common and segment signals for an LCD panel would deteriorate it. To avoid this problem, this LCD panel is driven with AC voltage.

(1) Common signals

Each common signal is selected sequentially according to the specified number of time slices at the timing listed in Table 13-4. This cycle of operation is performed repeatedly.

In the three-time-slice mode, leave the COM3 pin open.

Table 13-4. COM Signals

COM Signal / Number of Time Slices	COM0	COM1	COM2	COM3
Three-time-slice mode				Open
Four-time-slice mode				

(2) Segment signals

The segment signals correspond to LCD display data memory. Bits 0, 1, 2, and 3 of each byte are read in synchronization with COM0, COM1, COM2, and COM3, respectively. If the contents of each bit are 1, that bit is converted to the select voltage, and if 0, it is converted to the deselect voltage. The conversion results are output to the segment pins.

Check, with the information given above, what combination of the front-surface electrodes (corresponding to the segment signals) and the rear-surface electrodes (corresponding to the common signals) forms display patterns in the LCD display data memory, and write the bit data that corresponds to the desired display pattern on a one-to-one basis.

Bit 3 of the LCD display data memory is not used for LCD display in the three-time-slice mode. So this bit can be used for purposes other than display.

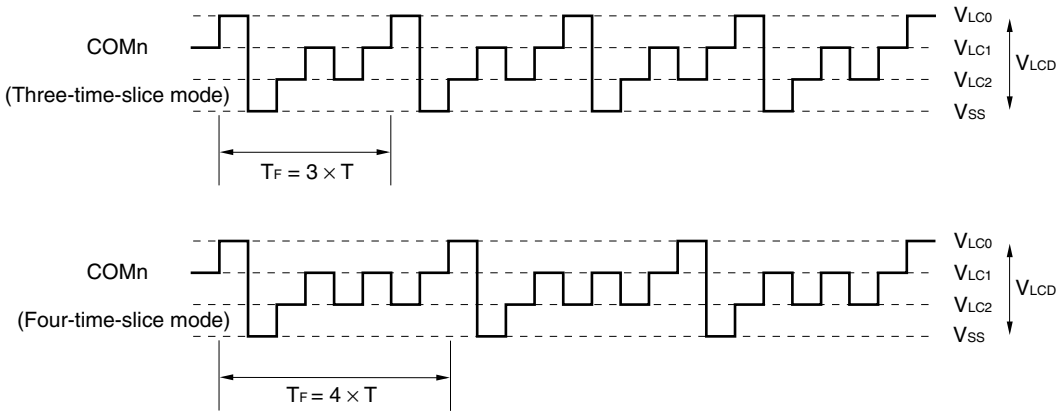
LCD display data memory bits 4 to 7 are fixed to 0.

(3) Output waveforms of common and segment signals

When both common and segment signals are at the select voltage, a display-on voltage of $\pm V_{LCD}$ is obtained. The other combinations of the signals correspond to the display-off voltage.

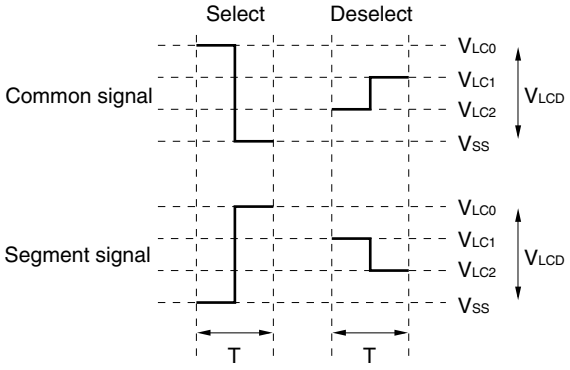
Figure 13-6 shows the common signal waveforms, and Figure 13-7 shows the voltages and phases of the common and segment signals.

Figure 13-6. Common Signal Waveforms



T: One LCD clock period TF: Frame frequency

Figure 13-7. Voltages and Phases of Common and Segment Signals



T: One LCD clock period

13.7 Display Modes

13.7.1 Three-time-slice display example

Figure 13-9 shows how a nine-digit LCD panel having the display pattern shown in Figure 13-8 is connected to the segment signals (S0 to S26) and the common signals (COM0 to COM2) of the μPD789479 Subseries chip. This example displays the data "123456.789" in the LCD panel. The contents of the display data memory (addresses FA00H to FA1AH) correspond to this display.

The following description focuses on numeral "6." (6.) displayed as the fourth digit from the right. To display "6." in the LCD panel, it is necessary to apply the select or deselect voltage to the S9 to S11 pins according to Table 13-5 at the timing of the common signals COM0 to COM2; see Figure 13-8 for the relationship between the segment signals and LCD segments.

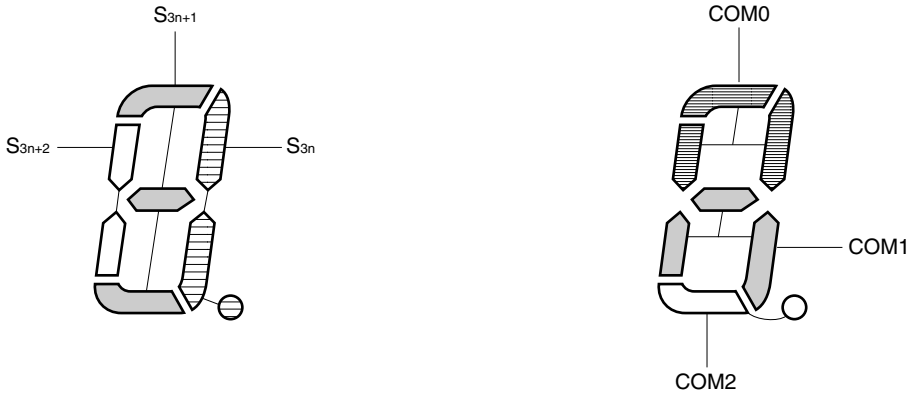
Table 13-5. Select and Deselect Voltages (COM0 to COM2)

Segment / Common	S9	S10	S11
COM0	Deselect	Select	Select
COM1	Select	Select	Select
COM2	Select	Select	–

According to Table 13-5, it is determined that the display data memory location (FA09H) that corresponds to S9 must contain x110.

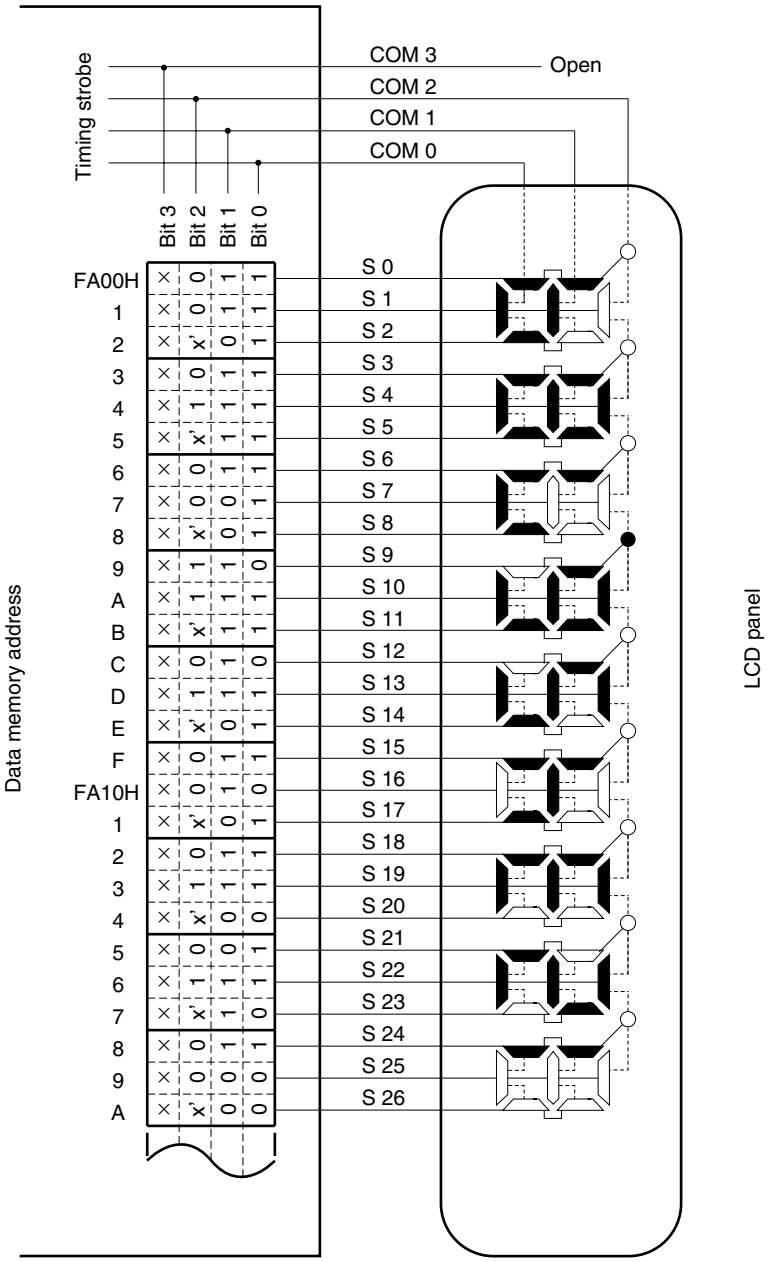
Figure 13-10 shows an example of LCD drive waveforms between the S9 signal and each common signal. When the select voltage is applied to S9 at the timing of COM1 or COM2, an alternate rectangle waveform, $+V_{LCD}/-V_{LCD}$, is generated to turn on the corresponding LCD segment.

Figure 13-8. Three-Time-Slice LCD Display Pattern and Electrode Connections



Remark n = 0 to 8

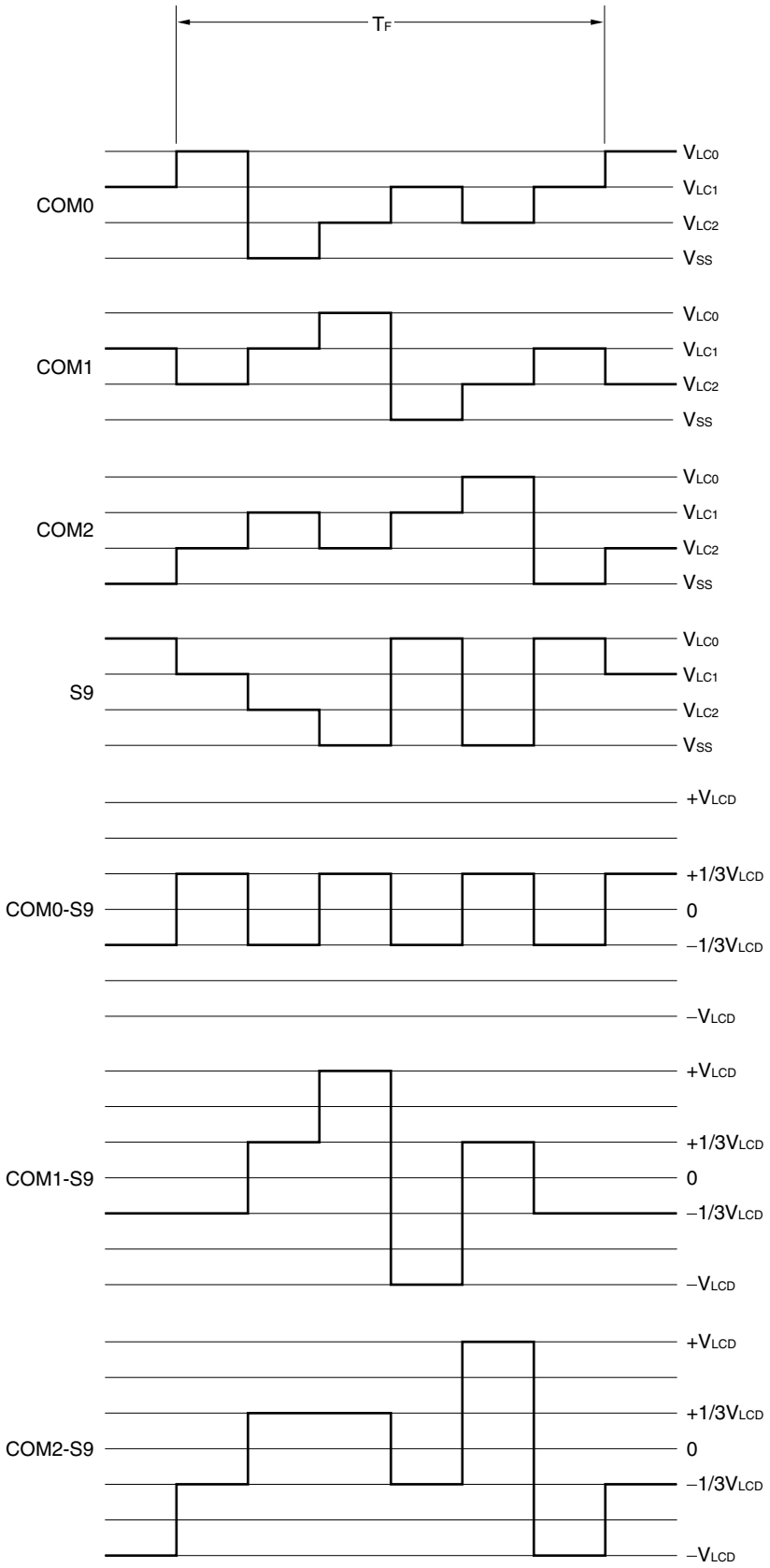
Figure 13-9. Example of Connecting Three-Time-Slice LCD Panel



x': Can be used to store any data because there is no corresponding segment in the LCD panel.

×: Can always be used to store any data because the three-time-slice mode is being used.

Figure 13-10. Three-Time-Slice LCD Drive Waveform Examples (1/3 Bias Method)



13.7.2 Four-time-slice display example

Figure 13-12 shows how a 14-digit LCD panel having the display pattern shown in Figure 13-11 is connected to the segment signals (S0 to S27) and the common signals (COM0 to COM3) of the μPD789479 Subseries chip. This example displays the data "123456.78901234" in the LCD panel. The contents of the display data memory (addresses FA00H to FA1BH) correspond to this display.

The following description focuses on numeral "6." (6.) displayed as the ninth digit from the right. To display "6." in the LCD panel, it is necessary to apply the select or deselect voltage to the S16 and S17 pins according to Table 13-6 at the timing of the common signals COM0 to COM3; see Figure 13-11 for the relationship between the segment signals and LCD segments.

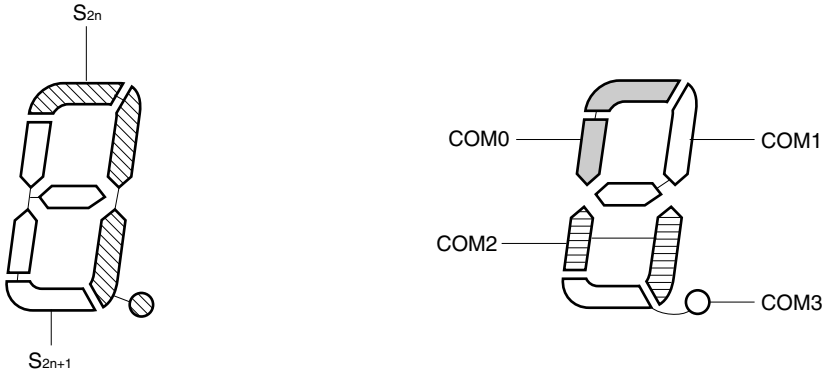
Table 13-6. Select and Deselect Voltages (COM0 to COM3)

Segment / Common	S16	S17
COM0	Select	Select
COM1	Deselect	Select
COM2	Select	Select
COM3	Select	Select

According to Table 13-6, it is determined that the display data memory location (FA10H) that corresponds to S16 must contain 1101.

Figure 13-13 shows examples of LCD drive waveforms between the S16 signal and each common signal. When the select voltage is applied to S16 at the timing of COM0, an alternate rectangle waveform, $+V_{LCD}/-V_{LCD}$, is generated to turn on the corresponding LCD segment.

Figure 13-11. Four-Time-Slice LCD Display Pattern and Electrode Connections



Remark n = 0 to 13

Figure 13-12. Example of Connecting Four-Time-Slice LCD Panel

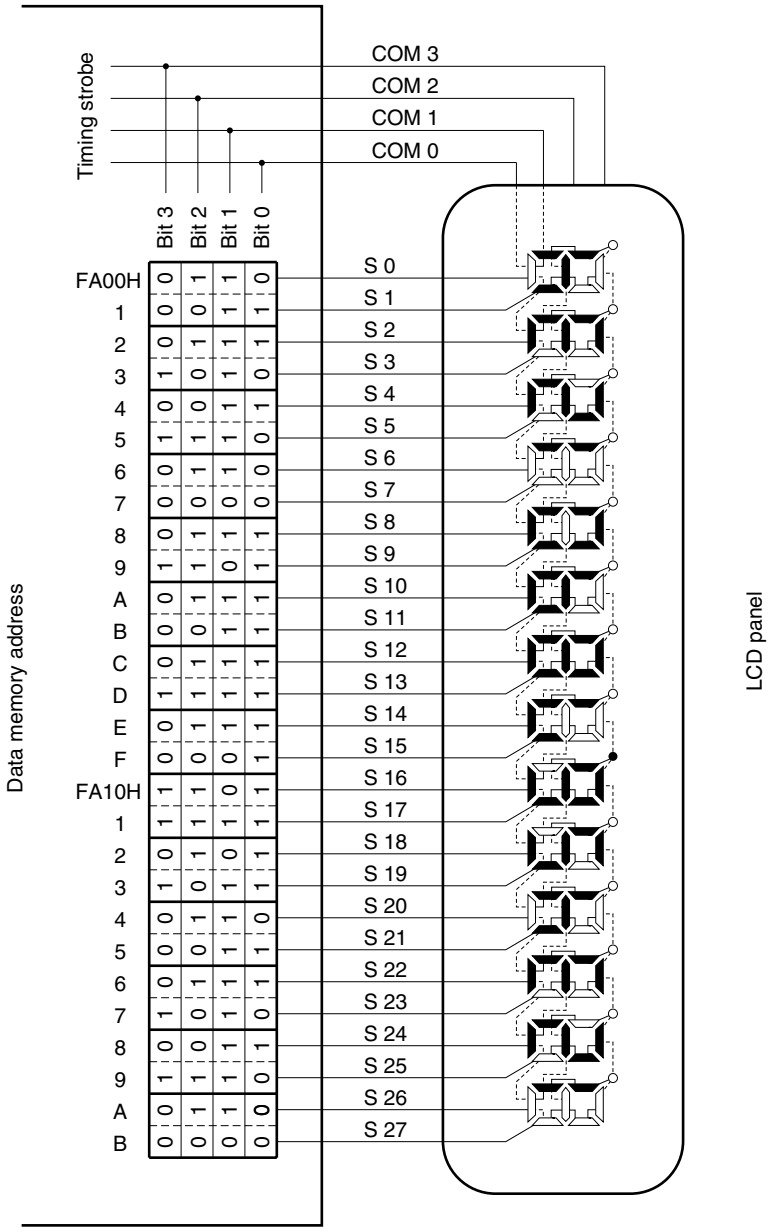
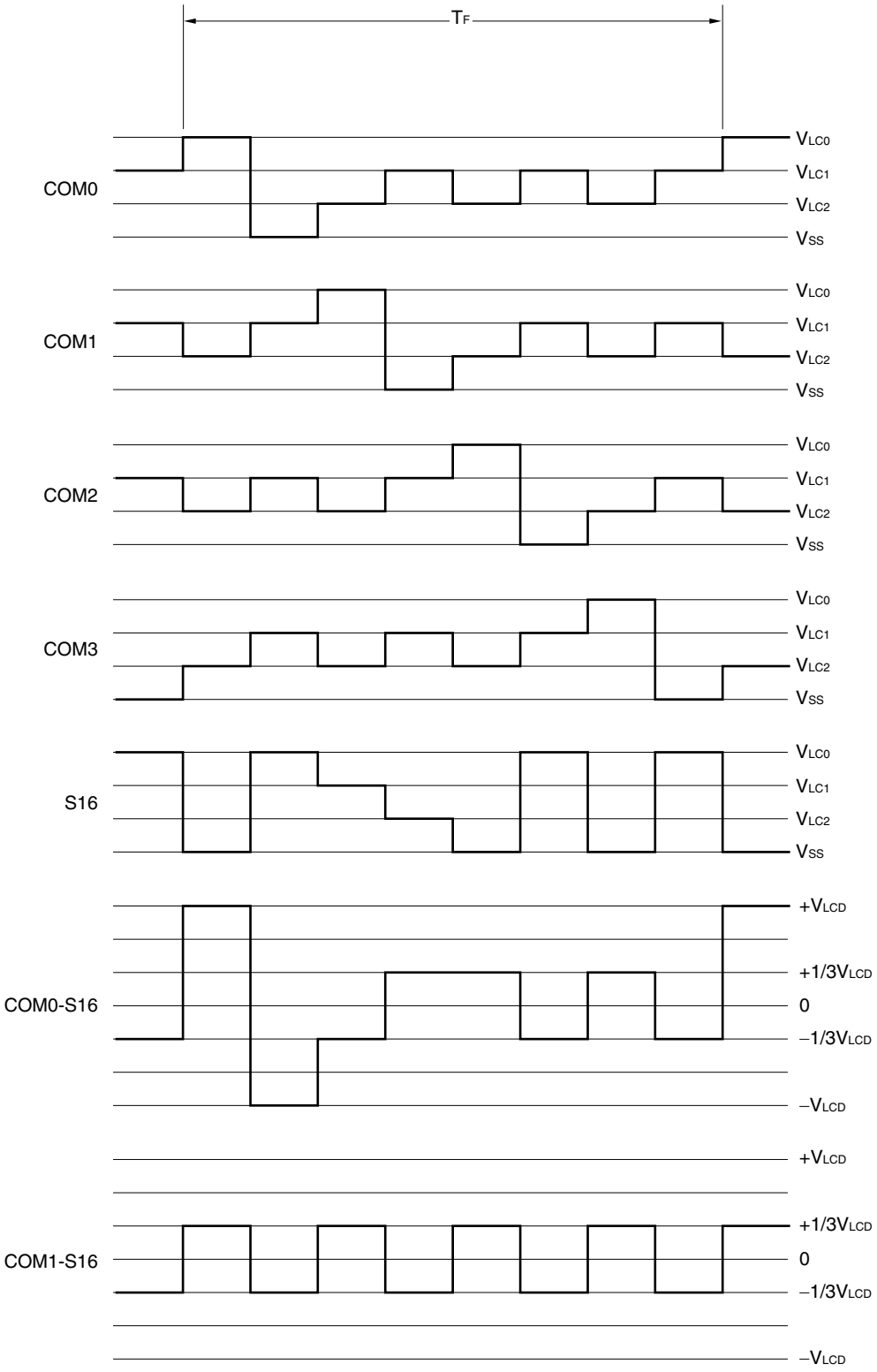


Figure 13-13. Four-Time-Slice LCD Drive Waveform Examples (1/3 Bias Method)



Remark The waveforms of COM2-S16 and COM3-S16 are not shown.

13.8 Examples of LCD Drive Power Connections

Since the μPD789479 Subseries employs a divider resistor system for generating LCD drive power, it requires external voltage divider resistors. Figure 13-14 shows an example of LCD drive power connections.

The LCD drive voltage is supplied to V_{LC0}, and 2/3 and 1/3 of V_{LC0} are supplied to the V_{LC1} and V_{LC0} pins, respectively.

Figure 13-14. Example of LCD Drive Power Connections

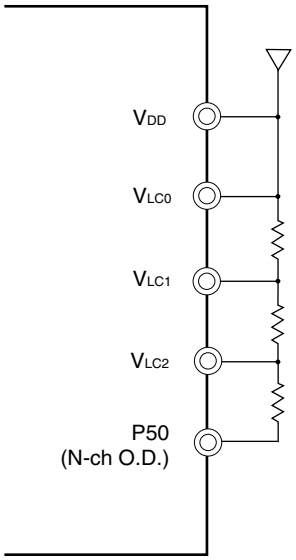


Figure 13-14 shows an example in which the current flow through voltage divider resistors is cut off by using P50 when the LCD is not used. The following explains this case.

<When turning on display>

<1> Set P50 to the output mode (PM50 = 0)

<2> Set the P50 output latch to 0, and output a low level from P50.

→ The LCD drive voltage is supplied to V_{LC0}, V_{LC1}, and V_{LC2}.

<3> Set LCDON0 (bit 7 of LCDM0) to 1, and turn on the display.

<When turning off display>

<1> Clear LCDON0 (bit 7 of LCDM0) to 1, and turn off the display.

<2> Set the P50 output latch to 1, and output a high level from P50 (Hi-Z).

→ When the display is off, the current flow between V_{DD} and P50 is cut off by setting the P50 (N-ch open drain) output level to high impedance.

CHAPTER 14 MULTIPLIER

14.1 Multiplier Function

The multiplier has the following function.

- Calculation of 8 bits × 8 bits = 16 bits

14.2 Multiplier Configuration

(1) 16-bit multiplication result storage register 0 (MUL0)

This register stores the 16-bit result of multiplication.

This register holds the result of multiplication after 16 CPU clocks have elapsed.

MUL0 is set with a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes this register undefined.

Caution Although this register is manipulated with a 16-bit memory manipulation instruction, it can also be manipulated with an 8-bit memory manipulation instruction. When using an 8-bit memory manipulation instruction, however, access the register by means of direct addressing.

(2) Multiplication data registers A and B (MRA0 and MRB0)

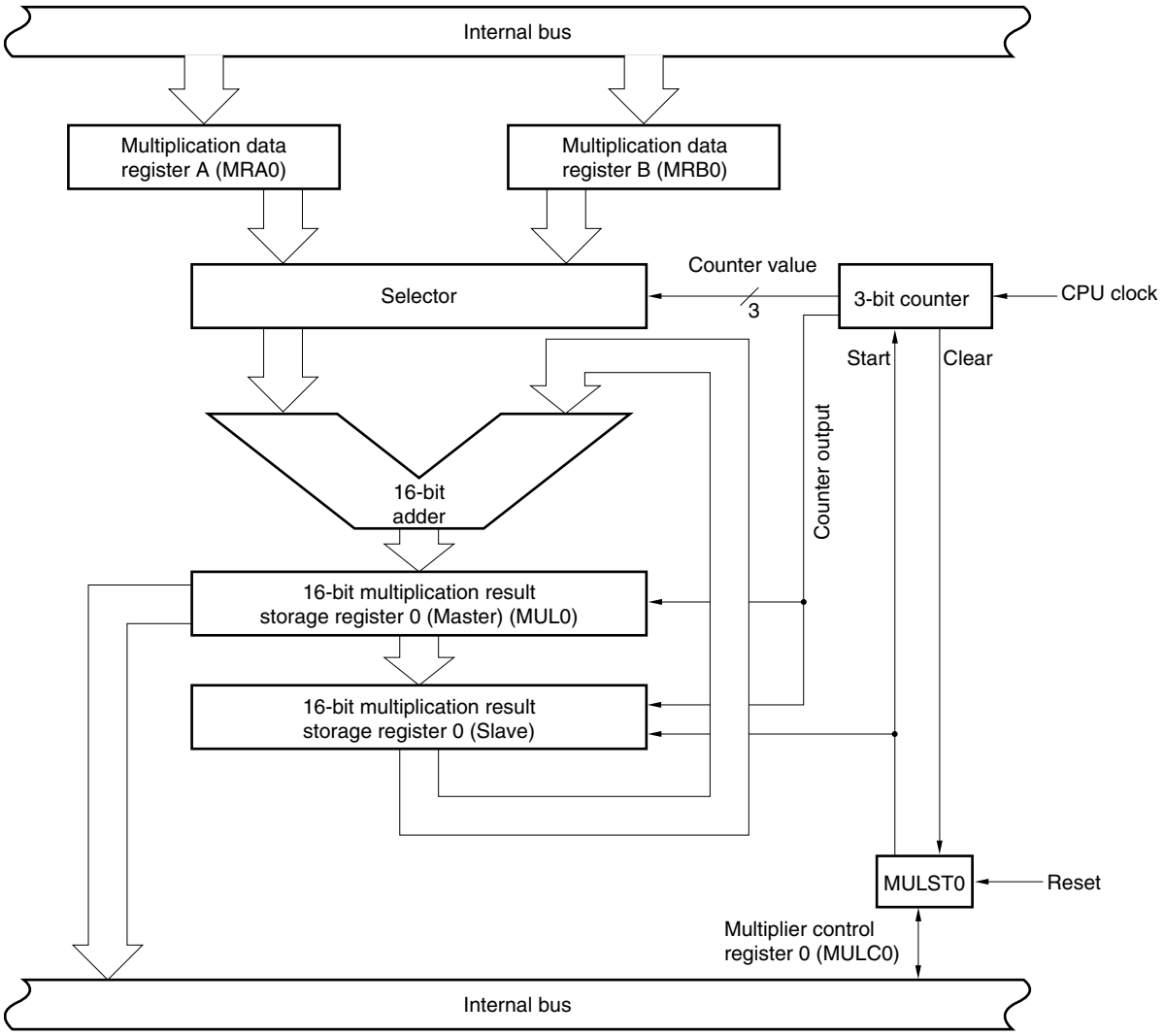
These are 8-bit multiplication data storage registers. The multiplier multiplies the values of MRA0 and MRB0.

MRA0 and MRB0 are set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes these registers undefined.

Figure 14-1 shows the block diagram of the multiplier.

Figure 14-1. Block Diagram of Multiplier



14.3 Multiplier Control Register

The multiplier is controlled by the following register.

- Multiplier control register 0 (MULC0)

(1) Multiplier control register 0 (MULC0)

MULC0 indicates the operating status of the multiplier after operation, as well as controls the multiplier.

MULC0 is set with a 1-bit or 8-bit memory manipulation instruction.

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

Figure 14-2. Format of Multiplier Control Register 0

Symbol	7	6	5	4	3	2	1	<0>	Address	After reset	R/W
MULC0	0	0	0	0	0	0	0	MULST0	FFD2H	00H	R/W

MULST0	Multiplier operation start control bit	Operating status of multiplier
0	Stop operation after resetting counter to 0.	Operation stopped
1	Enable operation	Operation in progress

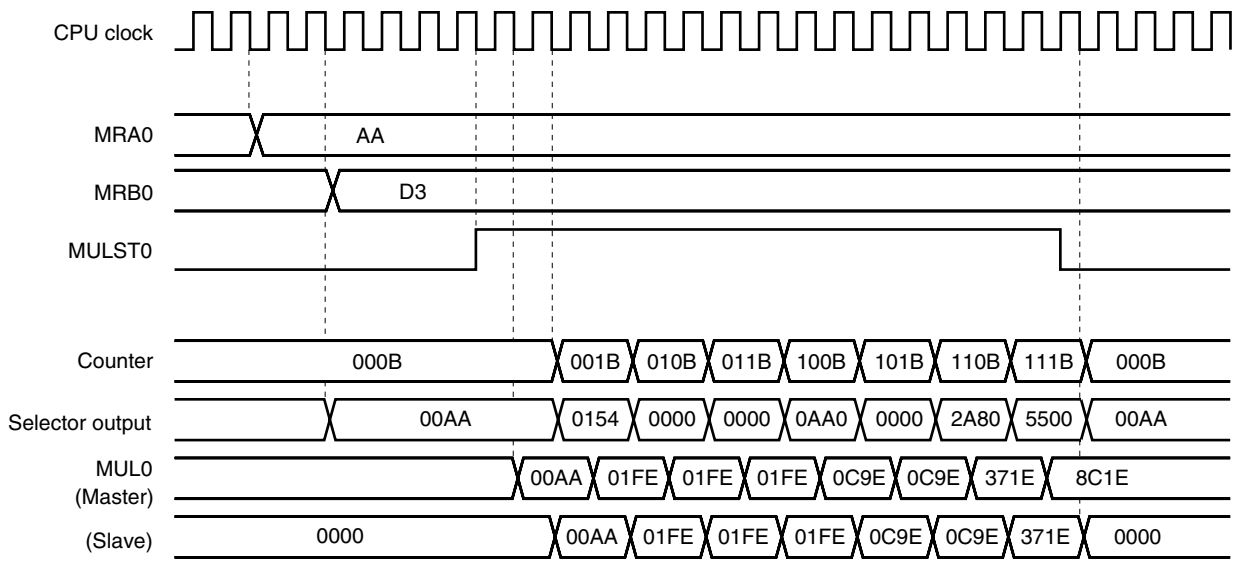
Caution Be sure to set bits 1 to 7 to 0.

14.4 Multiplier Operation

The multiplier of the μPD789479 Subseries can execute the calculation of 8 bits × 8 bits = 16 bits. Figure 14-3 shows the operation timing of the multiplier where MRA0 is set to AAH and MRB0 is set to D3H.

<1> Counting is started by setting MULST0.
<2> The data generated by the selector is added to the data of MUL0 at each CPU clock, and the counter value is incremented by one.
<3> If MULST0 is cleared when the counter value is 111B, the operation is stopped. At this time, MUL0 holds the data.
<4> While MULST0 is low, the counter and slave are cleared.

Figure 14-3. Multiplier Operation Timing (Example of AAH × D3H)



CHAPTER 15 REMOTE CONTROLLER RECEIVER

15.1 Remote Controller Receiver Functions

The remote controller receiver uses the following remote controller modes.

- Type A reception mode … Guide pulse (half clock) provided

15.2 Remote Controller Receiver Configuration

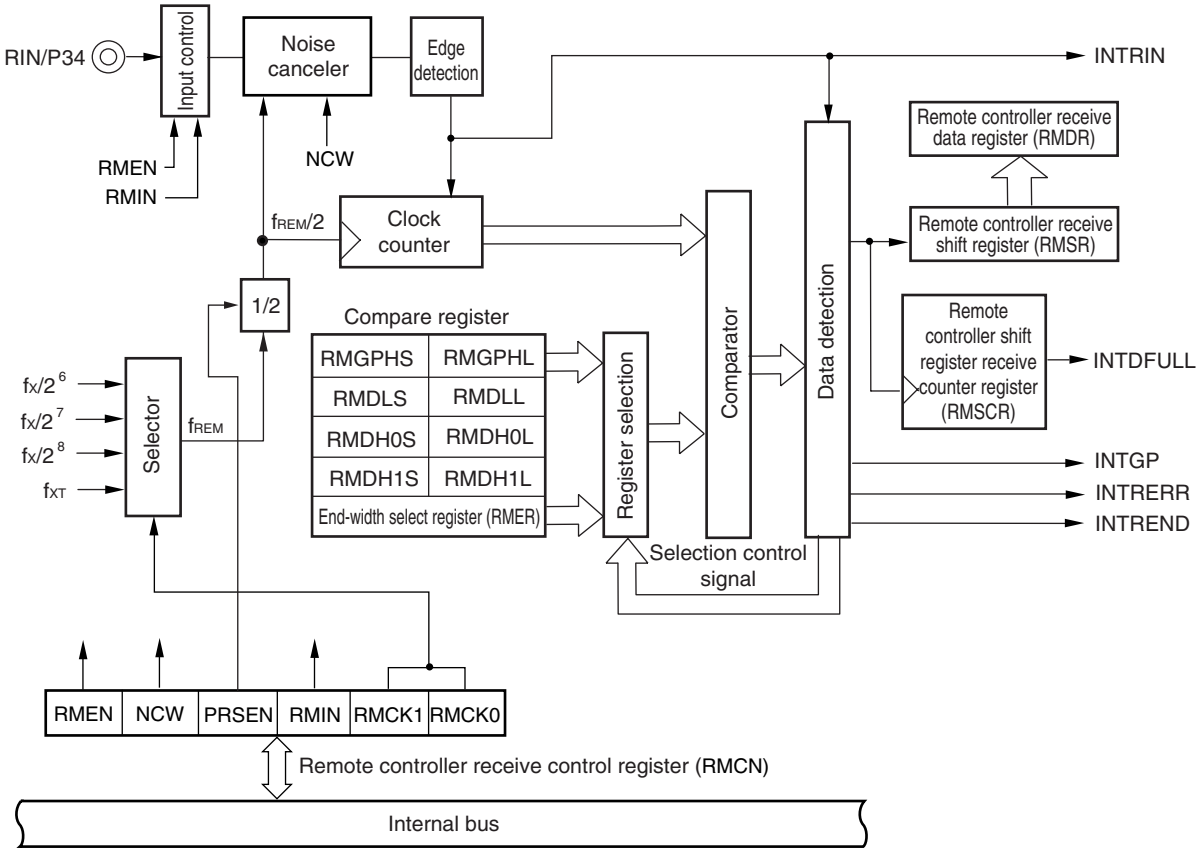
The remote controller receiver includes the following hardware.

