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

μPD789306, 789316 Subseries

8-Bit Single-Chip Microcontrollers

μPD789304
μPD789306
μPD789314
μPD789316
μPD78F9306
μPD78F9316

Document No. U14800EJ3V0UD00 (3rd edition)
Date Published May 2005 NS CP(K)

[MEMO]

NOTES FOR CMOS DEVICES

① VOLTAGE APPLICATION WAVEFORM AT INPUT PIN

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

② HANDLING OF UNUSED INPUT PINS

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

③ PRECAUTION AGAINST ESD

A strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it when it has occurred. Environmental control must be adequate. When it is dry, a humidifier should be used. It is recommended to avoid using insulators that easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors should be grounded. The operator should be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with mounted semiconductor devices.

④ STATUS BEFORE INITIALIZATION

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

⑤ POWER ON/OFF SEQUENCE

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

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

⑥ INPUT OF SIGNAL DURING POWER OFF STATE

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

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

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

- **The information in this document is current as of May, 2005. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC Electronics sales representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC Electronics does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.
- NEC Electronics products are classified into the following three quality grades: "Standard", "Special" and "Specific".

  The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of an NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product before using it in a particular application.

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

  "Special": Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support).

  "Specific": Aircraft, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems and medical equipment for life support, etc.

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

(Note)

(1) "NEC Electronics" as used in this statement means NEC Electronics Corporation and also includes its majority-owned subsidiaries.

(2) "NEC Electronics products" means any product developed or manufactured by or for NEC Electronics (as defined above).

M8E 02. 11-1

Regional Information

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

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

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

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

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

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

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

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

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

- **Branch The Netherlands**
  Eindhoven, The Netherlands
  Tel: 040-244 58 45

- **Tyskland Filial**
  Taeby, Sweden
  Tel: 08-63 80 820

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

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

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

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

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

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

J04.1

INTRODUCTION

Target Readers

This manual is intended to give user engineers an understanding of the functions of the μPD789306 and μPD789316 Subseries to design and develop its application systems and programs.

Target products:

- μPD789306 Subseries: μPD789304, 789306, 78F9306
- μPD789316 Subseries: μPD789314, 789316, 78F9316

For the main system clock frequency, f_X is applied to ceramic/crystal oscillation (μPD789306 Subseries) and f_{CC} is applied to RC oscillation (μPD789316 Subseries).

Purpose

This manual is designed to deepen your understanding of the following functions using the following organization.

Organization

Two manuals are available for the μPD789306 and μPD789316 Subseries:
This manual and the instruction manual (common to the 78K/0S Series).

μPD789306, 789316 Subseries User's Manual	78K/0S Series User's Manual Instructions
• Pin functions	• CPU function
• Internal block functions	• Instruction set
• Interrupts	• Instruction description
• Other internal peripheral functions	
• Electrical specifications	

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 μPD789306 and μPD789316 Subseries
  → Read this manual in the order of the **CONTENTS**.
  The mark ★ shows major revised points.
- 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 in the C compiler.
- To learn the detailed functions of a register whose register name is known
  → See **APPENDIX C**.
- 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 the electrical specifications of the μPD789306 and μPD789316 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.
μPD789306, 789316 Subseries User's Manual	This manual
78K/0S Series Instructions User's Manual	U11047E

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

Document Name		Document No.
RA78K0S Assembler Package	Operation	U17391E
	Language	U17390E
	Structured Assembly Language	U17389E
CC78K0S C Compiler	Operation	U16654E
	Language	U16655E
SM+ System Simulator	Operation	U17246E
	User Open Interface	U17247E
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 plus Ver.5.20		U16934E

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

Document Name	Document No.
IE-78K0S-NS In-Circuit Emulator	U13549E
IE-78K0S-NS-A In-Circuit Emulator	U15207E
IE-789306-NS-EM1 Emulation Board	U16115E

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

CHAPTER 1 GENERAL (μPD789306 SUBSERIES)

1.1 Features

- Main system clock: Ceramic/crystal oscillation
- Minimum instruction execution time can be changed from high-speed (0.4 μs: @ 5.0 MHz operation with main system clock) to ultra-low-speed (122 μs: @ 32.768 kHz operation with subsystem clock)
- ROM and RAM capacities

Item / Part Number	Program Memory (ROM)		Data Memory	
			Internal High-Speed RAM	LCD Display RAM
μPD789304	Mask ROM	8 KB	512 bytes	24 × 4 bits
μPD789306		16 KB		
μPD78F9306	Flash memory	16 KB		

- I/O ports: 23
- Serial interface: 2 channels
  Switchable between 3-wire serial I/O mode and UART mode: 1 channel
  3-wire serial I/O mode: 1 channel
- Timer: 5 channels
  - 16-bit timer: 1 channel
  - 8-bit timer/event counter: 2 channels
  - Watch timer: 1 channel
  - Watchdog timer: 1 channel
- LCD controller/driver
  Segment signals: 24, common signals: 4
- Vectored interrupt sources: 15
- Power supply voltage: V_{DD} = 1.8 to 5.5 V
- Operating ambient temperature: T_A = –40 to +85°C

1.2 Applications

Remote controllers, healthcare equipment, etc.

1.3 Ordering Information

	Part Number	Package	Internal ROM
	μPD789304GC-×××-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
	μPD789304GK-×××-9ET	64-pin plastic TQFP (12 × 12 mm)	Mask ROM
★	μPD789304GK-×××-9ET-A	64-pin plastic TQFP (12 × 12 mm)	Mask ROM
★	μPD789304GC-×××-8BS-A [Note]	64-pin plastic LQFP (14 × 14 mm)	Mask ROM
	μPD789306GC-×××-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
	μPD789306GK-×××-9ET	64-pin plastic TQFP (12 × 12 mm)	Mask ROM
★	μPD789306GK-×××-9ET-A	64-pin plastic TQFP (12 × 12 mm)	Mask ROM
★	μPD789306GC-×××-8BS-A [Note]	64-pin plastic LQFP (14 × 14 mm)	Mask ROM
	μPD78F9306GC-AB8	64-pin plastic QFP (14 × 14 mm)	Flash memory
	μPD78F9306GK-9ET	64-pin plastic TQFP (12 × 12 mm)	Flash memory
★	μPD78F9306GK-9ET-A	64-pin plastic TQFP (12 × 12 mm)	Flash memory
★	μPD78F9306GC-8BS-A [Note]	64-pin plastic LQFP (14 × 14 mm)	Flash memory

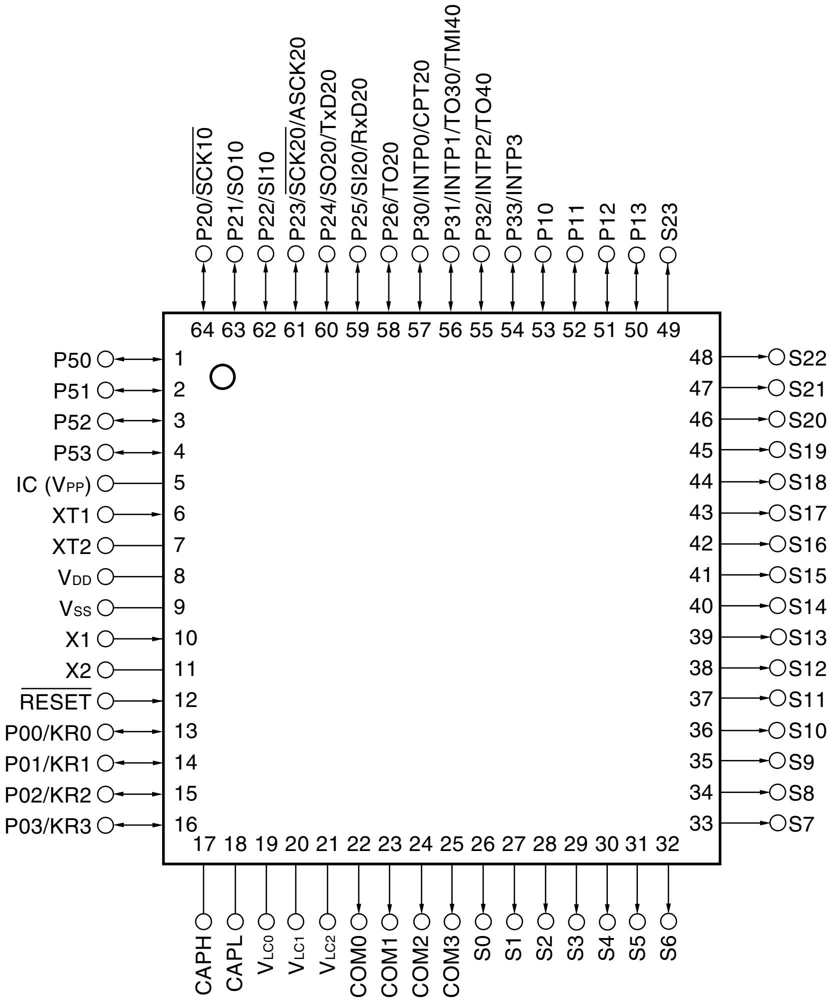
Note To be development.

Remarks 1. ××× indicates ROM code suffix.

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

1.4 Pin Configuration (Top View)

- 64-pin plastic QFP (14 × 14 mm)
  μPD789304GC-×××-AB8
  μPD789306GC-×××-AB8
  μPD78F9306GC-AB8
- ★ 64-pin plastic LQFP (14 × 14 mm)
  ★ μPD789304GC-×××-8BS-A Note
  ★ μPD789306GC-×××-8BS-A Note
  ★ μPD78F9306GC-8BS-A Note
- 64-pin plastic TQFP (12 × 12 mm)
  μPD789304GK-×××-9ET
  μPD789306GK-×××-9ET
  μPD78F9306GK-9ET
  μPD789304GK-×××-9ET-A
  μPD789306GK-×××-9ET-A
  μPD78F9306GK-9ET-A



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

Note To be development.

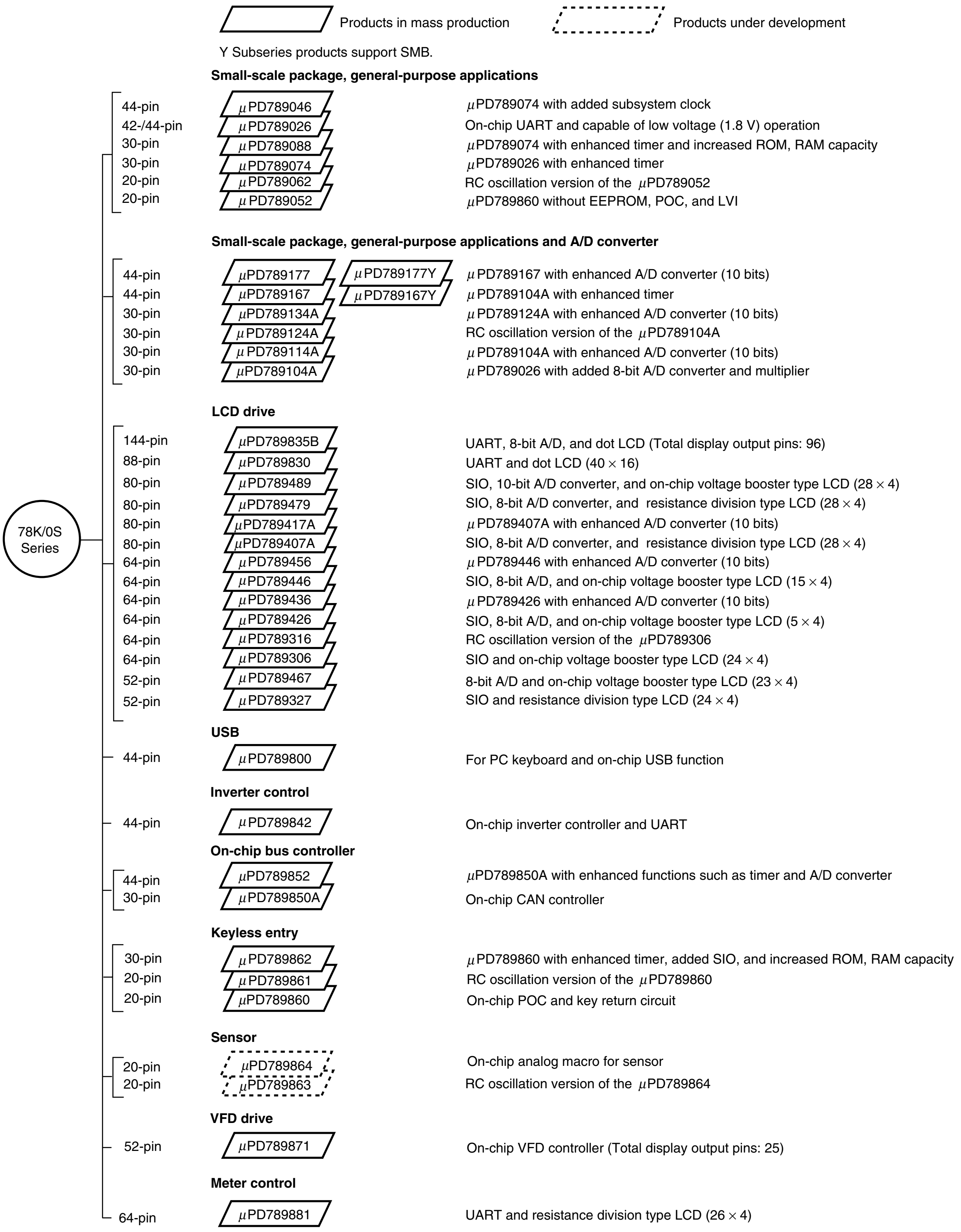
Remarks **1.** The parenthesized values apply to μPD78F9306.

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

ASCK20:	Asynchronous Serial Input
CAPH, CAPL:	LCD Power Supply Capacitance Control
COM0 to COM3:	Common Output
CPT20:	Capture Trigger Input
IC:	Internally Connected
INTP0 to INTP3:	External interrupt Input
KR0 to KR3:	Key Return
P00 to P03:	Port 0
P10 to P13:	Port 1
P20 to P26:	Port 2
P30 to P33:	Port 3
P50 to P53:	Port 5
$\overline{\text{RESET}}$:	Reset
RxD20:	Receive Data
S0 to S23:	Segment Output
$\overline{\text{SCK10}}$, $\overline{\text{SCK20}}$:	Serial Clock
SI10, SI20:	Serial Input
SO10, SO20:	Serial Output
TMI40:	Timer Input
TO20, TO30, TO40:	Timer Output
TxD20:	Transmit Data
V_{DD}:	Power Supply
V_{LC0} to V_{LC2}:	LCD Power Supply
V_{PP}:	Programming Power Supply
V_{SS}:	Ground
X1, X2:	Crystal/ceramic Oscillator
XT1, XT2:	Crystal Oscillator

★ 1.5 78K/0S Series Lineup

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



Y Subseries products support SMB.

Small-scale package, general-purpose applications

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

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

44-pin	μPD789177	μPD789177Y	μPD789167 with enhanced A/D converter (10 bits)
44-pin	μPD789167	μPD789167Y	μPD789104A with enhanced timer
30-pin	μPD789134A		μPD789124A with enhanced A/D converter (10 bits)
30-pin	μPD789124A		RC oscillation version of the μPD789104A
30-pin	μPD789114A		μPD789104A with enhanced A/D converter (10 bits)
30-pin	μPD789104A		μPD789026 with added 8-bit A/D converter and multiplier

LCD drive

144-pin	μPD789835B	UART, 8-bit A/D, and dot LCD (Total display output pins: 96)
88-pin	μPD789830	UART and dot LCD (40 × 16)
80-pin	μPD789489	SIO, 10-bit A/D converter, and on-chip voltage booster type LCD (28 × 4)
80-pin	μPD789479	SIO, 8-bit A/D converter, and resistance division type LCD (28 × 4)
80-pin	μPD789417A	μPD789407A with enhanced A/D converter (10 bits)
80-pin	μPD789407A	SIO, 8-bit A/D converter, and resistance division type LCD (28 × 4)
64-pin	μPD789456	μPD789446 with enhanced A/D converter (10 bits)
64-pin	μPD789446	SIO, 8-bit A/D, and on-chip voltage booster type LCD (15 × 4)
64-pin	μPD789436	μPD789426 with enhanced A/D converter (10 bits)
64-pin	μPD789426	SIO, 8-bit A/D, and on-chip voltage booster type LCD (5 × 4)
64-pin	μPD789316	RC oscillation version of the μPD789306
64-pin	μPD789306	SIO and on-chip voltage booster type LCD (24 × 4)
52-pin	μPD789467	8-bit A/D and on-chip voltage booster type LCD (23 × 4)
52-pin	μPD789327	SIO and resistance division type LCD (24 × 4)

USB

44-pin	μPD789800	For PC keyboard and on-chip USB function

Inverter control

44-pin	μPD789842	On-chip inverter controller and UART

On-chip bus controller

44-pin	μPD789852	μPD789850A with enhanced functions such as timer and A/D converter
30-pin	μPD789850A	On-chip CAN controller

Keyless entry

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

Sensor

20-pin	μPD789864	On-chip analog macro for sensor
20-pin	μPD789863	RC oscillation version of the μPD789864

VFD drive

52-pin	μPD789871	On-chip VFD controller (Total display output pins: 25)

Meter control

64-pin	μPD789881	UART and resistance division type LCD (26 × 4)

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

The major functional differences between the subseries are listed below.

Series for General-purpose applications and LCD drive

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

Note Flash memory version: 3.0 V

Series for ASSP

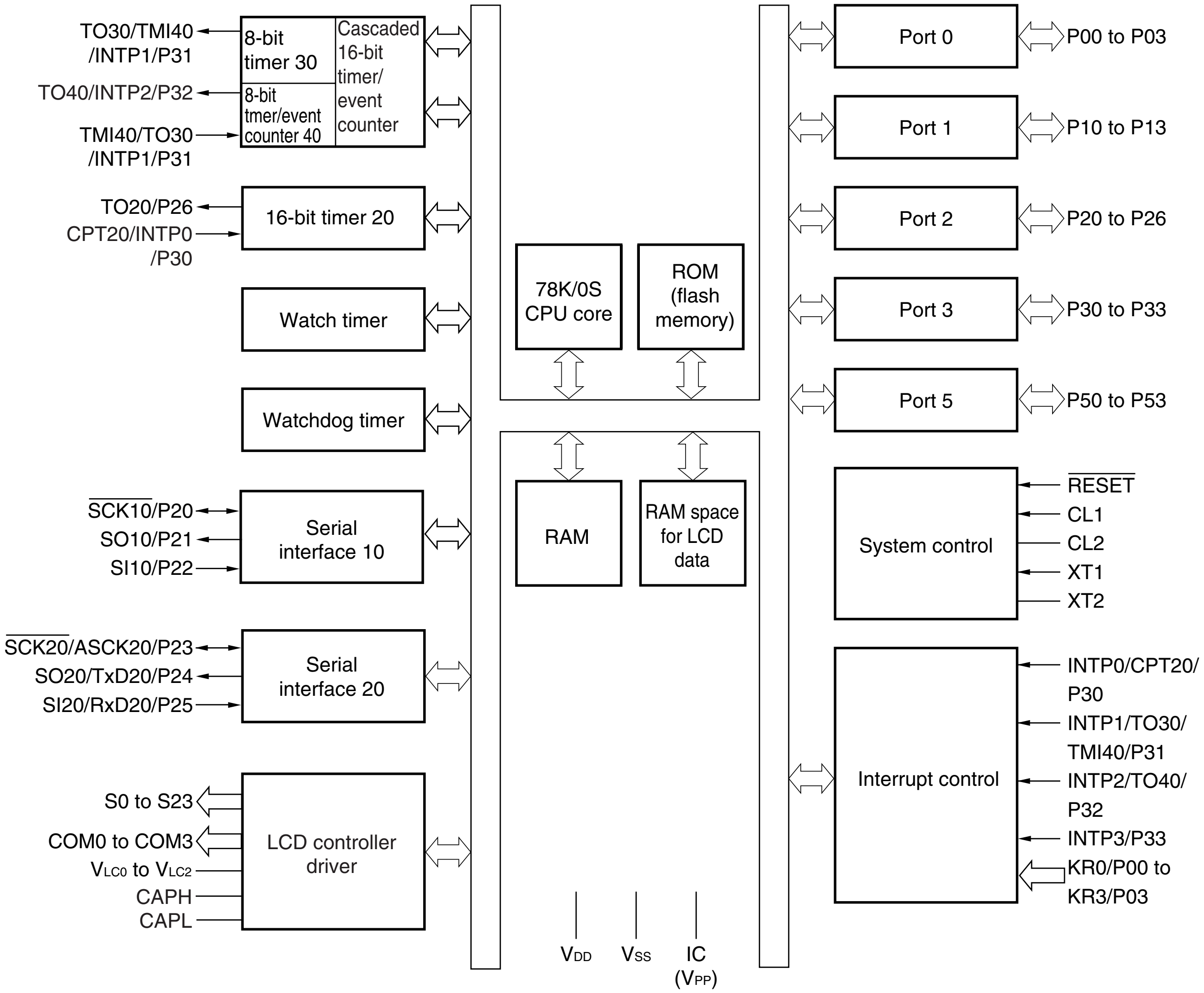
Function / Subseries Name		ROM Capacity	Timer				8-Bit A/D	10-Bit A/D	Serial Interface	I/O	V_{DD}	Remarks
			8-Bit	16-Bit	Watch	WDT					MIN. Value	
USB	μPD789800	8 KB	2 ch	–	–	1 ch	–	–	2 ch (USB: 1 ch)	31	4.0 V	–
Inverter control	μPD789842	8 KB to 16 KB	3 ch	Note 1	1 ch	1 ch	8 ch	–	1 ch (UART: 1 ch)	30	4.0 V	–
On-chip bus controller	μPD789852	24 KB to 32 KB	3 ch	1 ch	–	1 ch	–	8 ch	3 ch (UART: 2 ch)	31	4.0 V	–
	μPD789850A	16 KB	1 ch				4 ch	–	2 ch (UART: 1 ch)	18		
Keyless entry	μPD789861	4 KB	2 ch	–	–	1 ch	–	–	–	14	1.8 V	RC oscillation version, on-chip EEPROM
	μPD789860											On-chip EEPROM
	μPD789862	16 KB	1 ch	2 ch					1 ch (UART: 1 ch)	22		
Sensor	μPD789864	4 KB	1 ch	Note 2	–	1 ch	–	4 ch	–	5	1.9 V	On-chip EEPROM
	μPD789863											RC oscillation version, on-chip EEPROM
VFD drive	μPD789871	4 KB to 8 KB	3 ch	–	1 ch	1 ch	–	–	1 ch	33	2.7 V	–
Meter control	μPD789881	16 KB	2 ch	1 ch	–	1 ch	–	–	1 ch (UART: 1 ch)	28	2.7 V[Note 3]	–