Table 15-1. Remote Controller Receiver Configuration

Item	Configuration
Registers	Remote controller receive shift register (RMSR) Remote controller receive data register (RMDR) Remote controller shift register receive counter register (RMSCR) Remote controller receive GPHS compare register (RMGPHS) Remote controller receive GPHL compare register (RMGPHL) Remote controller receive DLS compare register (RMDLS) Remote controller receive DLL compare register (RMDLL) Remote controller receive DH0S compare register (RMDH0S) Remote controller receive DH0L compare register (RMDH0L) Remote controller receive DH1S compare register (RMDH1S) Remote controller receive DH1L compare register (RMDH1L) Remote controller receive end width select register (RMER)
Control register	Remote controller receive control register (RMCN)

<R>

Figure 15-1. Block Diagram of Remote Controller Receiver



(1) Remote controller receive shift register (RMSR)

This is an 8-bit register for reception of remote controller data.

Data is stored in bit 7 first. Each time new data is stored, the stored data is shifted to the lower bits. Therefore, the latest data is stored in bit 7, and the first data is stored in bit 0.

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

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

Also, RMSR is cleared to 00H under any of the following conditions.

- Remote controller stops operation (RMEN = 0).
- Error is detected (INTRERR is generated).
- INTDFULL is generated.
- RMSR is read after INTREND has been generated.

Caution Reading RMSR is disabled during remote controller reception. Complete reception, then read RMSR. When the reading operation is complete, RMSR is cleared. Therefore, values once read are not guaranteed.

(2) Remote controller receive data register (RMDR)

This register holds the remote controller reception data. When the remote controller receive shift register (RMSR) overflows, the data in RMSR is transferred to RMDR. Bit 7 stores the last data, and bit 0 stores the first data. INTDFULL is generated at the same time as data is transferred from RMSR to RMDR.

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

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

When the remote controller operation is disabled (RMEN = 0), RMSR is cleared to 00H.

Caution When INTDFULL has been generated, read RMDR before the next 8-bit data is received. If the next INTDFULL is generated before the read operation is complete, RMDR is overwritten.

(3) Remote controller shift register receive counter register (RMSCR)

This is an 8-bit counter register used to indicate the number of valid bits remaining in the remote controller receive shift register (RMSR) when remote controller reception is complete (INTREND is generated). Reading the values of this register allows confirmation of the number of bits, even if the received data is in a format other than an integral multiple of 8 bits.

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

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

It is cleared to 00H under any of the following conditions.

- Remote controller stops operation (RMEN = 0).
- Error is detected (INTRERR is generated).
- RMSR is read after INTREND has been generated.

Caution When INTREND has been generated, immediately read RMSCR before reading RMSR. If reading occurs at another timing, the value is not guaranteed.

Figure 15-2. Operation Examples of RMSR, RMSCR, and RMDR Registers When Receiving 1010101011111111B (16 Bits)

	RMSR								RMSCR	RMDR
	7	6	5	4	3	2	1	0		
After reset	0	0	0	0	0	0	0	0	00H	00000000B
Receiving 1 bit	1	0	0	0	0	0	0	0	01H	00000000B
Receiving 2 bits	0	1	0	0	0	0	0	0	02H	00000000B
Receiving 3 bits	1	0	1	0	0	0	0	0	03H	00000000B
…	…	…	…	…	…	…	…	…	…	…
Receiving 7 bits	1	0	1	0	1	0	1	0	07H	00000000B
Receiving 8 bits	0	1	0	1	0	1	0	1	00H	00000000B
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
RMDR transfer	0	0	0	0	0	0	0	0	00H	01010101B
Receiving 9 bits	1	0	0	0	0	0	0	0	01H	01010101B
Receiving 10 bits	1	1	0	0	0	0	0	0	02H	01010101B
…	…	…	…	…	…	…	…	…	…	…
Receiving 16 bits	1	1	1	1	1	1	1	1	00H	01010101B
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
RMDR transfer	0	0	0	0	0	0	0	0	00H	11111111B

(4) Remote controller receive GPHS compare register (RMGPHS)

This register is used to detect the high level of a remote controller guide pulse (short side).

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

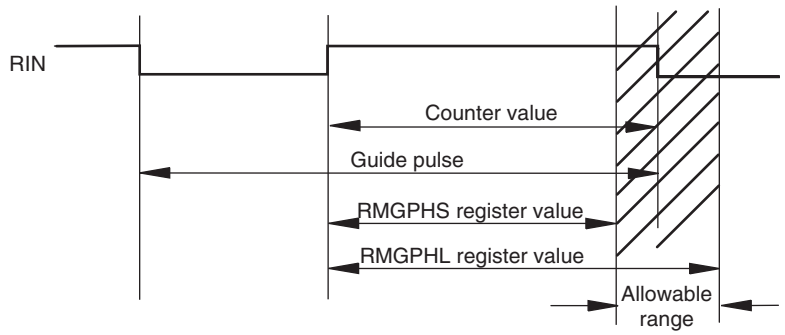
$\overline{\text{RESET}}$ input sets RMGPHS to 00H.

(5) Remote controller receive GPHL compare register (RMGPHL)

This register is used to detect the high level of a remote controller guide pulse (long side).

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

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



If RMGPHS ≤ counter value < RMGPHL is satisfied, it is assumed that the high level of the guide pulse has been successfully received.

(6) Remote controller DLS compare register (RMDLS)

This register is used to detect the low level of a remote controller data (short side).

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

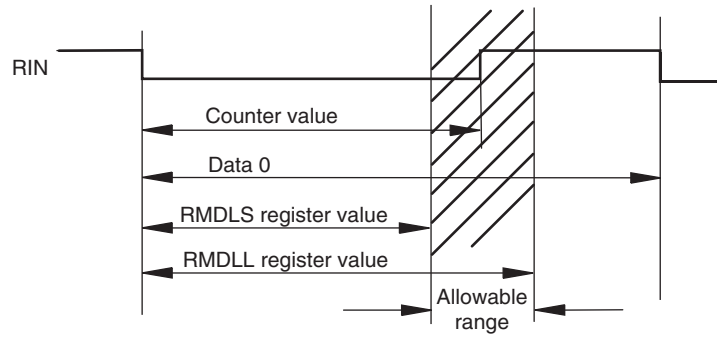
$\overline{\text{RESET}}$ input sets RMDLS to 00H.

(7) Remote controller receive DLL compare register (RMDLL)

This register is used to detect the low level of a remote controller data (long side).

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

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



If RMDLS ≤ counter value < RMDLL is satisfied, it is assumed that the low level of data 0 or data 1 has been successfully received.

(8) Remote controller receive DH0S compare register (RMDH0S)

This register is used to detect the high level of remote controller data 0 (short side).

RMDH0S is set with an 8-bit memory manipulation instruction.

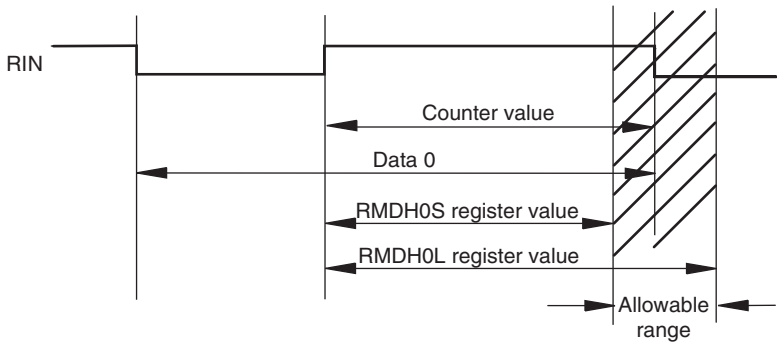
$\overline{\text{RESET}}$ input sets RMDH0S to 00H.

(9) Remote controller receive DH0L compare register (RMDH0L)

This register is used to detect the high level of remote controller data 0 (long side).

RMDH0L is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets RMDH0L to 00H.



If RMDH0S ≤ counter value < RMDH0L is satisfied, it is assumed that the high level of data 0 has been successfully received, and therefore RMSR receives the data.

(10) Remote controller receive DH1S compare register (RMDH1S)

This register is used to detect the high level of remote controller data 1 (short side).

RMDH1S is set with an 8-bit memory manipulation instruction.

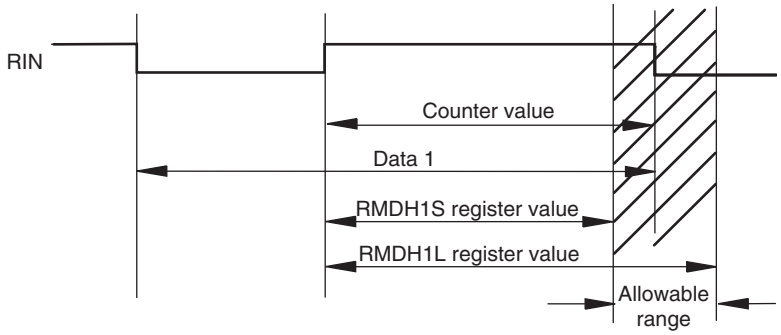
$\overline{\text{RESET}}$ input sets RMDH1S to 00H.

(11) Remote controller receive DH1L compare register (RMDH1L)

This register is used to detect the high level of remote controller data 1 (long side).

RMDH1L is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets RMDH1L to 00H.



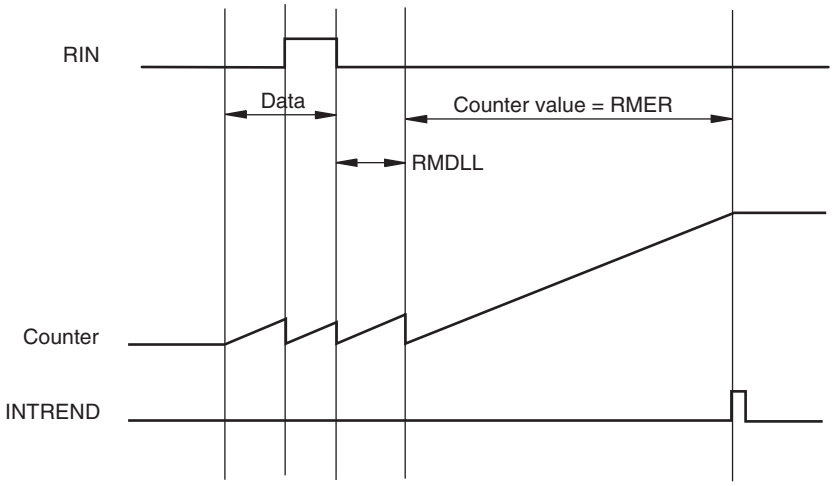
If RMDH1S ≤ counter value < RMDH1L is satisfied, it is assumed that the high level of data 1 has been successfully received, and therefore RMSR receives the data.

(12) Remote controller receive end-width select register (RMER)

This register determines the interval between the timing at which the INTREND signal is output.

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

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



Caution For RMER and all the remote controller receive compare registers (RMGPHS, RMGPHL, RMDLS, RMDLL, RMDH0S, RMDH0L, RMDH1S, and RMDH1L), disable remote controller reception (bit 7 (RMEN) of the remote controller receive control register (RMCN) = 0) first, and then change the value.

15.3 Registers to Control Remote Controller Receiver

The remote controller receiver is controlled by the following register.

- Remote controller receive control register (RMCN)

(1) Remote controller receive control register (RMCN)

This register is used to enable/disable remote controller reception and to set the noise elimination width, clock internal division, input invert signal, and source clock.

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

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

Figure 15-3. Format of Remote Controller Receive Control Register (1/2)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
RMCN	RMEN	NCW	PRSEN	RMIN	0	0	RMCK1	RMCK0	FF60H	00H	R/W

RMEN	Control of remote controller receive operation
0	Disable remote controller reception
1	Enable remote controller reception

NCW	Noise elimination width control signal
0	Eliminate noise less than $1/f_{PRS}$
1	Eliminate noise less than $2/f_{PRS}$

PRSEN	Internal clock division control signal
0	Clock not divided internally ($f_{PRS} = f_{REM}$)
1	Clock internally divided into two ($f_{PRS} = f_{REM}/2$)

RMIN	Remote controller input invert signal
0	Input positive phase
1	Input negative phase

Cautions 1. Always set bits 2 and 3 to 0.

2. To change the values of NCW, PRSEN, RMIN, RMCK1, and RMCK0, disable remote controller reception (RMEN = 0) first.

Remarks 1. f_{REM}: Source clock of remote controller counter (selected by bits 0 and 1 (RMCK0 and RMCK1)

2. f_{PRS}: Operation clock inside remote controller receiver

<R> **Figure 15-3. Format of Remote Controller Receive Control Register (2/2)**

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
RMCN	RMEN	NCW	PRSEN	RMIN	0	0	RMCK1	RMCK0	FF60H	00H	R/W

RMCK1	RMCK0	Selection of source clock (f_{REM}) of remote controller counter
0	0	$f_X/2^6$ (62.5 kHz)
0	1	$f_X/2^7$ (31.3 kHz)
1	0	$f_X/2^8$ (15.6 kHz)
1	1	f_{XT} (32.768 kHz)

Cautions 1. Always set bits 2 and 3 to 0.

2. To change the values of NCW, PRSEN, RMIN, RMCK1, and RMCK0, disable remote controller reception (RMEN = 0) first.

Remarks 1. f_X: Oscillation frequency of main system clock

2. f_{XT}: Oscillation frequency of subsystem clock

3. The parenthesized values apply to operation at f_X = 4.0 MHz and f_{XT} = 32.768 kHz.

15.4 Operation of Remote Controller Receiver

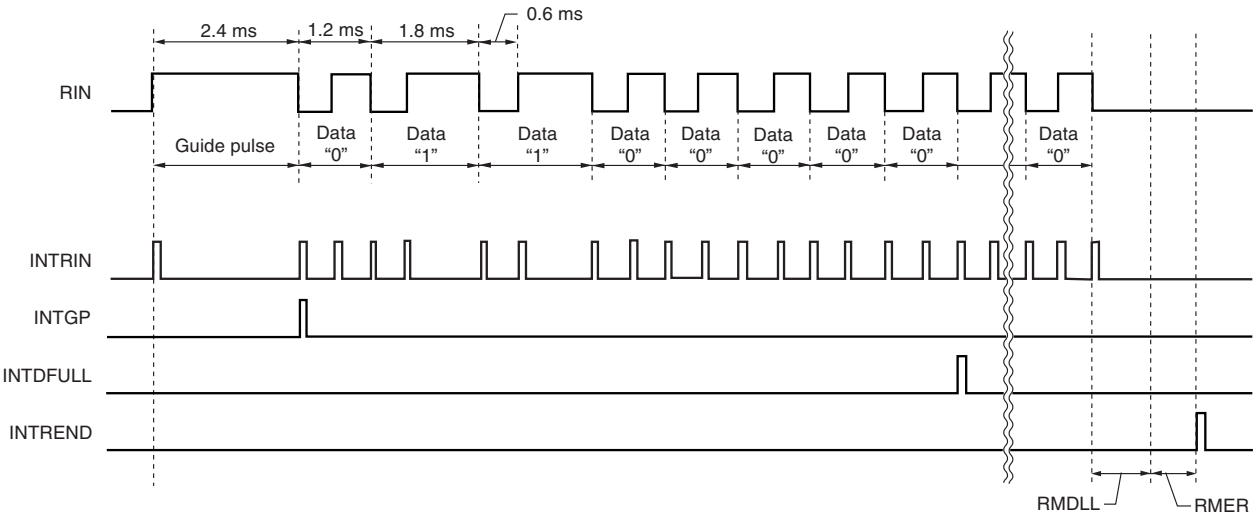
The following remote controller reception mode is used for this remote controller receiver.

- Type A reception mode with guide pulse (half clock)

15.4.1 Format of type A reception mode

Figure 15-4 shows the data format for type A.

Figure 15-4. Example of Type A Data Format



15.4.2 Operation flow of type A reception mode

Figure 15-5 shows the operation flow.

Cautions 1. When INTRERR is generated, RMSR and RMSCR are automatically cleared immediately.

2. When data has been set to all the bits of RMSR, the following processing is automatically performed.

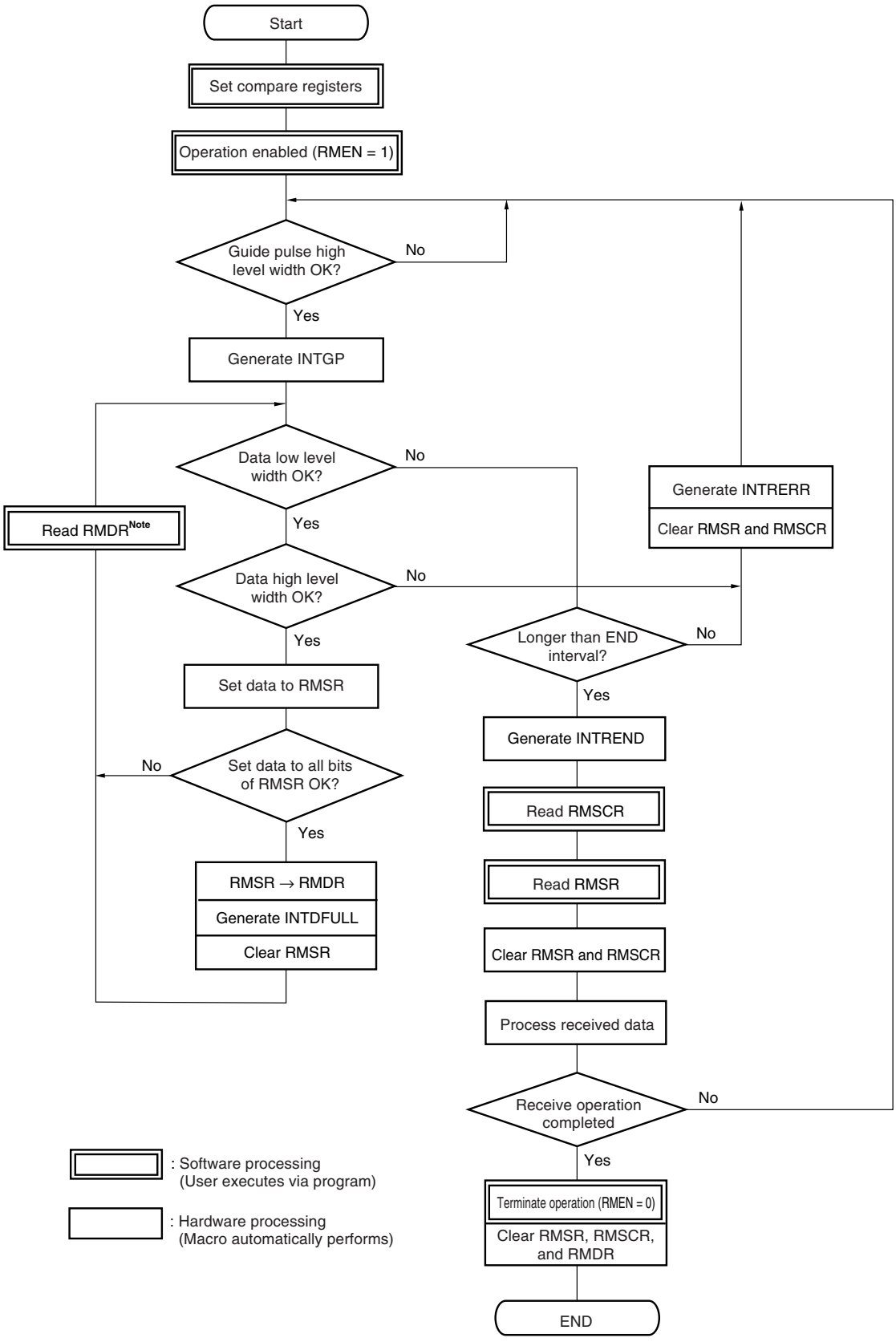
- **The value of RMSR is transferred to RMDR.**
- **INTDFULL is generated.**
- **RMSR is cleared.**

RMDR must then be read before the next data is set to all the bits of RMSR.

3. When INTREND has been generated, read RMSCR first followed by RMSR.
When RMSR has been read, RMSCR and RMSR are automatically cleared.
If INTREND is generated, the next data cannot be received until RMSR is read.

4. RMSR, RMSCR, and RMDR are cleared simultaneously to operation termination (RMEN = 0).

Figure 15-5. Operation Flow of Type A Reception Mode

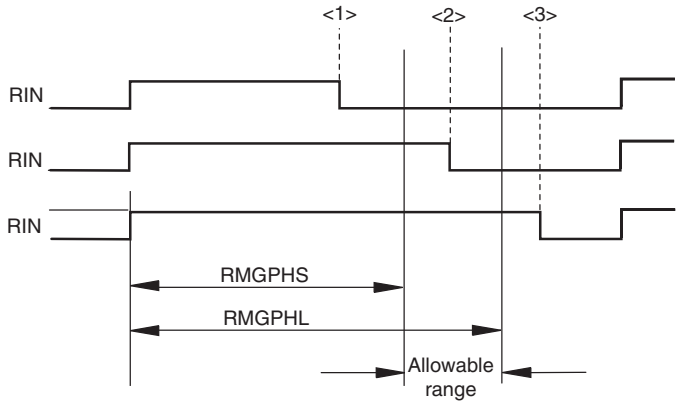


Note Read RMDR before data has been set to all the bits of RMSR.

15.4.3 Timing

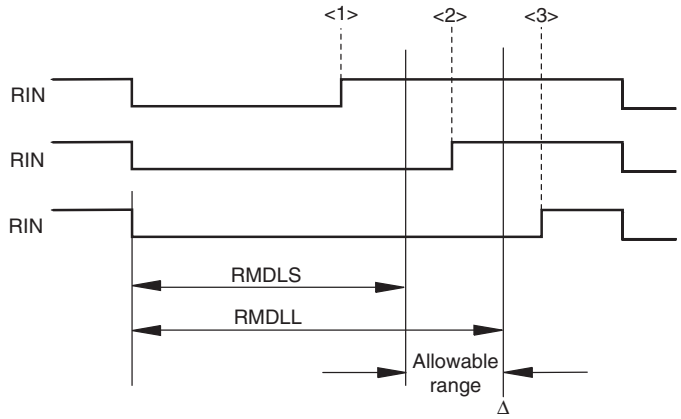
Operation varies depending on the positions of the PIN input waveform below.

(1) Guide pulse high level width determination



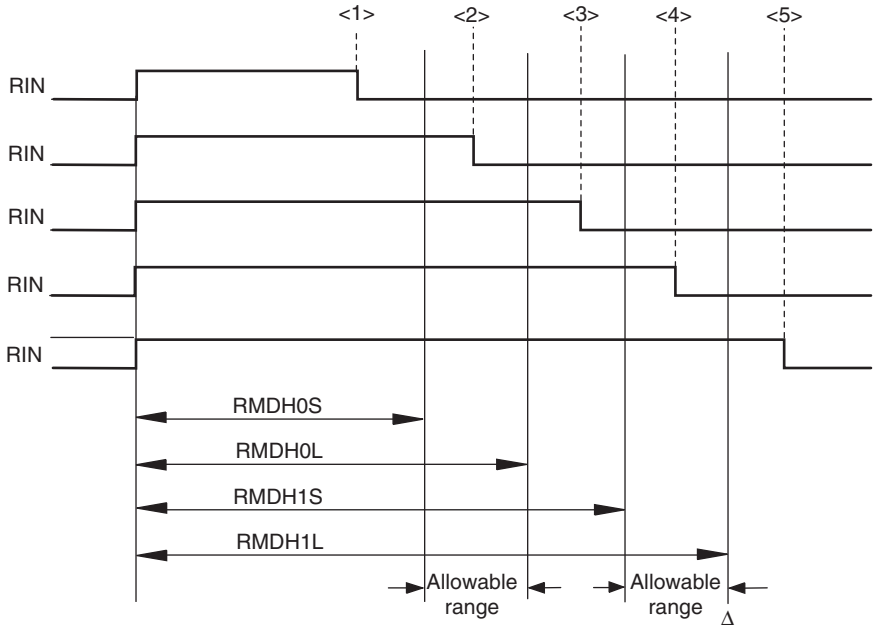
Relationship Between RMGPHS/RMGPHL/Counter	Position of Waveform	Corresponding Operation
Counter < PMGPHS	<1>: Short	Measuring guide pulse high-level width is started from the next rising edge.
PMGPHS ≤ counter < PMGPHL	<2>: Within the range	INTGP is generated. Data measurement is started.
PMGPHL ≤ counter	<3>: Long	Measuring guide pulse high-level width is started from the next rising edge.

(2) Data low level width determination



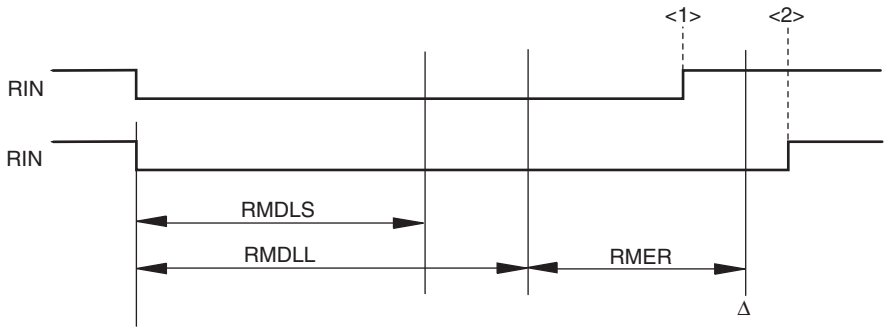
Relationship Between RMDLS/RMDLL/Counter	Position of Waveform	Corresponding Operation
Counter < RMDLS	<1>: Short	Error interrupt INTRERR is generated. Measuring guide pulse high-level width is started.
RMDLS ≤ counter < RMDLL	<2>: Within the range	Measuring data high-level width is started.
RMDLL ≤ counter	<3>: Long	Measuring the end width is started from the Δ point.

(3) Data high level width determination



Relationship Between RMDH0S/RMDH0L/RMDH1S/RMDH1L/Counter	Position of Waveform	Corresponding Operation
Counter < RMDH0S	<1>: Short	Error interrupt INTRERR is generated. Measuring the guide pulse high-level width is started at the next rising edge.
RMDH0S ≤ counter < RMDH0L	<2>: Within the range	Data 0 is received. Measuring data low-level width is started.
RMDH0L ≤ counter < RMDH1S	<3>: Outside of the range	Error interrupt INTRERR is generated. Measuring the guide pulse high-level width is started at the next rising edge.
RMDH1S ≤ counter < RMDH1L	<4>: Within the range	Data 1 is received. Measuring the data low-level width is started.
RMDH1L ≤ counter	<5>: Long	Error interrupt INTRERR is generated at the Δ point. Measuring the guide pulse high-level width is started at the next rising edge.

(4) End width determination



Relationship Between RMER/Counter	Position of Waveform	Corresponding Operation
Counter < RMER	<1>: Short	Error interrupt INTRERR is generated. Measuring the guide pulse high-level width is started.
RMER ≤ counter	<2>: Long	INTREND is generated at the Δ point. Reception via circuit stops until RMSR is read.

15.4.4 Compare register setting

This remote controller receiver has the following 9 types of compare registers.

- Remote controller receive GPHS compare register (RMGPHS)
- Remote controller receive GPHL compare register (RMGPHL)
- Remote controller receive DLS compare register (RMDLS)
- Remote controller receive DLL compare register (RMDLL)
- Remote controller receive DH0S compare register (RMDH0S)
- Remote controller receive DH0L compare register (RMDH0L)
- Remote controller receive DH1S compare register (RMDH1S)
- Remote controller receive DH1L compare register (RMDH1L)
- Remote controller receive end width select register (RMER)

Use formulas (1) to (3) below to set the value of each compare register.

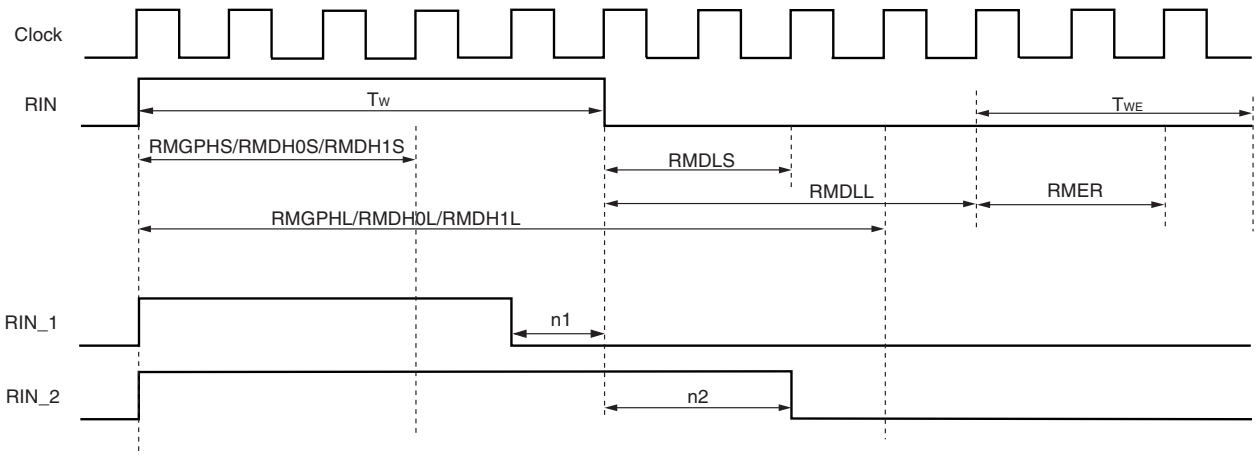
Making allowances for tolerance enables a normal reception operation, even if the RIN input waveform is RIN_1 or RIN_2 shown in Figure 15-6 due to the effect of noise.

Cautions 1. Always set each compare register while remote controller reception is disabled (RMEN = 0).

2. Set the set values so that they satisfy all the following three conditions.

- **RMGPHS < RMGPHL**
- **RMDLS < RMDLL**
- **RMDH0S < RMDH0L ≤ RMDH1S < RMDH1L**

Figure 15-6. Setting Example (Where n1 = 1, n2 = 2)



(1) Formula for RMGPHS, RMDLS, RMDH0S, and RMDH1S

$$\left(\frac{T_W \times (1 - a/100)}{1/f_{PRS}} \right)_{INT} - 2 - n1$$

(2) Formula for RMGPHL, RMDLL, RMDH0L, and RMDH1L

$$\left(\frac{T_W \times (1 + a/100)}{1/f_{PRS}} \right)_{INT} + 1 + n2$$

(3) Formula for RMER

$$\left(\frac{T_{WE} \times (1 - a/100)}{1/f_{PRS}} \right)_{INT} - 1$$

T_W: Width of RIN input waveform

$1/f_{PRS}$: Width of internal operation clock cycle after division control by PRSEN

a: Tolerance (%)

$[\ ]_{INT}$: Round down the fractional portion of the value produced by the formula in the brackets.

n1, n2: Variables of waveform change caused by noise[Note1]

T_{WE}: End width of RIN input[Note2]

Notes 1. Set the values of n1 and n2 as required to meet the user's system specification.

2. This end width is counted after RMDLL.
The low-level width actually required after the last data has been received is as follows:
(RMDLL + 1 + RMER + 1) × (width of internal operation clock cycle after division control by PRSEN)

15.4.5 Error interrupt generation timing

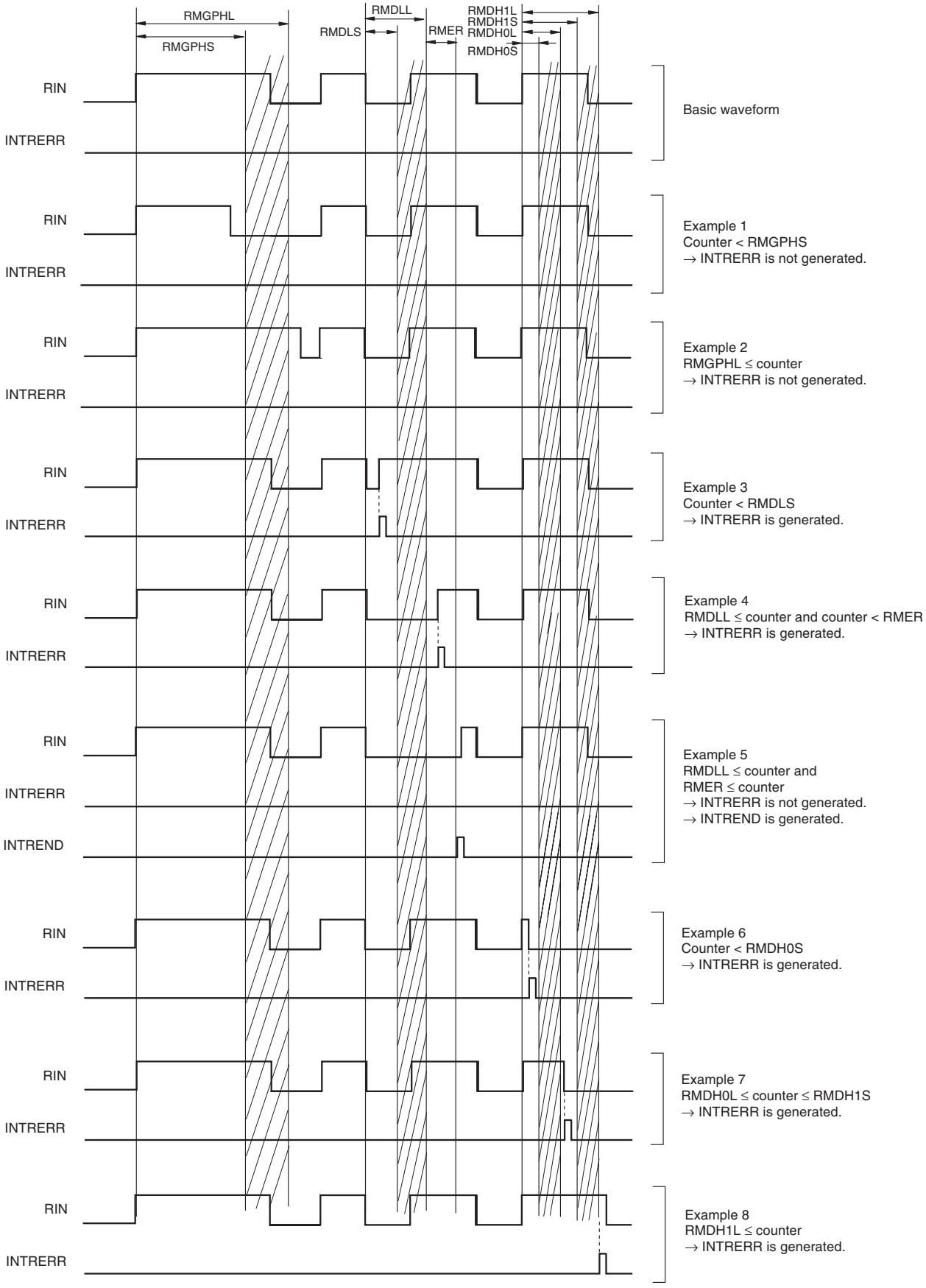
After the guide pulse has been detected normally, the INTRERR signal is generated under any of the following conditions.

- Counter < RMDLS at the rising edge of RIN
- RMDLL ≤ counter and counter after RMDLL < RMER at the rising edge of RIN
- Counter < RMDH0S at the falling edge of RIN
- RMDH0L ≤ counter < RMDH1S at the falling edge of RIN
- Register changes so that RMDH1L ≤ counter while RIN is at high level

The INTRERR signal is not generated until the guide pulse is detected.

Once the INTRERR signal has been generated, it will not be generated again until the next guide pulse is detected. The generation timing of the INTRERR signal is shown in Figure 15-7.

Figure 15-7. Generation Timing of INTRERR Signal



15.4.6 Noise elimination

This remote controller receiver provides a function that supplies the signals input from the outside to the RIN pin after eliminating noise.

Noise width can be eliminated by setting bit 5 (PRSEN) and bit 6 (NCW) of the remote controller receive control register (RMCN) as shown in Figure 15-2.

Table 15-2. Noise Elimination Width

PRSEN Division Control Signal	NCW Noise Elimination Width Control Signal	Internal Operation Clock Cycle After Division Control by PRSEN ($1/f_{PRS}$)	Eliminatable Noise Width
0	0	$1/f_{REM}$	Less than $1/f_{REM}$
0	1	$1/f_{REM}$	Less than $2/f_{REM}$
1	0	$2/f_{REM}$	Less than $2/f_{REM}$
1	1	$2/f_{REM}$	Less than $4/f_{REM}$

Remark f_{REM}: Source clock of remote controller counter

A noise elimination operation is performed by using the internal operation clock after division control by PRSEN. Then, after the external input signal from RIN pin has been synchronized with the clock,

If NCW = 0, the signal after sampling is performed twice is processed as a RIN input in the circuit.

If NCW = 1, the signal after sampling is performed three times is processed as a RIN input in the circuit.

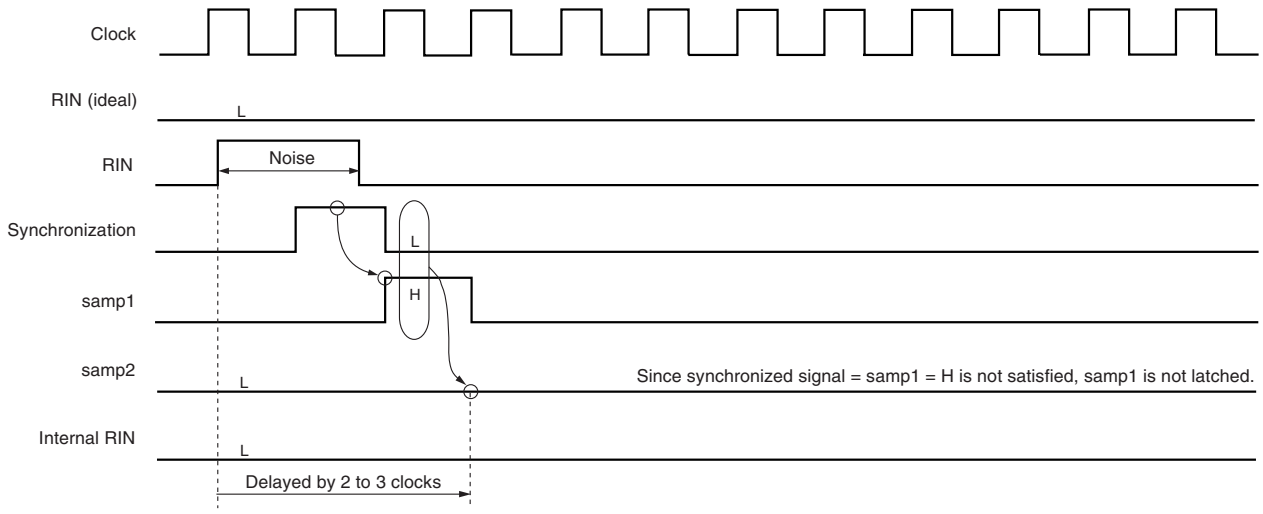
The following shows the flow of a noise elimination operation.

<1> Select whether or not the internal operation clock is divided by PRSEN.
PRSEN = 0: Not divided ($f_{PRS} = f_{REM}$)
PRSEN = 1: Divided ($f_{PRS} = f_{REM}/2$)
<2> Synchronize the external input signal from the RIN pin with the internal operation clock.
<3> Generate a signal (samp1) sampling the synchronized signal for the first time.
(The signal is later than the synchronized signal by one clock.)
<4> Generate a signal (samp2) sampling the synchronized signal and samp1 for the second time.
(When synchronized signal = samp1 = H, samp1 is latched.)
<5> Generate a signal (samp3) sampling the synchronized signal and samp2 for the third time.
(When synchronized signal = samp2 = H, samp2 is latched.)
<6> Select a signal to be the RIN input in the circuit using NCW.
NCW = 0: samp2 is processed as the RIN input in the circuit.
NCW = 1: samp3 is processed as the RIN input in the circuit.

Figure 15-8 shows an example of a noise elimination operation.

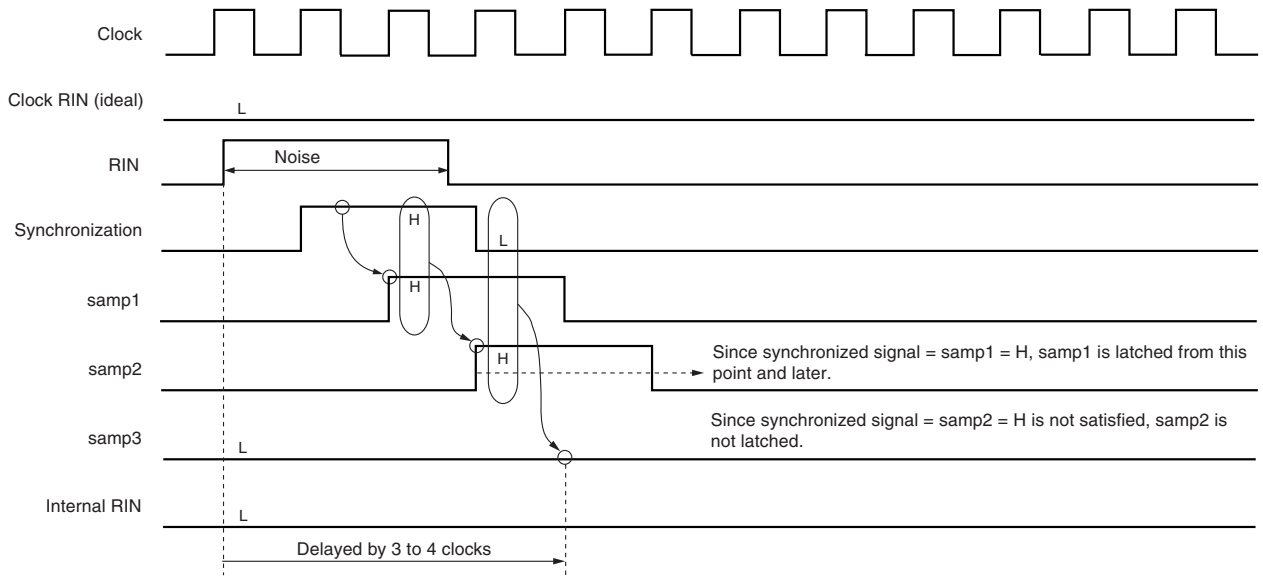
Figure 15-8. Noise Elimination Operation Example (1/2)

(a) 1-clock noise elimination (PRSEN = 0, NCW = 0)



Remark Internal RIN is a signal after synchronization and sampling are performed twice, and is therefore later than the actual signal input from the outside to the RIN pin by two to three clocks.

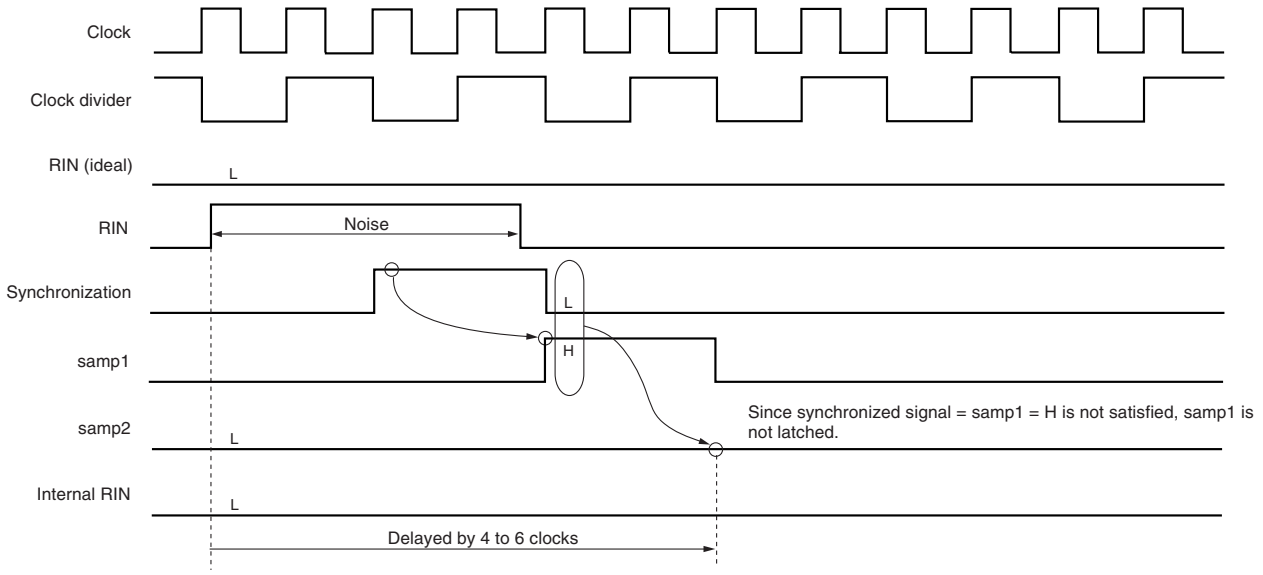
(b) 2-clock noise elimination (PRSEN = 0, NCW = 1)



Remark Internal RIN is a signal after synchronization and sampling are performed three times, and is therefore later than the actual signal input from the outside to the RIN pin by 3 to 4 clocks.

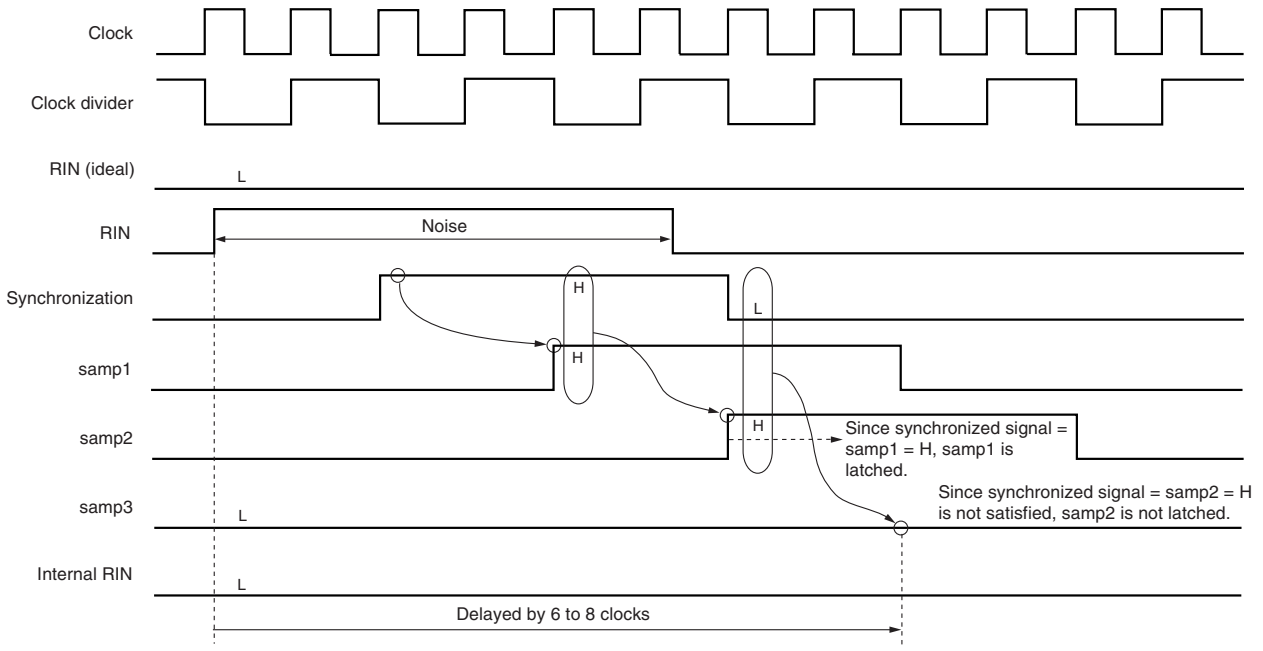
Figure 15-8. Noise Elimination Operation Example (2/2)

(c) 2-clock noise elimination (PRSEN = 1, NCW = 0)



Remark Internal RIN is a signal after synchronization and sampling are performed twice, and is therefore later than the actual signal input from the outside to the RIN pin by 4 to 6 clocks.

(d) 4-clock noise elimination (PRSEN = 1, NCW = 1)



Remark Internal RIN is a signal after synchronization and sampling are performed three times, and is therefore later than the actual signal input from the outside to the RIN pin by 6 to 8 clocks.

CHAPTER 16 INTERRUPT FUNCTIONS

16.1 Interrupt Function Types

The following two types of interrupt functions are used.

(1) Non-maskable interrupt

This interrupt is acknowledged unconditionally. It does not undergo interrupt priority control and is given top priority over all other interrupt requests.

A standby release signal is generated.

One interrupt source from the watchdog timer is incorporated as a non-maskable interrupt.

(2) Maskable interrupt

This interrupt undergoes mask control. If two or more interrupts with the same priority are simultaneously generated, each interrupt has a predetermined priority as shown in Table 16-1.

A standby release signal is generated.

For the μPD789477, 789478, and 78F9478, 5 external and 16 internal interrupt sources are incorporated as maskable interrupts.

For the μPD789479 and 78F9479, 6 external and 16 internal interrupt sources are incorporated as maskable interrupts.

16.2 Interrupt Sources and Configuration

A total of 22 non-maskable and maskable interrupts are incorporated as interrupt sources for the μPD789477, 789478, and 78F9478, and a total of 23 for the μPD789479 and 78F9479. (see **Table 16-1**).

Table 16-1. Interrupt Sources

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 (INTP0) input edge detection	External	0006H	(C)
	2	INTP1	Pin (INTP1) input edge detection		0008H	
	3	INTP2	Pin (INTP2) input edge detection		000AH	
	4	INTP3	Pin (INTP3) input edge detection		000CH	
	5	INTRIN	Remote controller edge detection	Internal	000EH	(B)
	6	INTSR20	UART reception completion		0010H	
		INTCSI20	End of 3-wire SIO transfer for serial interface 20			
	7	INTCSI10	End of 3-wire SIO transfer for serial interface 1A0		0012H	
	8	INTST20	End of UART transmission for serial interface 20		0014H	
	9	INTWTI	Standard time interval signal of watch timer (WT)		0016H	
	10	INTTM20	Match between TM20 and CR20		0018H	
	11	INTTM50	Match between TM50 and CR50		001AH	
	12	INTTM60	Match between TM60 and CR60 (in 8-bit counter mode), and between TM50, TM60 and CR50, CR60 (in 16-bit timer mode)		001CH	
	13	INTTM61	Match between TM61 and CR61		001EH	
	14	INTAD0	End of A/D conversion		0020H	
	15	INTWT	Watch timer (WT) overflow		0022H	
	16	INTKR00	Key return signal detection	External	0024H	(C)
	17	INTRERR	Remote controller reception error occurrence	Internal	0026H	(B)
	18	INTGP	Remote controller guide pulse detection		0028H	
	19	INTREND	Remote controller data reception completion		002AH	
	20	INTDFULL	Read request for remote controller 8-bit shift data		002CH	
	21	INTKR01[Note 3]	Key return signal detection	External	002EH	(C)

Notes 1. Default Priority is the priority order when more than one maskable interrupt request is generated at the same time. 0 is the highest priority and 21 is the lowest.

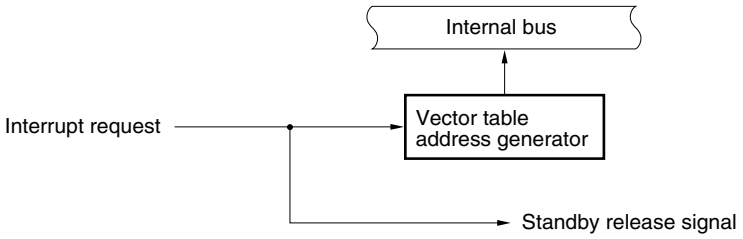
2. Basic configuration types (A), (B), and (C) correspond to (A), (B), and (C) in Figure 16-1.

3. μPD789479 and 78F9479 only.

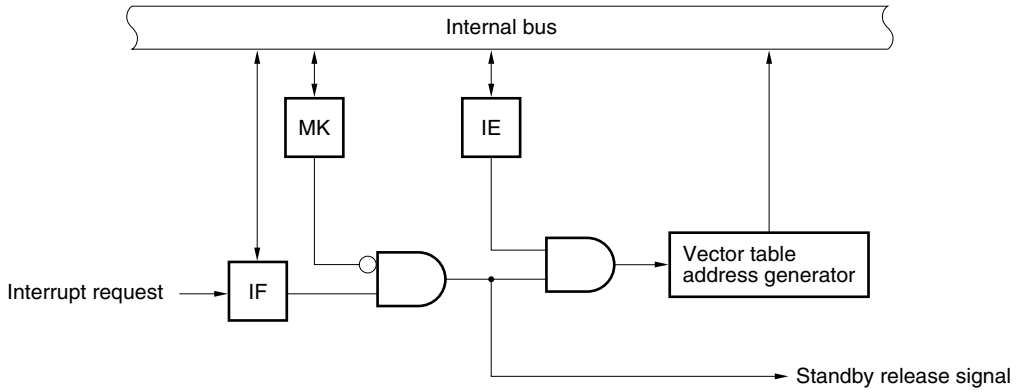
Remark Only one of the two watchdog timer interrupt (INTWDT) sources, non-maskable or maskable (internal), can be selected.

Figure 16-1. Basic Configuration of Interrupt Function

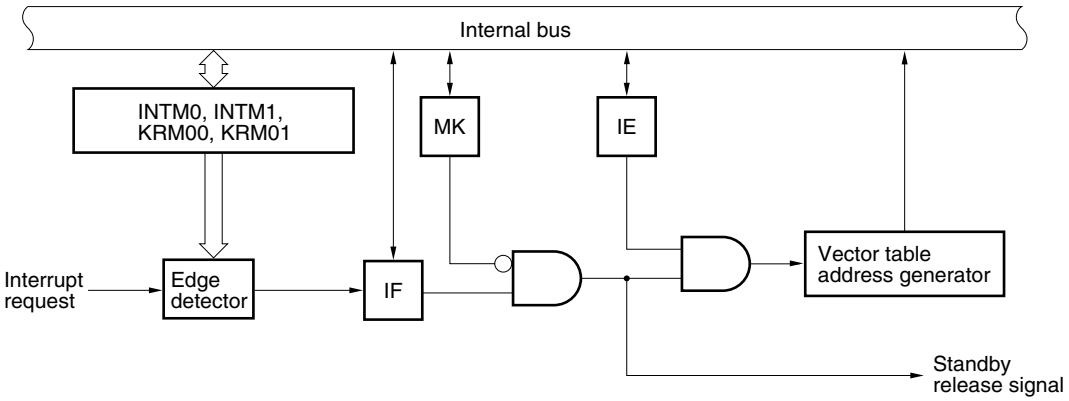
(A) Internal non-maskable interrupt



(B) Internal maskable interrupt



(C) External maskable interrupt



INTM0: External interrupt mode register 0
INTM1: External interrupt mode register 1
KRM00: Key return mode register 00
KRM01: Key return mode register 01
IF: Interrupt request flag
IE: Interrupt enable flag
MK: Interrupt mask flag

16.3 Registers Controlling Interrupt Function

The following five types of registers are used to control the interrupt functions.

- Interrupt request flag registers (IF0 to IF2)
- Interrupt mask flag registers (MK0 to MK2)
- External interrupt mode registers (INTM0 and INTM1)
- Program status word (PSW)
- Key return mode registers (KRM00 and KRM01)

Table 16-2 gives a listing of interrupt request flag and interrupt mask flag names corresponding to interrupt requests.

Table 16-2. Flags Corresponding to Interrupt Request Signal Names

Interrupt Request Signal	Interrupt Request Flag	Interrupt Mask Flag
INTWDT	WDTIF	WDTMK
INTP0	PIF0	PMK0
INTP1	PIF1	PMK1
INTP2	PIF2	PMK2
INTP3	PIF3	PMK3
INTRIN	RINIF	RINMK
INTSR20/INTCSI20	SRIF20	SRMK20
INTCSI10	CSIIF10	CSIMK10
INTST20	STIF20	STMK20
INTWTI	WTIIF	WTIMK
INTTM20	TMIF20	TMMK20
INTTM50	TMIF50	TMMK50
INTTM60	TMIF60	TMMK60
INTTM61	TMIF61	TMMK61
INTAD0	ADIF0	ADMK0
INTWT	WTIF	WTMK
INTKR00	KRIF00	KRMK00
INTRERR	RERRIF	RERRMK
INTGP	GPIF	GPMK
INTREND	RENDIF	RENDMK
INTDFULL	DFULLIF	DFULLMK
INTKR01[Note]	KRIF01[Note]	KRMK01[Note]

Note μPD789479 and 78F9479 only

(1) Interrupt request flag registers (IF0 to IF2)

An interrupt request flag is set (1) when the corresponding interrupt request is generated, or when an instruction is executed. It is cleared (0) when the interrupt request is acknowledged, when the $\overline{\text{RESET}}$ signal is input, or when an instruction is executed.

IF0 to IF2 are set with a 1-bit or 8-bit memory manipulation instruction.

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

Figure 16-2. Format of Interrupt Request Flag Registers

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
IF0	CSIIF10	SRIF20	RINIF	PIF3	PIF2	PIF1	PIF0	WDTIF	FFE0H	00H	R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
IF1	WTIF	ADIF0	TMIF61	TMIF60	TMIF50	TMIF20	WTIIF	STIF20	FFE1H	00H	R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
IF2	0	0	KRIF01[Note]	DFULLIF	RENDIF	GPIF	RERRIF	KRIF00	FFE2H	00H	R/W

××IF×	Interrupt request flag
0	No interrupt request signal generated
1	An interrupt request signal is generated and an interrupt request made

Note μPD789479 and 78F9479 only

Cautions 1. The WDTIF flag can be read/written only when the watchdog timer is being used as an interval timer. It must be cleared to 0 if the watchdog timer is used in watchdog timer mode 1 or 2.

2. Because P30 to P33 function alternately as external interrupts, when the output level changes after the output mode of the port function is specified, the interrupt request flag will be inadvertently set. Therefore, be sure to preset the interrupt mask flag (PMK0 to PMK3) before using the port in output mode.

(2) Interrupt mask flag registers (MK0 to MK2)

Interrupt mask flags are used to enable and disable the corresponding maskable interrupts.

MK0 to MK2 are set with a 1-bit or 8-bit memory manipulation instruction.

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

Figure 16-3. Format of Interrupt Mask Flag Registers

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
MK0	CSIMK10	SRMK20	RINMK	PMK3	PMK2	PMK1	PMK0	WDTMK	FFE4H	FFH	R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
MK1	WTMK	ADMK0	TMMK61	TMMK60	TMMK50	TMMK20	WTIMK	STMK20	FFE5H	FFH	R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
MK2	1	1	KRMK01[Note]	DFULLMK	RENDMK	GPMK	RERRMK	KRMK00	FFE6H	FFH	R/W

××MK×	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Note μPD789479 and 78F9479 only

Cautions 1. When the watchdog timer is being used in watchdog timer mode 1 or 2, any attempt to read the WDTMK flag results in an undefined value being detected.

2. Because P30 to P33 function alternately as external interrupts, when the output level changes after the output mode of the port function is specified, the interrupt request flag will be inadvertently set. Therefore, be sure to preset the interrupt mask flag (PMK0 to PMK3) before using the port in output mode.

(3) External interrupt mode registers (INTM0, INTM1)

These registers are used to specify the valid edge for INTP0 to INTP3.

INTM0 and INTM1 are set with an 8-bit memory manipulation instruction.

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

Figure 16-4. Format of External Interrupt Mode Registers

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
INTM0	ES21	ES20	ES11	ES10	ES01	ES00	0	0	FFECH	00H	R/W

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
INTM1	0	0	0	0	0	0	ES31	ES30	FFEDH	00H	R/W

ESn1	ESn0	INTPn valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

Remark n = 0, 1, 2, and 3

Cautions 1. Always set bits 0 and 1 of INTM0, and 2 to 7 of INTM1 to 0.

2. Before setting INTM0 and INTM1, set (1) the interrupt mask flags (PMK0 to PMK3) to disable interrupts.

To enable interrupts, clear (0) the interrupt request flags (PIF0 to PIF3), then clear (0) the interrupt mask flags (PMK0 to PMK3).

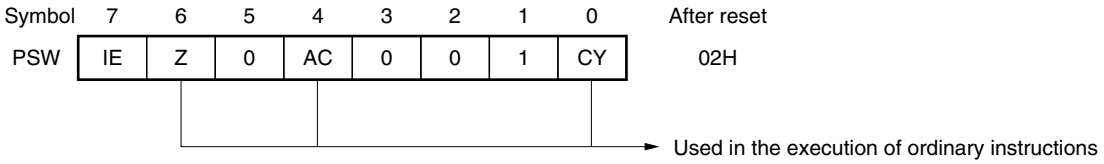
(4) Program status word (PSW)

The program status word is used to hold the instruction execution results and the current status of the interrupt requests. The IE flag, used to enable and disable maskable interrupts, is mapped to the PSW.

The PSW can be read and written in 8-bit units, and can be manipulated by using bit manipulation instructions and dedicated instructions (EI and DI). When a vectored interrupt is acknowledged, the PSW is automatically saved to the stack, and the IE flag is reset (0).

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

Figure 16-5. Program Status Word Configuration



IE	Interrupt acknowledgment enable/disable
0	Disabled
1	Enabled

(5) Key return mode register 00 (KRM00)

This register is used to set the pin that is to detect the key return signal (falling edge of port 0).

KRM00 is set with a 1-bit or 8-bit memory manipulation instruction.

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

Figure 16-6. Format of Key Return Mode Register 00

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
KRM00	KRM007	KRM006	KRM005	KRM004	0	0	0	KRM000	FFF5H	00H	R/W

KRM000	Control of key return signal detection
0	Key return signal not detected
1	Key return signal detected (P00 to P03 falling edge detection)

KRM00n	Control of key return signal detection
0	Key return signal not detected
1	Key return signal detected (P0n falling edge detection)

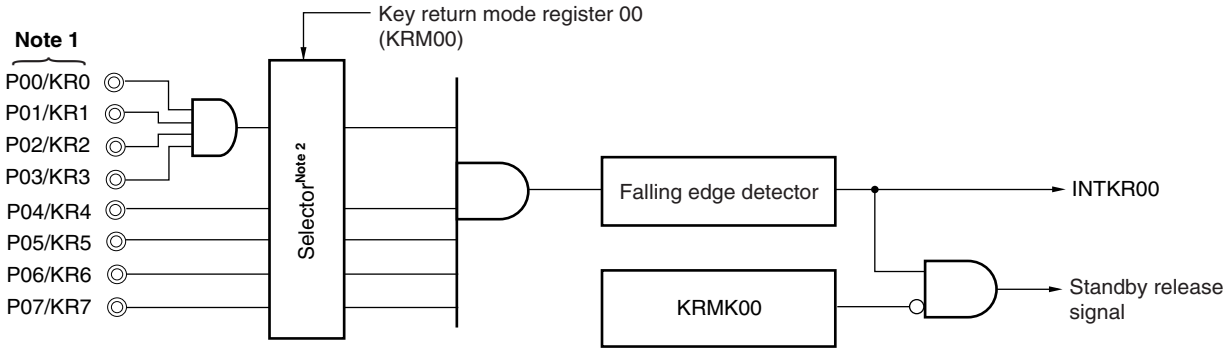
Remark n = 4 to 7

Cautions 1. Always set bits 1 to 3 to 0.

2. Before setting KRM00, set (1) bit 0 (KRMK00) of MK2 to disable interrupts. To enable interrupts, clear (0) KRMK00 after clearing (0) bit 0 (KRIF00) of IF2.

3. On-chip pull-up resistors are not automatically connected in input mode even when key return signal detection is specified. Therefore, when detecting the key return signal, connect the pull-up resistor of the corresponding bit using pull-up resistor option register B0 (PUB0). Although these resistors are disconnected when the mode changes to output, key return signal detection continues unchanged.

Figure 16-7. Block Diagram of Falling Edge Detector



Notes 1. The pin names are P00/KR00 to P07/KR07 in the μPD789479 and 78F9479.

2. For selecting the pin to be used as falling edge input.

(6) Key return mode register 01 (KRM01) (μPD789479 and 78F9479 only)

This register is used to set the pin that is to detect the key return signal (falling edge of port 6).

KRM01 is set with a 1-bit or 8-bit memory manipulation instruction.

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

Figure 16-8. Format of Key Return Mode Register 01

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
KRM01	KRM017	KRM016	KRM015	KRM014	0	0	0	KRM010	FFF4H	00H	R/W

KRM010	Control of key return signal detection
0	Key return signal not detected
1	Key return signal detected (P60 to P63 falling edge detection)

KRM01n	Control of key return signal detection
0	Key return signal not detected
1	Key return signal detected (P6n falling edge detection)

Remark n = 4 to 7

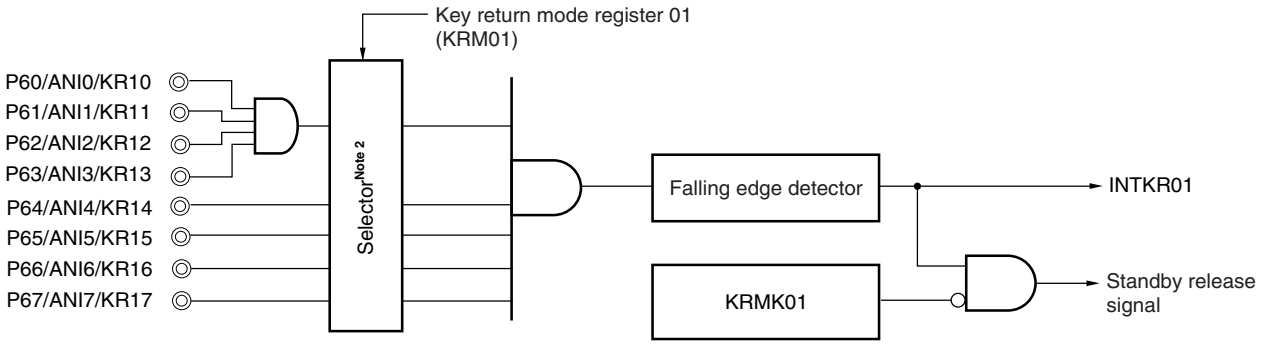
Cautions 1. Always set bits 1 to 3 to 0.

2. Before setting KRM01, set (1) bit 5 (KRMK01) of MK2 to disable interrupts. To enable interrupts, clear (0) KRMK01 after clearing (0) bit 5 (KRIF01) of IF2.

3. If any of the pins specified for key return signal detection is low level, the key return signal cannot be detected even if a falling edge is generated at other key return pins.

4. When even one of the P60/ANI0/KR10/ to P67/ANI7/KR17 pins is used as an A/D input, set KRM010 and KRM014 to KRM017 to 0.

Figure 16-9. Block Diagram of Falling Edge Detector



Notes For selecting the pin to be used as falling edge input.

16.4 Interrupt Servicing Operation

16.4.1 Non-maskable interrupt request acknowledgment operation

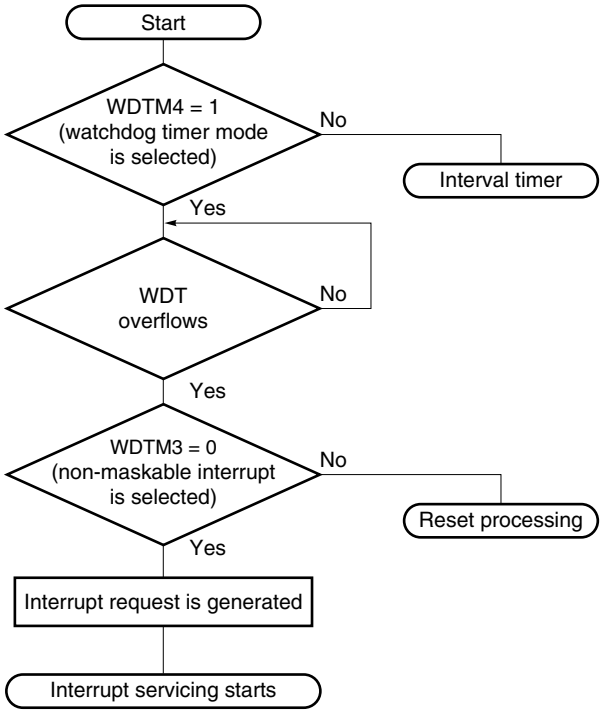
The non-maskable interrupt request is unconditionally acknowledged even when interrupts are disabled. It is not subject to interrupt priority control and takes precedence over all other interrupts.

When the non-maskable interrupt request is acknowledged, the PSW and PC are saved to the stack in that order, the IE flag is reset to 0, the contents of the vector table are loaded the PC, and then program execution branches.

Figure 16-10 shows the flow from non-maskable interrupt request generation to acknowledgment, Figure 16-11 shows the timing of non-maskable interrupt acknowledgment, and Figure 16-12 shows the acknowledgment operation when a number of non-maskable interrupts are generated.

Caution During non-maskable interrupt service program execution, do not input another non-maskable interrupt request; if it is input, the service program will be interrupted and the new non-maskable interrupt request will be acknowledged.

Figure 16-10. Flow from Generation of Non-Maskable Interrupt Request to Acknowledgment



WDTM: Watchdog timer mode register

WDT: Watchdog timer

Figure 16-11. Timing of Non-Maskable Interrupt Request Acknowledgment

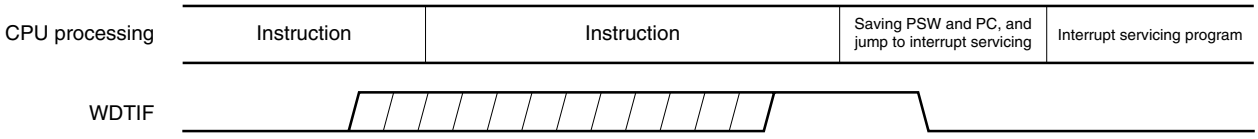
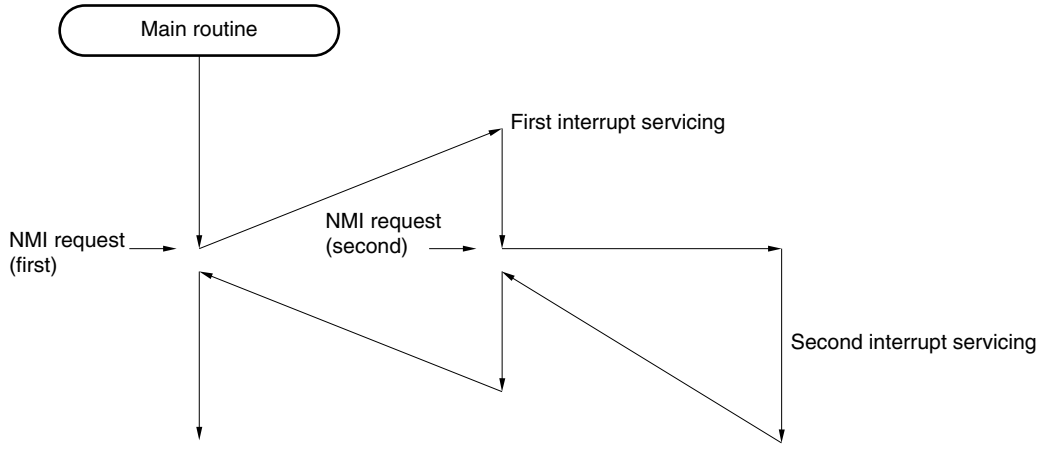


Figure 16-12. Non-Maskable Interrupt Request Acknowledgment



16.4.2 Maskable interrupt request acknowledgment operation

A maskable interrupt request can be acknowledged when the interrupt request flag is set to 1 and the corresponding interrupt mask flag is cleared to 0. A vectored interrupt is acknowledged in the interrupt enabled status (when the IE flag is set to 1).

The time required to start the interrupt servicing after a maskable interrupt request has been generated is shown in Table 16-3.

Refer to Figures 16-14 and 16-15 for the timing of interrupt request acknowledgment.

Table 16-3. Time from Generation of Maskable Interrupt Request to Servicing

Minimum Time	Maximum Time[Note]
9 clocks	19 clocks

Note The wait time is maximum when an interrupt request is generated immediately before the BT or BF instruction.

Remark 1 clock: $\frac{1}{f_{CPU}}$ (f_{CPU}: CPU clock)

When two or more maskable interrupt requests are generated at the same time, they are acknowledged starting from the one assigned the highest priority by the priority specification flag.

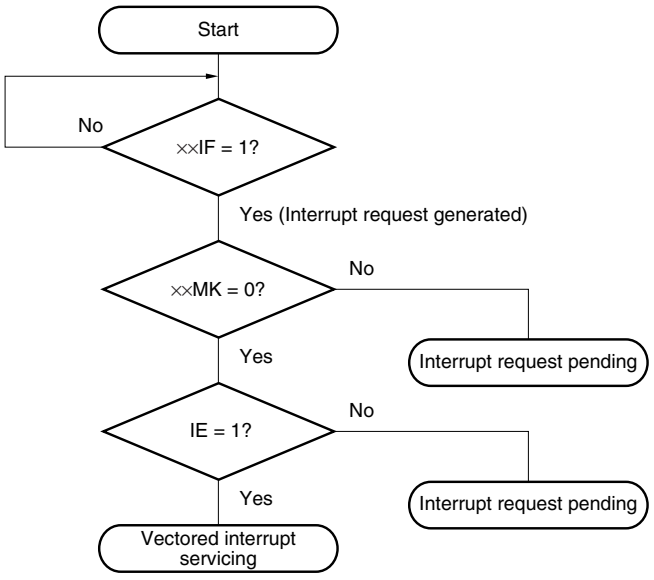
A pending interrupt is acknowledged when the status in which it can be acknowledged is set.