Notes 1. 10-bit timer: 1 channel

2. 12-bit timer: 1 channel

3. Flash memory version: 3.0 V

1.6 Block Diagram



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

2. The parenthesized values apply to μPD78F9306.

1.7 Overview of Functions

Part Number Item		µPD789304	µPD789306	µPD78F9306
Internal memory	ROM	Mask ROM		Flash memory
		8 KB	16 KB	16 KB
	High-speed RAM	512 bytes		
	LCD display RAM	24 × 4 bits		
System clock		Ceramic/crystal oscillation		
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)		
General-purpose registers		8 bits × 8 registers		
Instruction set		• 16-bit operations • Bit manipulations (such as set, reset, and test)		
Multiplier		8 bits × 8 bits = 16 bits		
I/O ports		Total: 23 • CMOS I/O: 19 • N-ch open-drain: 4		
Serial interfaces		• Switchable between 3-wire serial I/O mode and UART mode: 1 channel • 3-wire serial I/O mode: 1 channel		
Timers		• 16-bit timer: 1 channel • 8-bit timer/event counter: 2 channels • Watch timer: 1 channel • Watchdog timer: 1 channel		
Timer outputs		3		
LCD controller/driver		• Segment signal outputs: 24 max. • Common signal outputs: 4 max.		
Vectored interrupt sources	Maskable	Internal: 9, external: 5		
	Non-maskable	Internal: 1		
Power supply voltage		V_{DD} = 1.8 to 5.5 V		
Operating ambient temperature		T_A = –40 to +85°C		
Package		• 64-pin plastic QFP (14 × 14 mm) • 64-pin plastic TQFP (12 × 12 mm) ★ • 64-pin plastic LQFP (14 × 14 mm) (To be development)		

An outline of the timer is shown below.

		16-Bit Timer 20	8–Bit Timer 30	8-Bit Timer/Event Counter 40	Watch Timer	Watchdog Timer
Operation mode	Interval timer	–	1 channel	1 channel	1 channel[Note 1]	1 channel[Note 2]
	External event counter	–	–	1 channel	–	–
Function	Timer outputs	1	1	1	–	–
	Square-wave outputs	–	1	1	–	–
	Capture	1 input	–	–	–	–
	Interrupt sources	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 the watchdog timer and interval timer functions. However, use the watchdog timer by selecting either the watchdog timer function or interval timer function.

CHAPTER 2 GENERAL (μPD789316 SUBSERIES)

2.1 Features

- Main system clock: RC oscillation
- Minimum instruction execution time can be changed from high-speed (0.5 μs: @ 4.0 MHz operation with main system clock) to ultra-low-speed (122 μs: @ 32.768 kHz operation with subsystem clock)
- ROM and RAM capacities

Item Part Number	Program Memory (ROM)		Data Memory	
			Internal High-Speed RAM	LCD Display RAM
μPD789314	Mask ROM	8 KB	512 bytes	24 × 4 bits
μPD789316		16 KB		
μPD78F9316	Flash memory	16 KB		

- I/O ports: 23
- Serial interface: 2 channels

  Switchable between 3-wire serial I/O mode and UART mode: 1 channel
  3-wire serial I/O mode: 1 channel
- Timer: 5 channels
  - 16-bit timer: 1 channel
  - 8-bit timer/event counter: 2 channels
  - Watch timer: 1 channel
  - Watchdog timer: 1 channel
- LCD controller/driver

  Segment signals: 24, common signals: 4
- Vectored interrupt sources: 15
- Power supply voltage: V_{DD} = 1.8 to 5.5 V
- Operating ambient temperature: T_A = –40 to +85°C

2.2 Applications

Remote controllers, healthcare equipment, etc.

2.3 Ordering Information

	Part Number	Package	Internal ROM
	μPD789314GC-×××-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
	μPD789314GK-×××-9ET	64-pin plastic TQFP (12 × 12 mm)	Mask ROM
★	μPD789314GK-×××-9ET-A	64-pin plastic TQFP (12 × 12 mm)	Mask ROM
★	μPD789314GC-×××-8BS-A Note	64-pin plastic LQFP (14 × 14 mm)	Mask ROM
	μPD789316GC-×××-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
	μPD789316GK-×××-9ET	64-pin plastic TQFP (12 × 12 mm)	Mask ROM
★	μPD789316GK-×××-9ET-A	64-pin plastic TQFP (12 × 12 mm)	Mask ROM
★	μPD789316GC-×××-8BS-A Note	64-pin plastic LQFP (14 × 14 mm)	Mask ROM
	μPD78F9316GC-AB8	64-pin plastic QFP (14 × 14 mm)	Flash memory
	μPD78F9316GK-9ET	64-pin plastic TQFP (12 × 12 mm)	Flash memory
★	μPD78F9316GK-9ET-A	64-pin plastic TQFP (12 × 12 mm)	Flash memory
★	μPD78F9316GC-8BS-A Note	64-pin plastic LQFP (14 × 14 mm)	Flash memory

Note To be development.

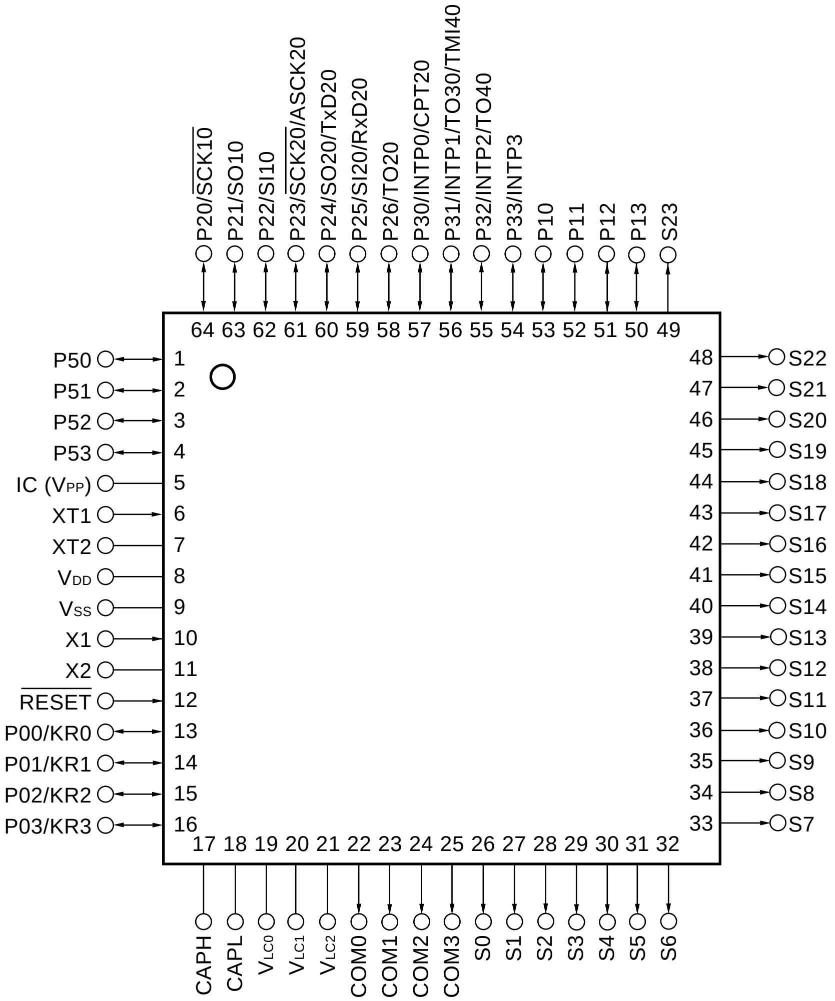
Remarks 1. ××× indicates ROM code suffix.

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

2.4 Pin Configuration (Top View)

- 64-pin plastic QFP (14 × 14 mm)
  μPD789314GC-×××-AB8
  μPD789316GC-×××-AB8
  μPD78F9316GC-AB8
- ★ 64-pin plastic LQFP (14 × 14 mm)
  ★ μPD789314GC-×××-8BS-A Note
  ★ μPD789316GC-×××-8BS-A Note
  ★ μPD78F9316GC-8BS-A Note

- 64-pin plastic TQFP (12 × 12 mm)
  μPD789314GK-×××-9ET
  μPD789316GK-×××-9ET
  μPD78F9316GK-9ET
  μPD789314GK-×××-9ET-A
  μPD789316GK-×××-9ET-A
  μPD78F9316GK-9ET-A



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

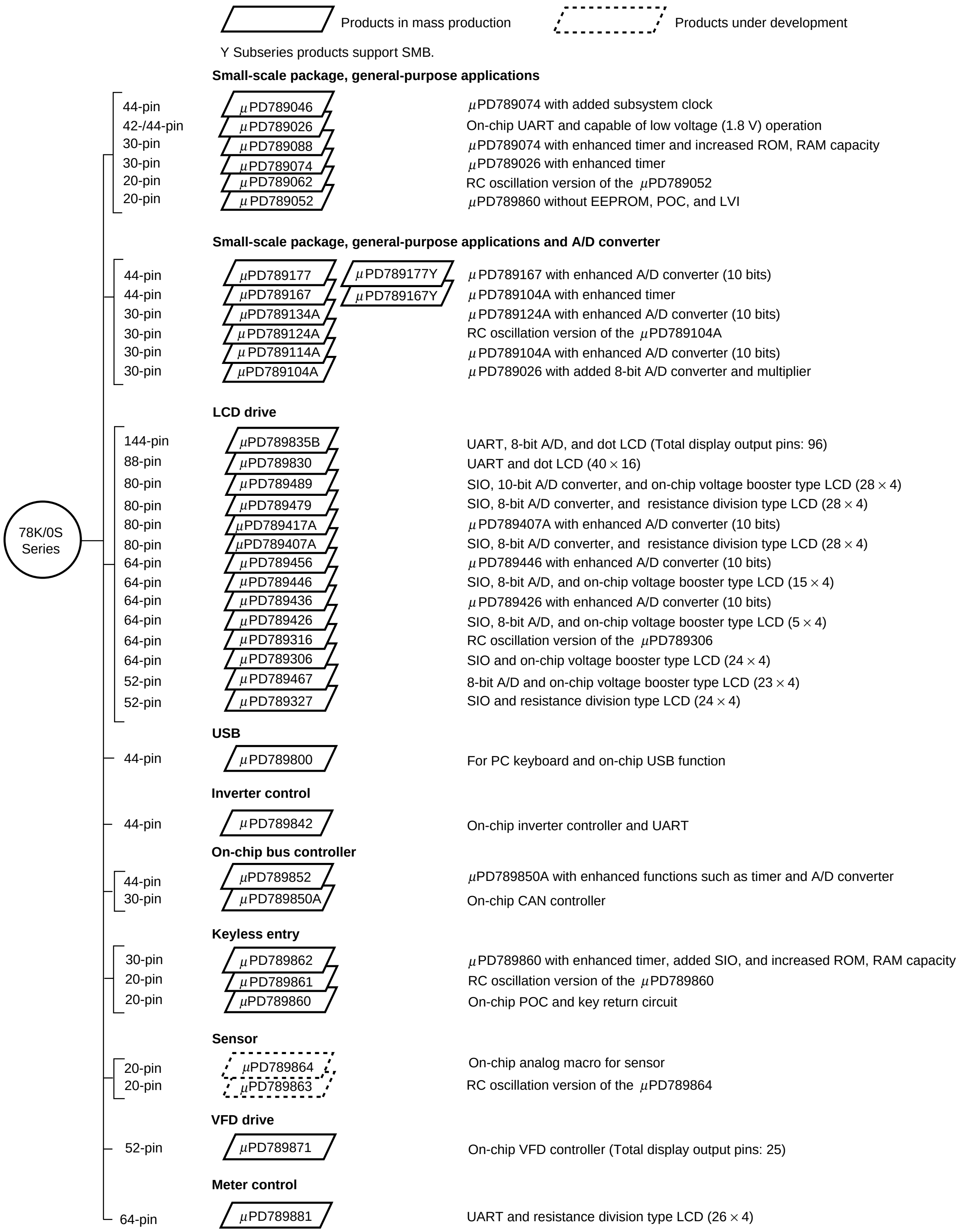
Note To be development.

Remark The parenthesized values apply to μPD78F9316.

ASCK20:	Asynchronous Serial Input	RxD20:	Receive Data
CAPH, CAPL:	LCD Power Supply Capacitance Control	S0 to S23:	Segment Output
CL1, CL2:	RC Oscillator	$\overline{\mathrm{SCK10}}$, $\overline{\mathrm{SCK20}}$:	Serial Clock
COM0 to COM3:	Common Output	SI10, SI20:	Serial Input
CPT20:	Capture Trigger Input	SO10, SO20:	Serial Output
IC:	Internally Connected	TMI40:	Timer Input
INTP0 to INTP3:	External interrupt Input	TO20, TO30, TO40:	Timer Output
KR0 to KR3:	Key Return	TxD20:	Transmit Data
P00 to P03:	Port 0	V_{DD}:	Power Supply
P10 to P13:	Port 1	V_{LC0} to V_{LC2}:	LCD Power Supply
P20 to P26:	Port 2	V_{PP}:	Programming Power Supply
P30 to P33:	Port 3	V_{SS}:	Ground
P50 to P53:	Port 5	XT1, XT2:	Crystal Oscillator
$\overline{\mathrm{RESET}}$:	Reset		

★ 2.5 78K/0S Series Lineup

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



Y Subseries products support SMB.

Small-scale package, general-purpose applications

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

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

44-pin	μPD789177	μPD789177Y	μPD789167 with enhanced A/D converter (10 bits)
44-pin	μPD789167	μPD789167Y	μPD789104A with enhanced timer
30-pin	μPD789134A		μPD789124A with enhanced A/D converter (10 bits)
30-pin	μPD789124A		RC oscillation version of the μPD789104A
30-pin	μPD789114A		μPD789104A with enhanced A/D converter (10 bits)
30-pin	μPD789104A		μPD789026 with added 8-bit A/D converter and multiplier

LCD drive

144-pin	μPD789835B	UART, 8-bit A/D, and dot LCD (Total display output pins: 96)
88-pin	μPD789830	UART and dot LCD (40 × 16)
80-pin	μPD789489	SIO, 10-bit A/D converter, and on-chip voltage booster type LCD (28 × 4)
80-pin	μPD789479	SIO, 8-bit A/D converter, and resistance division type LCD (28 × 4)
80-pin	μPD789417A	μPD789407A with enhanced A/D converter (10 bits)
80-pin	μPD789407A	SIO, 8-bit A/D converter, and resistance division type LCD (28 × 4)
64-pin	μPD789456	μPD789446 with enhanced A/D converter (10 bits)
64-pin	μPD789446	SIO, 8-bit A/D, and on-chip voltage booster type LCD (15 × 4)
64-pin	μPD789436	μPD789426 with enhanced A/D converter (10 bits)
64-pin	μPD789426	SIO, 8-bit A/D, and on-chip voltage booster type LCD (5 × 4)
64-pin	μPD789316	RC oscillation version of the μPD789306
64-pin	μPD789306	SIO and on-chip voltage booster type LCD (24 × 4)
52-pin	μPD789467	8-bit A/D and on-chip voltage booster type LCD (23 × 4)
52-pin	μPD789327	SIO and resistance division type LCD (24 × 4)

USB

44-pin	μPD789800	For PC keyboard and on-chip USB function

Inverter control

44-pin	μPD789842	On-chip inverter controller and UART

On-chip bus controller

44-pin	μPD789852	μPD789850A with enhanced functions such as timer and A/D converter
30-pin	μPD789850A	On-chip CAN controller

Keyless entry

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

Sensor

20-pin	μPD789864	On-chip analog macro for sensor
20-pin	μPD789863	RC oscillation version of the μPD789864

VFD drive

52-pin	μPD789871	On-chip VFD controller (Total display output pins: 25)

Meter control

64-pin	μPD789881	UART and resistance division type LCD (26 × 4)

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

The major functional differences between the subseries are listed below.

Series for General-purpose applications and LCD drive

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

Note Flash memory version: 3.0 V

Series for ASSP

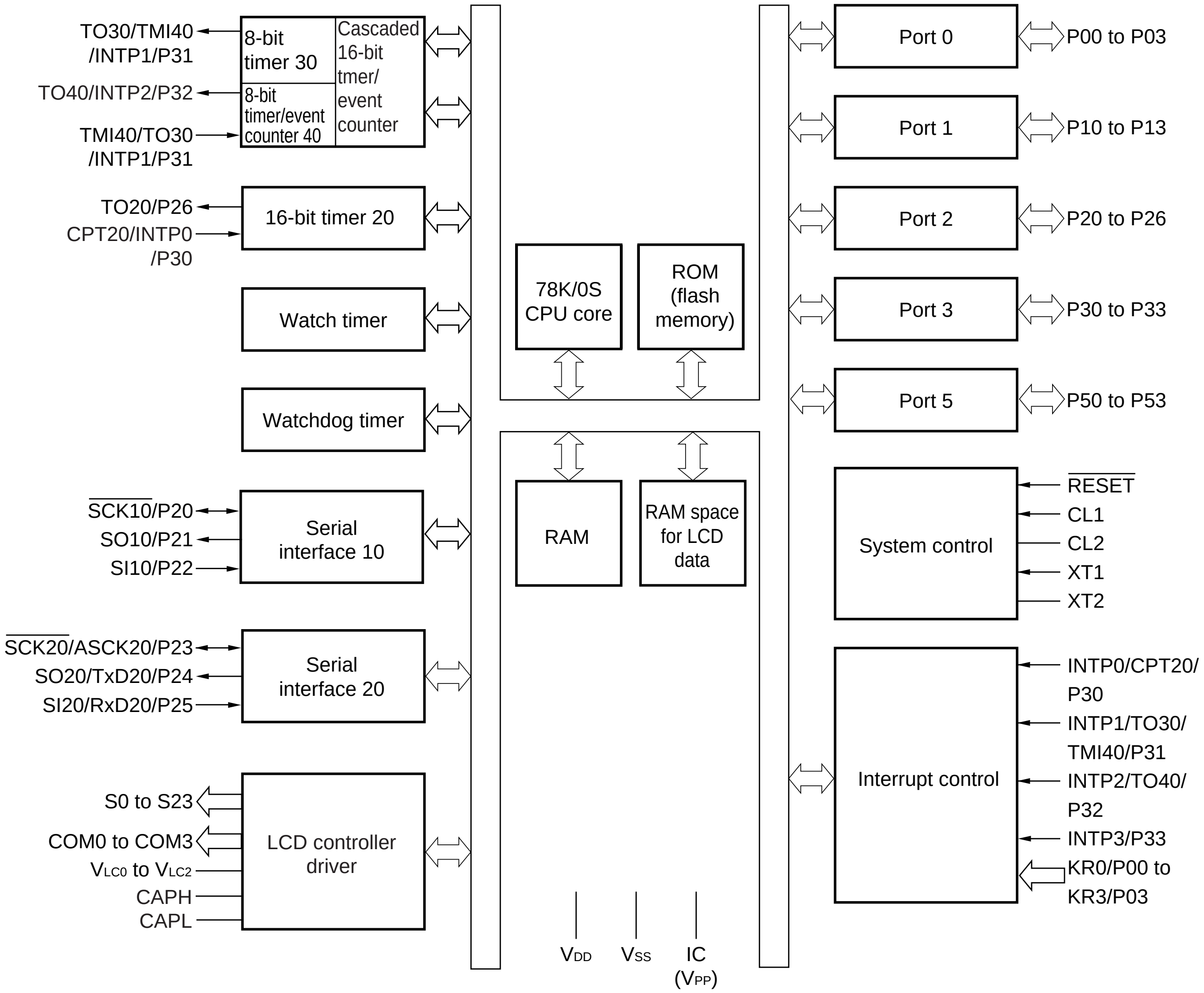
Function / Subseries Name		ROM Capacity	Timer				8-Bit A/D	10-Bit A/D	Serial Interface	I/O	V_{DD}	Remarks
			8-Bit	16-Bit	Watch	WDT					MIN. Value	
USB	μPD789800	8 KB	2 ch	–	–	1 ch	–	–	2 ch (USB: 1 ch)	31	4.0 V	–
Inverter control	μPD789842	8 KB to 16 KB	3 ch	Note 1	1 ch	1 ch	8 ch	–	1 ch (UART: 1 ch)	30	4.0 V	–
On-chip bus controller	μPD789852	24 KB to 32 KB	3 ch	1 ch	–	1 ch	–	8 ch	3 ch (UART: 2 ch)	31	4.0 V	–
	μPD789850A	16 KB	1 ch				4 ch	–	2 ch (UART: 1 ch)	18		
Keyless entry	μPD789861	4 KB	2 ch	–	–	1 ch	–	–	–	14	1.8 V	RC oscillation version, on-chip EEPROM
	μPD789860											On-chip EEPROM
	μPD789862	16 KB	1 ch	2 ch					1 ch (UART: 1 ch)	22		
Sensor	μPD789864	4 KB	1 ch	Note 2	–	1 ch	–	4 ch	–	5	1.9 V	On-chip EEPROM
	μPD789863											RC oscillation version, on-chip EEPROM
VFD drive	μPD789871	4 KB to 8 KB	3 ch	–	1 ch	1 ch	–	–	1 ch	33	2.7 V	–
Meter control	μPD789881	16 KB	2 ch	1 ch	–	1 ch	–	–	1 ch (UART: 1 ch)	28	2.7 V Note 3	–