Figure 16-13 shows the algorithm of interrupt request acknowledgment.

When a maskable interrupt request is acknowledged, the PSW and PC are saved to the stack in that order, the IE flag is reset to 0, the data in the vector table determined for each interrupt request is loaded to the PC, and execution branches.

To return from interrupt servicing, use the RETI instruction.

Figure 16-13. Interrupt Request Acknowledgment Program Algorithm

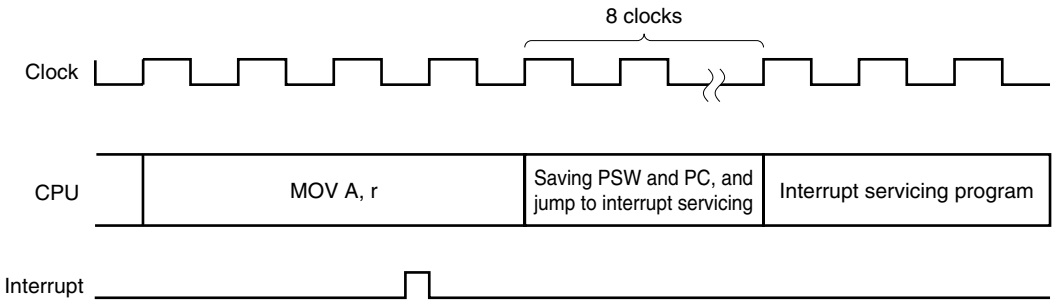


××IF: Interrupt request flag

××MK: Interrupt mask flag

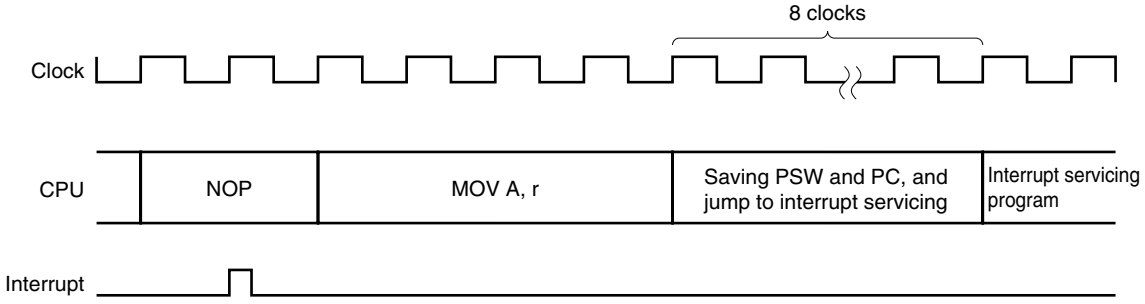
IE: Flag to control maskable interrupt request acknowledgment (1 = enable, 0 = disable)

Figure 16-14. Interrupt Request Acknowledgment Timing (Example: MOV A, r)



If the interrupt request has generated an interrupt request flag (××IF×) by the time the instruction clocks under execution, n clocks (n = 4 to 10), are n − 1, interrupt request acknowledgment processing will start following the completion of the instruction under execution. Figure 16-14 shows an example using the 8-bit data transfer instruction MOV A, r. Because this instruction is executed in 4 clocks, if an interrupt request is generated between the start of execution and the 3rd clock, interrupt request acknowledgment processing will take place following the completion of MOV A, r.

Figure 16-15. Interrupt Request Acknowledgment Timing
(When Interrupt Request Flag Is Generated in Final Clock Under Execution)



If the interrupt request flag (××IF×) is generated in the final clock of the instruction, interrupt request acknowledgment processing will begin after execution of the next instruction is complete.

Figure 16-15 shows an example whereby an interrupt request was generated in the 2nd clock of NOP (a 2-clock instruction). In this case, the interrupt request will be processed after execution of MOV A, r, which follows NOP, is complete.

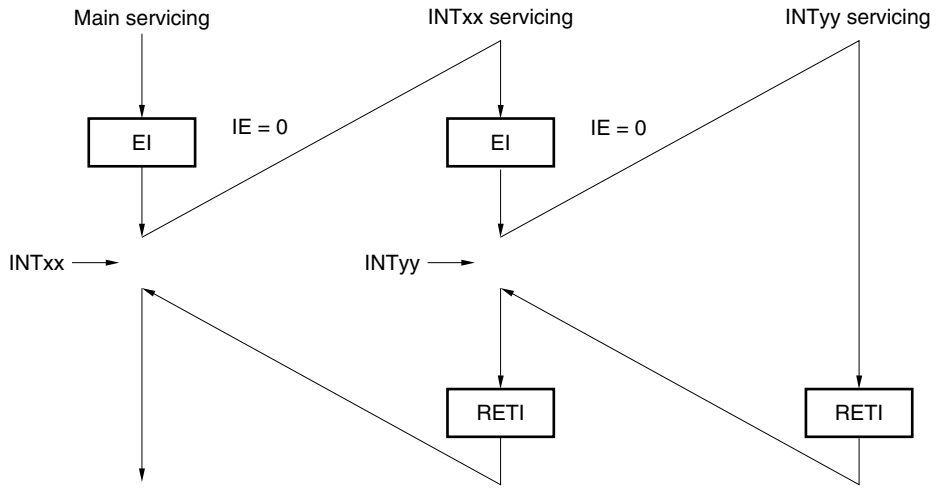
Caution When interrupt request flag registers (IF0 to IF2), or interrupt mask flag registers (MK0 to MK2) are being accessed, interrupt requests will be held pending.

16.4.3 Multiple interrupt servicing

Multiple interrupt servicing, in which an interrupt request is acknowledged while another interrupt request being serviced, can be executed using the priority order. If multiple interrupts are generated at the same time, they are serviced in the order according to the priority assigned to each interrupt request in advance (refer to **Table 16-1**).

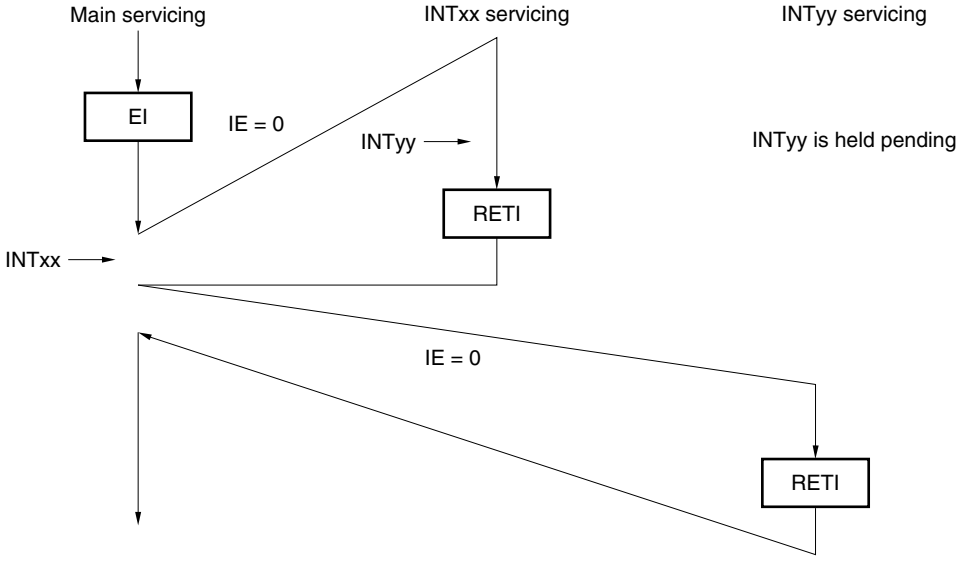
Figure 16-16. Example of Multiple Interrupt Servicing

Example 1. Acknowledging multiple interrupts



The interrupt request INTyy is acknowledged during the servicing of interrupt INTxx and multiple interrupt servicing is performed. Before each interrupt request is acknowledged, the EI instruction is issued and the interrupt request is enabled.

Example 2. Multiple interrupt servicing is not performed because interrupts are disabled



Because interrupt requests are disabled (the EI instruction has not been issued) in the INTxx interrupt servicing, the interrupt request INTyy is not acknowledged and multiple interrupt servicing is not performed. INTyy is held pending and is acknowledged after INTxx servicing is completed.

IE = 0: Interrupt requests disabled

16.4.4 Putting interrupt requests on hold

If an interrupt request (such as a maskable, non-maskable, or external interrupt) is generated when a certain type of instruction is being executed, the interrupt request will not be acknowledged until the instruction is completed. Such instructions (interrupt request pending instructions) are as follows.

- Instructions that manipulate interrupt request flag registers (IF0 to IF2)
- Instructions that manipulate interrupt mask flag registers (MK0 to MK2)

CHAPTER 17 STANDBY FUNCTION

17.1 Standby Function and Configuration

17.1.1 Standby function

The standby function is used to reduce the power consumption of the system and can be effected in the following two modes.

(1) HALT mode

This mode is set when the HALT instruction is executed. The HALT mode stops the operation clock of the CPU. The system clock oscillator continues oscillating. This mode does not reduce the power consumption as much as the STOP mode, but is useful for resuming processing immediately when an interrupt request is generated, or for intermittent operations.

(2) STOP mode

This mode is set when the STOP instruction is executed. The STOP mode stops the main system clock oscillator and stops the entire system. The power consumption of the CPU can be substantially reduced in this mode.

The data memory can be retained at a low voltage (V_{DD} = 1.8 V). Therefore, this mode is useful for retaining the contents of the data memory at extremely low power.

The STOP mode can be released by an interrupt request, so that this mode can be used for intermittent operation. However, some time is required until the system clock oscillator stabilizes after the STOP mode has been released. If processing must be resumed immediately by using an interrupt request, therefore, use the HALT mode.

In both modes, the previous contents of the registers, flags, and data memory before setting the standby mode are all retained. In addition, the statuses of the output latches of the I/O ports and output buffers are also retained.

Caution To set the STOP mode, be sure to stop the operations of the peripheral hardware, and then execute the STOP instruction.

17.1.2 Register controlling standby function

The wait time after the STOP mode is released upon interrupt request generation until oscillation stabilizes is controlled by the oscillation stabilization time selection register (OSTS).

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

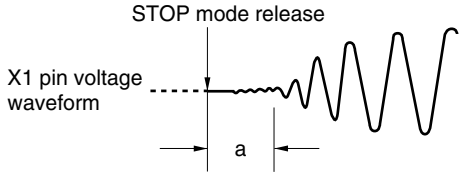
$\overline{\text{RESET}}$ input sets OSTS to 04H. However, it takes 2^{15}/fx, not 2^{17}/fx, to stabilize oscillation after $\overline{\text{RESET}}$ input.

Figure 17-1. Format of Oscillation Stabilization Time Selection Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0	FFFAH	04H	R/W

OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection
0	0	0	2^{12}/fx (819 μs)
0	1	0	2^{15}/fx (6.55 ms)
1	0	0	2^{17}/fx (26.2 ms)
Other than above			Setting prohibited

Caution The wait time after the STOP mode is released does not include the time from STOP mode release to clock oscillation start ("a" in the figure below), regardless of whether STOP mode is released by $\overline{\text{RESET}}$ input or by interrupt generation.



Remarks 1. fx: Main system clock oscillation frequency

2. The parenthesized values apply to operation at fx = 5.0 MHz.

17.2 Standby Function Operation

17.2.1 HALT mode

(1) HALT mode

The HALT mode is set by executing the HALT instruction.

The operation statuses in the HALT mode are shown in the following table.

Table 17-1. Operation Statuses in HALT Mode

Item	HALT Mode Operation Status During Main System Clock Operation		HALT Mode Operation Status During Subsystem Clock Operation	
	Subsystem Clock Operating	Subsystem Clock Stopped	Main System Clock Operating	Main System Clock Stopped
Clock generator	Oscillation enabled for both main system clock and subsystem clock, but clock supply to CPU is stopped			
Subsystem clock ×4 multiplication circuit	Operation stopped			
CPU	Operation stopped			
Ports (output latches)	Status before HALT mode setting retained			
16-bit timer 20	Operable			Operable[Note 1]
8-bit timer 50	Operable			Operable[Note 2]
8-bit timer 60	Operable			Operable[Note 3]
8-bit timer 61	Operable			Operable[Note 3]
Watch timer	Operable	Operable[Note 4]	Operable	Operable[Note 5]
Watchdog timer	Operable		Operation stopped	
Key return circuit	Operable			
Serial interface 20	Operable			Operable[Note 6]
Serial interface 1A0	Operable			Operable[Note 6]
LCD controller/driver	Operable[Note 7]	Operable[Notes 4, 7]	Operable[Note 7]	Operable[Notes 5, 7]
A/D converter	Operation stopped			
Multiplier	Operation stopped			
Remote controller receiver	Operable	Operable[Note 4]	Operable	Operable[Note 5]
External interrupts	Operable[Note 8]			

Notes 1. Operation is enabled when the 24-bit counter mode is selected.

2. Operation is enabled when either the subsystem clock or the input signal from timer 60 (when timer 60 is operable) is selected as the count clock.

3. Operation is enabled only when the external input clock is selected as the count clock.

4. Operation is enabled when the main system clock is selected.

5. Operation is enabled when the subsystem clock is selected.

6. Operation is enabled only when an external clock is selected.

7. The HALT instruction can be set after display instruction execution.

8. Operation is enabled only for a maskable interrupt that is not masked.

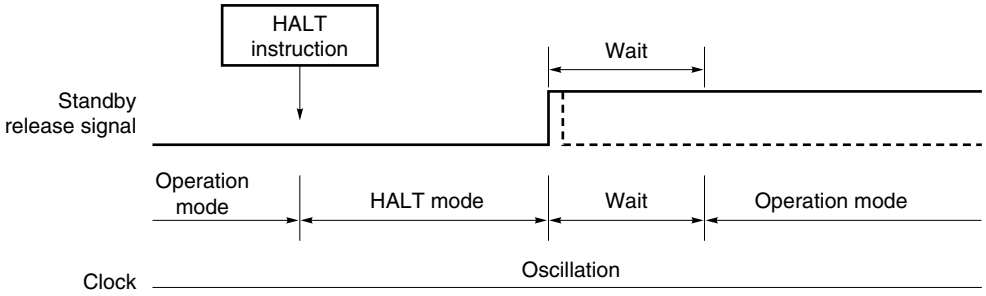
(2) Releasing HALT mode

The HALT mode can be released by the following three sources.

(a) Release by unmasked interrupt request

The HALT mode is released by an unmasked interrupt request. In this case, if interrupts are enabled to be acknowledged, vectored interrupt servicing is performed. If interrupts are disabled, the instruction at the next address is executed.

Figure 17-2. Releasing HALT Mode by Interrupt



Remarks 1. The broken lines indicate the case where the interrupt request that released the standby mode is acknowledged.

2. The wait time is as follows:

- When vectored interrupt servicing is performed: 9 to 10 clocks
- When vectored interrupt servicing is not performed: 1 to 2 clocks

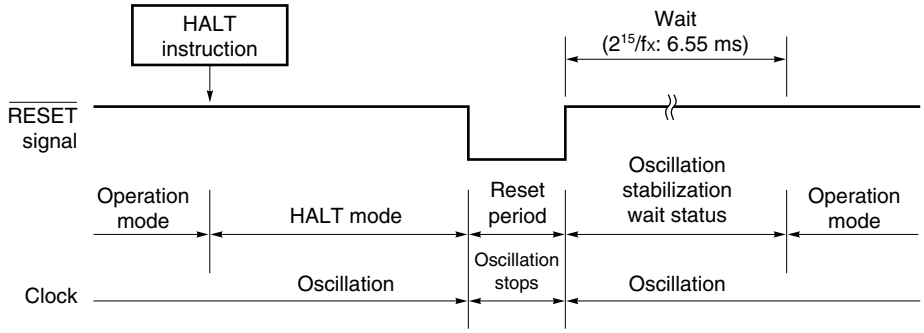
(b) Release by non-maskable interrupt request

The HALT mode is released regardless of whether interrupts are enabled or disabled, and vectored interrupt servicing is performed.

(c) Release by $\overline{\text{RESET}}$ input

When the HALT mode is released by the $\overline{\text{RESET}}$ signal, execution branches to the reset vector address in the same manner as the ordinary reset operation, and program execution is started.

Figure 17-3. Releasing HALT Mode by $\overline{\text{RESET}}$ Input



Remark fx: Main system clock oscillation frequency

Table 17-2. Operation After Releasing HALT Mode

Releasing Source	MK××	IE	Operation
Maskable interrupt request	0	0	Executes next address instruction
	0	1	Executes interrupt servicing
	1	×	Retains HALT mode
Non-maskable interrupt request	–	×	Executes interrupt servicing
$\overline{\text{RESET}}$ input	–	–	Reset processing

×: don't care

Caution Some constraints apply when the flash version (μPD78F9478 and 78F9479) is used in the HALT mode with the subclock multiplied by 4 as the CPU clock. For details, refer to 19.2 Cautions on μPD78F9478 and 78F9479.

17.2.2 STOP mode

(1) Setting and operation status of STOP mode

The STOP mode is set by executing the STOP instruction.

Caution Because the standby mode can be released by an interrupt request signal, the standby mode is released as soon as it is set if there is an interrupt source whose interrupt request flag is set and interrupt mask flag is reset. When the STOP mode is set, therefore, the HALT mode is set immediately after the STOP instruction has been executed, the wait time set by the oscillation stabilization time selection register (OSTS) elapses, and then the operation mode is set.

The operation statuses in the STOP mode are shown in the following table.

Table 17-3. Operation Statuses in STOP Mode

Item	STOP Mode Operation Status During Main System Clock Operation	
	Subsystem Clock Operating	Subsystem Clock Stopped
Main system clock	Oscillation stopped	
Subsystem clock ×4 multiplication circuit	Operation stopped	
CPU	Operation stopped	
Ports (output latches)	Status before STOP mode setting retained	
16-bit timer 20	Operation stopped	
8-bit timer 50	Operable[Note 1]	Operable[Note 2]
8-bit timer 60	Operable[Note 3]	
8-bit timer 61	Operable[Note 3]	
Watch timer	Operable[Note 4]	Operation stopped
Watchdog timer	Operation stopped	
Key return circuit	Operable	
Serial interface 20	Operable[Note 5]	
Serial interface 1A0	Operable[Note 5]	
LCD controller/driver	Operable[Note 4]	Operation stopped
A/D converter	Operation stopped	
Multiplier	Operation stopped	
Remote controller receiver	Operable[Note 4]	Operation stopped
External interrupts	Operable[Note 6]	

Notes 1. Operation is enabled when either the subsystem clock or the input signal from the timer 60 (when timer 60 is operable) is selected as the count clock.

2. Operation is enabled when the input signal from timer 60 (when timer 60 is operable) is selected as the count clock.

3. Operation is enabled when the external input clock is selected as the count clock.

4. Operation is enabled when the subsystem clock is selected.

5. Operation is enabled only when an external clock is selected.

6. Operation is enabled only for a maskable interrupt that is not masked

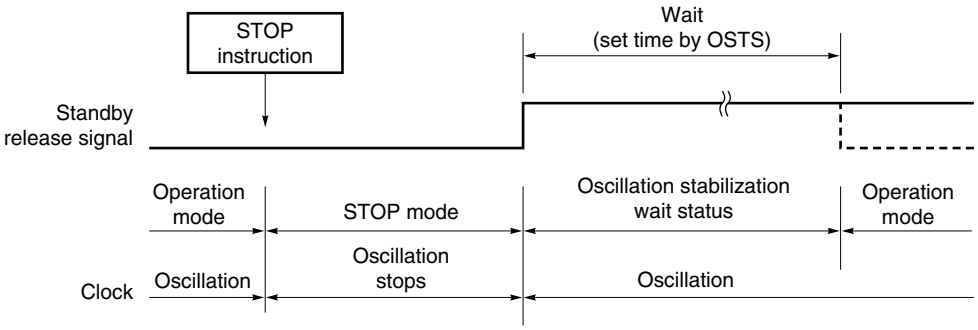
(2) Releasing STOP mode

The STOP mode can be released by the following two sources.

(a) Release by unmasked interrupt request

The STOP mode can be released by an unmasked interrupt request. In this case, if interrupts are enabled to be acknowledged, vectored interrupt servicing is performed, after the oscillation stabilization time has elapsed. If interrupts are disabled, the instruction at the next address is executed.

Figure 17-4. Releasing STOP Mode by Interrupt

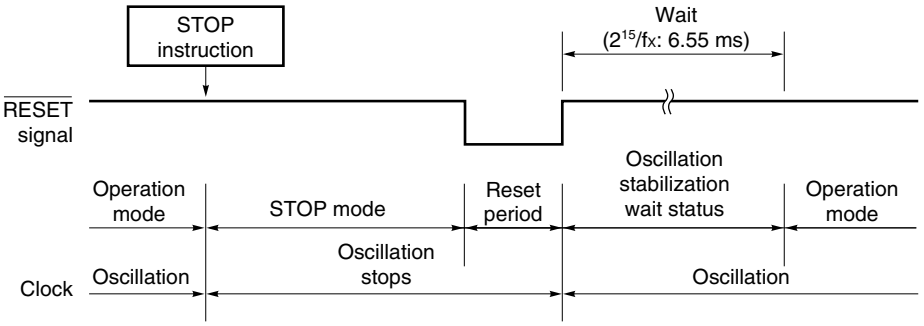


Remark The broken lines indicate the case where the interrupt request that released the standby mode is acknowledged.

(b) Release by $\overline{\text{RESET}}$ input

When the STOP mode is released by the $\overline{\text{RESET}}$ signal, the reset operation is performed after the oscillation stabilization time has elapsed.

Figure 17-5. Releasing STOP Mode by $\overline{\text{RESET}}$ Input



Remark fx: Main system clock oscillation frequency

Table 17-4. Operation After Releasing STOP Mode

Releasing Source	MK××	IE	Operation
Maskable interrupt request	0	0	Executes next address instruction
	0	1	Executes interrupt servicing
	1	×	Retains STOP mode
$\overline{\text{RESET}}$ input	–	–	Reset processing

×: don't care

CHAPTER 18 RESET FUNCTION

The following two operations are available to generate reset signals.

(1) External reset input by $\overline{\text{RESET}}$ pin
(2) Internal reset by watchdog timer program loop time detection

External and internal reset have no functional differences. In both cases, program execution starts at the address at 0000H and 0001H by $\overline{\text{RESET}}$ 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 18-1. 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 is started after the oscillation stabilization time (2^{15}/fx) has elapsed. The reset applied by the watchdog timer overflow is automatically released after reset, and program execution is started after the oscillation stabilization time (2^{15}/fx) has elapsed (see **Figures 18-2** to **18-4**.)

Cautions 1. For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin.

2. When the STOP mode is released by reset, the STOP mode contents are held during reset input. However, the port pins become high impedance.

Figure 18-1. Block Diagram of Reset Function

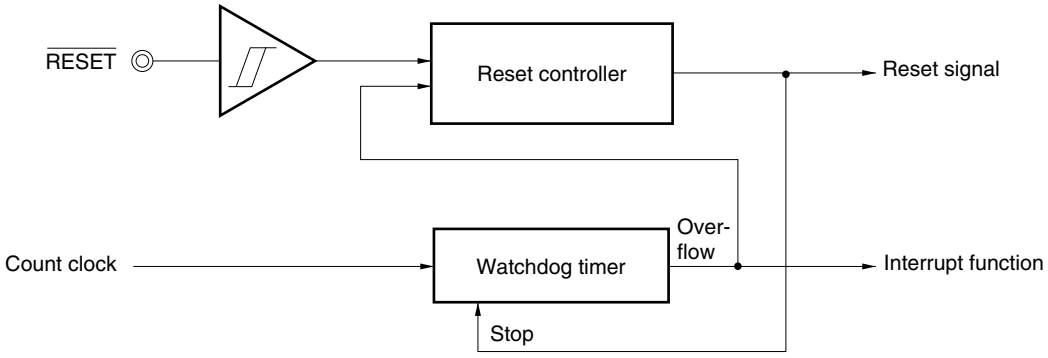


Figure 18-2. Reset Timing by $\overline{\text{RESET}}$ Input

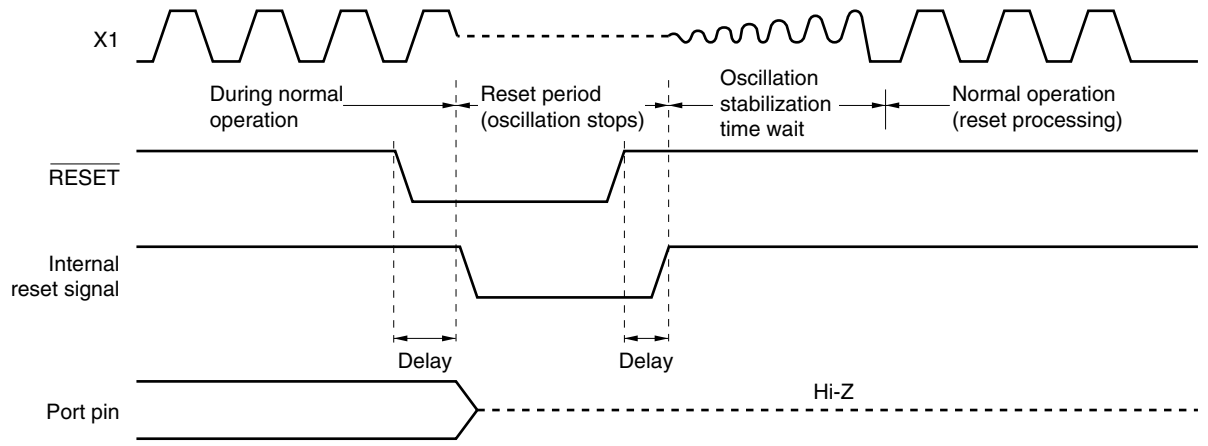


Figure 18-3. Reset Timing by Overflow in Watchdog Timer

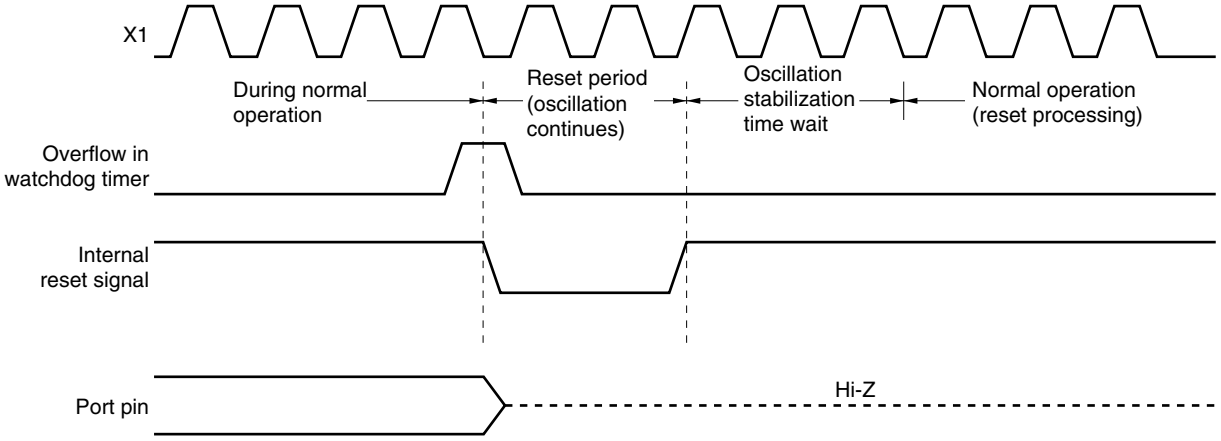


Figure 18-4. Reset Timing by $\overline{\text{RESET}}$ Input in STOP Mode

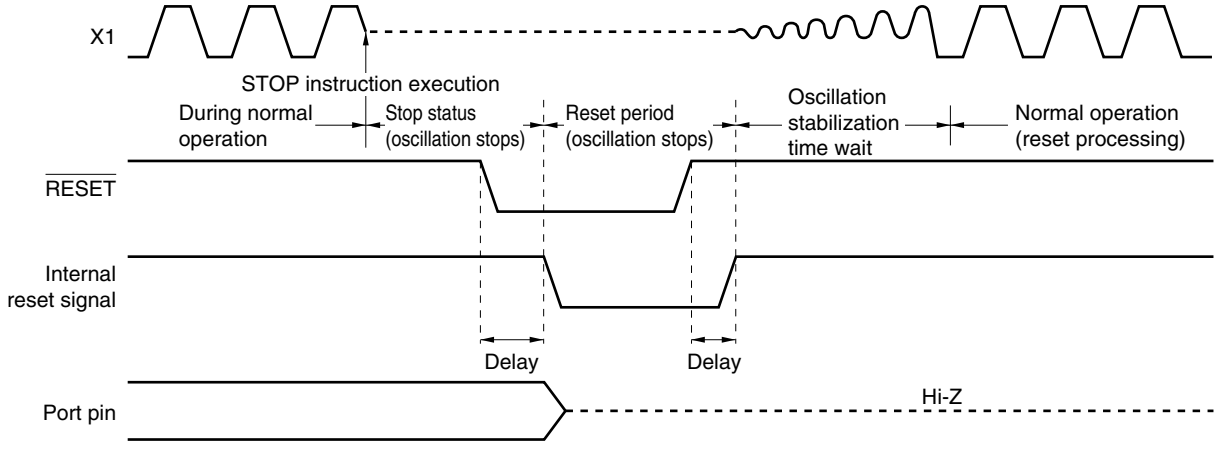


Table 18-1. Status of Hardware After Reset (1/2)

Hardware		Status After Reset
Program counter (PC)[Note 1]		Contents of reset vector table (0000H, 0001H) set
Stack pointer (SP)		Undefined
Program status word (PSW)		02H
RAM	Data memory	Undefined[Note 2]
	General-purpose registers	Undefined[Note 2]
Ports (P0 to P3, P5, P8[Note 3]) (output latches)		00H
Port mode registers (PM0 to PM3, PM5, PM8[Note 3])		FFH
Port function registers (PF7, PF8)		00H
Pull-up resistor option registers (PUB0 to PUB3)		00H
Processor clock control register (PCC)		02H
Subclock oscillation mode register (SCKM)		00H
Subclock selection register (SSCK)		Retained[Note 4]
Subclock control register (CSS)		00H
Oscillation stabilization time selection register (OSTS)		04H
16-bit timer 20	Timer counter (TM20)	0000H
	Compare register (CR20)	FFFFH
	Mode control register (TMC20)	00H
	Capture register (TCP20)	Undefined
8-bit timer 50, 60, 61	Timer counters (TM50, TM60, TM61)	00H
	Compare registers (CR50, CR60, CRH60, CR61, CRH61)	Undefined
	Mode control registers (TMC50, TMC60, TMC61)	00H
	Carrier generator output control register (TCA60)	00H
Watch timer	Mode control register (WTM)	00H
	Interrupt time selection register (WTIM)	00H
Watchdog timer	Clock selection register (WDCS)	00H
	Mode register (WDTM)	00H
Serial interface 20	Operation mode register (CSIM20)	00H
	Asynchronous serial interface mode register (ASIM20)	00H
	Asynchronous serial interface status register (ASIS20)	00H
	Baud rate generator control register (BRGC20)	00H
	Transmit shift register (TXS20)	FFH
	Receive buffer register (RXB20)	Undefined

Notes 1. While a reset signal is being input, and during the oscillation stabilization period, only the contents of the PC will be undefined; the remainder of the hardware will be the same state as after reset.

2. In standby mode, RAM enters the hold state after reset.

3. Port 8 is used only when the port function is specified by a mask option or port function register (refer to **CHAPTER 20 MASK OPTIONS** and **4.3 (3) Port function registers**).

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

Table 18-1. Status of Hardware After Reset (2/2)

Hardware		Status After Reset
Serial interface 1A0	Operation mode register (CSIM1A0)	00H
	Shift register (SIO1A0)	00H
	Buffer memory (SBMEM0 to SBMEMF)	Undefined
	Automatic data transmit/receive control register (ADTC0)	00H
	Automatic data transmit/receive address pointer (ADTP0)	Undefined
	Automatic data transmit/receive transfer interval specification register (ADTI0)	00H
A/D converter	Mode register (ADML0, ADML1)	00H
	Input channel specification register (ADS0)	00H
	Conversion result register (ADCRL0)	00H
LCD controller/driver	Display mode register (LCDM0)	00H
	Clock control register (LCDC0)	00H
Multiplier	16-bit result storage register (MUL0)	Undefined
	Data register (MRA0, MRB0)	Undefined
	Control register (MULC0)	00H
Remote controller receiver	Control register (RMCN)	00H
	Data register (RMDR)	00H
	Shift register reception counter register (RMSCR)	00H
	Shift register (RMSR)	00H
	Compare registers (RMGPHS, RMGPHL, RMDLS, RMDLL, RMDH0S, RMDH0L, RMDH1S, RMDH1L)	00H
	End width selection register (RMER)	00H
Interrupts	Request flag register (IF0 to IF2)	00H
	Mask flag register (MK0 to MK2)	FFH
	External interrupt mode register (INTM0, INTM1)	00H
	Key return mode registers (KRM00, KRM01[Note])	00H

Note KRM01 is only provided in the μPD789479 and 78F9479

CHAPTER 19 FLASH MEMORY VERSION

The μPD78F9478 is available as the flash memory version of the μPD789477 and 789478 (mask ROM versions).
The μPD78F9479 is available as the flash memory version of the μPD789479 (mask ROM version).
The differences between the μPD78F9478, 78F9479 and the mask ROM versions are shown in Table 19-1.

Table 19-1. Differences Between μPD78F9478, 78F9479, and Mask ROM Version

Item		Flash Memory Version		Mask ROM Version		
		μPD78F9478	μPD78F9479	μPD789477	μPD789478	μPD789479
Internal memory	ROM	32 KB (flash memory)	48 KB (flash memory)	24 KB	32 KB	48 KB
	Internal RAM	1,024 bytes	1,536 bytes	768 bytes	1,024 bytes	1,536 bytes
	LCD display RAM	28 × 4 bits				
Pin function selection S16 to S27 (LCD segment output) or P70 to P73 and P80 to P87 (general-purpose ports)		Selectable by a port function register (PF7 and PF8) in bit units		Selectable by a mask option in bit units		
Circuit to multiply subsystem clock by ×4		Use enabled/disabled by subclock select register (SSCK)		Use enabled/disabled by a mask option		
Pull-up resistor of port 5		None		Selectable by a mask option in 1-bit units		
Key return signal detection pins		P00/KR0 to P07/KR7	P00/KR00 to P07/KR07, P60/ANI0/KR10 to P67/ANI7/KR17	P00/KR0 to P07/KR7		P00/KR00 to P07/KR07, P60/ANI0/KR10 to P67/ANI7/KR17
Restrictions in HALT mode when using subclock ×4 clock		Refer to **19.2 Cautions on μPD78F9478 and 78F9479**.		None		
IC0 pin		Not provided		Provided		
V_{PP} pin		Provided		Not provided		
Electrical specifications		Refer to **CHAPTER 22 ELECTRICAL SPECIFICATIONS**				

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.

19.1 Flash Memory Characteristics

Flash memory programming is performed by connecting a dedicated flash programmer (Flashpro III (part no. FL-PR3, PG-FP3)/Flashpro IV (part no. FL-PR4, PG-FP4)) to the target system with the μPD78F9478 or 78F9479 mounted on the target system (on-board). A flash memory program adapter (FA adapter), which is a target board used exclusively for programming, is also provided.

Remark FL-PR3, FL-PR4, and the program adapter are products of Naito Densei Machida Mfg. Co., Ltd. (TEL +81-42-750-4172).

Programming using flash memory has the following advantages.

- Software can be modified after the microcontroller is solder-mounted on the target system.
- Distinguishing software facilitates low-quantity, varied model production
- Easy data adjustment when starting mass production

19.1.1 Programming environment

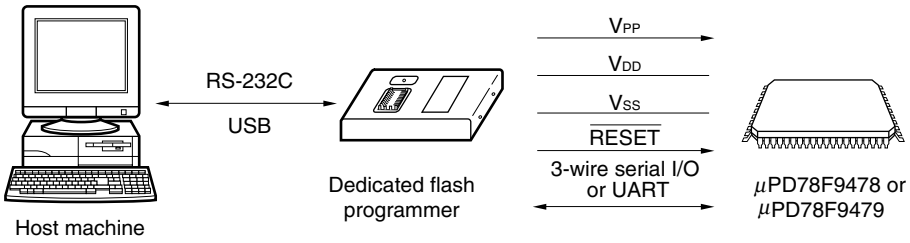
The following shows the environment required for μPD78F9478 and 78F9479 flash memory programming.

When Flashpro III (part no. FL-PR3, PG-FP3) or Flashpro IV (part no. FL-PR4, PG-FP4) is used as a dedicated flash programmer, a host machine is required to control the dedicated flash programmer. Communication between the host machine and flash programmer is performed via RS-232C/USB (Rev. 1.1).