Notes 1. 10-bit timer: 1 channel

2. 12-bit timer: 1 channel

3. Flash memory version: 3.0 V

2.6 Block Diagram



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

2. The parenthesized values apply to μPD78F9316.

2.7 Overview of Functions

Item	Part Number	μPD789314	μPD789316	μPD78F9316
Internal memory	ROM	Mask ROM		Flash memory
		8 KB	16 KB	16 KB
	High-speed RAM	512 bytes		
	LCD display RAM	24 × 4 bits		
System clock		RC oscillation		
Minimum instruction execution time		• 0.5 μs/2.0 μs (@ 4.0 MHz operation with main system clock) • 122 μs (@ 32.768 kHz operation with subsystem clock)		
General-purpose registers		8 bits × 8 registers		
Instruction set		• 16-bit operations • Bit manipulations (such as set, reset, and test)		
Multiplier		8 bits × 8 bits = 16 bits		
I/O ports		Total: 23 • CMOS I/O: 19 • N-ch open-drain: 4		
Serial interfaces		• Switchable between 3-wire serial I/O mode and UART mode: 1 channel • 3-wire serial I/O mode: 1 channel		
Timers		• 16-bit timer: 1 channel • 8-bit timer/event counter: 2 channels • Watch timer: 1 channel • Watchdog timer: 1 channel		
Timer outputs		3		
LCD controller/driver		• Segment signal outputs: 24 max. • Common signal outputs: 4 max.		
Vectored interrupt sources	Maskable	Internal: 9, external: 5		
	Non-maskable	Internal: 1		
Power supply voltage		V_{DD} = 1.8 to 5.5 V		
Operating ambient temperature		T_A = –40 to +85°C		
Package		• 64-pin plastic QFP (14 × 14 mm) • 64-pin plastic TQFP (12 × 12 mm) ★ • 64-pin plastic LQFP (14 × 14 mm) (To be development)		

An outline of the timer is shown below.

		16-Bit Timer 20	8–Bit Timer 30	8-Bit Timer/Event Counter 40	Watch Timer	Watchdog Timer
Operation mode	Interval timer	–	1 channel	1 channel	1 channelNote 1	1 channelNote 2
	External event counter	–	–	1 channel	–	–
Function	Timer outputs	1	1	1	–	–
	Square-wave outputs	–	1	1	–	–
	Capture	1 input	–	–	–	–
	Interrupt sources	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 the watchdog timer and interval timer functions. However, use the watchdog timer by selecting either the watchdog timer function or interval timer function.

3.1 List of Pin Functions

(1) Port pins

Pin Name	I/O	Function	After Reset	Alternate Function
P00 to P03	I/O	Port 0. 4-bit I/O port. Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be specified by means of pull-up resistor option register 0 (PU0) or key return mode register 00 (KRM00) in port units.	Input	KR0 to KR3
P10 to P13	I/O	Port 1. 4-bit I/O port. Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be specified by means of pull-up resistor option register 0 (PU0) in port units.	Input	–
P20	I/O	Port 2. 7-bit I/O port. Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be specified by means of pull-up resistor option register B2 (PUB2) in 1-bit units.	Input	$\overline{\text{SCK10}}$
P21				SO10
P22				SI10
P23				$\overline{\text{SCK20}}$/ASCK20
P24				SO20/TxD20
P25				SI20/RxD20
P26				TO20
P30	I/O	Port 3. 4-bit I/O port. Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be specified by means of pull-up resistor option register B3 (PUB3) in 1-bit units.	Input	INTP0/CPT20
P31				INTP1/TO30/TMI40
P32				INTP2/TO40
P33				INTP3
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 a mask ROM version, an on-chip pull-up resistor can be specified by the mask option in 1-bit units.	Input	–

(2) Non-port pins

Pin Name	I/O	Function	After Reset	Alternate Function
INTP0	Input	External interrupt input for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified	Input	P30/CPT20
INTP1				P31/TO30/TMI40
INTP2				P32/TO40
INTP3				P33
KR0 to KR3	Input	Key return signal detection	Input	P00 to P03
$\overline{\text{SCK10}}$	I/O	Serial interface 10 serial clock input/output	Input	P20
$\overline{\text{SCK20}}$		Serial interface 20 serial clock input/output		P23/ASCK20
SI10	Input	Serial interface 10 serial data input	Input	P22
SI20		Serial interface 20 serial data input		P25/RxD20
SO10	Output	Serial interface 10 serial data output	Input	P21
SO20		Serial interface 20 serial data output		P24/TxD20
ASCK20	Input	Serial clock input for asynchronous serial interface	Input	P23/$\overline{\text{SCK20}}$
RxD20	Input	Serial data input for asynchronous serial interface	Input	P25/SI20
TxD20	Output	Serial data output for asynchronous serial interface	Input	P24/SO20
TO20	Output	16-bit timer 20 output	Input	P26
CPT20	Input	Capture edge input	Input	P30/INTP0
TO30	Output	Timer 30 output	Input	P31/INTP1/TMI40
TO40	Output	Timer 40 output	Input	P32/INTP2
TMI40	Input	External count clock input to timer 40	Input	P31/INTP1/TO30
S0 to S23	Output	LCD controller/driver segment signal output	Output low level	–
COM0 to COM3	Output	LCD controller/driver common signal output	Output low level	–
V_{LC0} to V_{LC2}	–	LCD driving voltage	–	–
CAPH	–	Capacitor connection pin for LCD drive	–	–
CAPL	–		–	–
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	–	–
IC	–	Internally connected. Connect directly to V_{SS}.	–	–
V_{PP}	–	Sets flash memory programming mode. Applies high voltage when a program is written or verified.	–	–

3.2 Description of Pin Functions

3.2.1 P00 to P03 (Port 0)

These pins constitute a 4-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 a 4-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 0 (PU0) in port units.

(2) Control mode
In this mode, P00 to P03 function as key return signal detection pins (KR0 to KR3).

3.2.2 P10 to P13 (Port 1)

These pins constitute a 4-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 0 (PU0) in port units.

3.2.3 P20 to P26 (Port 2)

These pins constitute a 7-bit I/O port. In addition, these pins enable timer output, serial interface data I/O, and 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 P26 function as a 7-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 P26 function as the timer output, serial interface data I/O, and clock I/O.

(a) TO20
This is the timer output pin of 16-bit timer 20.

(b) SI10, SI20, SO10, SO20
These are the serial data I/O pins of the serial interface.

(c) $\overline{\text{SCK10}}$, $\overline{\text{SCK20}}$
These are the serial clock I/O pins of the serial interface.

(d) RxD20, TxD20
These are the serial data I/O pins of the asynchronous serial interface.

(e) ASCK20
This is the serial clock input pin of the asynchronous serial interface.

Caution When using P20 to P26 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 13-2 Settings of Serial Interface 10 Operating Mode and Table 14-2 Settings of Serial Interface 20 Operating Mode.

3.2.4 P30 to P33 (Port 3)

These pins constitute a 4-bit I/O port. In addition, they also function as timer I/O and external interrupt input.
Port 3 can be specified in the following operation mode in 1-bit units.

(1) Port mode

In this mode, port 3 functions as a 4-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, the pins function as timer I/O and external interrupt input.

(a) TMI40

This is the external clock input pin to timer 40.

(b) TO30, TO40

These are the timer output pins of timer 30 and timer 40

(c) 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.

3.2.5 P50 to P53 (Port 5)

These pins function as 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.

3.2.6 S0 to S23

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

3.2.7 COM0 to COM3

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

3.2.8 V_{LC0} to V_{LC2}

These pins are power supply voltage pins to drive the LCD.

3.2.9 CAPH, CAPL

These pins are capacitor connection pins to drive the LCD.

3.2.10 $\overline{\text{RESET}}$

This pin inputs an active-low system reset signal.

3.2.11 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.

3.2.12 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.

3.2.13 V_{DD}

This is the positive power supply pin.

3.2.14 V_{SS}

This is the ground pin.

3.2.15 V_{PP} (μPD78F9306 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.

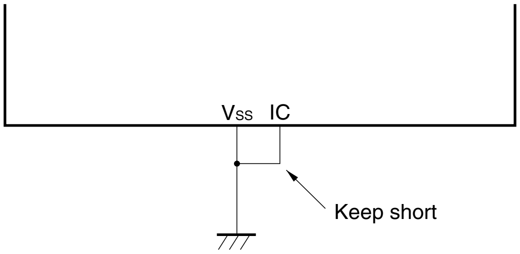
If the wiring between the V_{PP} pin and V_{SS} pin is long, or external noise is superimposed on the V_{PP} pin, the user program may not run correctly.

3.2.16 IC (mask ROM version only)

The IC (Internally Connected) pin is used to set the μPD789304 and μPD789306 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 a potential difference is generated between the IC pin and V_{SS} pin due to a long wiring length, or an external noise superimposed on the IC pin, the user program may not run correctly.

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



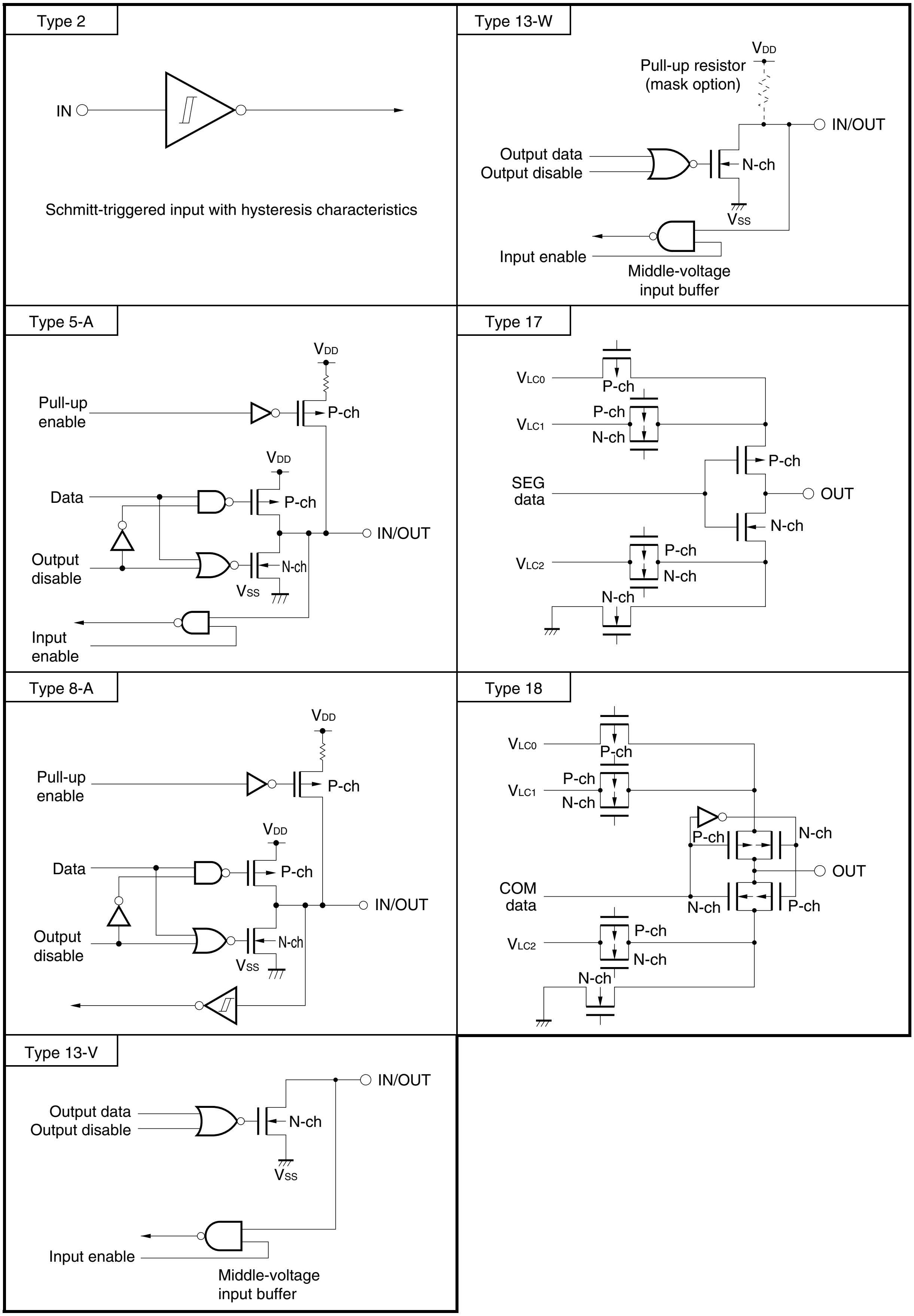
3.3 Pin Input/Output Circuits and Recommended Connection of Unused Pins

The input/output circuit type of each pin and recommended connection of unused pins are shown in Table 3-1. For the input/output circuit configuration of each type, see Figure 3-1.

Table 3-1. Types of Pin Input/Output Circuits

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00/KR0 to P03/KR3	8-A	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P10 to P13	5-A		
P20/$\overline{\text{SCK10}}$	8-A		
P21/SO10			
P22/SI10			
P23/$\overline{\text{SCK20}}$/ASCK20			
P24/SO20/TxD20			
P25/SI20/RxD20			
P26/TO20			
P30/INPT0/CPT20			Input: Independently connect to V_{SS} via a resistor. Output: Leave open.
P31/INPT1/TO30/TMI40			
P32/INPT2/TO40			
P33/INPT3			
P50 to P53 (Mask ROM version)	13-W		Input: Independently connect to V_{SS} via a resistor. Output: Leave this pin open at low-level output after clearing the output latch of the port to 0.
P50 to P53 (μPD78F9306)	13-V		
S0 to S23	17	Output	Leave open.
COM0 to COM3	18		
V_{LC0} to V_{LC2}	–	–	
CAPH, CAPL	–		
XT1	–	Input	Connect to V_{SS}.
XT2		–	Leave open.
$\overline{\text{RESET}}$	2	Input	–
IC (Mask ROM version)	–	–	Connect directly to V_{SS}.
V_{PP} (μPD78F9306)			Independently connect to a 10 kΩ pull-down resistor or connect directly to V_{SS}.

Figure 3-1. Pin Input/Output Circuits



4.1 List of Pin Functions

(1) Port pins

Pin Name	I/O	Function	After Reset	Alternate Function
P00 to P03	I/O	Port 0. 4-bit I/O port. Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be specified by means of pull-up resistor option register 0 (PU0) or key return mode register 00 (KRM00) in port units.	Input	KR0 to KR3
P10 to P13	I/O	Port 1. 4-bit I/O port. Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be specified by means of pull-up resistor option register 0 (PU0) in port units.	Input	–
P20	I/O	Port 2. 7-bit I/O port. Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be specified by means of pull-up resistor option register B2 (PUB2) in 1-bit units.	Input	$\overline{\text{SCK10}}$
P21				SO10
P22				SI10
P23				$\overline{\text{SCK20}}$/ASCK20
P24				SO20/TxD20
P25				SI20/RxD20
P26				TO20
P30	I/O	Port 3. 4-bit I/O port. Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be specified by means of pull-up resistor option register B3 (PUB3).	Input	INTP0/CPT20
P31				INTP1/TO30/TMI40
P32				INTP2/TO40
P33				INTP3
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 a mask ROM version, an on-chip pull-up resistor can be specified by the mask option in 1-bit units.	Input	–

(2) Non-port pins

Pin Name	I/O	Function	After Reset	Alternate Function
INTP0	Input	External interrupt input for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified	Input	P30/CPT20
INTP1				P31/TO30/TMI40
INTP2				P32/TO40
INTP3				P33
KR0 to KR3	Input	Key return signal detection	Input	P00 to P03
$\overline{\text{SCK10}}$	I/O	Serial interface 10 serial clock input/output	Input	P20
$\overline{\text{SCK20}}$		Serial interface 20 serial clock input/output		P23/ASCK20
SI10	Input	Serial interface 10 serial data input	Input	P22
SI20		Serial interface 20 serial data input		P25/RxD20
SO10	Output	Serial interface 10 serial data output	Input	P21
SO20		Serial interface 20 serial data output		P24/TxD20
ASCK20	Input	Serial clock input for asynchronous serial interface	Input	P23/$\overline{\text{SCK20}}$
RxD20	Input	Serial data input for asynchronous serial interface	Input	P25/SI20
TxD20	Output	Serial data output for asynchronous serial interface	Input	P24/SO20
TO20	Output	16-bit timer 20 output	Input	P26
CPT20	Input	Capture edge input	Input	P30/INTP0
TO30	Output	Timer 30 output	Input	P31/INTP1/TMI40
TO40	Output	Timer 40 output	Input	P32/INTP2
TMI40	Input	External count clock input to timer 40	Input	P31/INTP1/TO30
S0 to S23	Output	LCD controller/driver segment signal output	Output low level	–
COM0 to COM3	Output	LCD controller/driver common signal output	Output low level	–
V_{LC0} to V_{LC2}	–	LCD driving voltage	–	–
CAPH	–	Capacitor connection pin for LCD drive	–	–
CAPL	–		–	–
CL1	Input	Connecting resistor (R) and capacitor (C) for main system clock oscillation	–	–
CL2	–		–	–
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	–	–
IC	–	Internally connected. Connect directly to V_{SS}.	–	–
V_{PP}	–	Sets flash memory programming mode. Applies high voltage when a program is written or verified.	–	–

4.2 Description of Pin Functions

4.2.1 P00 to P03 (Port 0)

These pins constitute a 4-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 a 4-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 0 (PU0) in port units.

(2) Control mode

In this mode, P00 to P03 function as key return signal detection pins (KR0 to KR3). These pins constitute a 4-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 0 (PU0).

4.2.2 P10 to P13 (Port 1)

These pins constitute a 4-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 0 (PU0) in port units.

4.2.3 P20 to P26 (Port 2)

These pins constitute a 7-bit I/O port. In addition, these pins enable timer output, serial interface data I/O, and 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 P26 function as a 7-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 P26 function as the timer output, serial interface data I/O, and clock I/O.

(a) TO20

This is the timer output pin of 16-bit timer 20.

(b) SI10, SI20, SO10, SO20

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

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

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

(d) RxD20, TxD20

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

(e) ASCK20

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

Caution When using P20 to P26 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 13-2 Settings of Serial Interface 10 Operating Mode and Table 14-2 Settings of Serial Interface 20 Operating Mode.

4.2.4 P30 to P33 (Port 3)

These pins constitute a 4-bit I/O port. In addition, they also function as timer I/O and external interrupt input.
Port 3 can be specified in the following operation modes in 1-bit units.

(1) Port mode

In this mode, port 3 functions as a 4-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, the pins function as timer I/O and external interrupt input.

(a) TMI40

This is the external clock input pin to timer 40.

(b) TO30, TO40

These are the timer output pins of timer 30 and timer 40

(c) CPT20

This is the capture edge input pin for 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.

4.2.5 P50 to P53 (Port 5)

These pins function as 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.

4.2.6 S0 to S23

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

4.2.7 COM0 to COM3

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

4.2.8 V_{LC0} to V_{LC2}

These pins are power supply voltage pins to drive the LCD.

4.2.9 CAPH, CAPL

These pins are capacitor connection pins to drive the LCD.

4.2.10 $\overline{\text{RESET}}$

This pin inputs an active-low system reset signal.

4.2.11 CL1, CL2

These pins are used to connect a resistor (R) and capacitor (C) for main system clock oscillation.
To supply an external clock, input the clock to CL1 and input the inverted signal to CL2.

4.2.12 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.

4.2.13 V_{DD}

This is the positive power supply pin.

4.2.14 V_{SS}

This is the ground pin.

4.2.15 V_{PP} (μPD78F9316 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.

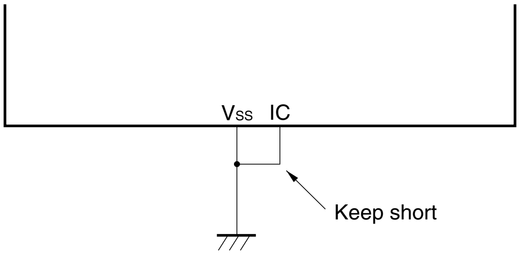
If the wiring between the V_{PP} pin and V_{SS} pin is long, or external noise is superimposed on the V_{PP} pin, the user program may not run correctly.

4.2.16 IC (mask ROM version only)

The IC (Internally Connected) pin is used to set the μPD789314 and μPD789316 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 a potential difference is generated between the IC pin and V_{SS} pin due to a long wiring length, or an external noise superimposed on the IC pin, the user program may not run correctly.

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



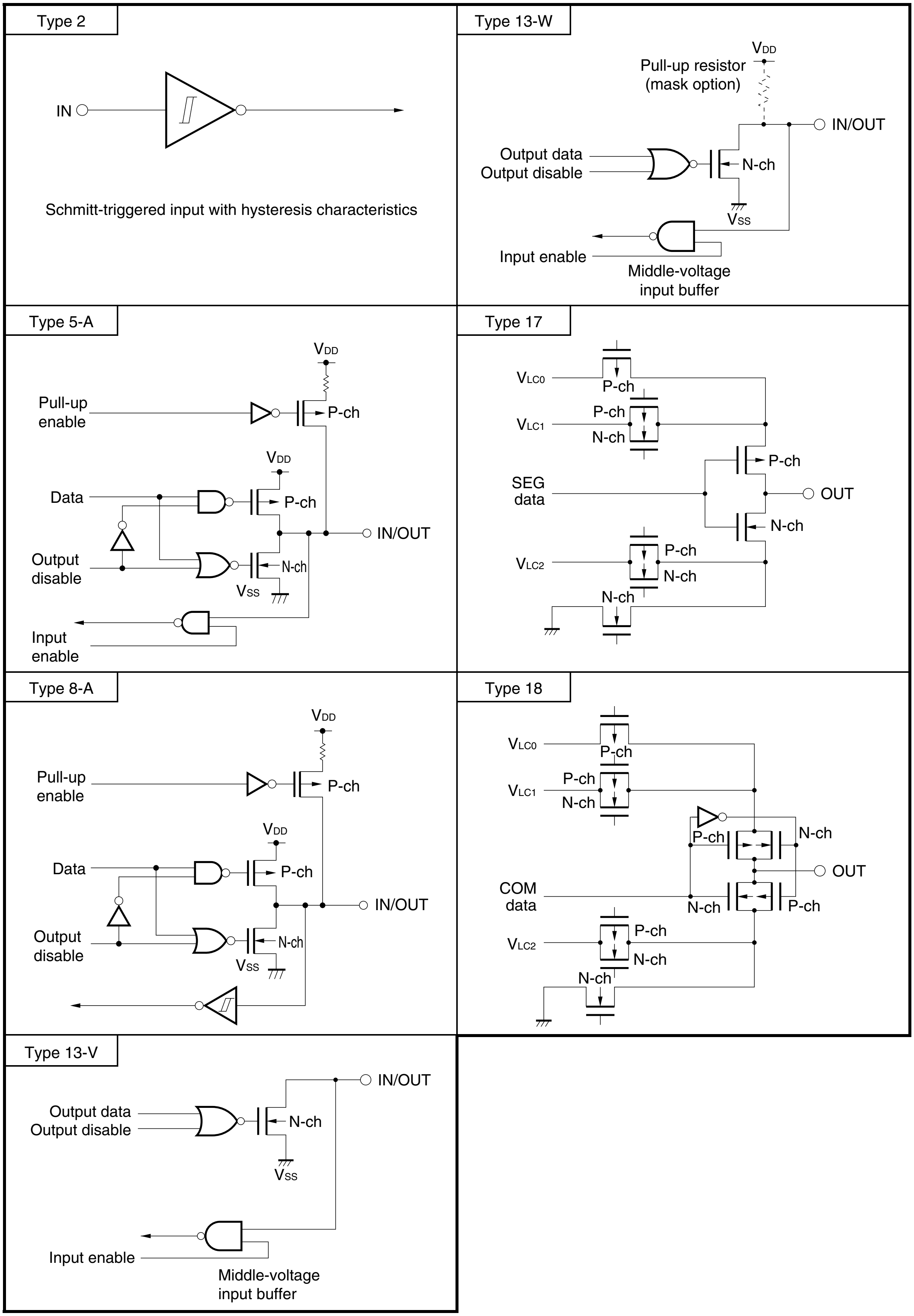
4.3 Pin Input/Output Circuits and Recommended Connection of Unused Pins

The input/output circuit type of each pin and recommended connection of unused pins are shown in Table 4-1. For the input/output circuit configuration of each type, see Figure 4-1.

Table 4-1. Types of Pin Input/Output Circuits

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00/KR0 to P03/KR3	8-A	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P10 to P13	5-A		
P20/$\overline{\text{SCK10}}$	8-A		
P21/SO10			
P22/SI10			
P23/$\overline{\text{SCK20}}$/ASCK20			
P24/SO20/TxD20			
P25/SI20/RxD20			
P26/TO20			
P30/INTP0/CPT20			Input: Independently connect to V_{SS} via a resistor. Output: Leave open.
P31/INTP1/TO30/TMI40			
P32/INTP2/TO40			
P33/INTP3			
P50 to P53 (Mask ROM version)	13-W		Input: Independently connect to V_{SS} via a resistor. Output: Leave this pin open at low-level output after clearing the output latch of the port to 0.
P50 to P53 (μPD78F9316)	13-V		
S0 to S23	17	Output	Leave open.
COM0 to COM3	18		
V_{LC0} to V_{LC2}	–	–	
CAPH, CAPL	–		
XT1	–	Input	Connect to V_{SS}.
XT2		–	Leave open.
$\overline{\text{RESET}}$	2	Input	–
IC (Mask ROM version)	–	–	Connect directly to V_{SS}.
V_{PP} (μPD78F9316)			Independently connect to a 10 kΩ pull-down resistor or connect directly to V_{SS}.

Figure 4-1. Pin Input/Output Circuits



5.1 Memory Space

The μPD789306 and μPD789316 Subseries can access 64 KB of memory space. Figures 5-1 through 5-3 show the memory maps.

Figure 5-1. Memory Map (μPD789304, 789314)

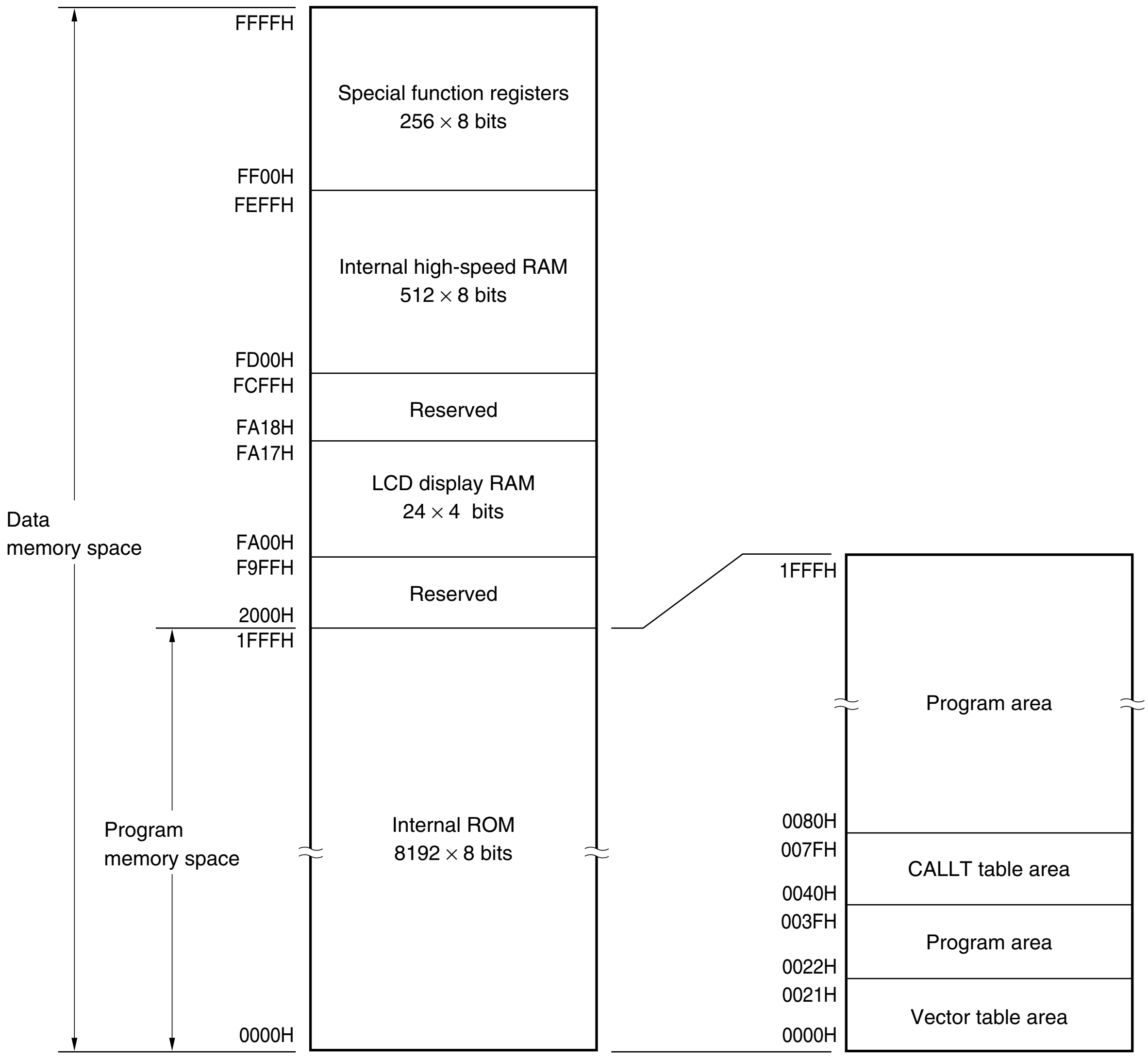


Figure 5-2. Memory Map (μPD789306, 789316)

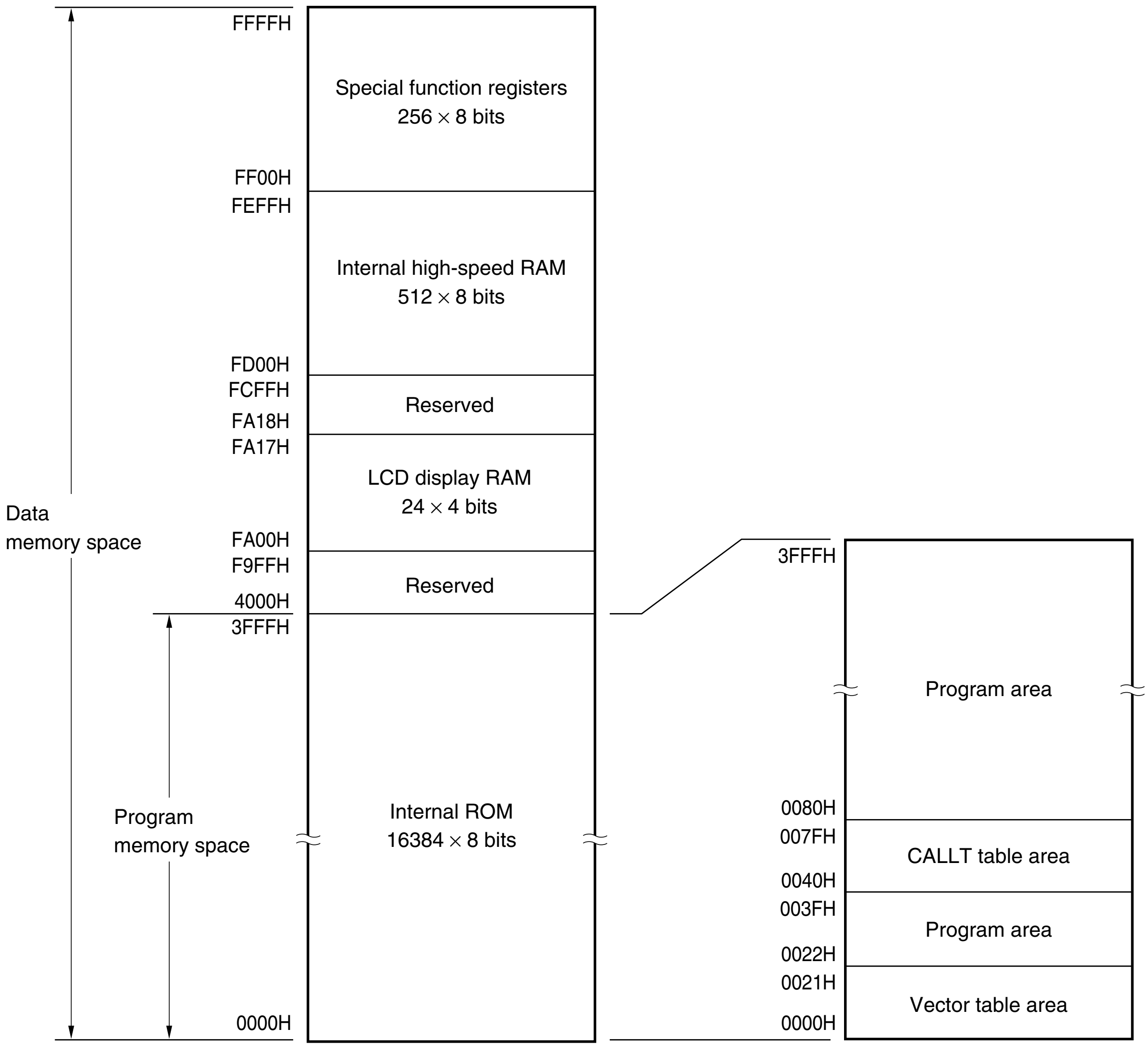
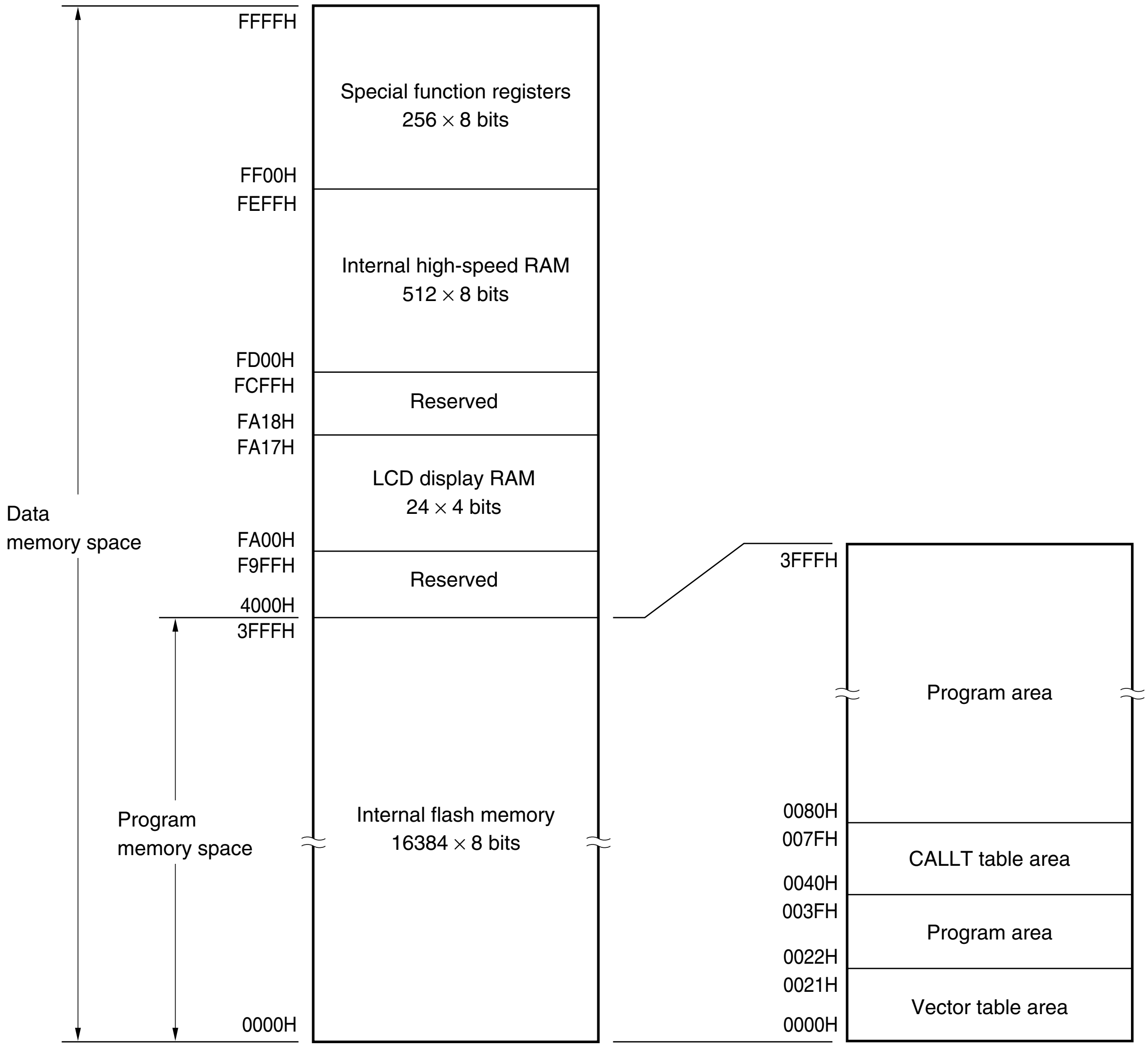


Figure 5-3. Memory Map (μPD78F9306, 78F9316)



5.1.1 Internal program memory space

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

The μPD789306 and μPD789316 Subseries provide internal ROM (or flash memory) with the following capacity for each product.

Table 5-1. Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
μPD789304, 789314	Mask ROM	8192 × 8 bits
μPD789306, 789316		16384 × 8 bits
μPD78F9306, 78F9316	Flash memory	16384 × 8 bits

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

(1) Vector table area

The 34-byte area of addresses 0000H to 0021H is reserved as a vector table area. This area stores program start addresses to be used when branching by the $\overline{\text{RESET}}$ input or an 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 5-2. Vector Table

Vector Table Address	Interrupt Request	Vector Table Address	Interrupt Request
0000H	$\overline{\text{RESET}}$ input	0012H	INTST20
0004H	INTWDT	0014H	INTWTI
0006H	INTP0	0016H	INTTM20
0008H	INTP1	0018H	INTTM30
000AH	INTP2	001AH	INTTM40
000CH	INTP3	001EH	INTWT
000EH	INTSR20/INTCSI20	0020H	INTKR00
0010H	INTCSI10		

(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.

5.1.2 Internal data memory (internal high-speed RAM) space

The μPD789306 and μPD789316 Subseries products incorporate the following RAM.

(1) Internal high-speed RAM

Internal high-speed RAM is incorporated in the area between FD00H and FEFFH.
The internal high-speed RAM is also used as a stack.

(2) LCD display RAM

LCD display RAM is allocated in the area between FA00H and FA17H. The LCD display RAM can also be used as ordinary RAM.

5.1.3 Special function register (SFR) area

Special function registers (SFRs) of on-chip peripheral hardware are allocated in the area between FF00H to FFFFH (see **Table 5-3**).

5.1.4 Data memory addressing

The μPD789306 and μPD789316 Subseries are provided with a variety of addressing modes to make memory manipulation as efficient as possible. At the addresses corresponding to data memory area (FD00H to FFFFH) especially, specific addressing modes that correspond to the particular function an area, such as the special function registers are available. Figures 5-4 through 5-6 show the data memory addressing modes.