For details, refer to the manuals of Flashpro III/Flashpro IV.

Remark USB is supported by Flashpro IV only.

Figure 19-1. Environment for Writing Program to Flash Memory



19.1.2 Communication mode

Use the communication mode shown in Table 19-2 to perform communication between the dedicated flash programmer and μPD78F9478 or μPD78F9479.

Table 19-2. Communication Mode List

Communication Mode	TYPE Setting[Note 1]					Pins Used	Number of V_{PP} Pulses
	COMM PORT	SIO Clock	CPU Clock		Multiple Rate		
			In Flashpro	On Target Board			
3-wire serial I/O	SIO ch-0 (3-wired, sync.)	100 Hz to 1.25 MHz[Note 2]	1, 2, 4, 5 MHz[Note 3]	1 to 5 MHz[Note 2]	1.0	SI20/RxD20/P22 SO20/TxD20/P21 $\overline{SCK20}$/ASCK20/P20	0
<R>						(SI20)[Note 6]/KR6/P06 (SO20)[Note 6]/KR5/P05 ($\overline{SCK20}$)[Note 6]/KR4/P04	1
3-wire serial I/O with handshake	SIO ch-3 + handshake					SI20/RxD20/P22 SO20/TxD20/P21 $\overline{SCK20}$/ASCK20/P20 P11 (HS)	3
UART	UART ch-0 (Async.)	4,800 to 76,800 bps [Notes 2, 4]	5 MHz[Note 5]	4.91 or 5 MHz[Note 2]	1.0	RxD20/SI20/P22 TxD20/SO20/P21	8

Notes 1. Selection items for TYPE settings on the dedicated flash programmer (Flashpro III (part no. FL-PR3, PG-FP3)/Flashpro IV (part no. FL-PR4, PG-FP4)).

2. The possible setting range differs depending on the voltage. For details, refer to **CHAPTER 22 ELECTRICAL SPECIFICATIONS**.

3. Only 2 MHz or 4 MHz can be selected for Flashpro III.

4. Because signal wave slew also affects UART communication, in addition to the baud rate error, thoroughly evaluate the slew.

5. Available for only Flashpro IV. However, when using Flashpro III, be sure to select the clock of the resonator on the board. UART cannot be used with the clock supplied by Flashpro III.

<R> **6.** Only when flash memory programming

Figure 19-2. Communication Mode Selection Format

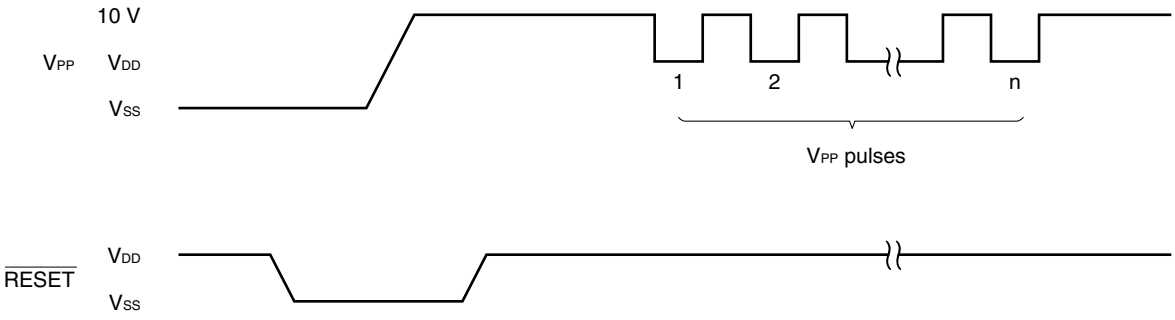
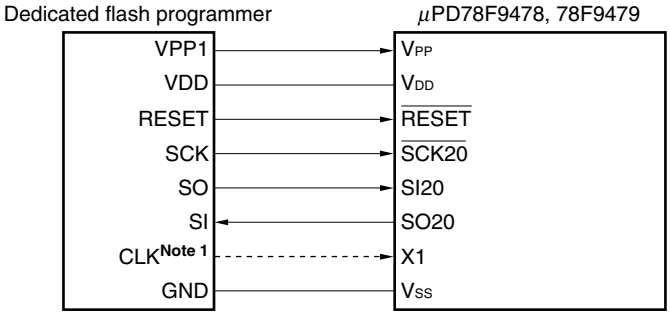
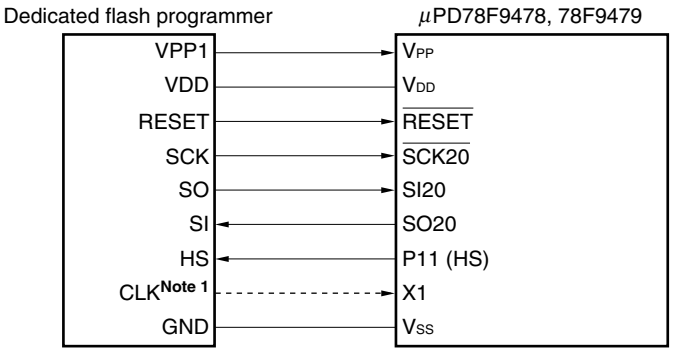


Figure 19-3. Example of Connection with Dedicated Flash Programmer

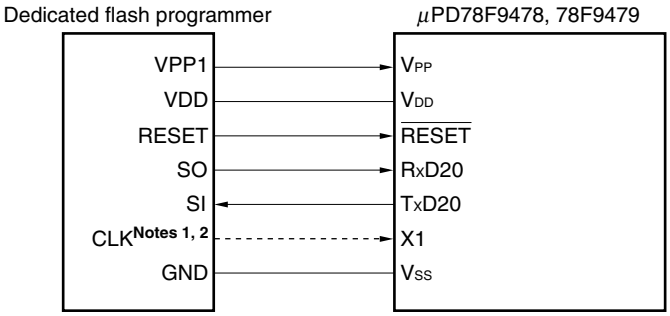
(a) 3-wire serial I/O



(b) 3-wire serial I/O with handshake



(c) UART



Notes 1. Connect the CLK pin to the X1 pin, and separate it from the on-board resonator when the system clock is supplied from the dedicated flash programmer. Do not connect to the CLK pin when the clock of the on-board resonator is used.

2. When using UART with Flashpro III, the clock of the resonator connected to the X1 pin must be used, so do not connect the CLK pin.

Caution The V_{DD} pin, if already connected to the power supply, must be connected to the VDD pin of the dedicated flash programmer. Before using the power supply connected to the V_{DD} pin, supply voltage before starting programming.

If Flashpro III (part no. FL-PR3, PG-FP3)/Flashpro IV (part no. FL-PR4, PG-FP4) is used as a dedicated flash programmer, the following signals are generated for the μPD78F9478 and 78F9479. For details, refer to the manual of Flashpro III/Flashpro IV.

Table 19-3. Pin Connection List

Signal Name	I/O	Pin Function	Pin Name	3-Wire Serial I/O	3-Wire Serial I/O with Handshake	UART
VPP1	Output	Write voltage	V_{PP}	◎	◎	◎
VPP2	–	–	–	×	×	×
VDD	I/O	V_{DD} voltage generation/ voltage monitoring	V_{DD}	◎Note	◎Note	◎Note
GND	–	Ground	V_{SS}	◎	◎	◎
CLK	Output	Clock output	X1	○	○	○
RESET	Output	Reset signal	$\overline{\text{RESET}}$	◎	◎	◎
SI	Input	Receive signal	SO20/TxD20	◎	◎	◎
SO	Output	Transmit signal	SI20/RxD20	◎	◎	◎
SCK	Output	Transfer clock	$\overline{\text{SCK20}}$	◎	◎	×
HS	Input	Handshake signal	P11 (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 does not need to be connected.
×: Pin does not need to be connected.

19.1.3 On-board pin processing

When performing programming on the target system, provide a connector on the target system to connect the dedicated flash programmer.

An on-board function that allows switching between normal operation mode and flash memory programming mode may be required in some cases.

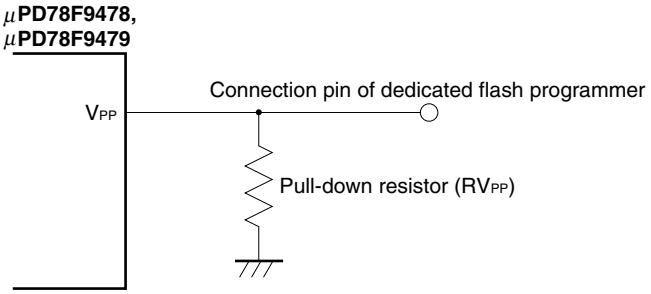
<V_{PP} pin>

In normal operation mode, input 0 V to the V_{PP} pin. In flash memory programming mode, a write voltage of 10.0 V (TYP.) is supplied to the V_{PP} pin, so perform either of the following.

(1) Connect a pull-down resistor (RV_{PP} = 10 kΩ) to the V_{PP} pin.
(2) Use the jumper on the board to switch the V_{PP} pin input to either the programmer or directly to GND.

A V_{PP} pin connection example is shown below.

Figure 19-4. V_{PP} Pin Connection Example



<Serial interface pin>

The following shows the pins used by the serial interface.

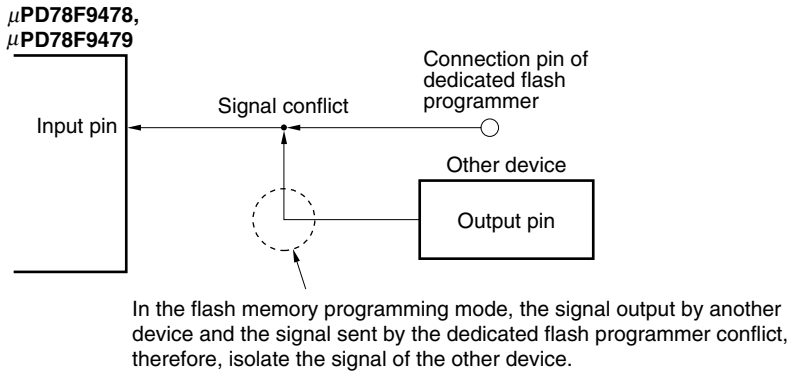
Serial Interface	Pins Used
3-wire serial I/O	SI20, SO20, $\overline{SCK20}$
3-wire serial I/O with handshake	SI20, SO20, $\overline{SCK20}$, P11 (HS)
UART	RxD20, TxD20

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.

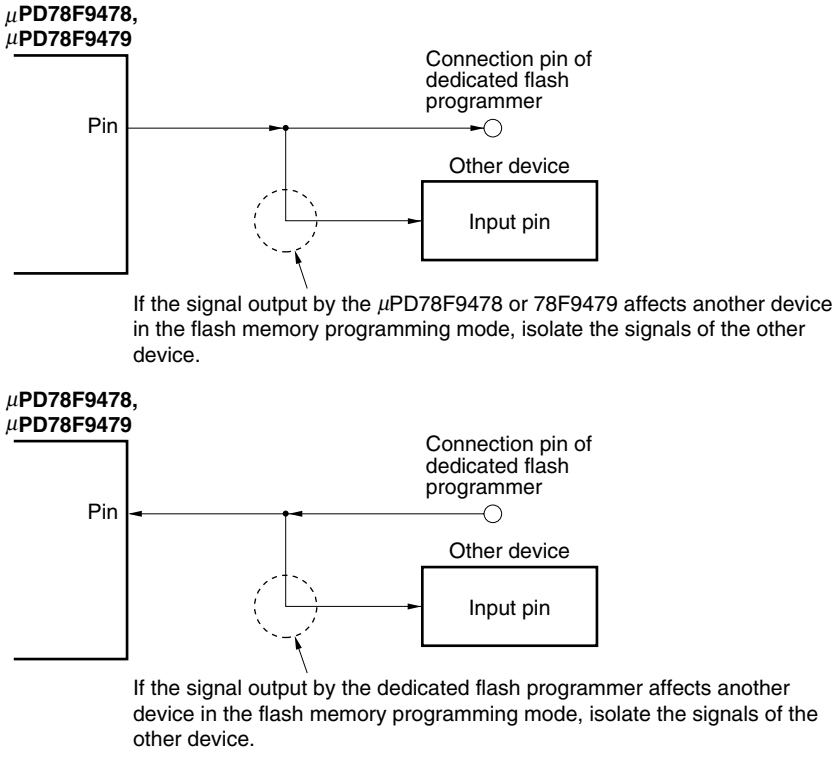
Figure 19-5. Signal Conflict (Input Pin of Serial Interface)



(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 19-6. Abnormal Operation of Other Device

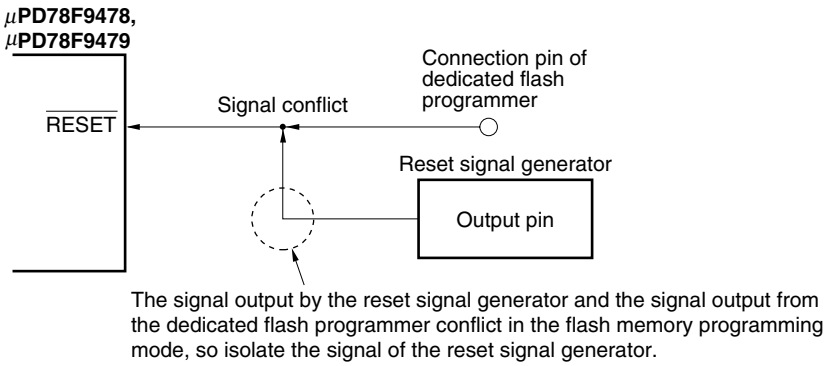


<$\overline{\text{RESET}}$ pin>

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 19-7. Signal Conflict ($\overline{\text{RESET}}$ Pin)



<Port pins>

When the μPD78F9478 enters the flash memory programming mode, all the pins other than those that communicate with flash programmer are in the same status as immediately after reset.

If the external device does not recognize initial statuses such as the output high impedance status, therefore, connect the external device to V_{DD} or V_{SS} via a resistor.

<Resonator>

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 resonator on-board, and leave the X2 pin open. The subsystem clock conforms to the normal operation mode.

<Power supply>

To use the power output from the flash programmer, connect the V_{DD} pin to VDD of the flash programmer, and V_{SS} pin to GND of the flash programmer, respectively.

To use the on-board power supply, make connection in accordance with the normal operation mode. However, because the voltage is monitored by the flash programmer, be sure to connect VDD of the flash programmer.

Supply the same power as in the normal operation mode to the other power pins (AV_{DD} and AV_{SS}).

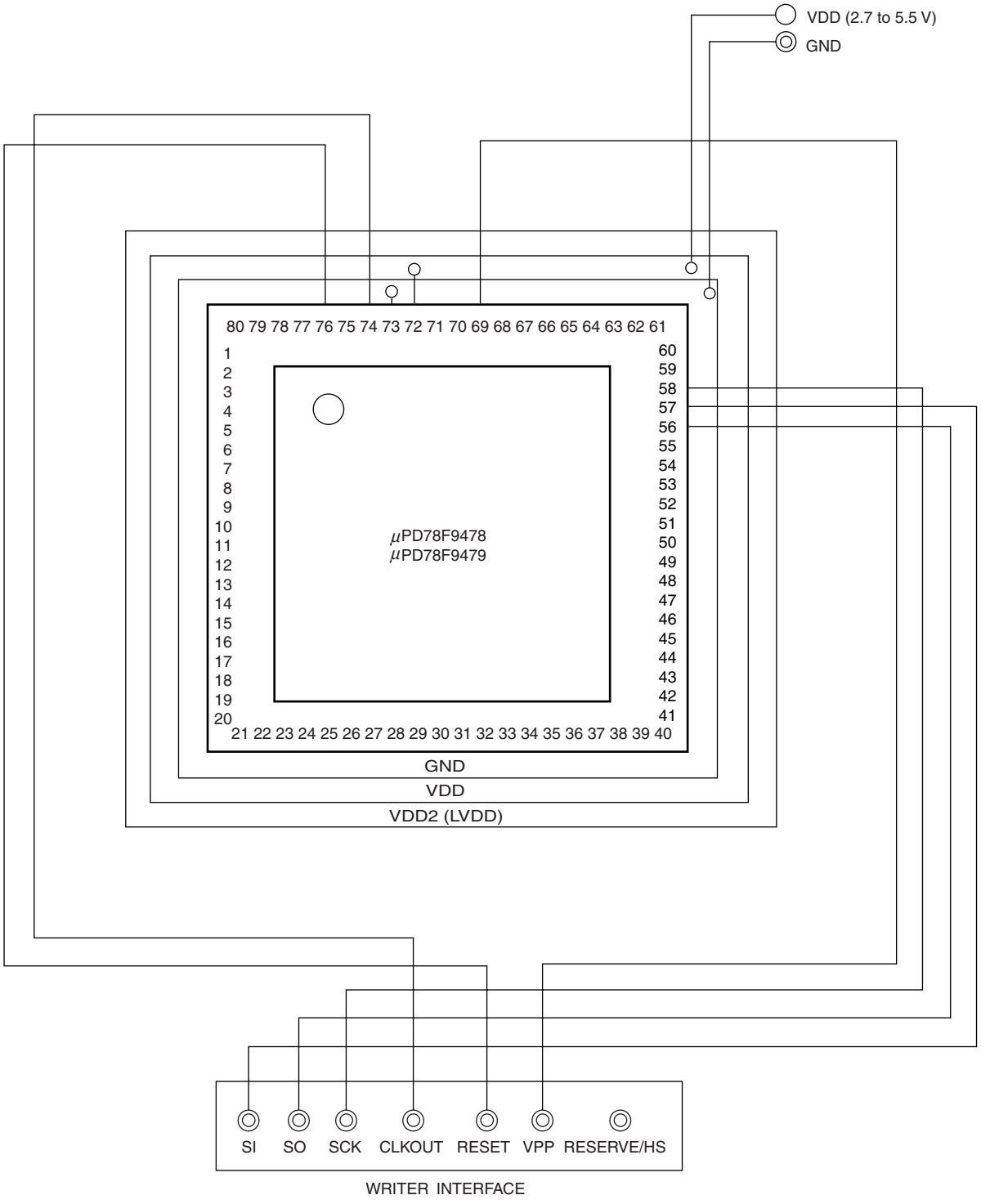
<Other pins>

Process the other pins (S0 to S27, COM0 to COM3, V_{LC0} to V_{LC2}, CAPH, and CAPL) in the same manner as in the normal operation mode.

19.1.4 Connection of adapter for flash writing

The following figure shows an example of recommended connection when the adapter for flash writing is used.

Figure 19-8. Wiring Example for Flash Writing Adapter with 3-Wire Serial I/O



<R> **Figure 19-9. Wiring Example for Flash Writing Adapter with 3-Wire Serial I/O with Handshake**

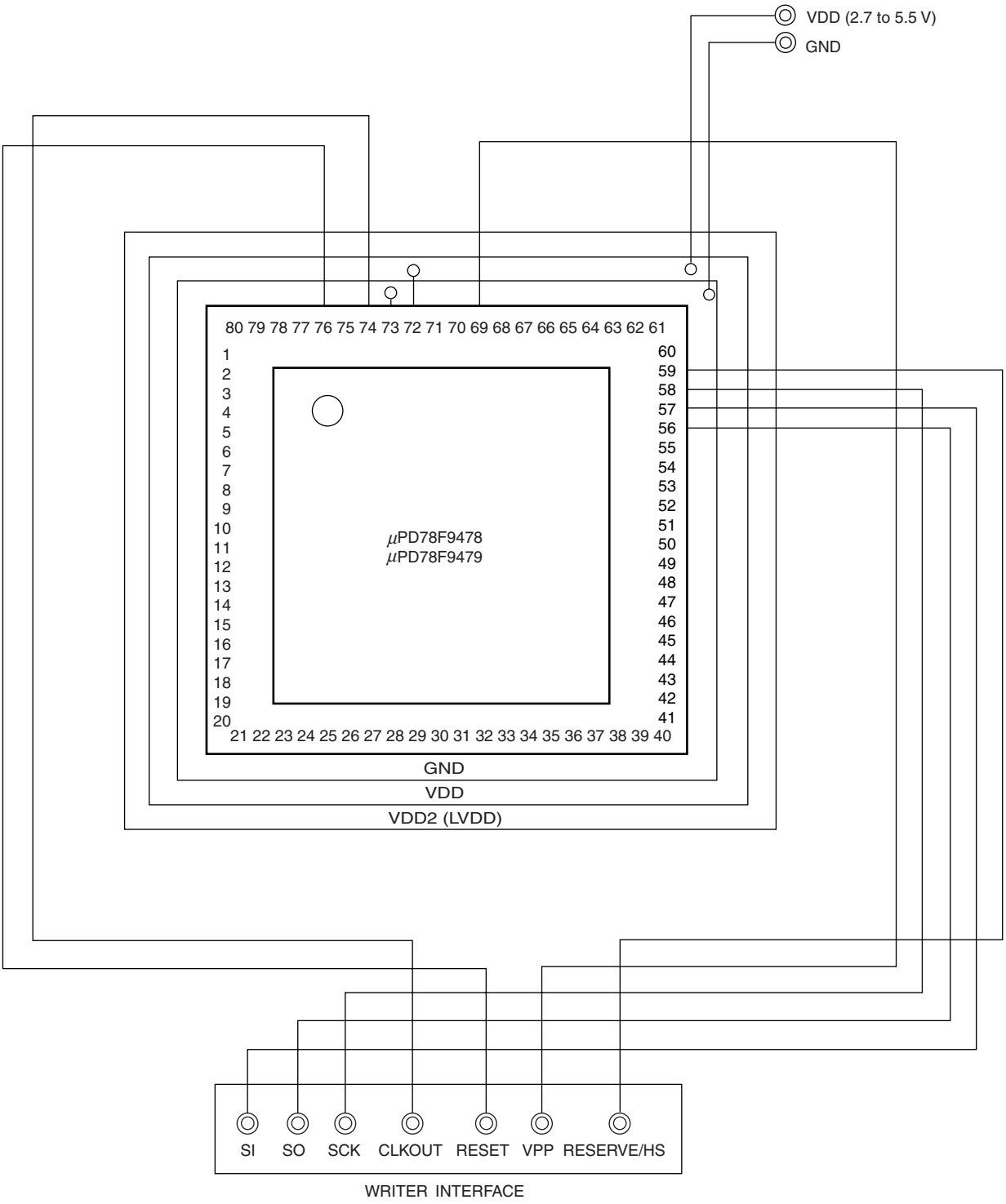
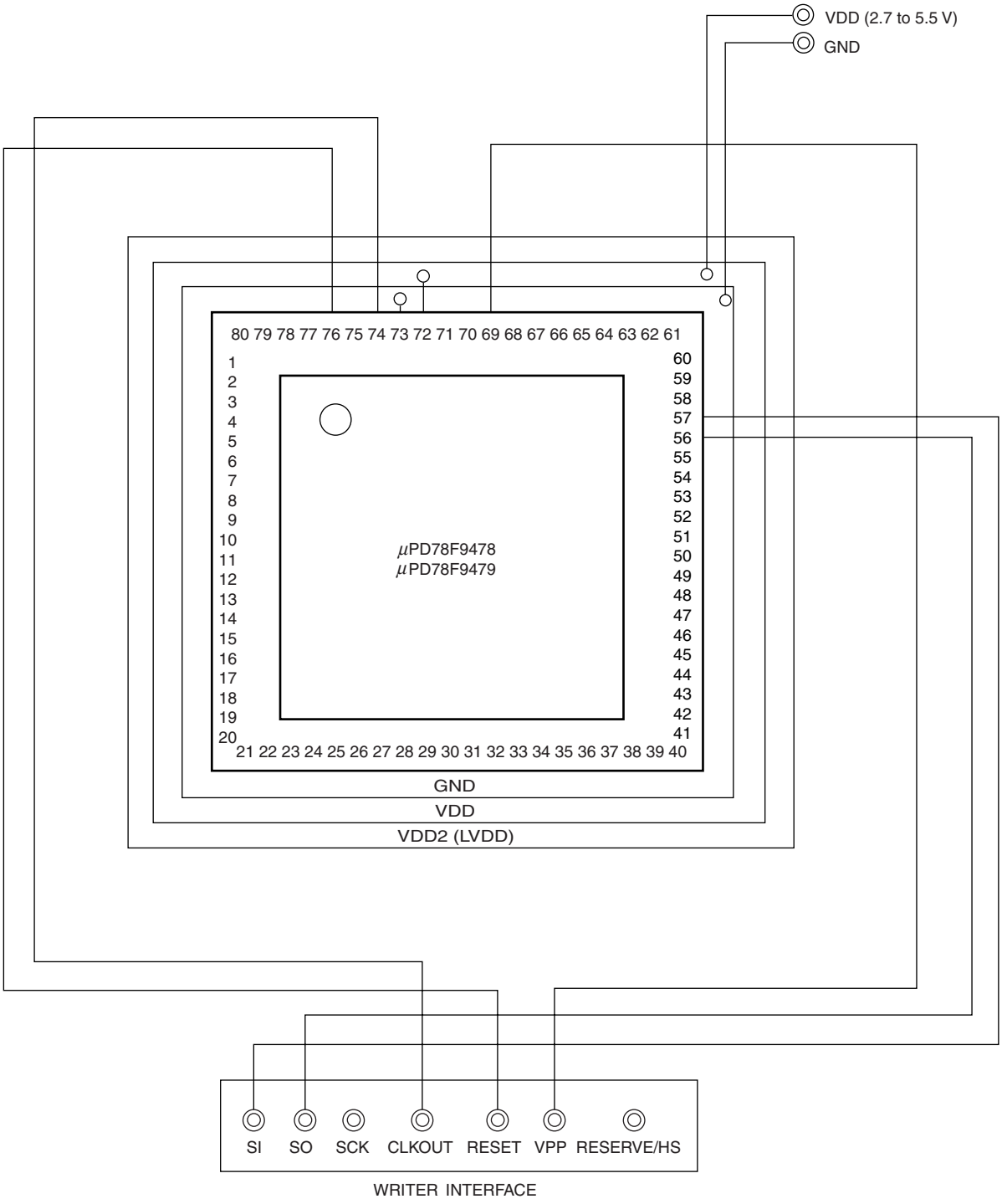


Figure 19-10. Wiring Example for Flash Writing Adapter with UART



19.2 Cautions on μPD78F9478 and 78F9479

(1) When using HALT mode with subclock multiplied by four

Observe the following constraints when using the flash version (μPD78F9478 and 78F9479) in the HALT mode with the subclock multiplied by 4 as the CPU clock.

- Be sure to insert the following number of NOP instructions immediately after the HALT instruction.

Operating Temperature	Number of NOP Instructions
T_A = −40 to +45°C	2
T_A = −40 to +80°C	3
T_A = −40 to +85°C	4

- Save the value of the A register to the internal high-speed RAM area before the HALT instruction is executed (because the value of the A register may be changed when the HALT mode is released).

CHAPTER 20 MASK OPTIONS

The μPD789477, 789478, and 789479 have the following mask options.

- Pin function
  The segment pins of the LCD and port 7 (input port) can be selected in 1-bit units.
  <1> S (16 + n)
  <2> P7n (n = 0 to 3)

  The segment pins of the LCD and port 8 (I/O port) can be selected in 1-bit units.
  <1> S (20 + m)
  <2> P8m (m = 0 to 7)

- Subsystem clock ×4 multiplication circuit
  The use of a circuit to multiply the subsystem clock (32.768 kHz) by 4 (131 kHz) is selected.
  <1> ×4 multiplication circuit is used
  <2> ×4 multiplication circuit is not used

- Pull-up resistor
  The connection of on-chip pull-up resistors for port 5 (I/O port) can be switched in 1-bit units.
  <1> Pull-up resistor is connected
  <2> Pull-up resistor is not connected

  Caution Mask options are not provided for flash memory products (μPD78F9478 and 78F9479).

CHAPTER 21 INSTRUCTION SET

This chapter lists the instruction set of the μPD789479 Subseries. For details of the operation and machine language (instruction code) of each instruction, refer to **78K/0S Series Instructions User's Manual (U11047E)**.

21.1 Operation

21.1.1 Operand identifiers and description methods

Operands are described in the "Operand" column of each instruction in accordance with the description method of the instruction operand identifier (refer to the assembler specifications for details). When there are two or more description methods, select one of them. Uppercase letters and the symbols #, !, $, and [ ] are keywords and are described as they are. Each symbol has the following meaning.

- #: Immediate data specification
- !: Absolute address specification
- $: Relative address specification
- [ ]: Indirect address specification

In the case of immediate data, describe an appropriate numeric value or a label. When using a label, be sure to describe the #, !, $ and [ ] symbols.

For operand register identifiers, r and rp, either functional names (X, A, C, etc.) or absolute names (names in parenthesis in the table below, R0, R1, R2, etc.) can be used for description.

Table 21-1. Operand Identifiers and Description Methods

Identifier	Description Method
r	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7)
rp	AX (RP0), BC (RP1), DE (RP2), HL (RP3)
sfr	Special function register symbol
saddr	FE20H to FF1FH Immediate data or labels
saddrp	FE20H to FF1FH Immediate data or labels (even addresses only)
addr16	0000H to FFFFH Immediate data or labels (only even addresses for 16-bit data transfer instructions)
addr5	0040H to 007FH Immediate data or labels (even addresses only)
word	16-bit immediate data or label
byte	8-bit immediate data or label
bit	3-bit immediate data or label

Remark See **Table 3-4 Special Function Registers** for symbols of special function registers.

21.1.2 Description of “Operation” column

A:	A register; 8-bit accumulator
X:	X register
B:	B register
C:	C register
D:	D register
E:	E register
H:	H register
L:	L register
AX:	AX register pair; 16-bit accumulator
BC:	BC register pair
DE:	DE register pair
HL:	HL register pair
PC:	Program counter
SP:	Stack pointer
PSW:	Program status word
CY:	Carry flag
AC:	Auxiliary carry flag
Z:	Zero flag
IE:	Interrupt request enable flag
( ):	Memory contents indicated by address or register contents in parenthesis
X_H, X_L:	Higher 8 bits and lower 8 bits of 16-bit register
$\wedge$:	Logical product (AND)
$\vee$:	Logical sum (OR)
V:	Exclusive logical sum (exclusive OR)
$\overline{\ }$:	Inverted data
addr16:	16-bit immediate data or label
jdisp8:	Signed 8-bit data (displacement value)

21.1.3 Description of “Flag” column

(Blank):	Unchanged
0:	Cleared to 0
1:	Set to 1
x:	Set/cleared according to the result
R:	Previously saved value is restored

21.2 Operation List

Mnemonic	Operands	Bytes	Clocks	Operation	Flag
					Z AC CY
MOV	r, #byte	3	6	r ← byte	
	saddr, #byte	3	6	(saddr) ← byte	
	sfr, #byte	3	6	sfr ← byte	
	A, r Note 1	2	4	A ← r	
	r, A Note 1	2	4	r ← A	
	A, saddr	2	4	A ← (saddr)	
	saddr, A	2	4	(saddr) ← A	
	A, sfr	2	4	A ← sfr	
	sfr, A	2	4	sfr ← A	
	A, !addr16	3	8	A ← (addr16)	
	!addr16, A	3	8	(addr16) ← A	
	PSW, #byte	3	6	PSW ← byte	× × ×
	A, PSW	2	4	A ← PSW	
	PSW, A	2	4	PSW ← A	× × ×
	A, [DE]	1	6	A ← (DE)	
	[DE], A	1	6	(DE) ← A	
	A, [HL]	1	6	A ← (HL)	
	[HL], A	1	6	(HL) ← A	
	A, [HL+byte]	2	6	A ← (HL + byte)	
	[HL+byte], A	2	6	(HL + byte) ← A	
XCH	A, X	1	4	A ↔ X	
	A, r Note 2	2	6	A ↔ r	
	A, saddr	2	6	A ↔ (saddr)	
	A, sfr	2	6	A ↔ sfr	
	A, [DE]	1	8	A ↔ (DE)	
	A, [HL]	1	8	A ↔ (HL)	
	A, [HL+byte]	2	8	A ↔ (HL + byte)	

Notes 1. Except r = A.

2. Except r = A, X.

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

Mnemonic	Operands	Bytes	Clocks	Operation	Flag
					Z AC CY
MOVW	rp, #word	3	6	rp ← word	
	AX, saddrp	2	6	AX ← (saddrp)	
	saddrp, AX	2	8	(saddrp) ← AX	
	AX, rp Note	1	4	AX ← rp	
	rp, AX Note	1	4	rp ← AX	
XCHW	AX, rp Note	1	8	AX ↔ rp	
ADD	A, #byte	2	4	A, CY ← A + byte	× × ×
	saddr, #byte	3	6	(saddr), CY ← (saddr) + byte	× × ×
	A, r	2	4	A, CY ← A + r	× × ×
	A, saddr	2	4	A, CY ← A + (saddr)	× × ×
	A, !addr16	3	8	A, CY ← A + (addr16)	× × ×
	A, [HL]	1	6	A, CY ← A + (HL)	× × ×
	A, [HL+byte]	2	6	A, CY ← A + (HL + byte)	× × ×
ADDC	A, #byte	2	4	A, CY ← A + byte + CY	× × ×
	saddr, #byte	3	6	(saddr), CY ← (saddr) + byte + CY	× × ×
	A, r	2	4	A, CY ← A + r + CY	× × ×
	A, saddr	2	4	A, CY ← A + (saddr) + CY	× × ×
	A, !addr16	3	8	A, CY ← A + (addr16) + CY	× × ×
	A, [HL]	1	6	A, CY ← A + (HL) + CY	× × ×
	A, [HL+byte]	2	6	A, CY ← A + (HL + byte) + CY	× × ×
SUB	A, #byte	2	4	A, CY ← A – byte	× × ×
	saddr, #byte	3	6	(saddr), CY ← (saddr) – byte	× × ×
	A, r	2	4	A, CY ← A – r	× × ×
	A, saddr	2	4	A, CY ← A – (saddr)	× × ×
	A, !addr16	3	8	A, CY ← A – (addr16)	× × ×
	A, [HL]	1	6	A, CY ← A – (HL)	× × ×
	A, [HL+byte]	2	6	A, CY ← A – (HL + byte)	× × ×

Note Only when rp = BC, DE, or HL.