Figure 5-4. Data Memory Addressing (μPD789304, 789314)

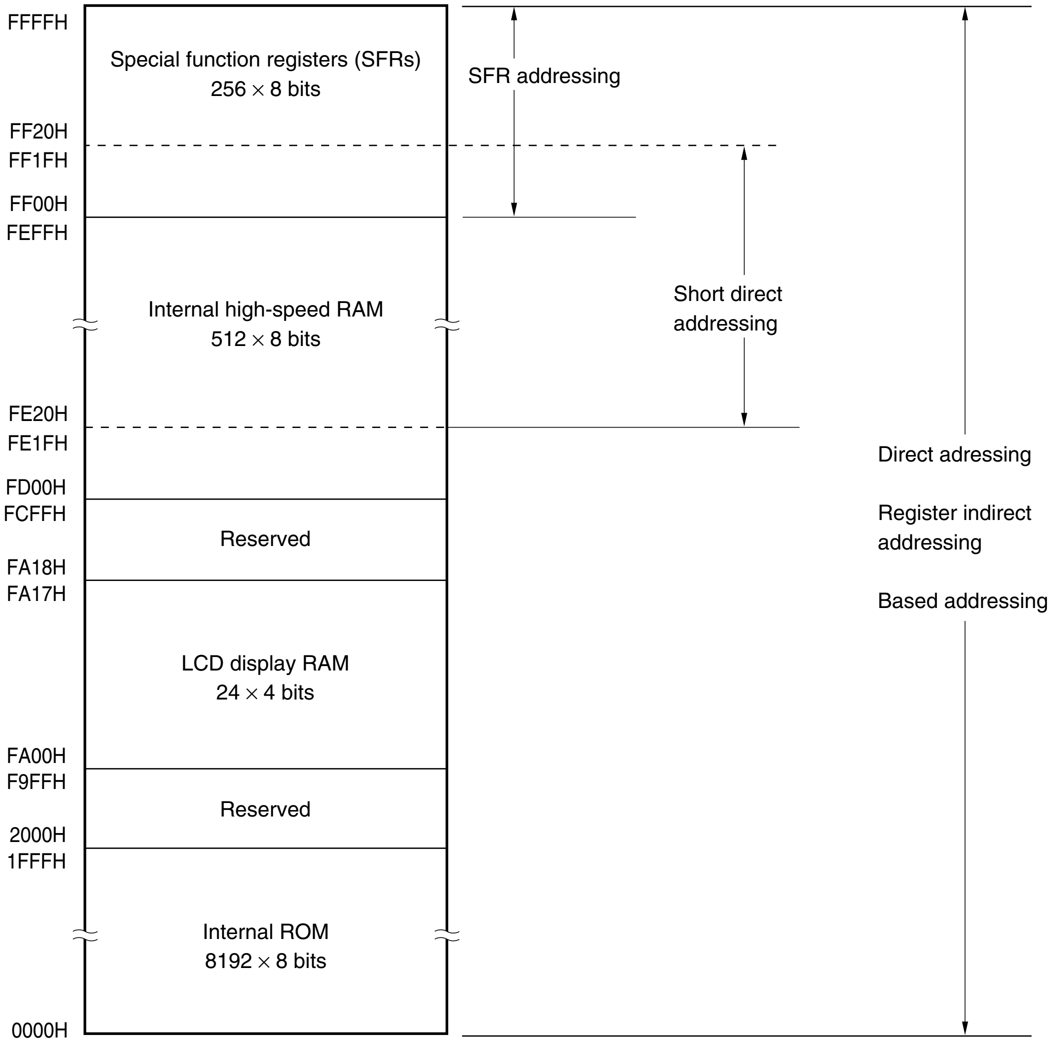


Figure 5-5. Data Memory Addressing (μPD789306, 789316)

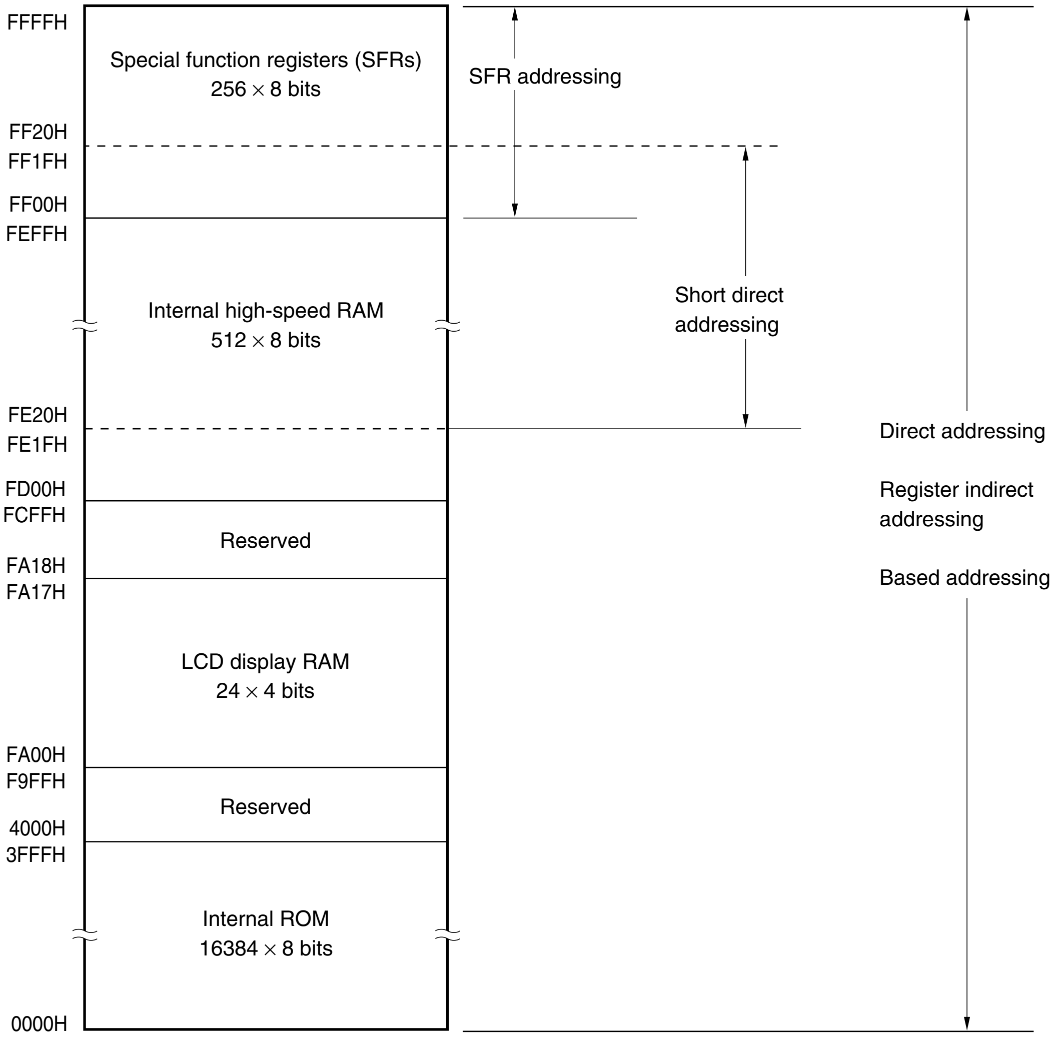
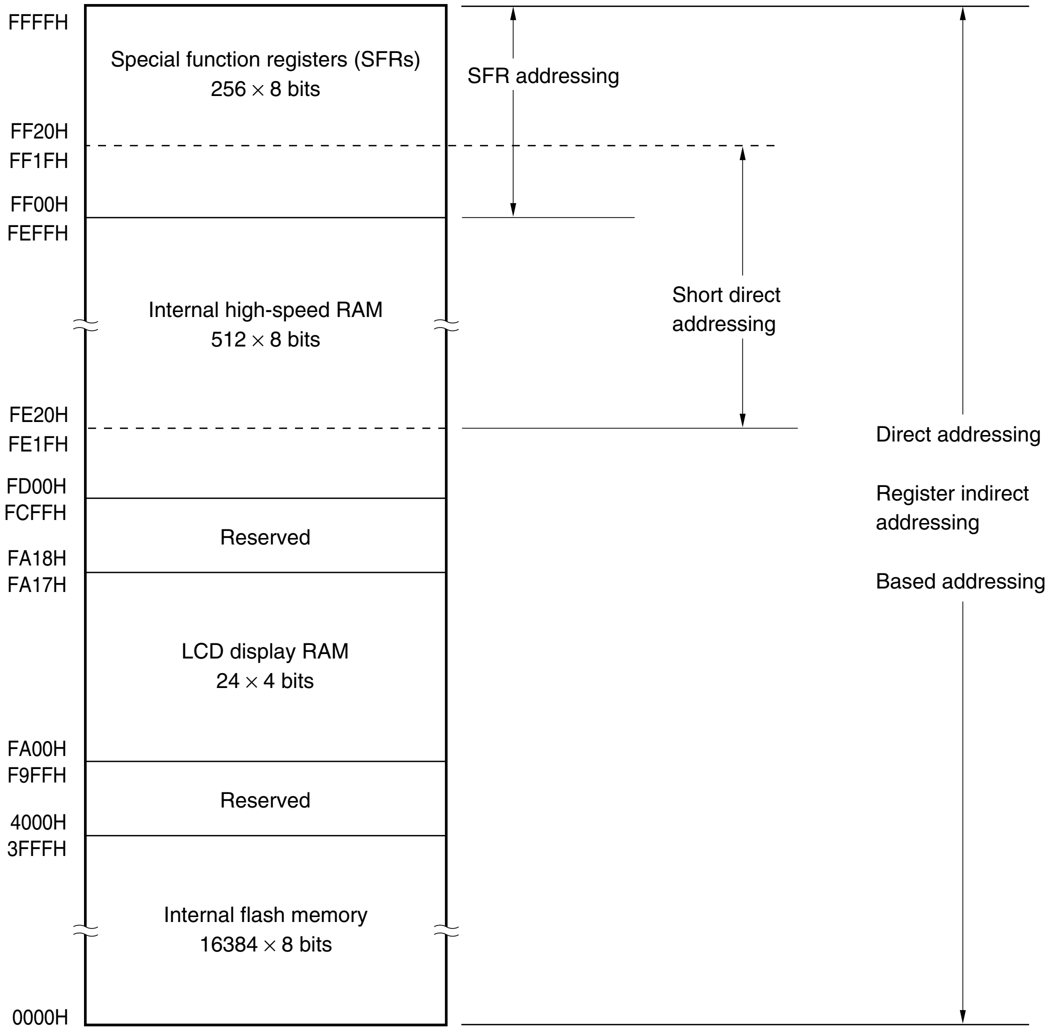


Figure 5-6. Data Memory Addressing (μPD78F9306, 78F9316)



5.2 Processor Registers

The μPD789306 and μPD789316 Subseries provide the following on-chip processor registers.

5.2.1 Control registers

The control registers contain special functions to control the program sequence statuses 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 program counter.

Figure 5-7. 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 5-8. 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 disable status (DI), and interrupt requests other than non-maskable interrupt are all disabled.

When 1, IE is set to the interrupt enable status (EI). Interrupt request acknowledgement enable is controlled with an interrupt mask flag for various interrupt sources.

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

(3) Stack pointer (SP)

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

Figure 5-9. 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 5-10 and 5-11.

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

Figure 5-10. Data to Be Saved to Stack Memory

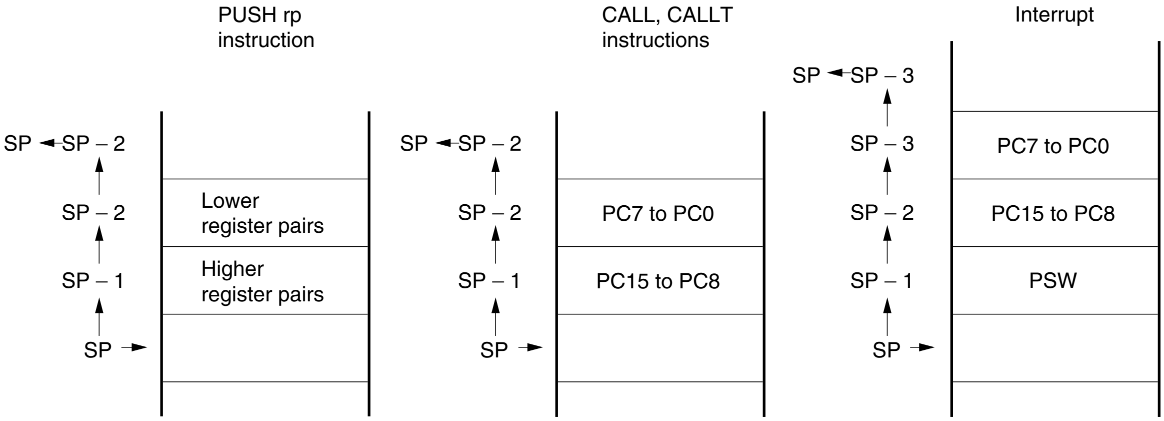
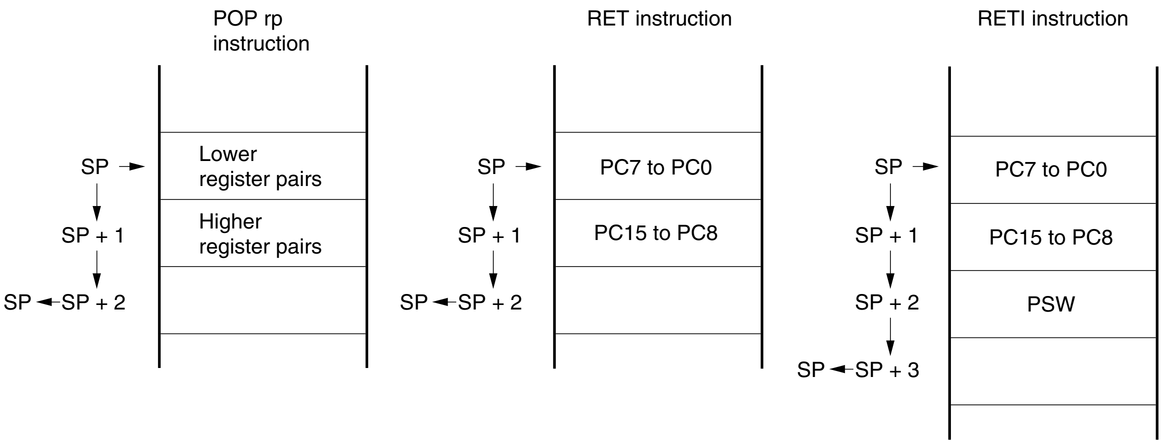


Figure 5-11. Data to Be Restored from Stack Memory



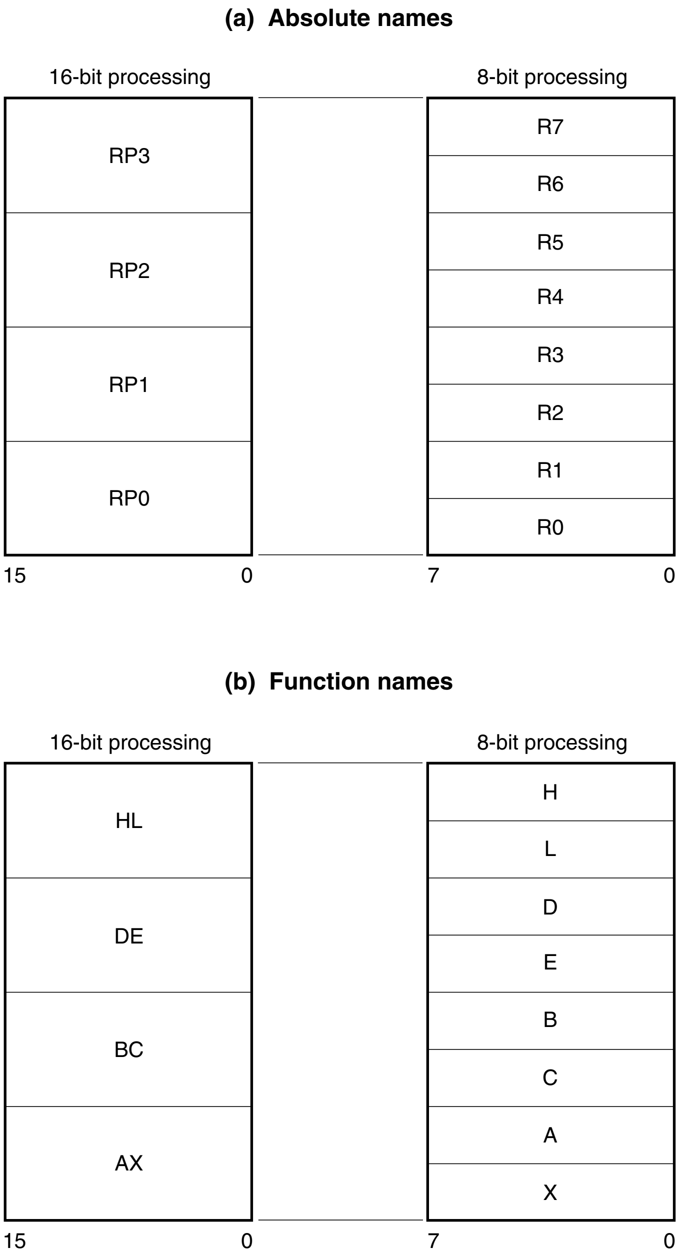
5.2.2 General-purpose registers

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 5-12. General-Purpose Register Configuration


(a) Absolute names

(b) Function names

5.2.3 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
  Describes a symbol reserved by the assembler for the 1-bit manipulation instruction operand (sfr.bit). This manipulation can also be specified with an address.

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

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

Table 5-3 lists the special function registers. The meanings of the symbols in this table are as follows:

- Symbol
  Indicates the addresses of the implemented special function registers. The symbols are reserved for the assembler and are defined as an sfr variable by the #pragma sfr directive for 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 manipulation unit
  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 5-3. Special Function Register List (1/2)

Address	Special Function Register (SFR) Name	Symbol		R/W	Bit Manipulation Unit			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			√	√	–	
FF0CH	8-bit compare register 40	CR40	CR4[Note1]	W	–	√	√[Notes 2, 3]	Undefined
FF0DH	8-bit compare register 30	CR30			–	√		
FF0EH	8-bit timer counter 40	TM40	TM4[Note1]	R	–	√	√[Notes 2, 3]	00H
FF0FH	8-bit timer counter 30	TM30			–	√		
FF10H	Transmit shift register 20	TXS20	SIO20	W	–	√	–	FFH
	Receive buffer register 20	RXB20		R	–	√	–	Undefined
FF11H	Serial shift register 10	SIO10		R/W	–	√	–	
FF16H	16-bit compare register 20	CR20[Note 1]		W	–	–	√[Notes 2, 3]	FFFFH
FF17H								
FF18H	16-bit timer counter 20	TM20[Note 1]		R	–	–	√[Notes 2, 3]	0000H
FF19H								
FF1AH	16-bit capture register 20	TCP20[Note 1]			–	–	√[Notes 2, 3]	Undefined
FF1BH								
FF20H	Port mode register 0	PM0		R/W	√	√	–	FFH
FF21H	Port mode register 1	PM1			√	√	–	
FF22H	Port mode register 2	PM2			√	√	–	
FF23H	Port mode register 3	PM3			√	√	–	
FF25H	Port mode register 5	PM5			√	√	–	
FF32H	Pull-up resistor option register B2	PUB2			√	√	–	00H
FF33H	Pull-up resistor option register B3	PUB3			√	√	–	
FF42H	Watchdog timer clock select register	WDCS			–	√	–	
FF48H	16-bit timer mode control register 20	TMC20			√	√	–	
FF4AH	Watch timer mode control register	WTM			√	√	–	
FF4CH	8-bit compare register H40	CRH40		W	–	√	–	Undefined

Notes 1. Name of SFR dedicated for 16-bit access.

2. Only in short direct addressing, 16-bit access is possible.

3. These are 16-bit access dedicated registers, however, 8-bit access is possible. When performing 8-bit access, access using direct addressing.

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

Address	Special Function Register (SFR) Name	Symbol	R/W	Bit Manipulation Unit			After Reset
				1 Bit	8 Bits	16 Bits	
FF4DH	8-bit timer mode control register 30	TMC30	R/W	√	√	–	00H
FF4EH	8-bit timer mode control register 40	TMC40		√	√	–	
FF4FH	Carrier generator output control register 40	TCA40	W	–	√	–	
FF70H	Asynchronous serial interface mode register 20	ASIM20	R/W	√	√	–	
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		–	√	–	
FF78H	Serial operation mode register 10	CSIM10		√	√	–	
FFB0H	LCD display mode register 0	LCDM0		√	√	–	
FFB2H	LCD clock control register 0	LCDC0		√	√	–	
FFB3H	LCD voltage amplification control register 0	LCDVA0		√	√	–	
FFE0H	Interrupt request flag register 0	IF0		√	√	–	
FFE1H	Interrupt request flag register 1	IF1		√	√	–	
FFE4H	Interrupt mask flag register 0	MK0		√	√	–	FFH
FFE5H	Interrupt mask flag register 1	MK1		√	√	–	
FFECH	External interrupt mode register 0	INTM0		–	√	–	00H
FFEDH	External interrupt mode register 1	INTM1		–	√	–	
FFF0H	Suboscillation mode register	SCKM		√	√	–	
FFF2H	Subclock control register	CSS		√	√	–	
FFF5H	Key return mode register 00	KRM00		√	√	–	
FFF7H	Pull-up resistor option register 0	PU0		√	√	–	
FFF9H	Watchdog timer mode register	WDTM		√	√	–	
FFFAH	Oscillation stabilization time select register[Note]	OSTS		–	√	–	04H
FFFBH	Processor clock control register	PCC		√	√	–	02H

Note μPD789306 Subseries only.

5.3 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)**).