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

Mnemonic	Operands	Bytes	Clocks	Operation	Flag		
					Z	AC	CY
SUBC	A, #byte	2	4	A, CY ← A – byte – CY	×	×	×
	saddr, #byte	3	6	(saddr), CY ← (saddr) – byte – CY	×	×	×
	A, r	2	4	A, CY ← A – r – CY	×	×	×
	A, saddr	2	4	A, CY ← A – (saddr) – CY	×	×	×
	A, !addr16	3	8	A, CY ← A – (addr16) – CY	×	×	×
	A, [HL]	1	6	A, CY ← A – (HL) – CY	×	×	×
	A, [HL+byte]	2	6	A, CY ← A – (HL + byte) – CY	×	×	×
AND	A, #byte	2	4	A ← A ∧ byte	×		
	saddr, #byte	3	6	(saddr) ← (saddr) ∧ byte	×		
	A, r	2	4	A ← A ∧ r	×		
	A, saddr	2	4	A ← A ∧ (saddr)	×		
	A, !addr16	3	8	A ← A ∧ (addr16)	×		
	A, [HL]	1	6	A ← A ∧ (HL)	×		
	A, [HL+byte]	2	6	A ← A ∧ (HL + byte)	×		
OR	A, #byte	2	4	A ← A ∨ byte	×		
	saddr, #byte	3	6	(saddr) ← (saddr) ∨ byte	×		
	A, r	2	4	A ← A ∨ r	×		
	A, saddr	2	4	A ← A ∨ (saddr)	×		
	A, !addr16	3	8	A ← A ∨ (addr16)	×		
	A, [HL]	1	6	A ← A ∨ (HL)	×		
	A, [HL+byte]	2	6	A ← A ∨ (HL + byte)	×		
XOR	A, #byte	2	4	A ← A $\overline{\vee}$ byte	×		
	saddr, #byte	3	6	(saddr) ← (saddr) $\overline{\vee}$ byte	×		
	A, r	2	4	A ← A $\overline{\vee}$ r	×		
	A, saddr	2	4	A ← A $\overline{\vee}$ (saddr)	×		
	A, !addr16	3	8	A ← A $\overline{\vee}$ (addr16)	×		
	A, [HL]	1	6	A ← A $\overline{\vee}$ (HL)	×		
	A, [HL+byte]	2	6	A ← A $\overline{\vee}$ (HL + byte)	×		

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

Mnemonic	Operands	Bytes	Clocks	Operation	Flag		
					Z	AC	CY
CMP	A, #byte	2	4	A – byte	×	×	×
	saddr, #byte	3	6	(saddr) – byte	×	×	×
	A, r	2	4	A – r	×	×	×
	A, saddr	2	4	A – (saddr)	×	×	×
	A, !addr16	3	8	A – (addr16)	×	×	×
	A, [HL]	1	6	A – (HL)	×	×	×
	A, [HL+byte]	2	6	A – (HL + byte)	×	×	×
ADDW	AX, #word	3	6	AX, CY ← AX + word	×	×	×
SUBW	AX, #word	3	6	AX, CY ← AX – word	×	×	×
CMPW	AX, #word	3	6	AX – word	×	×	×
INC	r	2	4	r ← r + 1	×	×	
	saddr	2	4	(saddr) ← (saddr) + 1	×	×	
DEC	r	2	4	r ← r – 1	×	×	
	saddr	2	4	(saddr) ← (saddr) – 1	×	×	
INCW	rp	1	4	rp ← rp + 1			
DECW	rp	1	4	rp ← rp – 1			
ROR	A, 1	1	2	(CY, A_7 ← A_0, A_{m-1} ← A_m) × 1			×
ROL	A, 1	1	2	(CY, A_0 ← A_7, A_{m+1} ← A_m) × 1			×
RORC	A, 1	1	2	(CY ← A_0, A_7 ← CY, A_{m-1} ← A_m) × 1			×
ROLC	A, 1	1	2	(CY ← A_7, A_0 ← CY, A_{m+1} ← A_m) × 1			×
SET1	saddr.bit	3	6	(saddr.bit) ← 1			
	sfr.bit	3	6	sfr.bit ← 1			
	A.bit	2	4	A.bit ← 1			
	PSW.bit	3	6	PSW.bit ← 1	×	×	×
	[HL].bit	2	10	(HL).bit ← 1			
CLR1	saddr.bit	3	6	(saddr.bit) ← 0			
	sfr.bit	3	6	sfr.bit ← 0			
	A.bit	2	4	A.bit ← 0			
	PSW.bit	3	6	PSW.bit ← 0	×	×	×
	[HL].bit	2	10	(HL).bit ← 0			
SET1	CY	1	2	CY ← 1			1
CLR1	CY	1	2	CY ← 0			0
NOT1	CY	1	2	CY ← $\overline{CY}$			×

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

Mnemonic	Operands	Bytes	Clocks	Operation	Flag
					Z AC CY
CALL	!addr16	3	6	$(SP-1) \leftarrow (PC+3)_H$, $(SP-2) \leftarrow (PC+3)_L$, $PC \leftarrow$ addr16, $SP \leftarrow SP-2$	
CALLT	[addr5]	1	8	$(SP-1) \leftarrow (PC+1)_H$, $(SP-2) \leftarrow (PC+1)_L$, $PC_H \leftarrow$ (00000000, addr5 + 1), $PC_L \leftarrow$ (00000000, addr5), $SP \leftarrow SP-2$	
RET		1	6	$PC_H \leftarrow (SP+1)$, $PC_L \leftarrow (SP)$, $SP \leftarrow SP+2$	
RETI		1	8	$PC_H \leftarrow (SP+1)$, $PC_L \leftarrow (SP)$, $PSW \leftarrow (SP+2)$, $SP \leftarrow SP+3$	R R R
PUSH	PSW	1	2	$(SP-1) \leftarrow PSW$, $SP \leftarrow SP-1$	
	rp	1	4	$(SP-1) \leftarrow rp_H$, $(SP-2) \leftarrow rp_L$, $SP \leftarrow SP-2$	
POP	PSW	1	4	$PSW \leftarrow (SP)$, $SP \leftarrow SP+1$	R R R
	rp	1	6	$rp_H \leftarrow (SP+1)$, $rp_L \leftarrow (SP)$, $SP \leftarrow SP+2$	
MOVW	SP, AX	2	8	$SP \leftarrow AX$	
	AX, SP	2	6	$AX \leftarrow SP$	
BR	!addr16	3	6	$PC \leftarrow$ addr16	
	$addr16	2	6	$PC \leftarrow PC+2+$ jdisp8	
	AX	1	6	$PC_H \leftarrow A$, $PC_L \leftarrow X$	
BC	$saddr16	2	6	$PC \leftarrow PC+2+$ jdisp8 if CY = 1	
BNC	$saddr16	2	6	$PC \leftarrow PC+2+$ jdisp8 if CY = 0	
BZ	$saddr16	2	6	$PC \leftarrow PC+2+$ jdisp8 if Z = 1	
BNZ	$saddr16	2	6	$PC \leftarrow PC+2+$ jdisp8 if Z = 0	
BT	saddr.bit, $addr16	4	10	$PC \leftarrow PC+4+$ jdisp8 if (saddr.bit) = 1	
	sfr.bit, $addr16	4	10	$PC \leftarrow PC+4+$ jdisp8 if sfr.bit = 1	
	A.bit, $addr16	3	8	$PC \leftarrow PC+3+$ jdisp8 if A.bit = 1	
	PSW.bit, $addr16	4	10	$PC \leftarrow PC+4+$ jdisp8 if PSW.bit = 1	
BF	saddr.bit, $addr16	4	10	$PC \leftarrow PC+4+$ jdisp8 if (saddr.bit) = 0	
	sfr.bit, $addr16	4	10	$PC \leftarrow PC+4+$ jdisp8 if sfr.bit = 0	
	A.bit, $addr16	3	8	$PC \leftarrow PC+3+$ jdisp8 if A.bit = 0	
	PSW.bit, $addr16	4	10	$PC \leftarrow PC+4+$ jdisp8 if PSW.bit = 0	
DBNZ	B, $addr16	2	6	$B \leftarrow B-1$, then $PC \leftarrow PC+2+$ jdisp8 if $B \neq 0$	
	C, $addr16	2	6	$C \leftarrow C-1$, then $PC \leftarrow PC+2+$ jdisp8 if $C \neq 0$	
	saddr, $addr16	3	8	$(saddr) \leftarrow (saddr)-1$, then $PC \leftarrow PC+3+$ jdisp8 if $(saddr) \neq 0$	
NOP		1	2	No Operation	
EI		3	6	$IE \leftarrow 1$ (Enable interrupt)	
DI		3	6	$IE \leftarrow 0$ (Disable interrupt)	
HALT		1	2	Set HALT mode	
STOP		1	2	Set STOP mode	

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

21.3 Instructions Listed by Addressing Type

(1) 8-bit instructions

MOV, XCH, ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP, INC, DEC, ROR, ROL, RORC, ROLC, PUSH, POP, DBNZ

1st Operand \ 2nd Operand	#byte	A	r	sfr	saddr	!addr16	PSW	[DE]	[HL]	[HL+byte]	$addr16	1	None
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV[Note] XCH[Note] ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP		ROR ROL RORC ROLC	
r	MOV	MOV											INC DEC
B, C											DBNZ		
sfr	MOV	MOV											
saddr	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV									DBNZ		INC DEC
!addr16		MOV											
PSW	MOV	MOV											PUSH POP
[DE]		MOV											
[HL]		MOV											
[HL+byte]		MOV											

Note Except r = A.

(2) 16-bit instructions

MOVW, XCHW, ADDW, SUBW, CMPW, PUSH, POP, INCW, DECW

1st Operand \ 2nd Operand	#word	AX	rp[Note]	saddrp	SP	None
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	
rp	MOVW	MOVW[Note]				INCW DECW PUSH POP
saddrp		MOVW				
sp		MOVW				

Note Only when rp = BC, DE, or HL.

(3) Bit manipulation instructions

SET1, CLR1, NOT1, BT, BF

1st Operand \ 2nd Operand	$addr16	None
A.bit	BT BF	SET1 CLR1
sfr.bit	BT BF	SET1 CLR1
saddr.bit	BT BF	SET1 CLR1
PSW.bit	BT BF	SET1 CLR1
[HL].bit		SET1 CLR1
CY		SET1 CLR1 NOT1

(4) Call instructions/branch instructions

CALL, CALLT, BR, BC, BNC, BZ, BNZ, DBNZ

1st Operand \ 2nd Operand	AX	!addr16	[addr5]	$addr16
Basic Instructions	BR	CALL BR	CALLT	BR BC BNC BZ BNZ
Compound Instructions				DBNZ

(5) Other instructions

RET, RETI, NOP, EI, DI, HALT, STOP

<R>

CHAPTER 22 ELECTRICAL SPECIFICATIONS

Absolute Maximum Ratings (T_A = 25°C)

Parameter	Symbol	Conditions		Ratings	Unit
Power supply voltage	V_{DD}	V_{DD} = AV_{DD}		–0.3 to +6.5	V
	AV_{DD}				
	V_{PP}	μPD78F9478, 78F9479 only[Note 1]		–0.3 to +10.5	V
Input voltage	V_{I1}	P00 to P07, P10, P11, P20 to P25, P30 to P34, P60 to P67, P70 to P73[Note 2], P80 to P87[Note 2], X1, X2, XT1, XT2, $\overline{\text{RESET}}$		–0.3 to V_{DD} + 0.3[Note 3]	V
	V_{I2}	P50 to P53	N-ch open drain	–0.3 to +13	V
			On-chip pull-up resistor	–0.3 to V_{DD} + 0.3[Note 3]	V
Output voltage	V_O	P00 to P07, P10, P11, P20 to P25, P30 to P34, P50 to P53, P80 to P87[Note 2]		–0.3 to V_{DD} + 0.3[Note 3]	V
		S0 to S15, S16 to S27[Note 2], COM0 to COM3		–0.3 to V_{LC0} + 0.3	V
Output current, high	I_{OH}	Per pin		–10	mA
		Total for all pins		–30	mA
Output current, low	I_{OL}	Per pin		30	mA
		Total for all pins		160	mA
Operating ambient temperature	T_A	Normal operation		–40 to +85	°C
		Flash memory programming		10 to 40	°C
Storage temperature	T_{stg}	μPD789477, 789478, 789479		–65 to +150	°C
		μPD78F9478, 78F9479		–40 to +125	°C

(See the next page for a description of the notes.)

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 otherwise specified, the characteristics of alternate-function pins are the same as those of port pins.

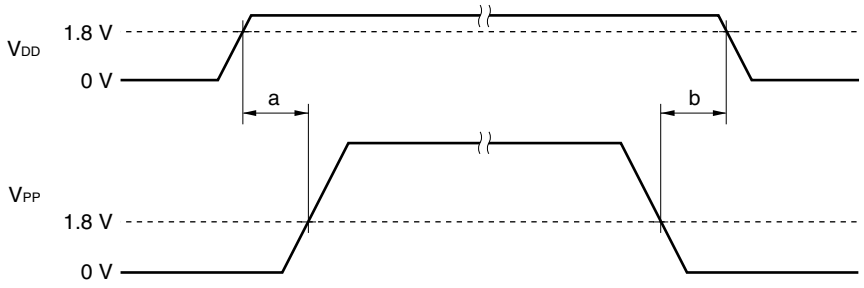
Notes 1. Make sure that the following conditions of the V_{PP} voltage application timing are satisfied when the flash memory is written.

- **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**

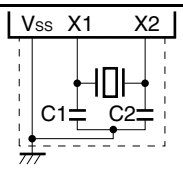
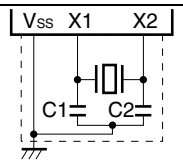
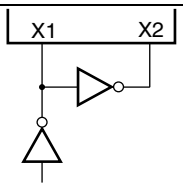
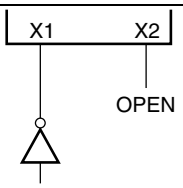
  V_{DD} must be lowered 10 μs or more after V_{PP} falls below the lower-limit value (1.8 V) of the operating voltage range of V_{DD} (see b in the figure below).



2. Only when selected by a mask option or port function register

3. 6.5 V or less

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]		1.0		5.0	MHz
		Oscillation stabilization time[Note 2]	After V_{DD} reaches oscillation voltage range MIN.			4	ms
Crystal resonator		Oscillation frequency(f_X)[Note 1]		1.0		5.0	MHz
		Oscillation stabilization time[Note 2]	V_{DD} = 4.5 to 5.5 V			10	ms
			V_{DD} = 1.8 to 5.5 V			30	ms
External clock		X1 input frequency (f_X)[Note 1]		1.0		5.0	MHz
		X1 input high-/low-level width (t_{XH}, t_{XL})		85		500	ns
		X1 input frequency (f_X)[Note 1]	V_{DD} = 2.7 to 5.5 V	1.0		5.0	MHz
		X1 input high-/low-level width (t_{XH}, t_{XL})	V_{DD} = 2.7 to 5.5 V	85		500	ns

Notes 1. Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

2. Time required to stabilize oscillation after reset or STOP mode release.

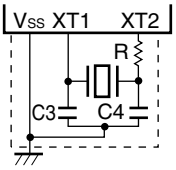
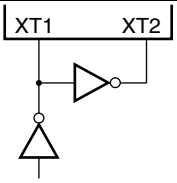
Cautions 1. When using the main system clock oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

- **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 signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as V_{SS}.**
- **Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

2. 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.

Remark For the resonator selection and oscillator constant, customers are required 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		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	
External clock		XT1 input frequency (f_{XT})Note 1		32		35	kHz
		XT1 input high-/low-level width (t_{XTH}, t_{XTL})		14.3		15.6	μs

Notes 1. Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

2. Time required to stabilize oscillation after V_{DD} reaches oscillation voltage range MIN.

Cautions 1. When using the subsystem clock oscillator, wire as follows in the area enclosed by the broken lines in the above figure to avoid an adverse effect from wiring capacitance.

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

2. The subsystem clock oscillator is designed as a low-amplitude circuit for reducing power 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 required to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

DC Characteristics (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V) (1/6)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Output current, low	I_{OL}	Per pin					10	mA
		All pins					80	mA
Output current, high	I_{OH}	Per pin					–1	mA
		All pins					–15	mA
Input voltage, high	V_{IH1}	P10, P11, P60 to P67		V_{DD} = 2.7 to 5.5 V	0.7V_{DD}		V_{DD}	V
				V_{DD} = 1.8 to 5.5 V	0.9V_{DD}		V_{DD}	V
	V_{IH2}	P50 to P53	N-ch open drain	V_{DD} = 2.7 to 5.5 V	0.7V_{DD}		12	V
				V_{DD} = 1.8 to 5.5 V	0.9V_{DD}		12	V
			On-chip pull-up resistor[Note 1]	V_{DD} = 2.7 to 5.5 V	0.7V_{DD}		V_{DD}	V
				V_{DD} = 1.8 to 5.5 V	0.9V_{DD}		V_{DD}	V
	V_{IH3}	$\overline{\text{RESET}}$, P00 to P07, P20 to P25, P30 to P34, P70 to P73[Note 2] P80 to P87[Note 2]		V_{DD} = 2.7 to 5.5 V	0.8V_{DD}		V_{DD}	V
				V_{DD} = 1.8 to 5.5 V	0.9V_{DD}		V_{DD}	V
	V_{IH4}	X1, X2, XT1, XT2		V_{DD} = 4.5 to 5.5 V	V_{DD} – 0.5		V_{DD}	V
				V_{DD} = 1.8 to 5.5 V	V_{DD} – 0.1		V_{DD}	V
Input voltage, low	V_{IL1}	P10, P11, P60 to P67		V_{DD} = 2.7 to 5.5 V	0		0.3V_{DD}	V
				V_{DD} = 1.8 to 5.5 V	0		0.1V_{DD}	V
	V_{IL2}	P50 to P53		V_{DD} = 2.7 to 5.5 V	0		0.3V_{DD}	V
				V_{DD} = 1.8 to 5.5 V	0		0.1V_{DD}	V
	V_{IL3}	$\overline{\text{RESET}}$, P00 to P07, P20 to P25, P30 to P34, P70 to P73[Note 2], P80 to P87[Note 2]		V_{DD} = 2.7 to 5.5 V	0		0.2V_{DD}	V
				V_{DD} = 1.8 to 5.5 V	0		0.1V_{DD}	V
	V_{IL4}	X1, X2, XT1, XT2		V_{DD} = 4.5 to 5.5 V	0		0.4	V
				V_{DD} = 1.8 to 5.5 V	0		0.1	V
Output voltage, high	V_{OH}	V_{DD} = 4.5 to 5.5 V, I_{OH} = –1 mA			V_{DD} – 1.0			V
		V_{DD} = 1.8 to 5.5 V, I_{OH} = –100 μA			V_{DD} – 0.5			V
Output voltage, low	V_{OL1}	P00 to P07, P10, P11, P20 to P25, P30 to P34, P80 to P87[Note 2]		4.5 ≤ V_{DD} ≤ 5.5 V, I_{OL} = 10 mA			1.0	V
				1.8 ≤ V_{DD} < 4.5 V, I_{OL} = 400 μA			0.5	V
	V_{OL2}	P50 to P53		4.5 ≤ V_{DD} ≤ 5.5 V, I_{OL} = 10 mA			1.0	V
				1.8 ≤ V_{DD} < 4.5 V, I_{OL} = 1.6 mA			0.4	V

Notes 1. μPD789477, 789478, and 789479 only

2. Only when selected by a mask option or port function register

Remark Unless otherwise specified, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V) (2/6)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	$V_I = V_{DD}$	P00 to P07, P10, P11, P20 to P25, P30 to P34, P60 to P67, P70 to P73[Note 1], P80 to P87[Note 1], $\overline{RESET}$			3	μA
	I_{LIH2}		X1, X2, XT1, XT2			20	μA
	I_{LIH3}	V_I = 12 V	P50 to P53 (N-ch open drain)			20	μA
Input leakage current, low	I_{LIL1}	V_I = 0 V	P00 to P07, P10, P11, P20 to P25, P30 to P34, P60 to P67, P70 to P73[Note 1], P80 to P87[Note 1], $\overline{RESET}$			–3	μA
	I_{LIL2}		X1, X2, XT1, XT2			–20	μA
	I_{LIL3}		P50 to P53 (N-ch open drain)			–3[Note 2]	μA
Output leakage current, high	I_{LOH}	$V_O = V_{DD}$				3	μA
Output leakage current, low	I_{LOL}	V_O = 0 V				–3	μA
Software pull-up resistor	R_1	V_I = 0 V	P00 to P07, P10, P11, P20 to P25, P30 to P34	50	100	200	kΩ
Mask option pull-up resistor[Note 3]	R_2	V_I = 0 V	P50 to P53	10	30	60	kΩ

Notes 1. Only when selected by a mask option or port function register

2. If there is no on-chip pull-up resistor for P50 to P53 (specified by a mask option) and if P50 to P53 have been set to input mode when a read instruction is executed to read from P50 to P53, a low-level input leakage current of up to –60 μA flows during only one cycle. At all other times, the maximum leakage current is –3 μA.

3. Mask ROM version only

Remark Unless otherwise specified, 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) (3/6)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Power supply current[Note 1] (μPD789477, 789478)	I_{DD1}	5.0 MHz crystal oscillation operation mode (C1 = C2 = 22 pF)		V_{DD} = 5.0 V ±10%[Note 2]		2	3.5	mA
				V_{DD} = 3.0 V ±10%[Note 3]		0.4	1	mA
				V_{DD} = 2.0 V ±10%[Note 3]		0.2	0.5	mA
	I_{DD2}	5.0 MHz crystal oscillation HALT mode (C1 = C2 = 22 pF)		V_{DD} = 5.0 V ±10%[Note 2]		0.96	1.92	mA
				V_{DD} = 3.0 V ±10%[Note 3]		0.26	0.76	mA
				V_{DD} = 2.0 V ±10%[Note 3]		0.1	0.34	mA
	I_{DD3}	32.768 kHz crystal oscillation operation mode[Note 4] (C3 = C4 = 22 pF, R1 = 220 kΩ)		V_{DD} = 5.0 V ±10%		33	67	μA
				V_{DD} = 3.0 V ±10%		10	31	μA
				V_{DD} = 2.0 V ±10%		5	16	μA
		32.768 kHz crystal oscillation operation × 4 multiplication operation mode[Note 4] (C3 = C4 = 22 pF, R1 = 220 kΩ)		V_{DD} = 5.0 V ±10%		130	200	μA
				V_{DD} = 3.0 V ±10%		50	110	μA
	I_{DD4}	32.768 kHz crystal oscillation HALT mode[Note 4] (C3 = C4 = 22 pF, R1 = 220 kΩ)	LCD not operating[Note 5]	V_{DD} = 5.0 V ±10%		25	60	μA
				V_{DD} = 3.0 V ±10%		8	28	μA
				V_{DD} = 2.0 V ±10%		5	13	μA
			LCD operating[Note 6]	V_{DD} = 5.0 V ±10%		27	66	μA
				V_{DD} = 3.0 V ±10%		9.8	33	μA
				V_{DD} = 2.0 V ±10%		6.6	17	μA
		32.768 kHz crystal oscillation × 4 multiplication HALT mode[Note 4] (C3 = C4 = 22 pF, R1 = 220 kΩ)	LCD not operating[Note 5]	V_{DD} = 5.0 V ±10%		25	60	μA
				V_{DD} = 3.0 V ±10%		8	28	μA
			LCD operating[Note 6]	V_{DD} = 5.0 V ±10%		27	66	μA
				V_{DD} = 3.0 V ±10%		9.8	33	μA
	I_{DD5}	STOP mode[Note 5]		V_{DD} = 5.0 V ±10%		0.1	10	μA
				V_{DD} = 3.0 V ±10%		0.05	5	μA
				V_{DD} = 2.0 V ±10%		0.05	3	μA
	I_{DD6}	5.0 MHz crystal oscillation A/D operating mode[Note 7] (C1 = C2 = 22 pF)		V_{DD} = 5.0 V ±10%[Note 2]		3	5.2	mA
				V_{DD} = 3.0 V ±10%[Note 3]		1.1	2	mA
				V_{DD} = 2.0 V ±10%[Note 3]		0.7	1.5	mA

Notes 1. The port current (including the current that flows to on-chip pull-up resistors) is not included.

2. High-speed mode operation (when the processor clock control register (PCC) is set to 00H)

3. Low-speed mode operation (when PCC is set to 02H)

4. When the main system clock is stopped

5. When the LCD is not operating (LCDON0 = 0, LIPS0 = 0)

6. Then the LCD is operating (LCDON0 = 1, LIPS0 = 1)

7. This is the total current that flows to V_{DD} and AV_{DD}.

Remark Unless otherwise specified, 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) (4/6)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Power supply current[Note 1] (μPD78F9477, 789478)	I_{DD1}	5.0 MHz crystal oscillation operation mode (C1 = C2 = 22 pF)		V_{DD} = 5.0 V ±10%[Note 2]		5.5	9.0	mA
				V_{DD} = 3.0 V ±10%[Note 3]		1.3	2.3	mA
				V_{DD} = 2.0 V ±10%[Note 3]		0.8	1.6	mA
	I_{DD2}	5.0 MHz crystal oscillation HALT mode (C1 = C2 = 22 pF)		V_{DD} = 5.0 V ±10%[Note 2]		1.5	2.1	mA
				V_{DD} = 3.0 V ±10%[Note 3]		0.41	0.85	mA
				V_{DD} = 2.0 V ±10%[Note 3]		0.2	0.43	mA
	I_{DD3}	32.768 kHz crystal oscillation operation mode[Note 4] (C3 = C4 = 22 pF, R1 = 220 kΩ)		V_{DD} = 5.0 V ±10%		115	200	μA
				V_{DD} = 3.0 V ±10%		85	140	μA
				V_{DD} = 2.0 V ±10%		70	110	μA
		32.768 kHz crystal oscillation operation × 4 multiplication operation mode[Note 4] (C3 = C4 = 22 pF, R1 = 220 kΩ)		V_{DD} = 5.0 V ±10%		315	480	μA
				V_{DD} = 3.0 V ±10%		200	300	μA
	I_{DD4}	32.768 kHz crystal oscillation HALT mode[Note 4] (C3 = C4 = 22 pF, R1 = 220 kΩ)	LCD not operating[Note 5]	V_{DD} = 5.0 V ±10%		25	65	μA
				V_{DD} = 3.0 V ±10%		7	29	μA
				V_{DD} = 2.0 V ±10%		4	20	μA
			LCD operating[Note 6]	V_{DD} = 5.0 V ±10%		27	71	μA
				V_{DD} = 3.0 V ±10%		8.8	34	μA
				V_{DD} = 2.0 V ±10%		5.6	24	μA
		32.768 kHz crystal oscillation × 4 multiplication HALT mode[Note 4] (C3 = C4 = 22 pF, R1 = 220 kΩ)	LCD not operating[Note 5]	V_{DD} = 5.0 V ±10%		25	65	μA
				V_{DD} = 3.0 V ±10%		7	29	μA
			LCD operating[Note 6]	V_{DD} = 5.0 V ±10%		27	71	μA
				V_{DD} = 3.0 V ±10%		8.8	34	μA
	I_{DD5}	STOP mode[Note 5]		V_{DD} = 5.0 V ±10%		0.1	10	μA
				V_{DD} = 3.0 V ±10%		0.05	5	μA
				V_{DD} = 2.0 V ±10%		0.05	3	μA
	I_{DD6}	5.0 MHz crystal oscillation A/D operating mode[Note 7] (C1 = C2 = 22 pF)		V_{DD} = 5.0 V ±10%[Note 2]		6.5	10.2	mA
				V_{DD} = 3.0 V ±10%[Note 3]		2.0	3.3	mA
				V_{DD} = 2.0 V ±10%[Note 3]		1.3	2.6	mA

Notes 1. The port current (including the current that flows to on-chip pull-up resistors) is not included.

2. High-speed mode operation (when the processor clock control register (PCC) is set to 00H)

3. Low-speed mode operation (when PCC is set to 02H)

4. When the main system clock is stopped

5. When the LCD is not operating (LCDON0 = 0, LIPS0 = 0)

6. Then the LCD is operating (LCDON0 = 1, LIPS0 = 1)

7. This is the total current that flows to V_{DD} and AV_{DD}.

Remark Unless otherwise specified, 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) (5/6)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Power supply current[Note 1] (μPD789479)	I_{DD1}	5.0 MHz crystal oscillation operation mode (C1 = C2 = 22 pF)		V_{DD} = 5.0 V ±10%[Note 2]		2.5	5.0	mA
				V_{DD} = 3.0 V ±10%[Note 3]		0.5	1.2	mA
				V_{DD} = 2.0 V ±10%[Note 3]		0.3	0.6	mA
	I_{DD2}	5.0 MHz crystal oscillation HALT mode (C1 = C2 = 22 pF)		V_{DD} = 5.0 V ±10%[Note 2]		1.0	2.0	mA
				V_{DD} = 3.0 V ±10%[Note 3]		0.35	0.8	mA
				V_{DD} = 2.0 V ±10%[Note 3]		0.2	0.4	mA
	I_{DD3}	32.768 kHz crystal oscillation operation mode[Note 4] (C3 = C4 = 22 pF, R1 = 220 kΩ)		V_{DD} = 5.0 V ±10%		38	100	μA
				V_{DD} = 3.0 V ±10%		13	50	μA
				V_{DD} = 2.0 V ±10%		7	25	μA
		32.768 kHz crystal oscillation operation × 4 multiplication operation mode[Note 4] (C3 = C4 = 22 pF, R1 = 220 kΩ)		V_{DD} = 5.0 V ±10%		150	250	μA
				V_{DD} = 3.0 V ±10%		75	160	μA
	I_{DD4}	32.768 kHz crystal oscillation HALT mode[Note 4] (C3 = C4 = 22 pF, R1 = 220 kΩ)	LCD not operating[Note 5]	V_{DD} = 5.0 V ±10%		25	70	μA
				V_{DD} = 3.0 V ±10%		8	32	μA
				V_{DD} = 2.0 V ±10%		5	15	μA
			LCD operating[Note 6]	V_{DD} = 5.0 V ±10%		27	76	μA
				V_{DD} = 3.0 V ±10%		9.8	37	μA
				V_{DD} = 2.0 V ±10%		6.6	24	μA
		32.768 kHz crystal oscillation × 4 multiplication HALT mode[Note 4] (C3 = C4 = 22 pF, R1 = 220 kΩ)	LCD not operating[Note 5]	V_{DD} = 5.0 V ±10%		25	70	μA
				V_{DD} = 3.0 V ±10%		8	32	μA
			LCD operating[Note 6]	V_{DD} = 5.0 V ±10%		27	76	μA
				V_{DD} = 3.0 V ±10%		9.8	37	μA
	I_{DD5}	STOP mode[Note 5]		V_{DD} = 5.0 V ±10%		0.1	10	μA
				V_{DD} = 3.0 V ±10%		0.05	5	μA
				V_{DD} = 2.0 V ±10%		0.05	3	μA
	I_{DD6}	5.0 MHz crystal oscillation A/D operating mode[Note 7] (C1 = C2 = 22 pF)		V_{DD} = 5.0 V ±10%[Note 2]		5.0	6.7	mA
				V_{DD} = 3.0 V ±10%[Note 3]		1.5	2.2	mA
				V_{DD} = 2.0 V ±10%[Note 3]		0.8	1.6	mA

Notes 1. The port current (including the current that flows to on-chip pull-up resistors) is not included.

2. High-speed mode operation (when the processor clock control register (PCC) is set to 00H)

6. Low-speed mode operation (when PCC is set to 02H)

7. When the main system clock is stopped

8. When the LCD is not operating (LCDON0 = 0, LIPS0 = 0)

6. Then the LCD is operating (LCDON0 = 1, LIPS0 = 1)

7. This is the total current that flows to V_{DD} and AV_{DD}.

Remark Unless otherwise specified, 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) (6/6)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Power supply current[Note 1] (μPD78F9479)	I_{DD1}	5.0 MHz crystal oscillation operation mode (C1 = C2 = 22 pF)		V_{DD} = 5.0 V ±10%[Note 2]		6.0	12.0	mA
				V_{DD} = 3.0 V ±10%[Note 3]		1.6	3.2	mA
				V_{DD} = 2.0 V ±10%[Note 3]		1.0	2.5	mA
	I_{DD2}	5.0 MHz crystal oscillation HALT mode (C1 = C2 = 22 pF)		V_{DD} = 5.0 V ±10%[Note 2]		1.6	3.0	mA
				V_{DD} = 3.0 V ±10%[Note 3]		0.5	1.2	mA
				V_{DD} = 2.0 V ±10%[Note 3]		0.3	0.6	mA
	I_{DD3}	32.768 kHz crystal oscillation operation mode[Note 4] (C3 = C4 = 22 pF, R1 = 220 kΩ)		V_{DD} = 5.0 V ±10%		130	250	μA
				V_{DD} = 3.0 V ±10%		90	180	μA
				V_{DD} = 2.0 V ±10%		80	160	μA
		32.768 kHz crystal oscillation operation × 4 multiplication operation mode[Note 4] (C3 = C4 = 22 pF, R1 = 220 kΩ)		V_{DD} = 5.0 V ±10%		330	550	μA
				V_{DD} = 3.0 V ±10%		250	400	μA
	I_{DD4}	32.768 kHz crystal oscillation HALT mode[Note 4] (C3 = C4 = 22 pF, R1 = 220 kΩ)	LCD not operating[Note 5]	V_{DD} = 5.0 V ±10%		25	70	μA
				V_{DD} = 3.0 V ±10%		8	32	μA
				V_{DD} = 2.0 V ±10%		5	25	μA
			LCD operating[Note 6]	V_{DD} = 5.0 V ±10%		27	76	μA
				V_{DD} = 3.0 V ±10%		9.8	37	μA
				V_{DD} = 2.0 V ±10%		6.6	24	μA
		32.768 kHz crystal oscillation × 4 multiplication HALT mode[Note 4] (C3 = C4 = 22 pF, R1 = 220 kΩ)	LCD not operating[Note 5]	V_{DD} = 5.0 V ±10%		25	70	μA
				V_{DD} = 3.0 V ±10%		8	32	μA
			LCD operating[Note 6]	V_{DD} = 5.0 V ±10%		27	76	μA
				V_{DD} = 3.0 V ±10%		9.8	37	μA
	I_{DD5}	STOP mode[Note 5]		V_{DD} = 5.0 V ±10%		0.1	10	μA
				V_{DD} = 3.0 V ±10%		0.05	5	μA
				V_{DD} = 2.0 V ±10%		0.05	3	μA
	I_{DD6}	5.0 MHz crystal oscillation A/D operating mode[Note 7] (C1 = C2 = 22 pF)		V_{DD} = 5.0 V ±10%[Note 2]		7.0	14.0	mA
				V_{DD} = 3.0 V ±10%[Note 3]		2.3	4.2	mA
				V_{DD} = 2.0 V ±10%[Note 3]		1.5	3.5	mA

Notes 1. The port current (including the current that flows to on-chip pull-up resistors) is not included.

2. High-speed mode operation (when the processor clock control register (PCC) is set to 00H)

3. Low-speed mode operation (when PCC is set to 02H)

4. When the main system clock is stopped

5. When the LCD is not operating (LCDON0 = 0, LIPS0 = 0)

6. Then the LCD is operating (LCDON0 = 1, LIPS0 = 1)

7. This is the total current that flows to V_{DD} and AV_{DD}.

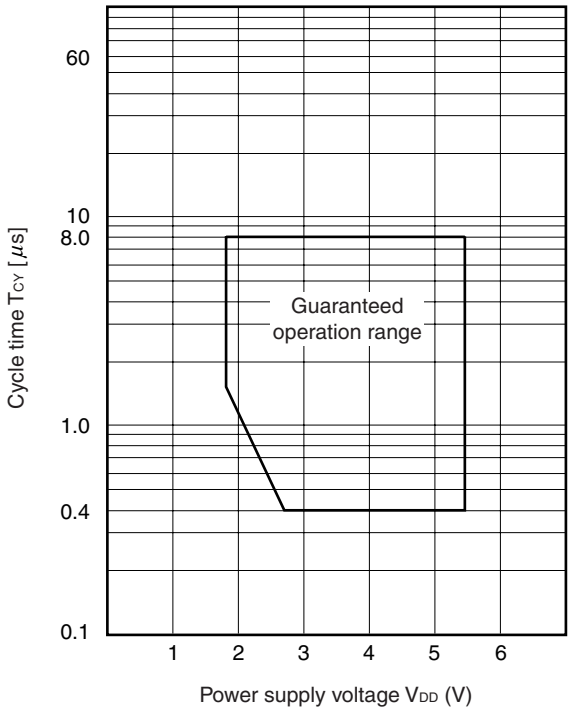
Remark Unless otherwise specified, the characteristics of alternate-function pins are the same as those of port pins.

AC Characteristics

(1) Basic operation (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Cycle time (minimum instruction execution time)	T_{CY}	Operating with main system clock		V_{DD} = 2.7 to 5.5 V	0.4		8.0	μs
				V_{DD} = 1.8 to 5.5 V	1.6		8.0	μs
		Operating with subsystem clock	Original oscillation operation	V_{DD} = 1.8 to 5.5 V	114	122	125	μs
			× 4 multiplication operation	V_{DD} = 2.7 to 5.5 V	14.3	15.3	15.6	μs
Capture input high-/low-level width	t_{CPTH}, t_{CPTL}	CPT20			10			μs
TMI60, TM61 input frequency	f_{TI}	V_{DD} = 2.7 to 5.5 V			0		4	MHz
		V_{DD} = 1.8 to 5.5 V			0		275	kHz
TMI60, TM61 input high-/low-level width	t_{TIH}, t_{TIL}	V_{DD} = 2.7 to 5.5 V			0.125			μs
		V_{DD} = 1.8 to 5.5 V			1.8			μs
Interrupt input high-/low-level width	t_{INTH}, t_{INTL}	INTP0 to INTP3			10			μs
Key return input low-level width	t_{KRL}	KR0 to KR7 (μPD789477, 789478, 78F9478)			10			μs
		KR00 to KR07, KR10 to KR17 (μPD789479, 78F9479)			10			μs
$\overline{\text{RESET}}$ low-level width	t_{RSL}				10			μs

T_{CY} vs. V_{DD} (main system clock)



(2) Serial interface 20 (SIO20) (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

(a) 3-wire serial I/O mode (internal clock output)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{SCK20}$ cycle time	t_{KCY1}	V_{DD} = 2.7 to 5.5 V		800			ns
		V_{DD} = 1.8 to 5.5 V		3200			ns
$\overline{SCK20}$ high-/low-level width	t_{KH1}, t_{KL1}	V_{DD} = 2.7 to 5.5 V		t_{KCY1}/2–50			ns
		V_{DD} = 1.8 to 5.5 V		t_{KCY1}/2–150			ns
SI20 setup time (to $\overline{SCK20}$↑)	t_{SIK1}	V_{DD} = 2.7 to 5.5 V		150			ns
		V_{DD} = 1.8 to 5.5 V		500			ns
SI20 hold time (from $\overline{SCK20}$↑)	t_{KSI1}	V_{DD} = 2.7 to 5.5 V		400			ns
		V_{DD} = 1.8 to 5.5 V		600			ns
Delay time from $\overline{SCK20}$↓ to SO20 output	t_{KSO1}	R = 1 kΩ, C = 100 pF[Note]	V_{DD} = 2.7 to 5.5 V	0		250	ns
			V_{DD} = 1.8 to 5.5 V	0		1000	ns

Note R and C are the load resistance and load capacitance of the SO20 output line.