5.3.1 Relative addressing

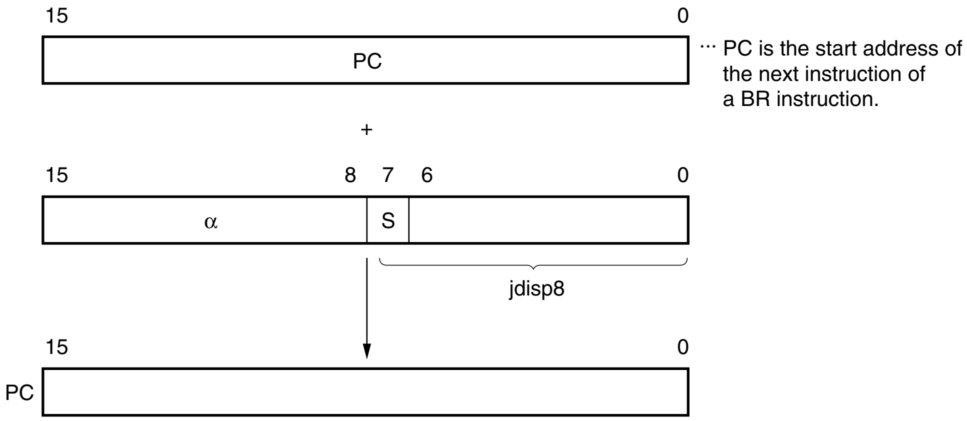
[Function]

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]


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

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

5.3.2 Immediate addressing

[Function]

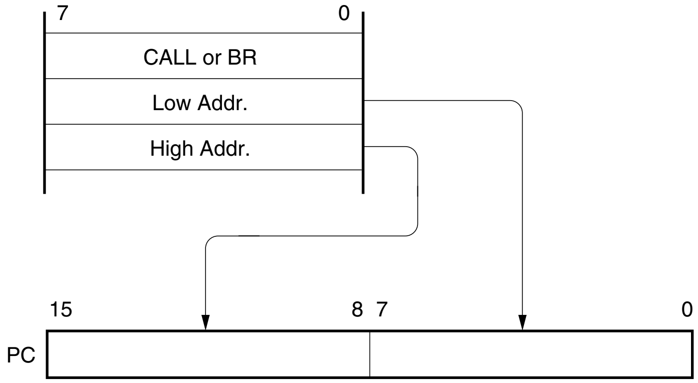
Immediate data in the instruction word is transferred to the program counter (PC) and branched.

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



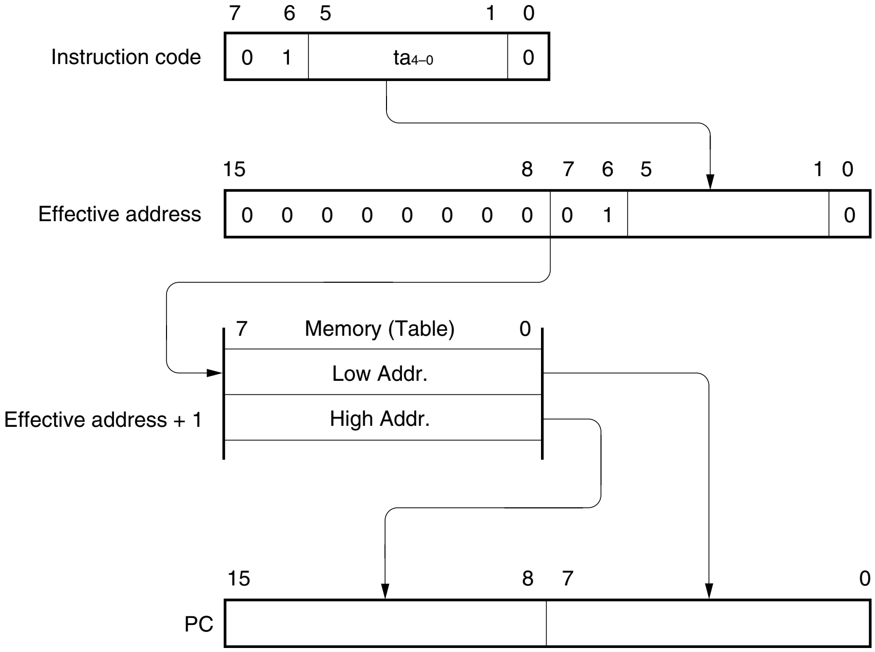
5.3.3 Table indirect addressing

[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]



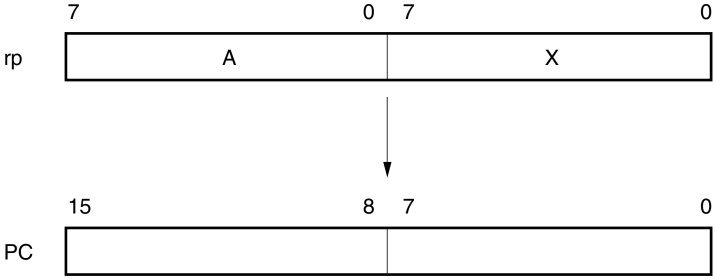
5.3.4 Register addressing

[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]



5.4 Operand Address Addressing

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

5.4.1 Direct addressing

[Function]

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

[Operand format]

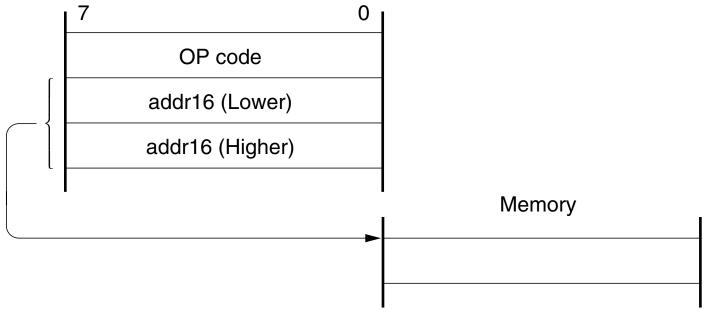
Identifier	Description
addr16	Label or 16-bit immediate data

[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]



5.4.2 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]

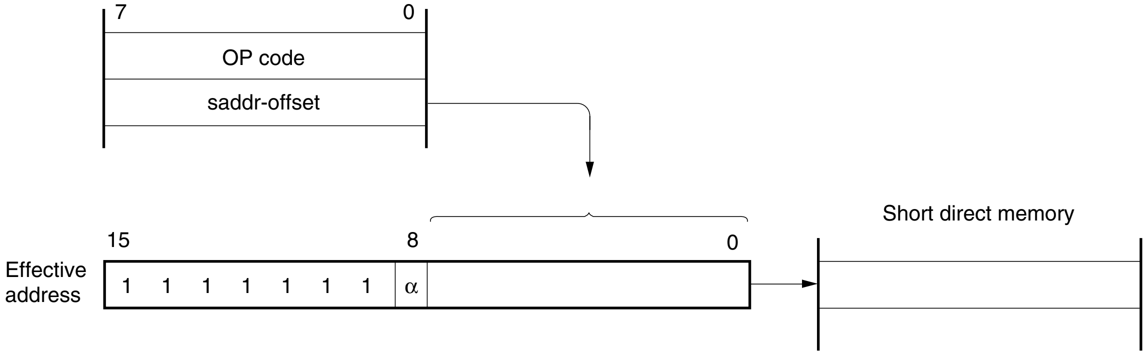
Identifier	Description
saddr	Label or FE20H to FF1FH immediate data
saddrp	Label or FE20H to FF1FH immediate data (even address only)

[Description example]

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

Instruction code	1	1	1	1	0	1	0	1	OP code
	1	0	0	1	0	0	0	0	90H (saddr-offset)
	0	1	0	1	0	0	0	0	50H (Immediate data)

[Illustration]



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

5.4.3 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]

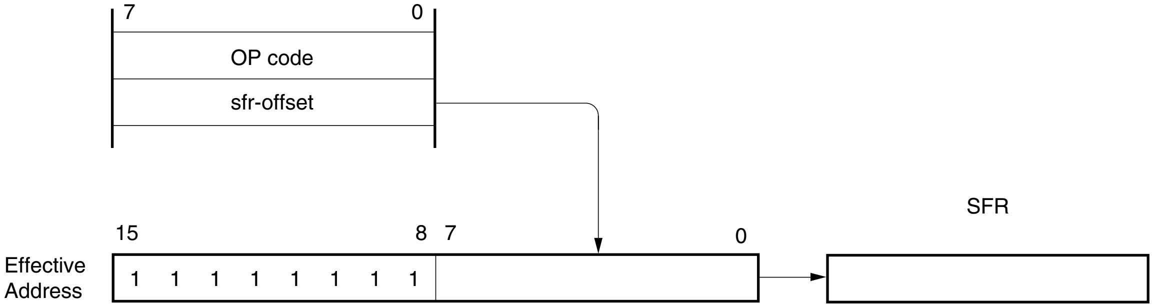
Identifier	Description
sfr	Special function register name

[Description example]

MOV PM0, A; When selecting PM0 for sfr

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

[Illustration]



5.4.4 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.

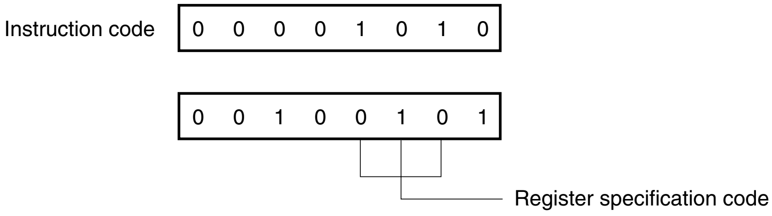
[Operand format]

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

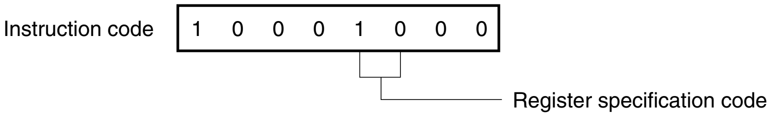
r and rp can be described with absolute names (R0 to R7 and RP0 to RP3) as well as function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL).

[Description example]

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



INCW DE; When selecting the DE register pair for rp



5.4.5 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]

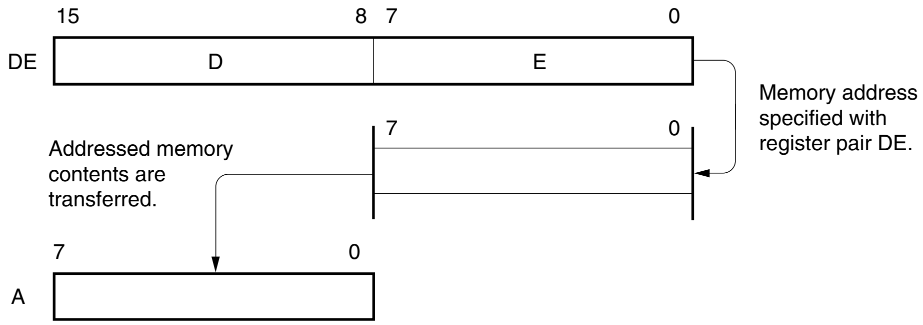
Identifier	Description
–	[DE], [HL]

[Description example]

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

Instruction code	0	0	1	0	1	0	1	1

[Illustration]


15
8
7
0
DE
D
E
Memory address specified with register pair DE.
7
0
Addressed memory contents are transferred.
7
0
A

5.4.6 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

5.4.7 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 6 PORT FUNCTIONS

6.1 Port Functions

The μPD789306 and μPD789316 Subseries provide the ports shown in Figure 6-1, enabling various methods of control.

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 3 PIN FUNCTIONS (μPD789306 SUBSERIES)** and **CHAPTER 4 PIN FUNCTIONS (μPD789316 SUBSERIES)**.

Figure 6-1. Port Types

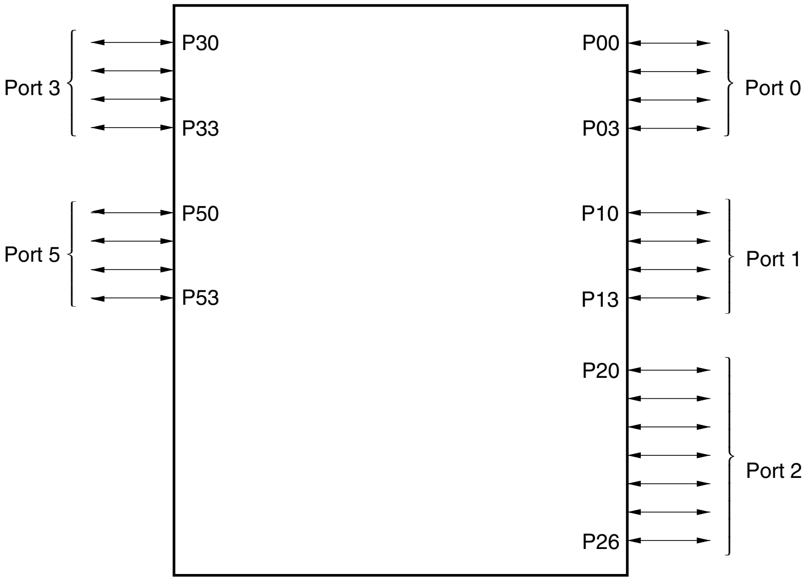


Table 6-1. Port Functions

Pin Name	I/O	Function	After Reset	Alternate Function
P00 to P03	I/O	Port 0. 4-bit I/O port. Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be specified by means of pull-up resistor option register 0 (PU0) or key return mode register 00 (KRM00) in port units.	Input	KR0 to KR3
P10 to P13	I/O	Port 1. 4-bit I/O port. Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be specified by pull-up resistor option register 0 (PU0) in port units.	Input	–
P20	I/O	Port 2. 7-bit I/O port. Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be specified by means of pull-up resistor option register B2 (PUB2) in 1-bit units.	Input	$\overline{\text{SCK10}}$
P21				SO10
P22				SI10
P23				$\overline{\text{SCK20}}$/ASCK20
P24				SO20/TxD20
P25				SI20/RxD20
P26				TO20
P30	I/O	Port 3. 4-bit I/O port. Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be specified by means of pull-up resistor option register B3 (PUB3) in 1-bit units.	Input	INTP0/CPT20
P31				INTP1/TO30/TMI40
P32				INTP2/TO40
P33				INTP3
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 a mask ROM version, an on-chip pull-up resistor can be specified by a mask option in 1 bit units.	Input	–

6.2 Port Configuration

Ports have the following hardware configuration.

Table 6-2. Configuration of Port

Item	Configuration
Control registers	Port mode register (PMm: m = 0 to 3, 5) Pull-up resistor option register 0 (PU0) Pull-up resistor option register B2, B3 (PUB2, PUB3)
Ports	Total: 23 (CMOS I/O: 19, N-ch open-drain I/O: 4)
Pull-up resistors	• Mask ROM version Total: 23 (software control: 19, mask option control: 4) • Flash memory version Total: 19 (software control only)

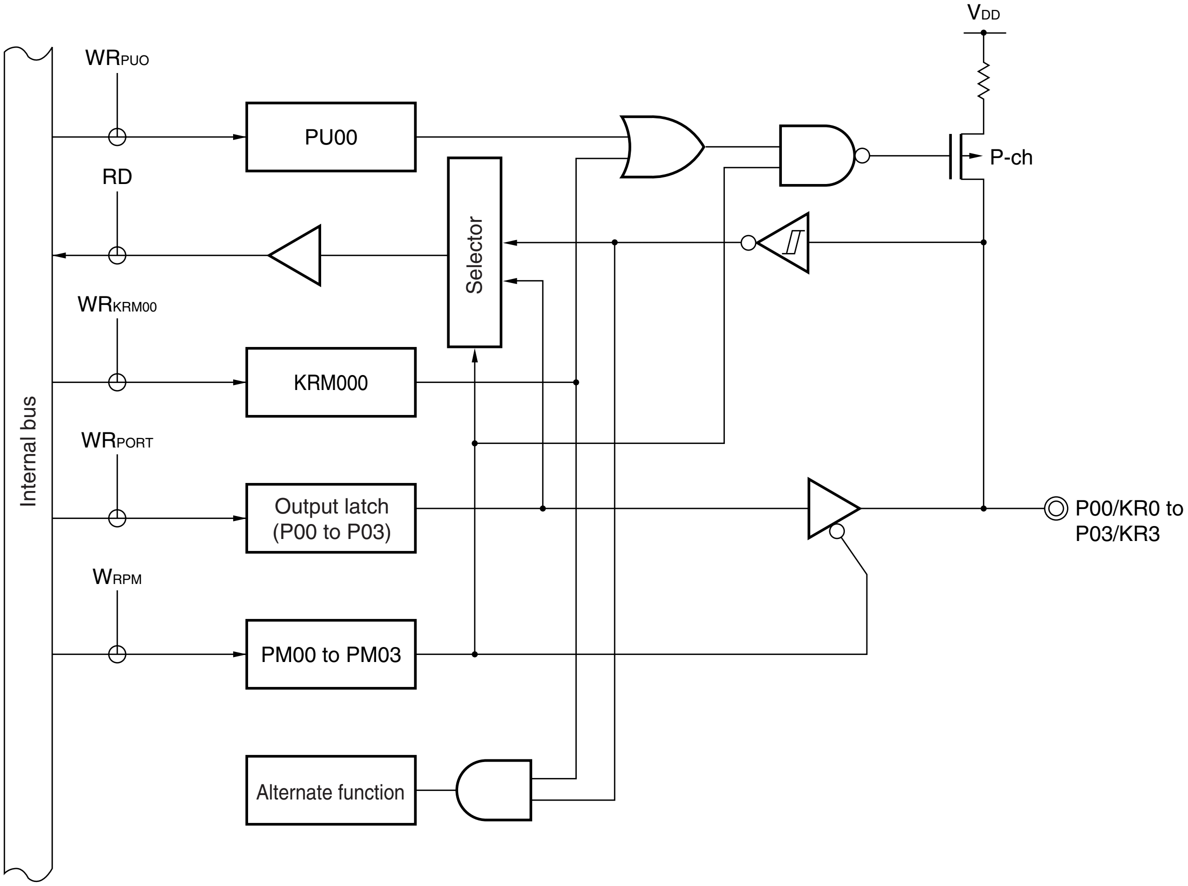
6.2.1 Port 0

This is a 4-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 the port mode register 0 (PM0). When the P00 to P03 pins are used as input port pins, on-chip pull-up resistors can be connected in 4-bit units by using pull-up resistor option register 0 (PU0).

Port 0 is set in the input mode when the $\overline{\text{RESET}}$ signal is input.

Figure 6-2 shows a block diagram of port 0.

Figure 6-2. Block Diagram of P00 to P03



KRM00:	Key return mode register 00
PU0:	Pull-up resistor option register 0
PM:	Port mode register
RD:	Port 0 read signal
WR:	Port 0 write signal

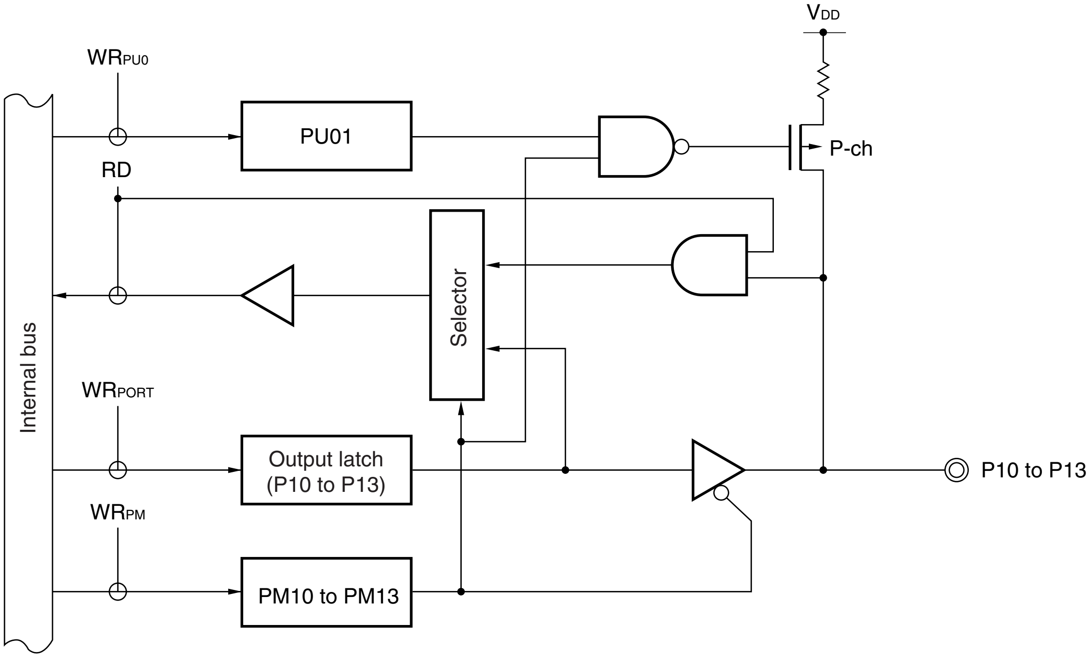
6.2.2 Port 1

This is a 4-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 to P13 pins as input port pins, on-chip pull-up resistors can be connected in 4-bit units by using pull-up resistor option register 0 (PU0).

This port is set in the input mode when the $\overline{\text{RESET}}$ signal is input.

Figure 6-3 shows a block diagram of port 1.

Figure 6-3. Block Diagram of P10 to P13



PU0: Pull-up resistor option register 0

PM: Port mode register

RD: Port 1 read signal

WR: Port 1 write signal

6.2.3 Port 2

This is a 7-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 P26 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).

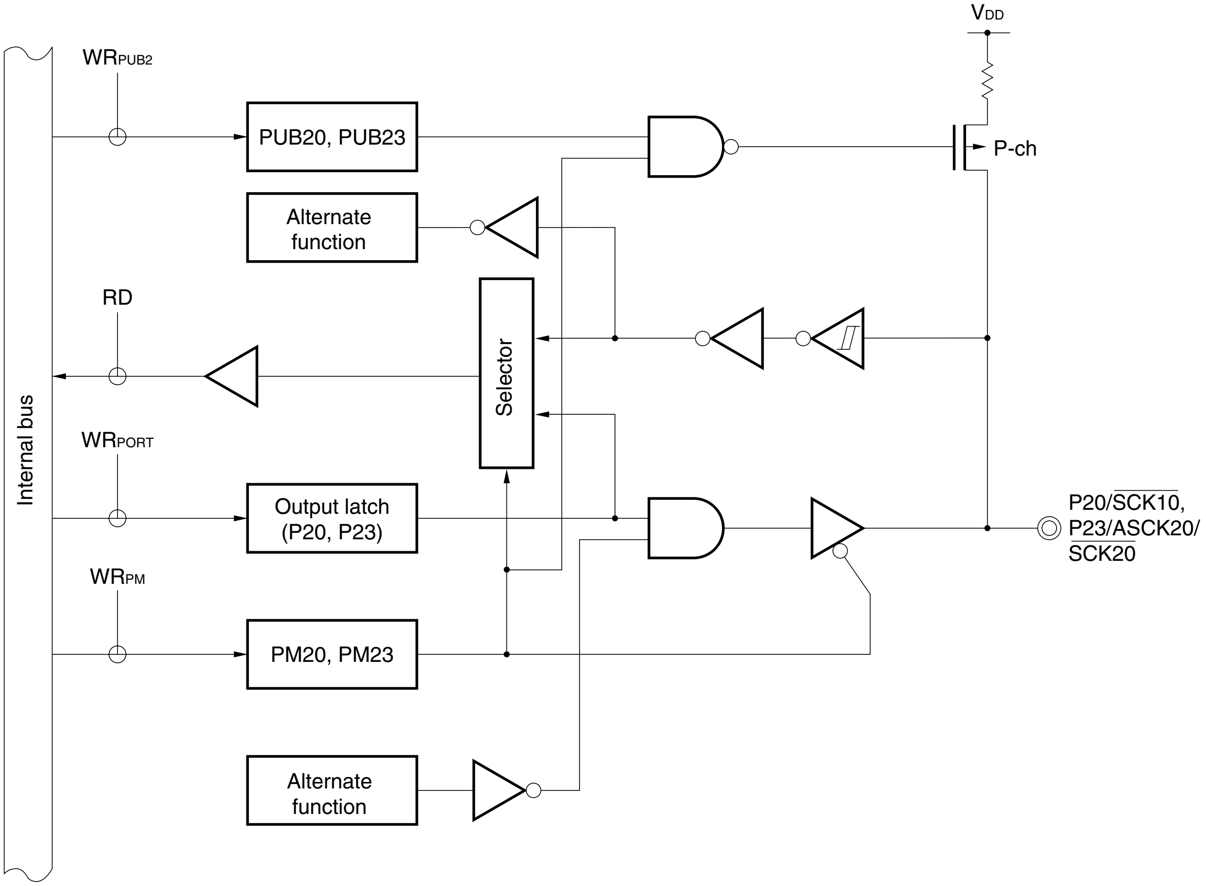
The port is also used as a data I/O and clock I/O to and from the serial interface and for timer output.

This port is set in the input mode when the $\overline{\text{RESET}}$ signal is input.

Figures 6-4 to 6-7 show block diagrams of port 2.

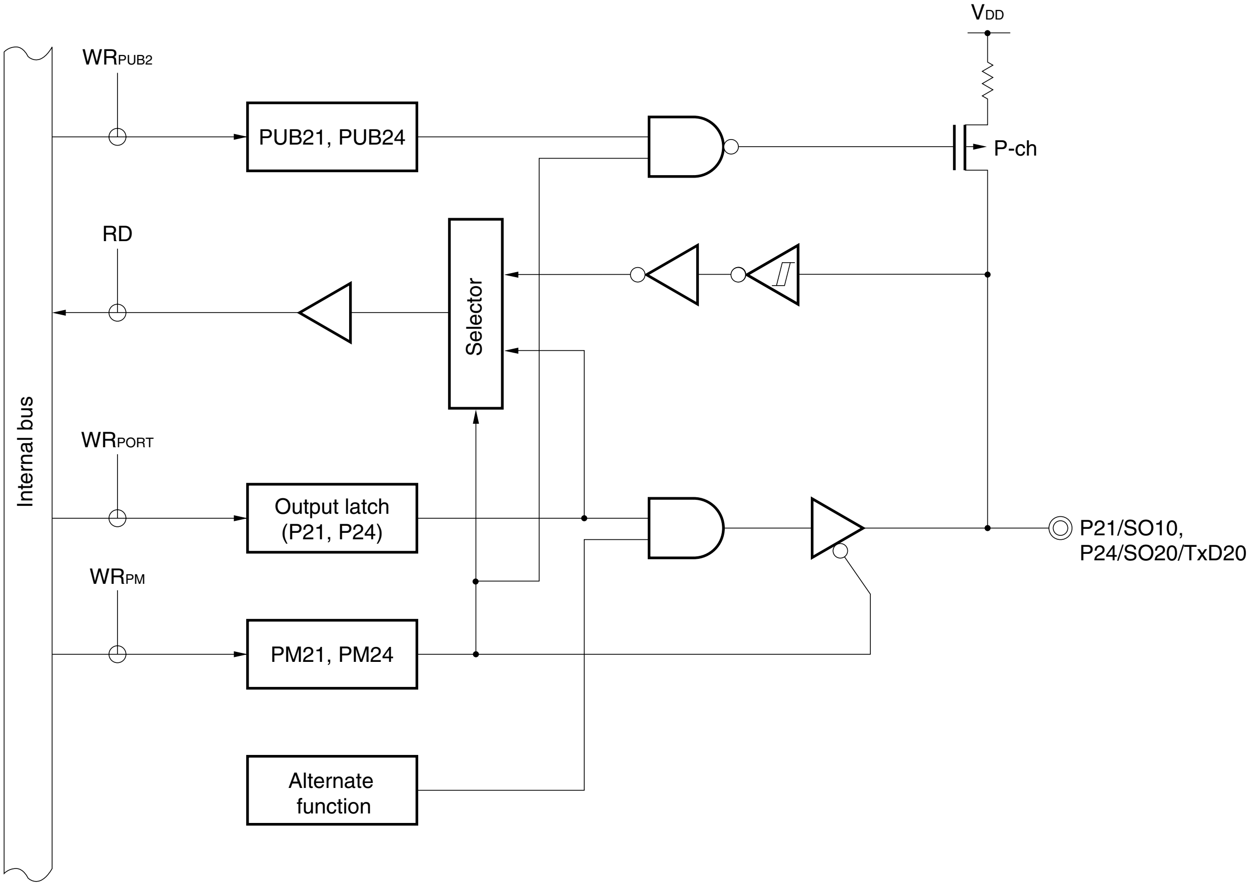
Caution When using the pins of port 2 as the serial interface pin, the I/O or output latch must be set according to the function to be used. For how to set the latches, see Table 13-2 Settings of Serial Interface 10 Operating Mode and Figure 14-2 Settings of Serial Interface 20 Operating Mode.

Figure 6-4. Block Diagram of P20 and P23



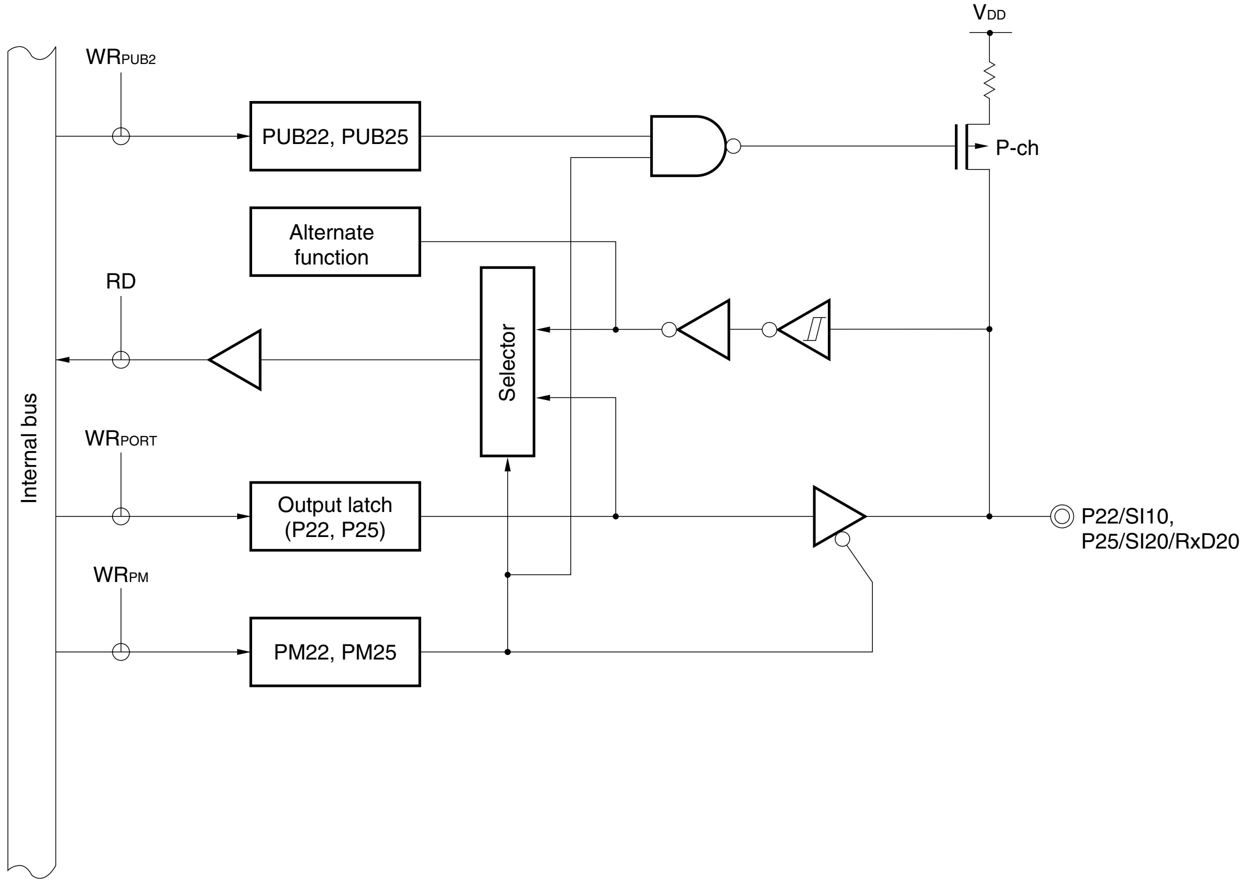
PUB2:	Pull-up resistor option register B2
PM:	Port mode register
RD:	Port 2 read signal
WR:	Port 2 write signal

Figure 6-5. Block Diagram of P21 and P24



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

Figure 6-6. Block Diagram of P22 and P25



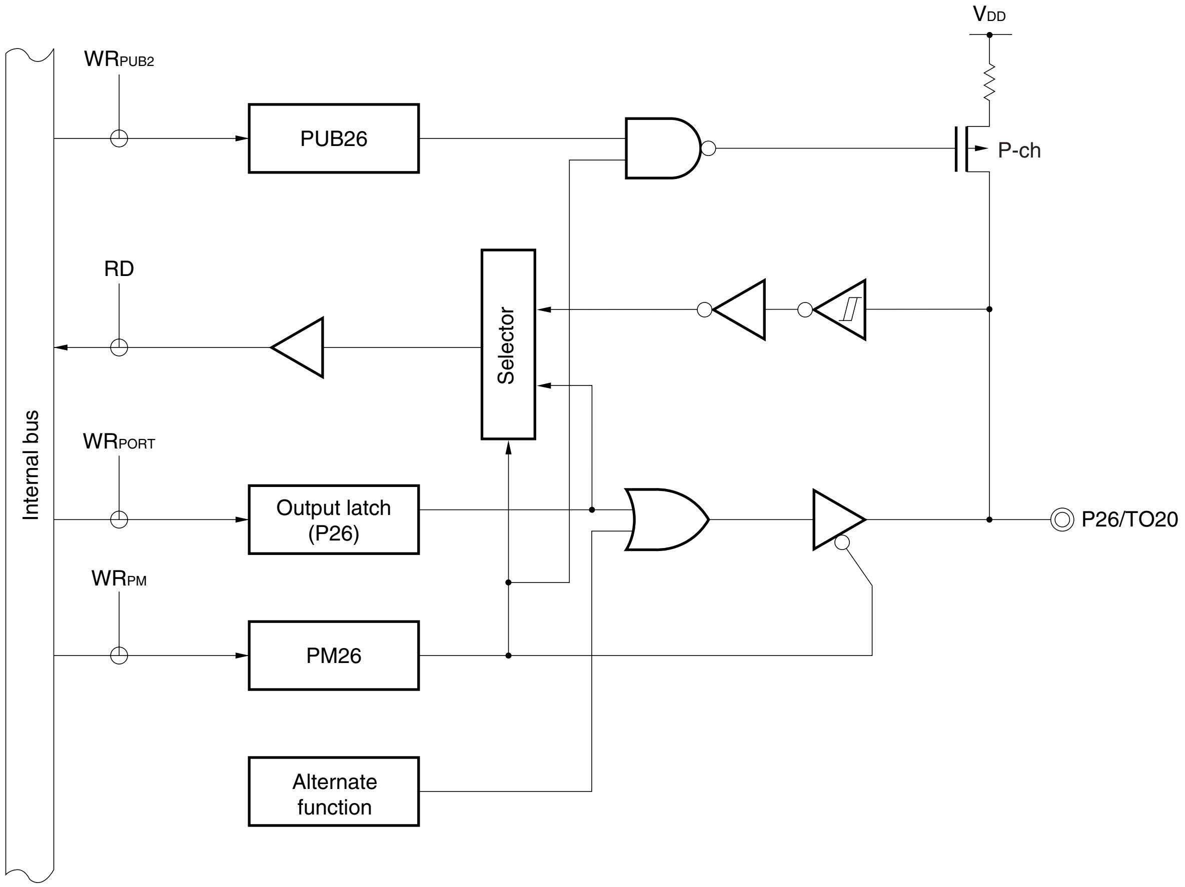
PUB2: Pull-up resistor option register B2

PM: Port mode register

RD: Port 2 read signal

WR: Port 2 write signal

Figure 6-7. Block Diagram of P26



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

6.2.4 Port 3

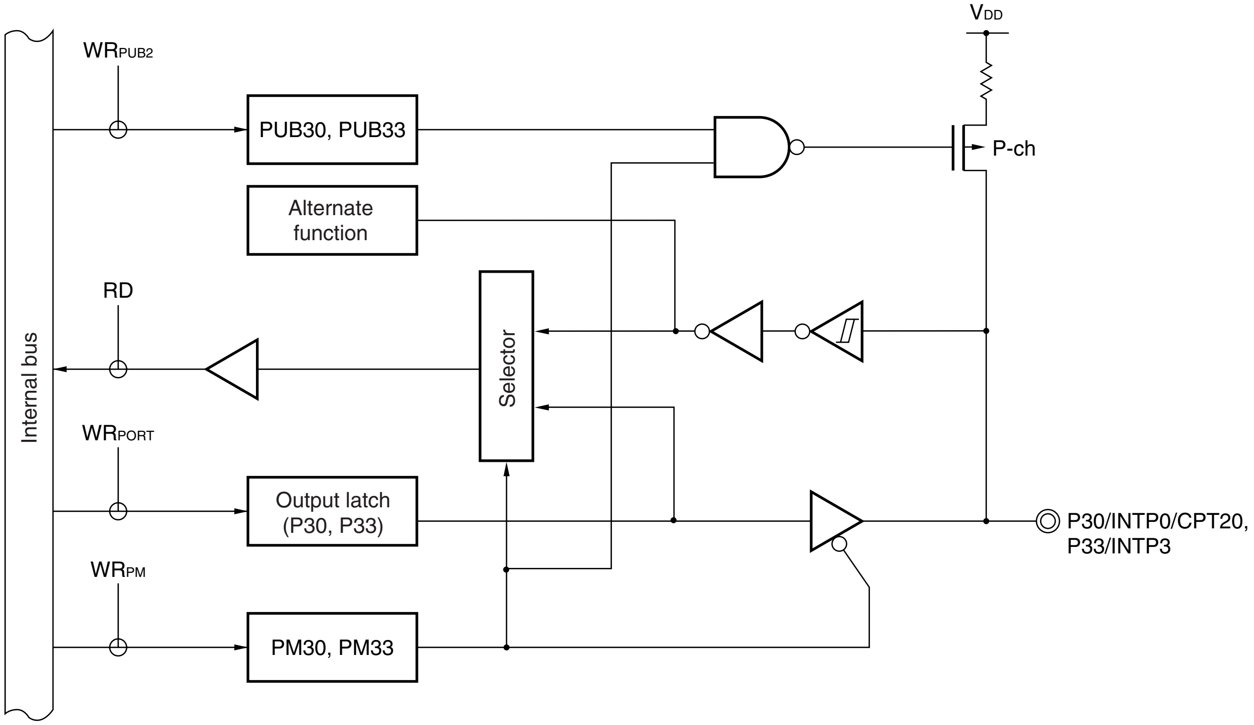
This is a 4-bit I/O port with an output latch. Port 3 can be specified in the input or output mode in 1-bit units by using port mode register 3 (PM3). When using the P30 to P33 pins as input port pins, on-chip pull-up resistors can be connected in 1-bit units by using pull-up resistor option register B3 (PUB3).

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

This port is set in the input mode when the $\overline{\text{RESET}}$ signal is input.

Figures 6-8 and 6-9 show block diagrams of port 3.

Figure 6-8. Block Diagram of P30 and P33



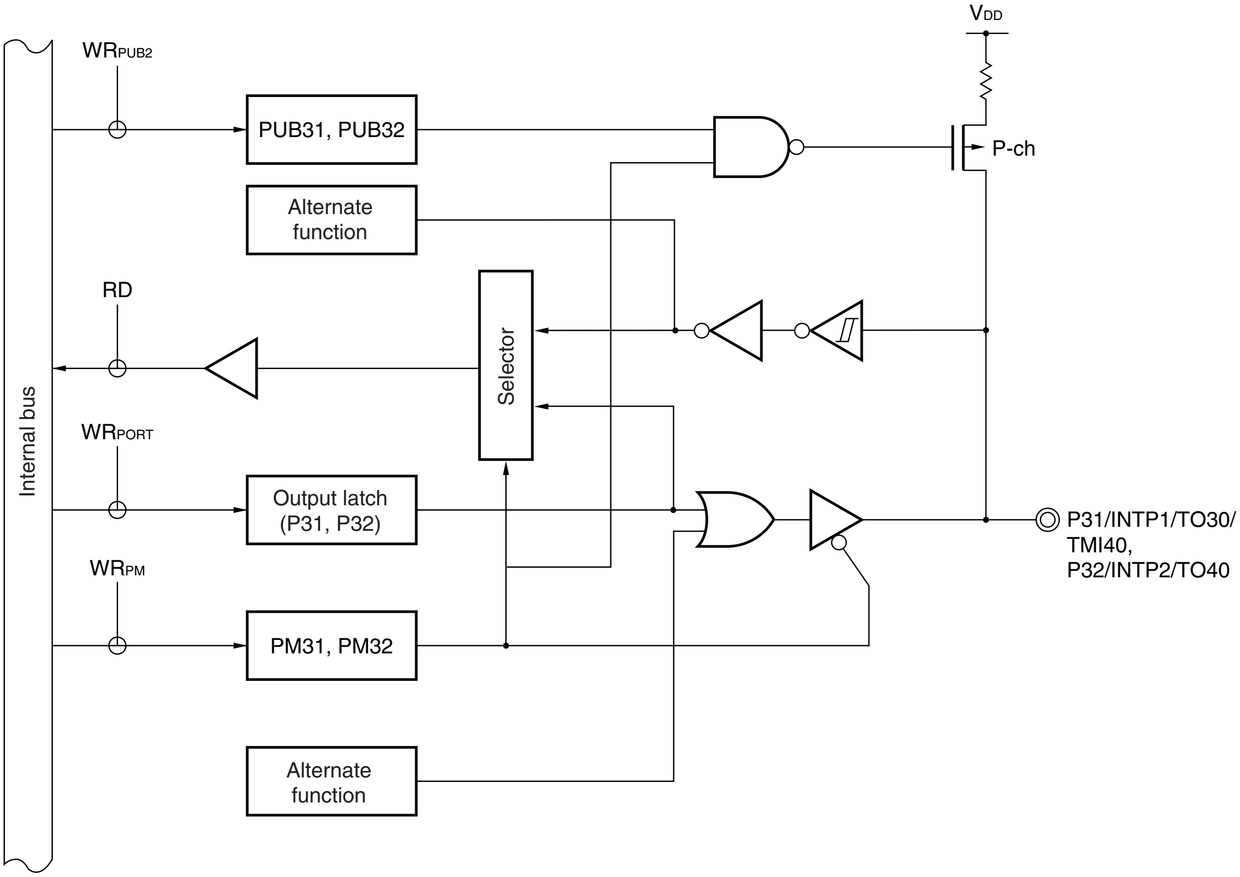
PUB3: Pull-up resistor option register B3

PM: Port mode register

RD: Port 3 read signal

WR: Port 3 write signal

Figure 6-9. Block Diagram of P31 and P32



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

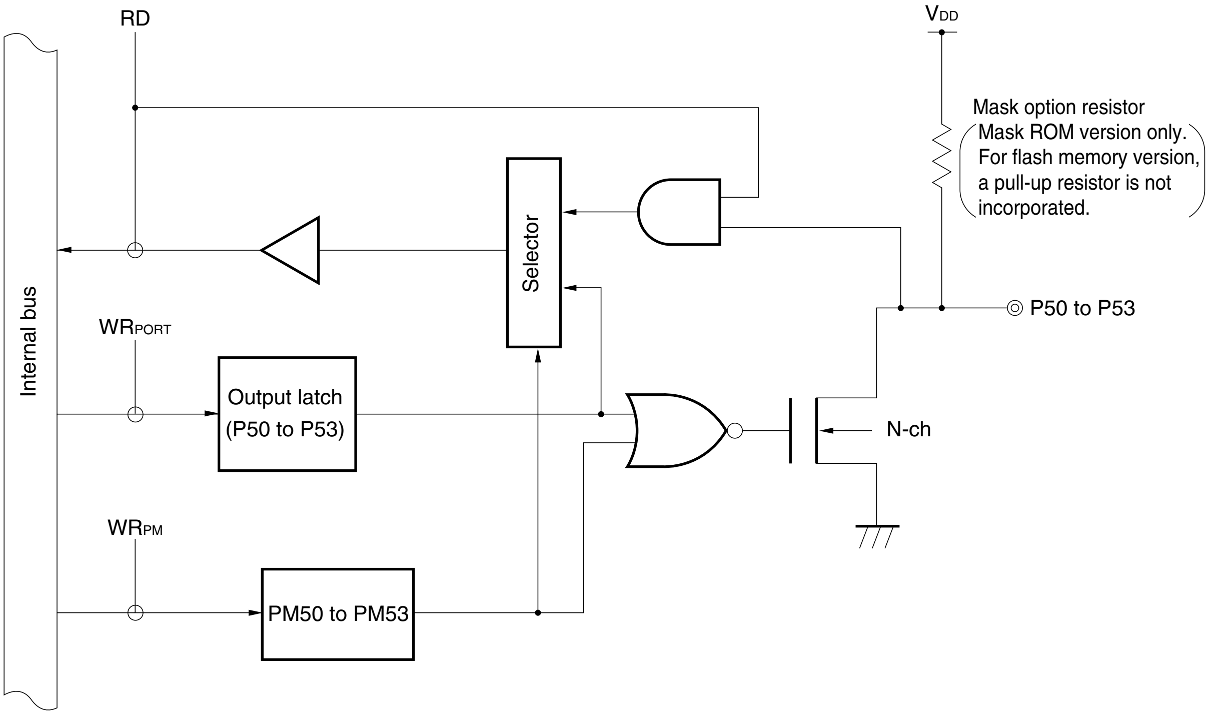
6.2.5 Port 5

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.