(b) 3-wire serial I/O mode (external clock input)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{SCK20}$ cycle time	t_{KCY2}	V_{DD} = 2.7 to 5.5 V		800			ns
		V_{DD} = 1.8 to 5.5 V		3200			ns
$\overline{SCK20}$ high-/low-level width	t_{KH2}, t_{KL2}	V_{DD} = 2.7 to 5.5 V		400			ns
		V_{DD} = 1.8 to 5.5 V		1600			ns
SI20 setup time (to $\overline{SCK20}$↑)	t_{SIK2}	V_{DD} = 2.7 to 5.5 V		100			ns
		V_{DD} = 1.8 to 5.5 V		150			ns
SI20 hold time (from $\overline{SCK20}$↑)	t_{KSI2}	V_{DD} = 2.7 to 5.5 V		400			ns
		V_{DD} = 1.8 to 5.5 V		600			ns
Delay time from $\overline{SCK20}$↓ to SO20 output	t_{KSO2}	R = 1 kΩ, C = 100 pF[Note]	V_{DD} = 2.7 to 5.5 V	0		300	ns
			V_{DD} = 1.8 to 5.5 V	0		1000	ns

Note R and C are the load resistance and load capacitance of the SO20 output line.

(c) UART mode (dedicated baud rate generator output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		V_{DD} = 2.7 to 5.5 V			78125	bps
		V_{DD} = 1.8 to 5.5 V			19531	bps

(d) UART mode (external clock input)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
ASCK20 cycle time	t_{KCY3}	V_{DD} = 2.7 to 5.5 V	800			ns
		V_{DD} = 1.8 to 5.5 V	3200			ns
ASCK20 high-/low-level width	t_{KH3}, t_{KL3}	V_{DD} = 2.7 to 5.5 V	400			ns
		V_{DD} = 1.8 to 5.5 V	1600			ns
Transfer rate		V_{DD} = 2.7 to 5.5 V			39063	bps
		V_{DD} = 1.8 to 5.5 V			9766	bps
ASCK20 rise/fall time	t_R, t_F				1	μs

(3) Serial interface 1A0 (SIO1A0) (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

(a) 3-wire serial I/O mode, 3-wire serial I/O mode with automatic transmit/receive function (internal clock output)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{SCK10}$ cycle time	t_{KCY4}	V_{DD} = 2.7 to 5.5 V		800			ns
		V_{DD} = 1.8 to 5.5 V		3200			ns
$\overline{SCK10}$ high-/low-level width	t_{KH4}, t_{KL4}	V_{DD} = 2.7 to 5.5 V		$t_{KCY4}/2-50$			ns
		V_{DD} = 1.8 to 5.5 V		$t_{KCY4}/2-150$			ns
SI10 setup time (to $\overline{SCK10}\uparrow$)	t_{SIK4}	V_{DD} = 2.7 to 5.5 V		150			ns
		V_{DD} = 1.8 to 5.5 V		500			ns
SI10 hold time (from $\overline{SCK10}\uparrow$)	t_{KSI4}	V_{DD} = 2.7 to 5.5 V		400			ns
		V_{DD} = 1.8 to 5.5 V		600			ns
Delay time from $\overline{SCK10}\downarrow$ to SO10 output	t_{KSO4}	R = 1 kΩ, C = 100 pF[Note]	V_{DD} = 2.7 to 5.5 V	0		250	ns
			V_{DD} = 1.8 to 5.5 V	0		1000	ns

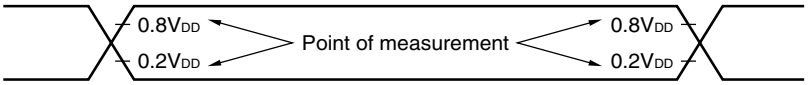
Note R and C are the load resistance and load capacitance of the SO10 output line.

(b) 3-wire serial I/O mode, 3-wire serial I/O mode with automatic transmit/receive function (external clock input)

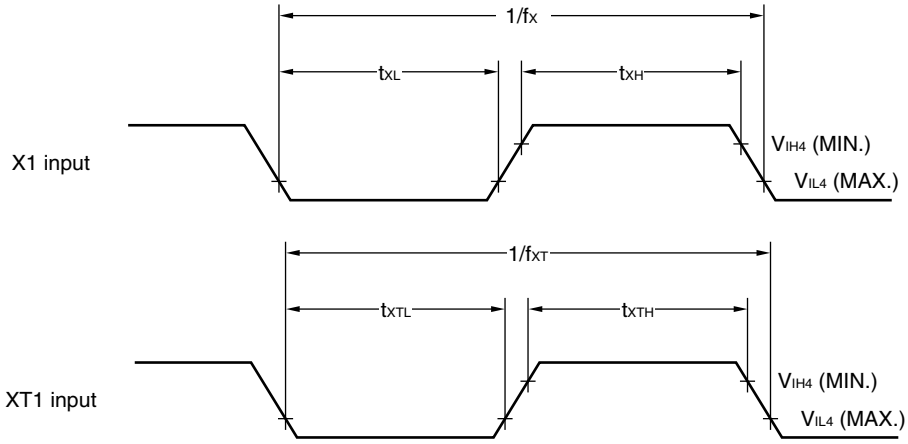
Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{SCK10}$ cycle time	t_{KCY5}	V_{DD} = 2.7 to 5.5 V		800			ns
		V_{DD} = 1.8 to 5.5 V		3200			ns
$\overline{SCK10}$ high-/low-level width	t_{KH5}, t_{KL5}	V_{DD} = 2.7 to 5.5 V		400			ns
		V_{DD} = 1.8 to 5.5 V		1600			ns
SI10 setup time (to $\overline{SCK10}\uparrow$)	t_{SIK5}	V_{DD} = 2.7 to 5.5 V		100			ns
		V_{DD} = 1.8 to 5.5 V		150			ns
SI10 hold time (from $\overline{SCK10}\uparrow$)	t_{KSI5}	V_{DD} = 2.7 to 5.5 V		400			ns
		V_{DD} = 1.8 to 5.5 V		600			ns
Delay time from $\overline{SCK10}\downarrow$ to SO10 output	t_{KSO5}	R = 1 kΩ, C = 100 pF[Note]	V_{DD} = 2.7 to 5.5 V	0		300	ns
			V_{DD} = 1.8 to 5.5 V	0		1000	ns

Note R and C are the load resistance and load capacitance of the SO10 output line.

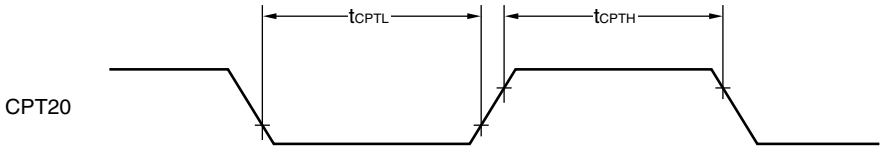
AC Timing Measurement Points (Excluding X1 and XT1 Inputs)



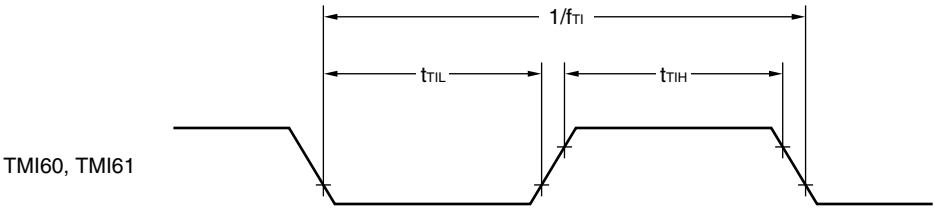
Clock Timing



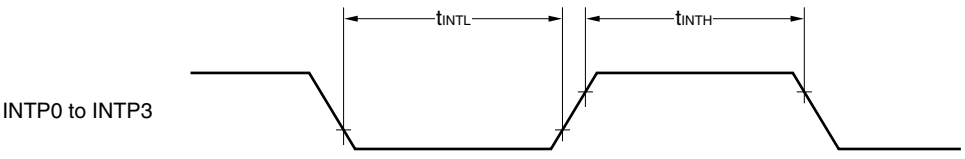
Capture Input Timing



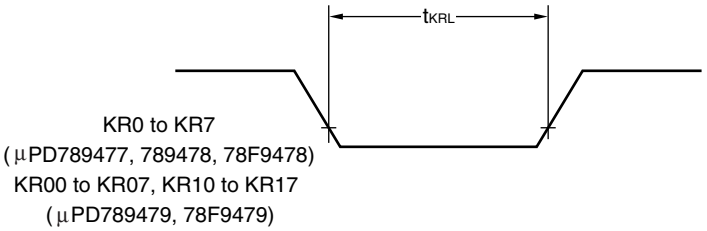
TMI Timing



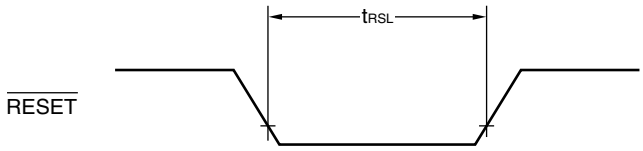
Interrupt Input Timing



Key Return Input Timing

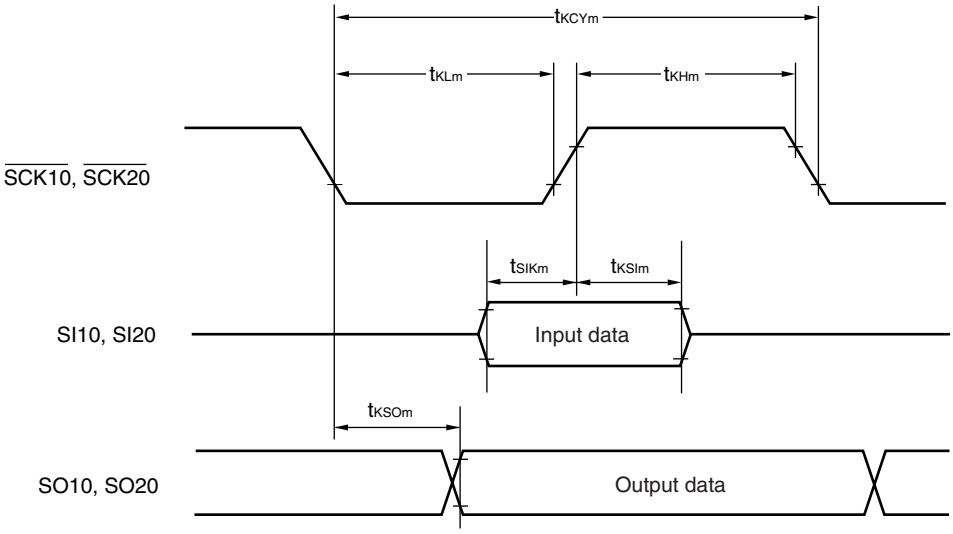


$\overline{\text{RESET}}$ Input Timing



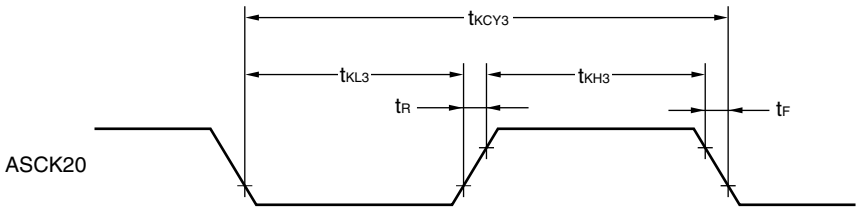
Serial Transfer Timing

3-wire serial I/O mode:



Remark m = 1, 2, 4, 5

UART mode (external clock input):



8-Bit A/D Converter Characteristics

(T_A = –40 to +85°C, 1.8 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution			8	8	8	bit
Overall error[Note 1]		AV_{DD} = 2.7 to 5.5 V			±0.6	%FSR
		AV_{DD} = 1.8 to 5.5 V			±1.2	%FSR
Conversion time	t_{CONV}	AV_{DD} = 2.7 to 5.5 V	14		100	μs
		AV_{DD} = 1.8 to 5.5 V	28		100	μs
		When × 4 subsystem clock is used (ADSEL1 = 1), AV_{DD} = 2.7 to 5.5 V	132[Note 2]			Clock
Analog input voltage	V_{IAN}		0		AV_{DD}	V

Notes 1. Excludes quantization error (±0.2%)

2. Number of clocks of ×4 subsystem clock

Remark FSR: Full scale range

LCD Characteristics (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

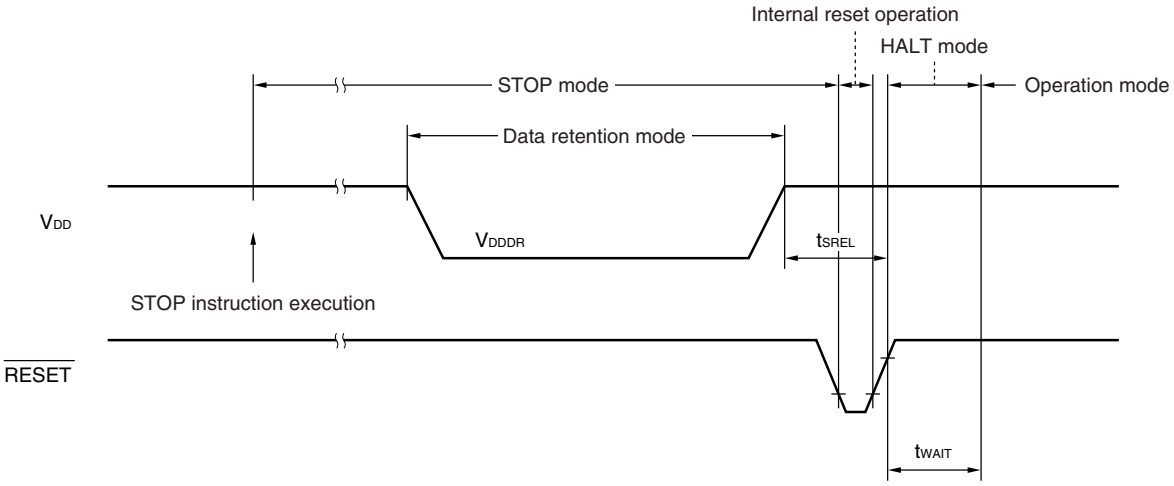
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
LCD drive voltage	V_{LCD}		2.7		V_{DD}	V
LCD output voltage differential[Note] (common)	V_{ODC}	I_O = ±5 μA	0		±0.2	V
LCD output voltage differential[Note] (segment)	V_{ODS}	I_O = ±1 μA	0		±0.2	V

Note The voltage differential is the difference between the segment and common signal output's actual and ideal output voltages.

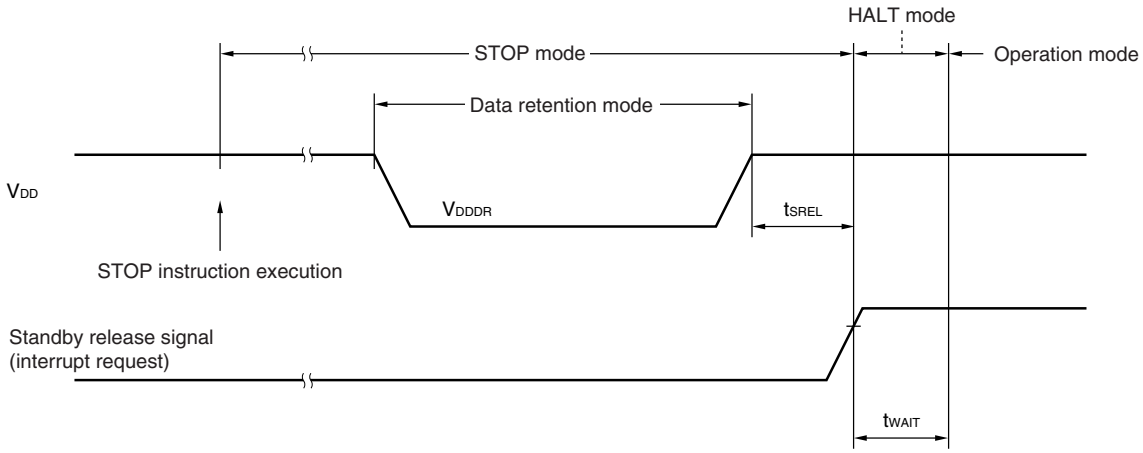
Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics (T_A = –40 to +85°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention power supply voltage	V_{DDDR}		1.8		5.5	V
Release signal set time	t_{SREL}		0			μs

Data Retention Timing (STOP Mode Release by $\overline{\text{RESET}}$)



Data Retention Timing (Standby Release Signal: STOP Mode Release by Interrupt Request Signal)



Oscillation Stabilization Wait Time (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Oscillation stabilization wait time[Note 1]	t_{WAIT}	Release by $\overline{\text{RESET}}$		$2^{15}/f_X$		s
		Release by interrupt		**Note 2**		s

Notes 1. Use a resonator whose oscillation stabilizes within the oscillation stabilization wait time.

2. Selection of $2^{12}/f_X$, $2^{15}/f_X$, or $2^{17}/f_X$ is possible using bits 0 to 2 (OSTS0 to OSTS2) of the oscillation stabilization time selection register (OSTS).

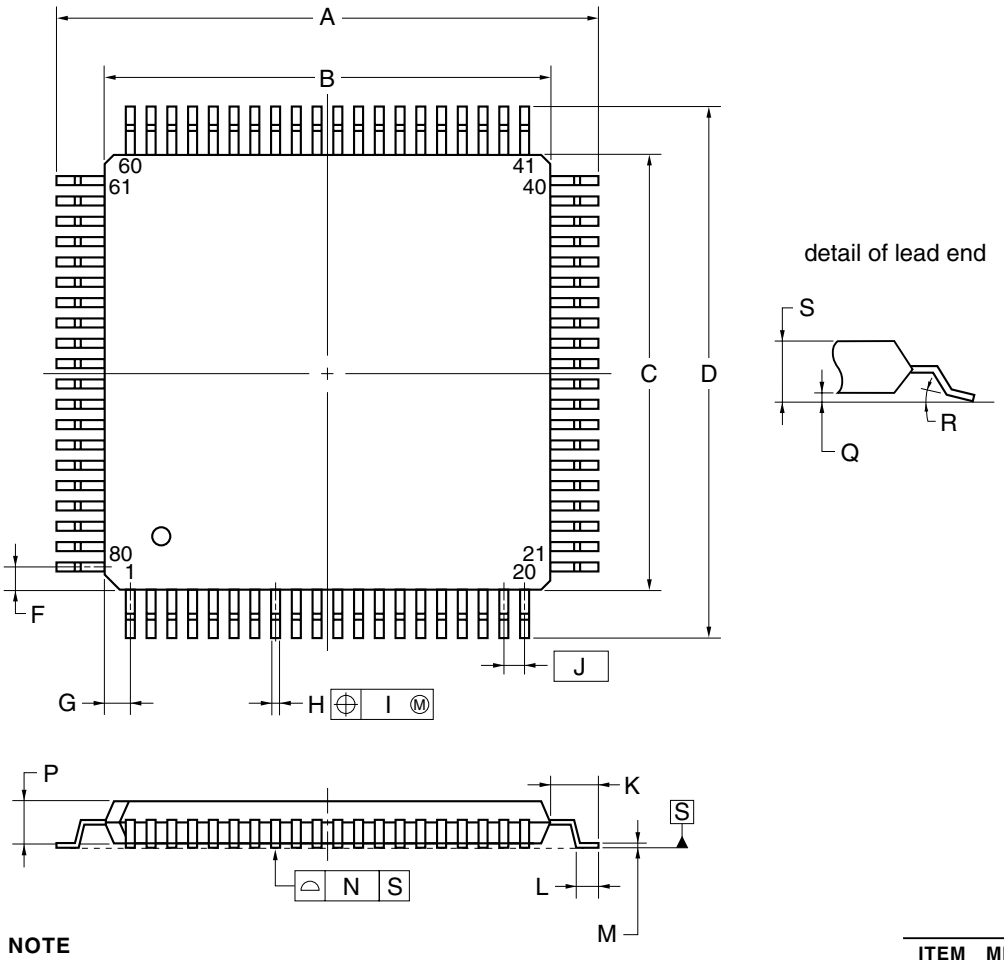
Remark f_X: Main system clock oscillation frequency

Writing and Erasing Characteristics (T_A = 10 to 40°C, V_{DD} = 1.8 to 5.5 V) (μPD78F9478, 78F9479 only)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Write operation frequency	f_X	V_{DD} = 2.7 to 5.5 V	1.0		5	MHz
		V_{DD} = 1.8 to 5.5 V	1.0		1.25	MHz
Write current (V_{DD} pin)[Note]	I_{DDW}	When V_{PP} supply voltage = V_{PP1} (at 5.0 MHz operation)			7	mA
Write current (V_{PP} pin)[Note]	I_{PPW}	When V_{PP} supply voltage = V_{PP1}			13	mA
Erase current (V_{DD} pin)[Note]	I_{DDE}	When V_{PP} supply voltage = V_{PP1} (at 5.0 MHz operation)			7	mA
Erase current (V_{PP} pin)[Note]	I_{PPE}	When V_{PP} supply voltage = V_{PP1}			100	mA
Unit erase time	t_{er}		0.5	1	1	s
Total erase time	t_{era}				20	s
Number of rewrites		Erase and write is considered as 1 cycle			20	times
V_{PP} supply voltage	V_{PP0}	Normal operation	0		0.2V_{DD}	V
	V_{PP1}	Flash memory programming	9.7	10.0	10.3	V

Note Excludes current flowing through ports (including on-chip pull-up resistors)

80-PIN PLASTIC QFP (14x14)



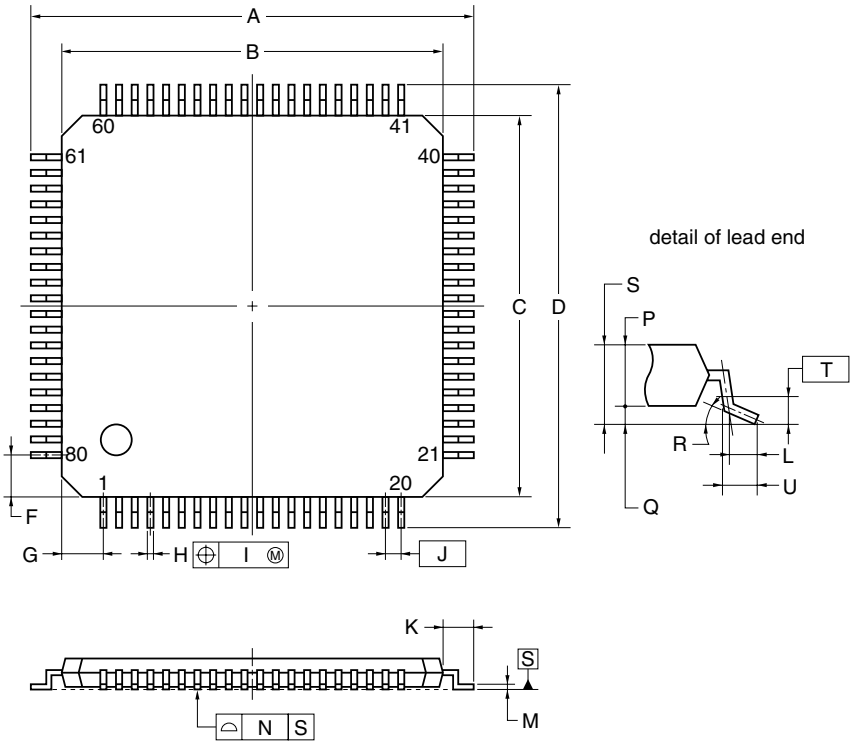
NOTE

Each lead centerline is located within 0.13 mm of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS
A	17.20±0.20
B	14.00±0.20
C	14.00±0.20
D	17.20±0.20
F	0.825
G	0.825
H	0.32±0.06
I	0.13
J	0.65 (T.P.)
K	1.60±0.20
L	0.80±0.20
M	$0.17^{+0.03}_{-0.07}$
N	0.10
P	1.40±0.10
Q	0.125±0.075
R	$3^{\circ}{}^{+7^{\circ}}_{-3^{\circ}}$
S	1.70 MAX.

P80GC-65-8BT-1

80-PIN PLASTIC TQFP (FINE PITCH) (12x12)



NOTE

Each lead centerline is located within 0.08 mm of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS
A	14.0±0.2
B	12.0±0.2
C	12.0±0.2
D	14.0±0.2
F	1.25
G	1.25
H	0.22±0.05
I	0.08
J	0.5 (T.P.)
K	1.0±0.2
L	0.5
M	0.145±0.05
N	0.08
P	1.0
Q	0.1±0.05
R	$3^{\circ}{}^{+4^{\circ}}_{-3^{\circ}}$
S	1.1±0.1
T	0.25
U	0.6±0.15

P80GK-50-9EU-1

<R>

CHAPTER 24 RECOMMENDED SOLDERING CONDITIONS

The μPD789479 subseries should be soldered and mounted under the following recommended conditions.

For soldering methods and conditions other than those recommended below, contact an NEC Electronics sales representative.

For technical information, see the following website.

Semiconductor Device Mount Manual (http://www.necel.com/pkg/en/mount/index.html)

Table 24-1. Surface Mounting Type Soldering Conditions (1/3)

(1) μPD789477GC-×××-8BT: 80-pin plastic QFP (14x14)
μPD789478GC-×××-8BT: 80-pin plastic QFP (14x14)
μPD789479GC-×××-8BT: 80-pin plastic QFP (14x14)
μPD78F9478GC-8BT: 80-pin plastic QFP (14x14)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 235°C, Time: 30 seconds max. (at 210°C or higher), Count: Twice or less	IR35-00-2
VPS	Package peak temperature: 215°C, Time: 40 seconds max. (at 200°C or higher), Count: Twice or less	VP15-00-2
Wave soldering	Solder bath temperature: 260°C max., Time: 10 seconds max., Count: Once, Preheating temperature: 120°C max. (package surface temperature)	WS60-00-1
Partial heating	Pin temperature: 350°C max., Time: 3 seconds max. (per pin row)	–

Caution Do not use different soldering methods together (except for partial heating).

(2) μPD789477GK-×××-9EU: 80-pin plastic TQFP (fine pitch) (12x12)
μPD789478GK-×××-9EU: 80-pin plastic TQFP (fine pitch) (12x12)
μPD789479GK-×××-9EU: 80-pin plastic TQFP (fine pitch) (12x12)
μPD78F9478GK-9EU: 80-pin plastic TQFP (fine pitch) (12x12)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Interface reflow	Package peak temperature: 235°C, Time:30 seconds max. (at 210°C or higher), Count: Twice or less, Exposure limit: 7 days[Note] (after that, prebake at 125°C for 10 to 72 hours)	IR35-107-2
VPS	Package peak temperature: 215°C, Time:40 seconds max. (at 200°C or higher), Count: Twice or less, Exposure limit: 7 days[Note] (after that, prebake at 125°C for 10 to 72 hours)	VP15-107-2
Partial heating	Pin temperature: 350°C max., Time: 3 seconds max. (per pin row)	–

Note After opening the dry peak, store it at 25°C or less and 65% RH or less for the allowable storage period.

Caution Do not use different soldering methods together (except for partial heating).

Table 24-1. Surface Mounting Type Soldering Conditions (2/3)

(3) μPD78F9479GC-8BT: 80-pin plastic QFP (14x14)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Interface reflow	Package peak temperature: 235°C, Time:30 seconds max. (at 210°C or higher), Count: Twice or less, Exposure limit: 7 days[Note] (after that, prebake at 125°C for 10 to 72 hours)	IR35-107-2
VPS	Package peak temperature: 215°C, Time:40 seconds max. (at 200°C or higher), Count: Twice or less, Exposure limit: 7 days[Note] (after that, prebake at 125°C for 10 to 72 hours)	VP15-107-2
Wave soldering	Solder bath temperature: 260°C max., Time: 10 seconds max., Count: Once, Preheating temperature: 120°C max. (package surface temperature), Exposure limit: 7 days[Note] (after that, prebake at 125°C for 10 to 72 hours)	WS60-107-1
Partial heating	Pin temperature: 350°C max., Time: 3 seconds max. (per pin row)	–

Note After opening the dry peak, 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).

(4) μPD78F9479GK-9EU: 80-pin plastic TQFP (fine pitch) (12x12)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Interface reflow	Package peak temperature: 235°C, Time:30 seconds max. (at 210°C or higher), Count: Twice or less, Exposure limit: 3 days[Note] (after that, prebake at 125°C for 10 to 72 hours)	IR35-103-2
VPS	Package peak temperature: 215°C, Time:40 seconds max. (at 200°C or higher), Count: Twice or less, Exposure limit: 3 days[Note] (after that, prebake at 125°C for 10 to 72 hours)	VP15-103-2
Partial heating	Pin temperature: 350°C max., Time: 3 seconds max. (per pin row)	–

Note After opening the dry peak, store it at 25°C or less and 65% RH or less for the allowable storage period.

Caution Do not use different soldering methods together (except for partial heating).

Table 24-1. Surface Mounting Type Soldering Conditions (3/3)

(5) μPD789477GC-×××-8BT-A: 80-pin plastic QFP (14x14)
μPD789478GC-×××-8BT-A: 80-pin plastic QFP (14x14)
μPD789479GC-×××-8BT-A: 80-pin plastic QFP (14x14)
μPD78F9478GC-8BT-A: 80-pin plastic QFP (14x14)
μPD78F9479GC-8BT-A: 80-pin plastic QFP (14x14)
μPD789477GK-×××-9EU-A: 80-pin plastic TQFP (fine pitch) (12x12)
μPD789478GK-×××-9EU-A: 80-pin plastic TQFP (fine pitch) (12x12)
μPD789479GK-×××-9EU-A: 80-pin plastic TQFP (fine pitch) (12x12)
μPD78F9478GK-9EU-A: 80-pin plastic TQFP (fine pitch) (12x12)
μPD78F9479GK-9EU-A: 80-pin plastic TQFP (fine pitch) (12x12)

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
Wave soldering	When the pin pitch of the package is 0.65 mm or more, wave soldering can also be performed. For details, contact an NEC Electronics sales representative.	–
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).

Remarks 1. Products that have the part numbers suffixed by "-A" are lead-free products.
2. For soldering methods and conditions other than those recommended above, contact an NEC Electronics sales representative.

APPENDIX A DEVELOPMENT TOOLS

The following development tools are available for development of systems using the μPD789479 Subseries.
Figure A-1 shows development tools.

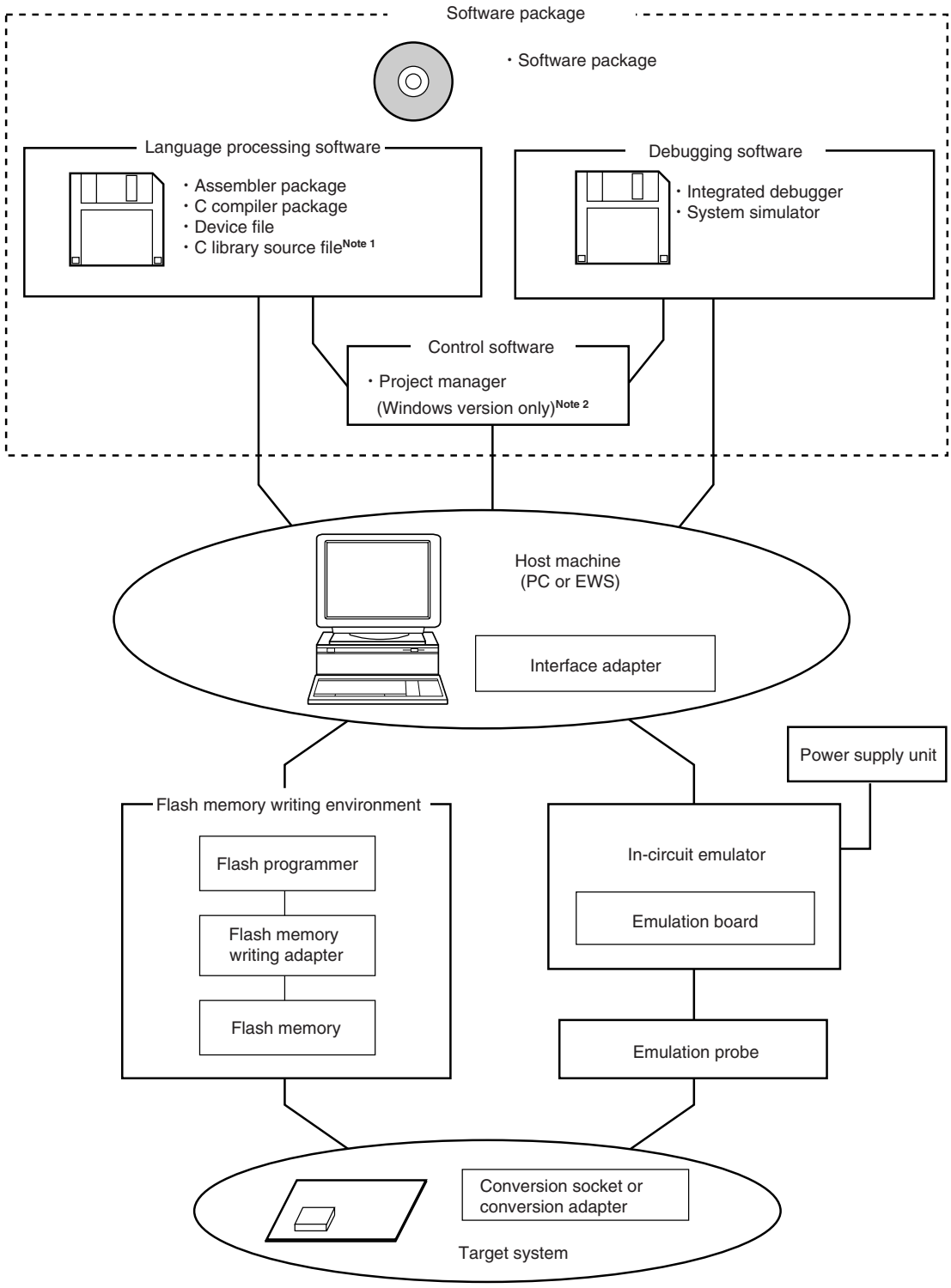
- Support for PC98-NX Series

  Unless specified otherwise, the products supported by IBM PC/AT™ compatibles can be used in the PC98-NX Series. When using the PC98-NX Series, refer to the explanation of IBM PC/AT compatibles.

- Windows™

  Unless specified otherwise, “Windows” indicates the following operating systems.
  - Windows 3.1
  - Windows 95
  - Windows 98
  - Windows 2000
  - Windows NT™ Ver.4.0
  - Windows XP

Figure A-1. Development Tools



Notes 1. The C library source file is not included in the software package.

2. The project manager is included in the assembler package.
The project manager is used only in the Windows environment.

A.1 Software Package

SP78K0S Software package	Software tools for development of the 78K/0S microcontrollers are combined in this package. The following tools are included. RA78K0S, CC78K0S, ID78K0S-NS, SM78K0S, and device files
	Part number: μS××××SP78K0S

Remark ×××× in the part number differs depending on the OS used

μS××××SP78K0S

××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Japanese Windows	CD-ROM
BB17		English Windows	

A.2 Language Processing Software

RA78K0S Assembler package	Program that converts program written in mnemonic into object codes that can be executed by a microcontroller. In addition, automatic functions to generate symbol tables and optimize branch instructions are also provided. Used in combination with a device file (DF789488) (sold separately). **<Caution when used in PC environment>** The assembler package is a DOS-based application but may be used in the Windows environment by using the project manager of Windows (included in the assembler package).
	Part number: μS××××RA78K0S
CC78K0S C compiler package	Program that converts program written in C language into object codes that can be executed by a microcontroller. Used in combination with an assembler package (RA78K0S) and device file (DF789488) (both sold separately). **<Caution when used under PC environment>** The C compiler package is a DOS-based application but may be used in the Windows environment by using the project manager of Windows (included in the assembler package).
	Part number: μS××××CC78K0S
DF789488[Note 1] Device file	File containing information inherent to the device. Used in combination with the RA78K0S, CC78K0S, ID78K0S-NS, and SM78K0S (all sold separately).
	Part number: μS××××DF789488
CC78K0S-L[Note 2] C library source file	Source file of functions for generating the object library included in C compiler package. Necessary for changing the object library included in the C compiler package according to the customer's specifications. Since this is a source file, its working environment does not depend on any particular operating system.
	Part number: μS××××CC78K0S-L

Notes 1. DF789488 is a common file that can be used with the RA78K0S, CC78K0S, ID78K0S-NS, and SM78K0S.

2. CC78K0S-L is not included in the software package (SP78K0S).

Remark ××××in the part number differs depending on the host machine and operating system to be used.

μS××××RA78K0S
μS××××CC78K0S

××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT compatible	Japanese Windows	3.5" 2HD FD
BB13		English Windows	
AB17		Japanese Windows	CD-ROM
BB17		English Windows	
3P17	HP9000 series 700™	HP-UX™ (Rel. 10.10)	
3K17	SPARCstation™	SunOS™ (Rel. 4.1.4), Solaris™ (Rel. 2.5.1)	

μS××××DF789488
μS××××CC78K0S-L

××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT compatible	Japanese Windows	3.5" 2HD FD
BB13		Japanese Windows	
3P16	HP9000 series 700	HP-UX™ (Rel. 10.10)	DAT
3K13	SPARCstation	SunOS™ (Rel. 4.1.4),	3.5" 2HD FD
3K15		Solaris™ (Rel. 2.5.1)	1/4-inch CGMT

A.3 Control Software

PM+ Project manager	Control software created for efficient development of the user program in the Windows environment. User program development operations such as editor startup, build, and debugger startup can be performed from the PM+. <Caution> The PM+ is included in the assembler package (RA78K0S). The PM+ is used only in the Windows environment.

A.4 Flash Memory Writing Tools

Flashpro III (FL-PR3, PG-FP3) Flashpro IV (FL-PR4, PG-FP4) Flash programmer	Dedicated flash programmer for microcontrollers incorporating flash memory
FA-80GC-8BT FA-80GK-9EU Flash memory writing adapter	Adapter for writing to flash memory and connected to Flashpro III or Flashpro IV. • FA-80GC-8BT: For 80-pin plastic QFP (GC-8BT type) • FA-80GK-9EU: For 80-pin plastic TQFP (GK-9EU type)

Remark The FL-PR3, FL-PR4, FA-80GC-8BT, and FA-80GK-9EU are products made by Naito Densei Machida Mfg. Co., Ltd. (TEL +81-42-750-4172).

A.5 Debugging Tools (Hardware)

IE-78K0S-NS In-circuit emulator		In-circuit emulator for debugging hardware and software of an application system using the 78K/0S microcontrollers. Can be used with the integrated debugger ID78K0S-NS. Used in combination with an AC adapter, emulation probe, and interface adapter for connecting the host machine.
IE-78K0S-NS-A In-circuit emulator		The IE-78K0S-NS-A provides a coverage function in addition to the IE-78K0S-NS functions, thus enhancing the debug functions, including the tracer and timer functions.
IE-70000-MC-PS-B AC adapter		Adapter for supplying power from AC 100 to 240 V outlet.
IE-70000-98-IF-C Interface adapter		Adapter necessary when using a PC-9800 series PC (except notebook type) as the host machine (C bus supported)
IE-70000-CD-IF-A PC card interface		PC card and interface cable necessary when using a notebook PC as the host machine (PCMCIA socket supported)
IE-70000-PC-IF-C Interface adapter		Interface adapter necessary when using an IBM PC/AT compatible as the host machine (ISA bus supported)
IE-70000-PCI-IF-A Interface adapter		Adapter necessary when using a personal computer incorporating a PCI bus as the host machine
IE-789488-NS-EM1 Emulation board		Board for emulating the peripheral hardware inherent to the device. Used in combination with in-circuit emulator.
NP-80GC Emulation probe		Cable to connect the in-circuit emulator and target system. Used in combination with the EV-9200GC-80.
	EV-9200GC-80 Conversion socket	Conversion socket to connect the NP-80GC and a target system board on which an 80-pin plastic QFP (GC-8BT type) can be mounted.
NP-80GC-TQ NP-H80GC-TQ Emulation probe		Cable to connect an in-circuit emulator to the target system. Used in combination with the TGC-080SBP.
	TGC-080SBP Conversion adapter	Conversion adapter to connect the NP-80GC-TQ or NP-H80GC-TQ to a target system board on which an 80-pin plastic QFP (GC-8BT type) can be mounted.
NP-80GK NP-H80GK-TQ Emulation probe		Cable to connect an in-circuit emulator to the target system. Used in combination with the TGK-080SDW.
	TGK-080SDW Conversion adapter	Conversion adapter to connect the NP-80GK or NP-H80GK-TQ to a target system board on which an 80-pin plastic TQFP (fine pitch) (GK-9EU type) can be mounted.

Remarks 1. The NP-80GC, NP-80GC-TQ, NP-H80GC-TQ, NP-80GK, and NP-H80GK-TQ are products of Naito Densei Machida Mfg. Co., Ltd. (TEL +81-42-750-4172).

2. The TGC-080SBP and TGK-080SDW are products of TOKYO ELETECH CORPORATION (TEL +81-3-5295-1661).

A.6 Debugging Tools (Software)

ID78K0S-NS Integrated debugger	This debugger supports the in-circuit emulators IE-78K0S-NS and IE-78K0S-NS-A for the 78K/0S microcontrollers. The ID78K0S-NS is Windows-based software. It has improved C-compatible debugging functions and can display the results of tracing with the source program using an integrating window function that associates the source program, disassemble display, and memory display with the trace result. Used in combination with a device file (DF789488) (sold separately).
	Part number: μS××××ID78K0S-NS
SM78K0S System simulator	This is a system simulator for the 78K/0S microcontrollers. The SM78K0S is Windows-based software. It can be used to debug the target system at C source level or assembler level while simulating the operation of the target system on the host machine. Using SM78K0S, the logic and performance of the application can be verified independently of hardware development. Therefore, the development efficiency can be enhanced and the software quality can be improved. Used in combination with a device file (DF789488) (sold separately).
	Part number: μS××××SM78K0S
DF789488[Note] Device file	File containing information inherent to the device. Used in combination with the RA78K0S, CC78K0S, ID78K0S-NS, and SM78K0S (all sold separately).
	Part number: μS××××DF789488

Note DF789488 is a common file that can be used with the RA78K0S, CC78K0S, ID78K0S-NS, and SM78K0S.

Remark ×××× in the part number differs depending on the operating system and supply medium to be used.

μS××××ID78K0S-NS
μS××××SM78K0S

××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series IBM PC/AT compatibles	Japanese Windows	3.5" 2HD FD
BB13		English Windows	
AB17		Japanese Windows	CD-ROM
BB17		English Windows	

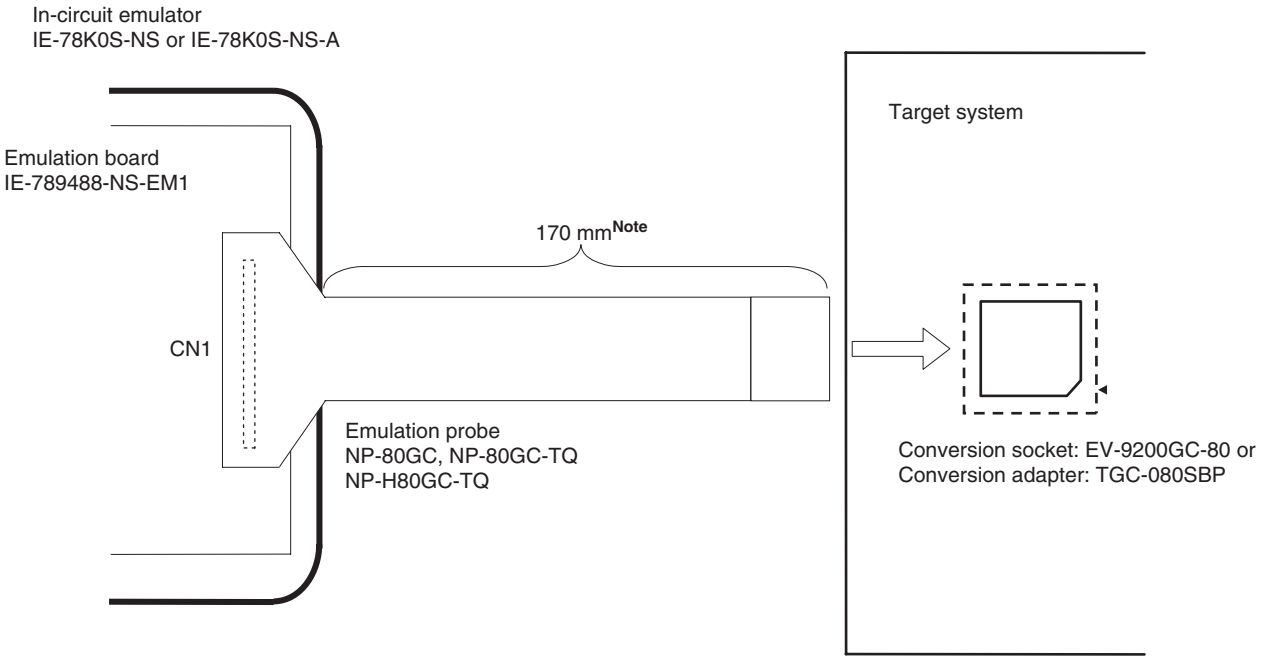
APPENDIX B NOTES ON TARGET SYSTEM DESIGN

Figures B-1 to B-6 show the conditions when connecting the emulation probe to the conversion adapter or conversion socket. Follow the configuration below and consider the shape of parts to be mounted on the target system when designing a system.

Of the products described in this chapter, the NP-80GC, NP-80GC-TQ, NP-H80GC-TQ, NP-80GK and NP-H80GK-TQ are products of Naito Densei Machida Mfg. Co., Ltd, and the TGC-080SBP and TGK-080SDW are products of TOKYO ELETECH CORPORATION.

(1) NP-80GC, NP-80GC-TQ, NP-H80GC-TQ

Figure B-1. Distance Between In-Circuit Emulator and Conversion Socket (80GC)



Note When NP-H80GC-TQ is used, the distance is 370 mm.

Figure B-2. Connection Conditions of Target System (When NP-80GC-TQ Is Used)

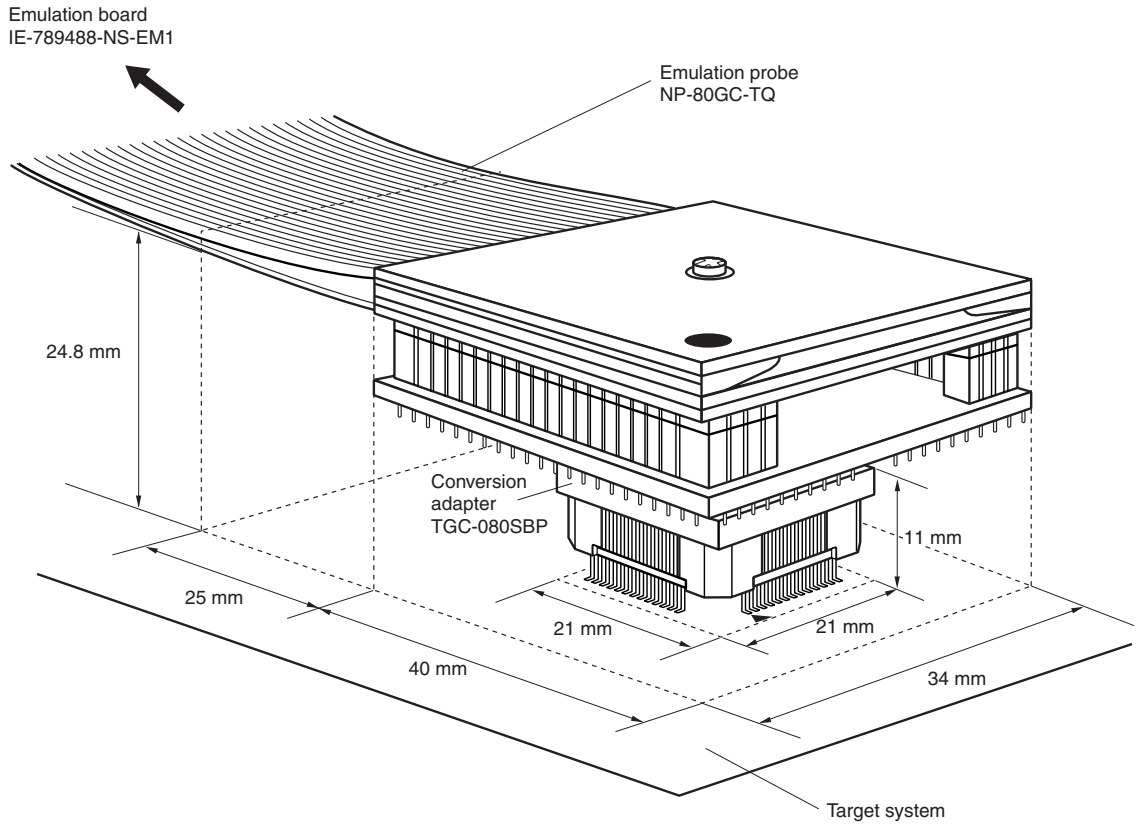
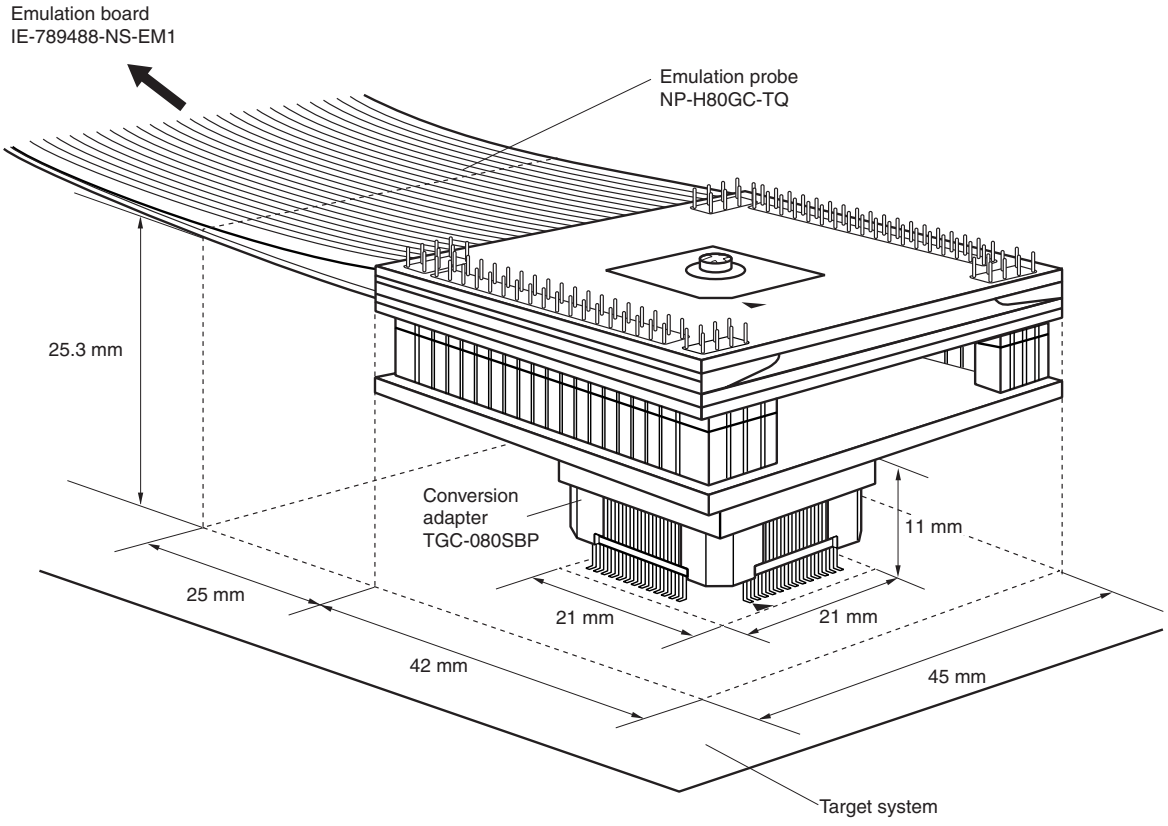
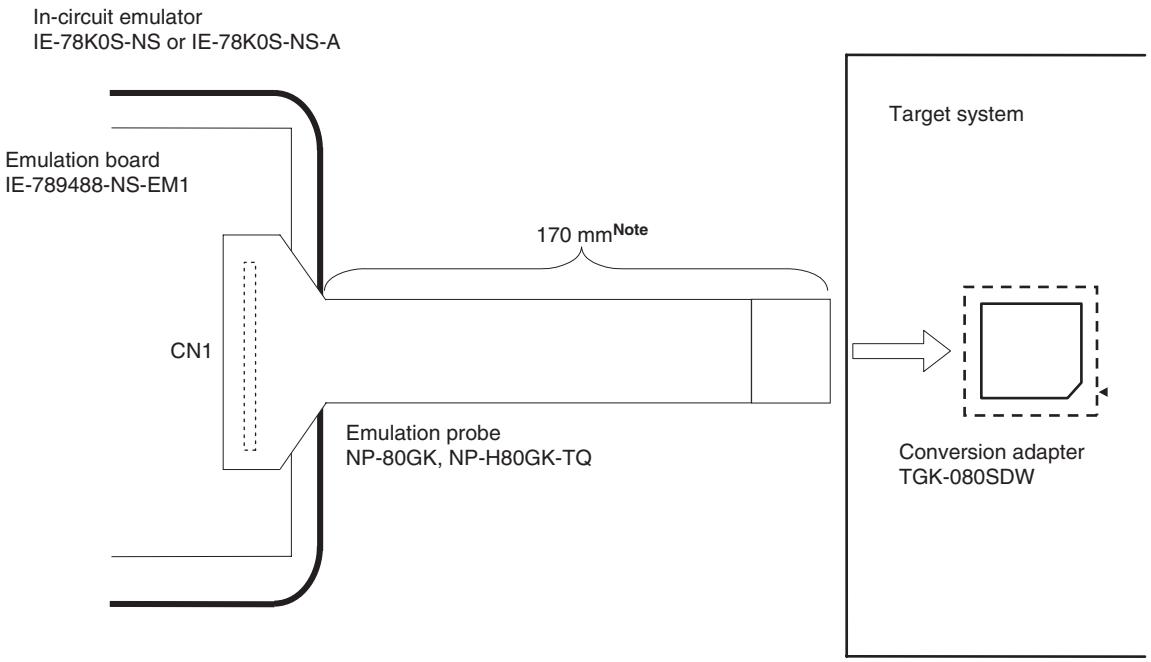


Figure B-3. Connection Conditions of Target System (When NP-H80GC-TQ Is Used)



(2) NP-80GK, NP-H80GK-TQ

Figure B-4. Distance Between In-Circuit Emulator and Conversion Adapter (80GK)



Note When NP-H80GK-TQ is used, the distance is 370 mm.

Figure B-5. Connection Conditions of Target System (When NP-80GK Is Used)

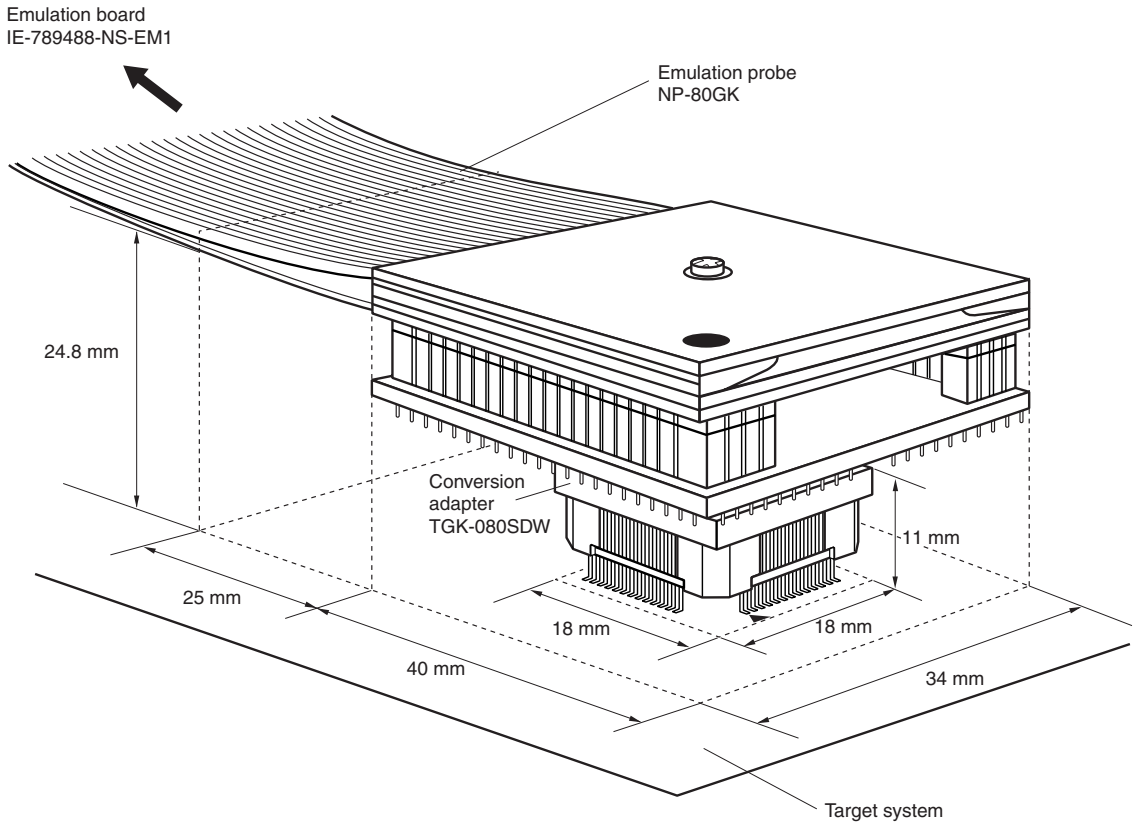
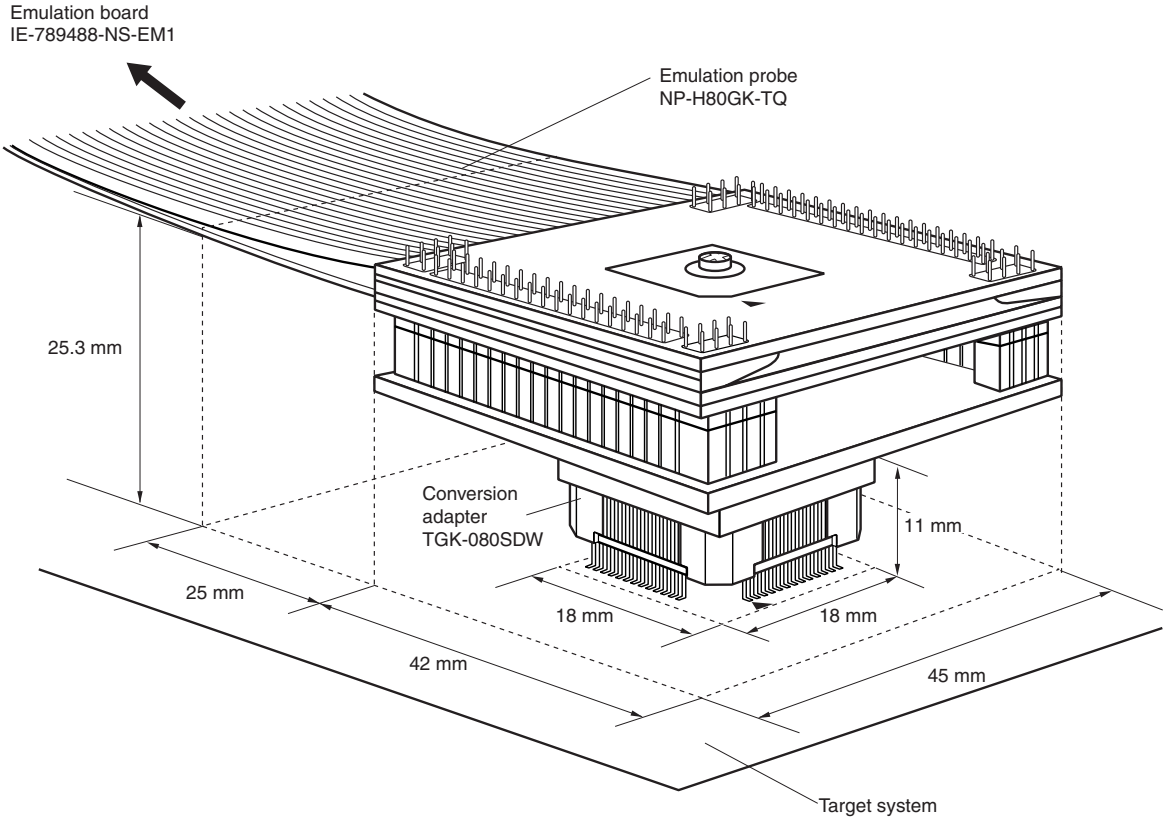


Figure B-6. Connection Conditions of Target System (When NP-H80GK-TQ Is Used)



APPENDIX C REGISTER INDEX

C.1 Register Index (Register Names in Alphabetic Order)

[A]

[B]

[C]

[E]

[I]

[K]

[S]

[T]

[W]

C.2 Register Index (Register Symbols in Alphabetic Order)

[T]

[W]

APPENDIX D REVISION HISTORY

D.1 Major Revisions in This Edition

(1/2)

Page	Description	Classification
CHAPTER 1 OUTLINE		
p. 25	Change of **1.3 Ordering Information**	(d)
p. 29	Change of **1.5 78K/0S Series Lineup**	(e)
CHAPTER 4 PORT FUNCTIONS		
p. 77	Modification of **Figure 4-2. Block Diagram of P00 to P07**	(a)
CHAPTER 7 8-BIT TIMER/EVENT COUNTERS 50, 60, AND 61		
p. 123	Modification of **Figure 7-2. Block Diagram of Timer 50**	(c)
p. 124	Modification of **Figure 7-3. Block Diagram of Timer 60**	(c)
p. 126	Modification of **Figure 7-5. Block Diagram of Output Controller (Timer 60)**	(a)
pp. 126, 127	Change of **(2)** and **(4)** of **7.2 Configuration of 8-Bit Timers 50, 60, and 61**	(c)
p. 136	Modification of **Figure 7-11. Timing of Interval Timer Operation with 8-Bit Resolution (Basic Operation)**	(a)
p. 137	Modification of **Figure 7-13. Timing of Interval Timer Operation with 8-Bit Resolution (When CRnm Is Set to FFH)**	(a)
p. 140	Modification of **Figure 7-17. Timing of Operation of External Event Counter with 8-Bit Resolution**	(c)
p. 141	Modification of <4> in **(3)** of **7.4.1 Operation as 8-bit timer counter**	(a)
p. 149	Addition of <9> and <10> to **7.4.3 Operation as carrier generator**	(c)
p. 150	Modification of **Figure 7-22. Timing of Carrier Generator Operation (When CR60 = N, CRH60 = M (M > N))**	(c)
p. 151	Modification of **Figure 7-23. Timing of Carrier Generator Operation (When CR60 = N, CRH60 = M (M < N))**	(c)
p. 153	Change of **7.4.4 PWM output mode operation (timer 50)**	(c)
p. 157	Change of **7.4.5 Operation as PPG output mode (timer 60 and timer 61)**	(c)
p. 160	Change of (1) of **7.5 Cautions on Using 8-Bit Timers 50, 60, and 61**	(c)
CHAPTER 10 8-BIT A/D CONVERTER		
p. 174	Change of **Figure 10-1. Block Diagram of 8-Bit A/D Converter**	(c)
p. 183	Change of **(1)** of **10.5 Cautions Related to 8-Bit A/D Converter**	(c)
CHAPTER 11 SERIAL INTERFACE 20		
p. 188	Change of **Figure 11-1. Block Diagram of Serial Interface 20**	(a, c)
p. 191	Addition of **Cautions 3** and **4** to **Figure 11-3. Format of Serial Operation Mode Register 20**	(c)
p. 195	Change of **Cautions 2** and **3** of **Figure 11-6. Format of Baud Rate Generator Control Register 20**	(c)
p. 196	Change of **Caution** of **Table 11-3. Example of Relationship Between System Clock and Baud Rate**	(c)

Remark "Classification" in the above table classifies revisions as follows.
(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note, (d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

(2/2)

Page	Description	Classification
CHAPTER 11 SERIAL INTERFACE 20		
p. 200	Addition of **Caution 2** to **(a)** of **11.4.2 Asynchronous serial interface (UART) mode**	(c)
p. 203	Change of **Cautions 2 and 3** in **(d)** of **11.4.2 Asynchronous serial interface (UART) mode**	(c)
p. 204	Change of **Caution** of **Table 11-5. Example of Relationship Between System Clock and Baud Rate**	(c)
p. 212	Addition of **Cautions 2 and 3** to **(a)** of **11.4.3 3-wire serial I/O mode**	(c)
CHAPTER 15 REMOTE CONTROLLER RECEIVER		
p. 271	Change of **Figure 15-1. Block Diagram of Remote Controller Receiver**	(c)
p. 277	Modification of **Figure 15-3. Format of Remote Controller Receive Control Register (2/2)**	(a)
CHAPTER 19 FLASH MEMORY VERSION		
p. 319	Change of and addition of **Note 6** to **Table 19-2. Communication Mode List**	(b, c)
p. 326	Modification of **Figure 19-9. Wiring Example for Flash Writing Adapter with 3-Wire Serial I/O with Handshake**	(a)
CHAPTER 22 ELECTRICAL SPECIFICATIONS		
p. 340	Addition of formal spec of μPD789479, 78F9479	(b)
CHAPTER 24 RECOMMENDED SOLDERING CONDITIONS		
p. 361	Revision of chapter	(d)
CHAPTER 23 ELECTRICAL SPECIFICATIONS (TARGET) (μPD789479, 78F9479)		
Old edition	Deletion of chapter	(b)

Remark “Classification” in the above table classifies revisions as follows.
(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note, (d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

D.2 Revision History of Preceding Editions

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.

(1/2)

Edition	Major Revision from Previous Edition	Applied to:
2nd	Addition of μPD789478	Throughout
	Change of V_{PP} pin handling	**CHAPTER 2 PIN FUNCTIONS**
	Addition of **Figure 3-2 Memory Map (μPD789478)** and **Figure 3-5 Data Memory Addressing (μPD789478)**	**CHAPTER 3 CPU ARCHITECTURE**
	Change of block diagrams of P23 and P24	**CHAPTER 4 PORT FUNCTIONS**
	Addition of **Note** on feedback resistor	**CHAPTER 5 CLOCK GENERATOR**
	Modification of description on **6.4.1 Operation as timer interrupt** and **6.4.2 Operation as timer output**	**CHAPTER 6 16-BIT TIMER 20**
	Correction of bit name of bit 0 of timer mode control registers 60 and 61 (TMC60, TMC61)	**CHAPTER 7 8-BIT TIMERS 50, 60, AND 61**
	Addition of **Caution** on carrier generator output control register 60 (TCA60)	
	Correction of values in **Table 7-8 Square-Wave Output Range of Timer 61**	
	Addition of **10.5 (8) Input impedance of ANI0 to ANI7 pins**	**CHAPTER 10 10-BIT A/D CONVERTER**
	Modification of **Figure 11-1 Block Diagram of Serial Interface 20**	**CHAPTER 11 SERIAL INTERFACE 20**
	Modification of description on PE20 flag in **Figure 11-5 Format of Asynchronous Serial Interface Status Register 20**	
	Addition of description on UART receive data read	
	Change of **Figure 13-2 LCD Controller/Driver Block Diagram**	**CHAPTER 13 LCD CONTROLLER/DRIVER**
	Revision of contents about flash memory programming as **19.1 Flash Memory Characteristics**	**CHAPTER 19 μPD78F9478**
	Addition of electrical specifications	**CHAPTER 21 ELECTRICAL SPECIFICATIONS**
	Addition of package drawings	**CHAPTER 23 PACKAGE DRAWINGS**
	Addition of recommended soldering conditions	**CHAPTER 24 RECOMMENDED SOLDERING CONDITIONS**
	Revision of development tools Deletion of description on embedded software	**APPENDIX A DEVELOPMENT TOOLS**
	Addition of revision history	**APPENDIX C REVISION HISTORY**

(2/2)

Edition	Major Revision from Previous Edition	Applied to:
3rd	Addition of μPD789479 and 78F9479 Addition of 80-pin plastic TQFP (fine pitch) (12 ×12)	Throughout
	Update of series lineup diagram in **1.5 78K/0S Series Lineup**	**CHAPTER 1 GENERAL**
	Addition of **Table 3-3 Internal High-Speed RAM, Internal Low-Speed RAM Capacity**	**CHAPTER 3 CPU ARCHITECTURE**
	Modification of description of minimum instruction execution time in **Figure 5-3. Format of Processor Clock Control Register** and **Figure 5-5. Format of Subclock Control Register** Addition of **5.4.6 Subsystem clock ×4 multiplication circuit**	**CHAPTER 5 CLOCK GENERATOR**
	Addition of **6.5 Cautions on Using 16-Bit Timer 20**	**CHAPTER 6 16-BIT TIMER 20**
	Modification of **Figure 13-2 LCD Controller/Driver Block Diagram** Addition of **13.8 Examples of LCD Drive Power Connections**	**CHAPTER 13 LCD CONTROLLER/DRIVER**
	Addition of description of key return mode register 01 (KRM01)	**CHAPTER 16 INTERRUPT FUNCTIONS**
	Modification of description of CPU Clock in **Table 19-2 Communication Mode List** Change of description of **Note 1** in **Figure 19-3 Example of Connection with Dedicated Flash Programmer**	**CHAPTER 19 FLASH MEMORY VERSION**
	Addition of chapter	**CHAPTER 23 ELECTRICAL SPECIFICATIONS (TARGET) (μPD789479, 78F9479)**
	Addition of Flashpro IV and FA-80GK-9EU to **A.4 Flash Memory Writing Tools** Modification of **A.5 Debugging Tools (Hardware)**	**APPENDIX A DEVELOPMENT TOOLS**
	Addition of appendix	**APPENDIX B NOTES ON TARGET SYSTEM DESIGN**

For further information,
please contact:

NEC Electronics Corporation
1753, Shimonumabe, Nakahara-ku,
Kawasaki, Kanagawa 211-8668,
Japan
Tel: 044-435-5111
http://www.necel.com/

[America]

NEC Electronics America, Inc.
2880 Scott Blvd.
Santa Clara, CA 95050-2554, U.S.A.
Tel: 408-588-6000
800-366-9782
http://www.am.necel.com/

[Europe]

NEC Electronics (Europe) GmbH
Arcadiastrasse 10
40472 Düsseldorf, Germany
Tel: 0211-65030
http://www.eu.necel.com/

Hanover Office
Podbielskistrasse 166 B
30177 Hannover
Tel: 0 511 33 40 2-0

Munich Office
Werner-Eckert-Strasse 9
81829 München
Tel: 0 89 92 10 03-0

Stuttgart Office
Industriestrasse 3
70565 Stuttgart
Tel: 0 711 99 01 0-0

United Kingdom Branch
Cygnus House, Sunrise Parkway
Linford Wood, Milton Keynes
MK14 6NP, U.K.
Tel: 01908-691-133

Succursale Française
9, rue Paul Dautier, B.P. 52
78142 Velizy-Villacoublay Cédex
France
Tel: 01-3067-5800

Sucursal en España
Juan Esplandiu, 15
28007 Madrid, Spain
Tel: 091-504-2787

Tyskland Filial
Täby Centrum
Entrance S (7th floor)
18322 Täby, Sweden
Tel: 08 638 72 00

Filiale Italiana
Via Fabio Filzi, 25/A
20124 Milano, Italy
Tel: 02-667541

Branch The Netherlands
Steijgerweg 6
5616 HS Eindhoven
The Netherlands
Tel: 040 265 40 10

[Asia & Oceania]

NEC Electronics (China) Co., Ltd
7th Floor, Quantum Plaza, No. 27 ZhiChunLu Haidian
District, Beijing 100083, P.R.China
Tel: 010-8235-1155
http://www.cn.necel.com/

Shanghai Branch
Room 2509-2510, Bank of China Tower,
200 Yincheng Road Central,
Pudong New Area, Shanghai, P.R.China P.C:200120
Tel:021-5888-5400
http://www.cn.necel.com/

Shenzhen Branch
Unit 01, 39/F, Excellence Times Square Building,
No. 4068 Yi Tian Road, Futian District, Shenzhen,
P.R.China P.C:518048
Tel:0755-8282-9800
http://www.cn.necel.com/

NEC Electronics Hong Kong Ltd.
Unit 1601-1613, 16/F., Tower 2, Grand Century Place,
193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: 2886-9318
http://www.hk.necel.com/

NEC Electronics Taiwan Ltd.
7F, No. 363 Fu Shing North Road
Taipei, Taiwan, R. O. C.
Tel: 02-8175-9600
http://www.tw.necel.com/

NEC Electronics Singapore Pte. Ltd.
238A Thomson Road,
#12-08 Novena Square,
Singapore 307684
Tel: 6253-8311
http://www.sg.necel.com/

NEC Electronics Korea Ltd.
11F., Samik Lavied'or Bldg., 720-2,
Yeoksam-Dong, Kangnam-Ku,
Seoul, 135-080, Korea
Tel: 02-558-3737
http://www.kr.necel.com/

G0706