This port is set in the input mode when the $\overline{\text{RESET}}$ signal is input.

Figure 6-10 shows a block diagram of port 5.

Figure 6-10. Block Diagram of P50 to P53



PM:	Port mode register
RD:	Port 5 read signal
WR:	Port 5 write signal

6.3 Registers Controlling Port Function

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

- Port mode registers (PM0 to PM3, PM5)
- Pull-up resistor option registers (PU0, PUB2, PUB3)

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

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

The port mode registers are independently set with a 1-bit or 8-bit memory manipulation instruction.

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

When port pins are used as alternate-function pins, set the port mode register and output latch according to Table 6-3.

Caution As port 3 has an alternate function as external interrupt input, when the port function output mode is specified and the output level is changed, the interrupt request flag is set. When the output mode is used, therefore, the interrupt mask flag should be preset to 1.

Figure 6-11. Format of Port Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	1	1	1	PM03	PM02	PM01	PM00	FF20H	FFH	R/W
PM1	1	1	1	1	PM13	PM12	PM11	PM10	FF21H	FFH	R/W
PM2	1	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FF22H	FFH	R/W
PM3	1	1	1	1	PM33	PM32	PM31	PM30	FF23H	FFH	R/W
PM5	1	1	1	1	PM53	PM52	PM51	PM50	FF25H	FFH	R/W

PMmn	Pmn pin input/output mode selection (m = 0 to 3, 5, n = 0 to 6)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

Table 6-3. Port Mode Register and Output Latch Settings When Using Alternate Functions

Pin Name	Alternate Function		PMxx	Pxx
	Name	I/O		
P00 to P03	KR0 to KR3	Input	1	x
P26	TO20	Output	0	0
P30	INTP0	Input	1	x
	CPT20	Input	1	x
P31	INTP1	Input	1	x
	TO30	Output	0	0
	TMI40	Input	1	x
P32	INTP2	Input	1	x
	TO40	Output	0	0
P33	INTP3	Input	1	x

Caution When port 2 is used as a serial interface pin, the I/O latch or output latch must be set according to its function. For the setting method, see Table 13-2 Settings of Serial Interface 10 Operating Mode and Table 14-2 Settings of Serial Interface 20 Operating Mode.

Remark x: don’t care

PMxx: Port mode register

Pxx: Port output latch

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

Pull-up resistor option register 0 (PU0) sets whether an on-chip pull-up resistor on each port is used or not.

On the port specified to use an on-chip pull-up resistor by PU0, the pull-up resistor can be internally used only for the bits set in the input mode. No on-chip pull-up resistors can be used for the bits set in the output mode regardless of the setting of PU0. This also applies to cases when the pins are used for alternate functions.

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

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

Figure 6-12. Format of Pull-Up Resistor Option Register 0

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

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

Caution Bits 2 to 7 must be set to 0.

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

Pull-up resistor option register B2 (PUB2) sets whether on-chip pull-up resistors on P20 to P26 are used or not.

On the port specified to use an on-chip pull-up resistor by PUB2, the pull-up resistor can be internally used only for the bits set in the input mode. No on-chip pull-up resistors can be used for the bits set in the output mode regardless of the setting of PUB2. This also applies to cases when the pins are used for alternate functions.

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

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

Figure 6-13. Format of Pull-Up Resistor Option Register B2

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

PUB2n	P2n on-chip pull-up resistor selection (n = 0 to 6)
0	On-chip pull-up resistor not used
1	On-chip pull-up resistor used

Caution Bit 7 must be set to 0.

(4) Pull-up resistor option register B3 (PUB3)

Pull-up resistor option register B3 (PUB3) sets whether on-chip pull-up resistors on P30 to P33 are used or not.

On the port specified to use an on-chip pull-up resistor by PUB3, the pull-up resistor can be internally used only for the bits set in the input mode. No on-chip pull-up resistors can be used for the bits set in the output mode regardless of the setting of PUB3. This also applies to cases when the pins are used for alternate functions.

PUB3 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears PUB3 to 00H.

Figure 6-14. Format of Pull-Up Resistor Option Register B3

Symbol	7	6	5	4	<3>	<2>	<1>	<0>	Address	After reset	R/W
PUB3	0	0	0	0	PUB33	PUB32	PUB31	PUB30	FF33H	00H	R/W

PUB3n	P3n on-chip pull-up resistor selection (n = 0 to 3)
0	On-chip pull-up resistor not used
1	On-chip pull-up resistor used

Caution Bits 4 to 7 must be set to 0.

6.4 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.

6.4.1 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.

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

(2) In input mode

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

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

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

6.4.2 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.

6.4.3 Arithmetic operation of I/O port

(1) In output mode

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

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

(2) In input mode

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

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

7.1 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 (ceramic/crystal) 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).

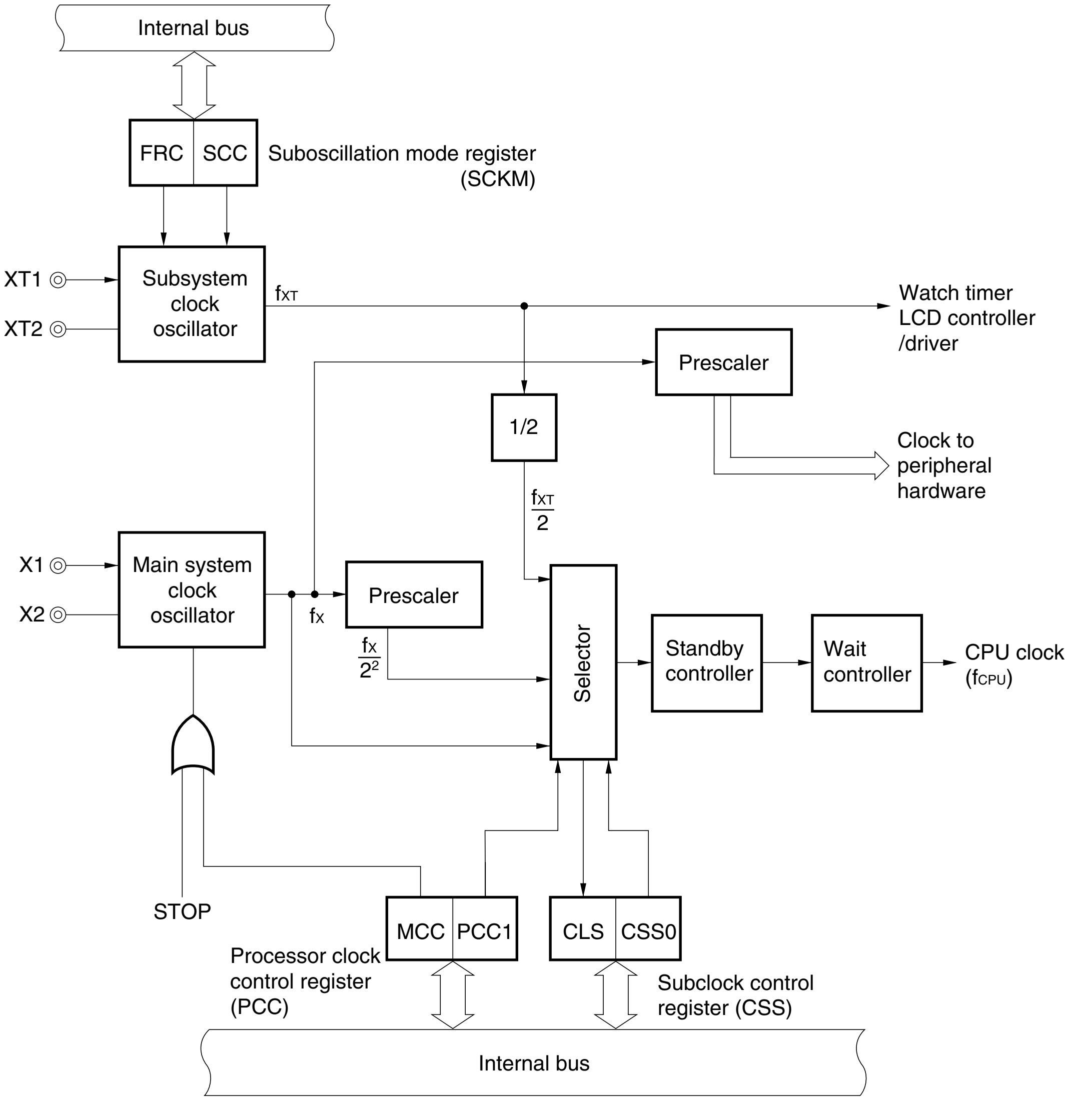
7.2 Clock Generator Configuration

The clock generator includes the following hardware.

Table 7-1. Configuration of Clock Generator

Item	Configuration
Control registers	Processor clock control register (PCC) Suboscillation mode register (SCKM) Subclock control register (CSS)
Oscillators	Main system clock oscillator Subsystem clock oscillator

Figure 7-1. Block Diagram of Clock Generator



7.3 Registers Controlling Clock Generator

The clock generator is controlled by the following registers.

- Processor clock control register (PCC)
- Suboscillation mode register (SCKM)
- Subclock control register (CSS)

(1) Processor clock control register (PCC)

PCC sets CPU clock selection and the division ratio.
PCC is set with a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets PCC to 02H.

Figure 7-2. 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	Control of main system clock oscillator operation
0	Operation enabled
1	Operation disabled

CSS0	PCC1	Selection of CPU clock (f_{CPU})Note	Minimum instruction execution time: $2/f_{CPU}$
			f_X = 5.0 MHz or f_{XT} = 32.768 kHz operation
0	0	f_X	0.4 μs
0	1	$f_X/2^2$	1.6 μs
1	0	$f_{XT}/2$	122 μs
1	1		

Note The CPU clock is selected according to a combination of the PCC1 flag in the processor clock control register (PCC) and the CSS0 flag in the subclock control register (CSS) (Refer to **7.3 (3) Subclock control register (CSS)**).

Cautions 1. Bits 0 and 2 to 6 must be set to 0.

2. The MCC can be set only when the subsystem clock has been selected as the CPU clock.

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

(2) Suboscillation 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 clears SCKM to 00H.

Figure 7-3. Format of Suboscillation 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. Only when 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 clears CSS to 00H.

Figure 7-4. 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.

7.4 System Clock Oscillators

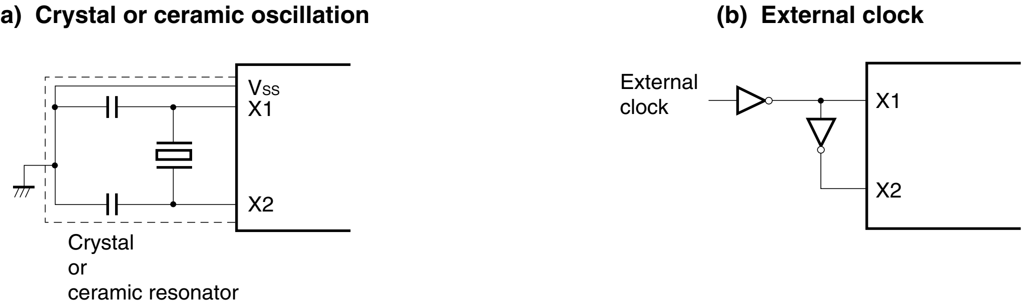
7.4.1 Main system clock oscillator

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

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


(a) Crystal or ceramic 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 7-5 and 7-6 to avoid an adverse effect from wiring capacitance.

- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines. 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 VSS. Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

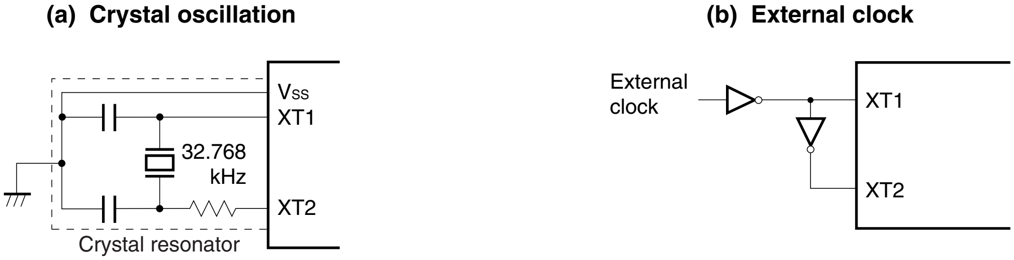
7.4.2 Subsystem clock oscillator

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 7-6 shows the external circuit of the subsystem clock oscillator.

Figure 7-6. External Circuit of Subsystem Clock Oscillator



Caution When using the main system or subsystem clock oscillator, wire as follows in the area enclosed by the broken lines in Figures 7-5 and 7-6 to avoid an adverse effect from wiring capacitance.

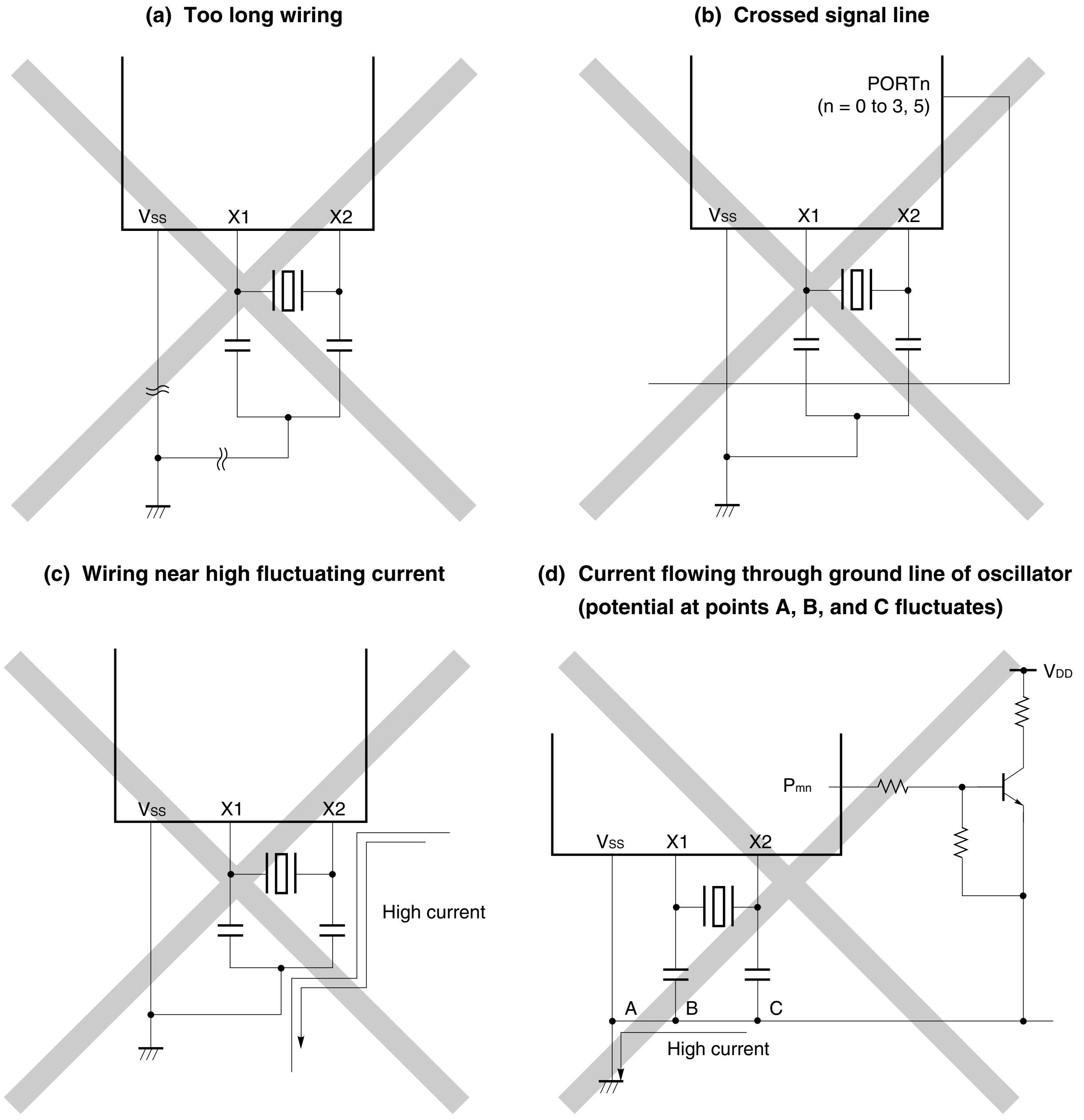
- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as V_{SS}. Do not ground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

7.4.3 Examples of incorrect resonator connection

Figure 7-7 shows examples of incorrect resonator connection.

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



(a) Too long wiring

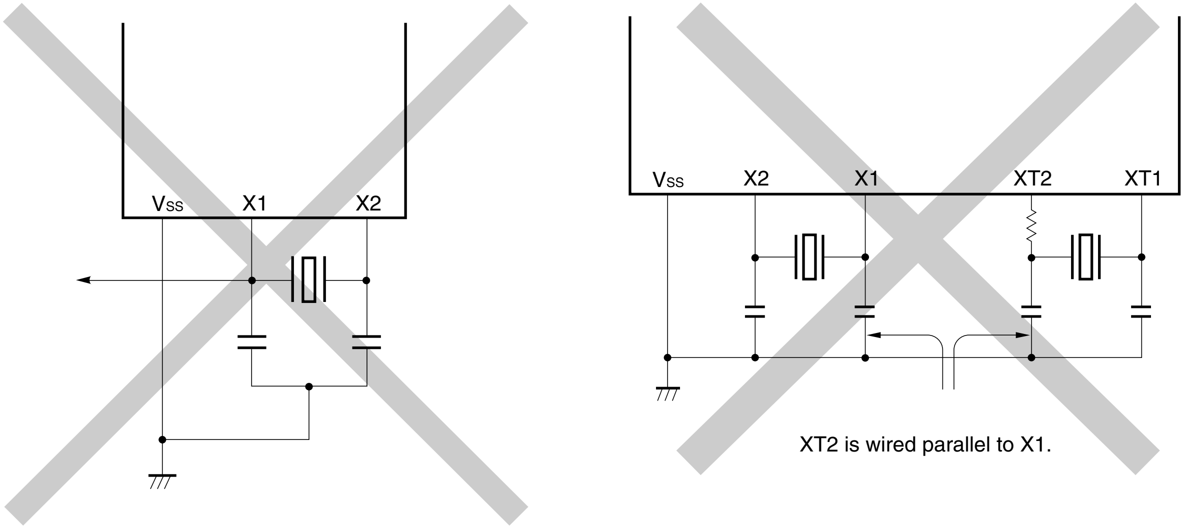
(b) Crossed signal line

(c) Wiring near high fluctuating current

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

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

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



(e) Signal is fetched

(f) Parallel and near signal lines of main system clock and subsystem clock

XT2 is wired parallel to X1.

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

Caution If the X1 wire is in parallel with the XT2 wire, crosstalk noise may occur between the X1 and XT2, resulting in a malfunction.
To avoid this, do not lay the X1 and XT2 wires in parallel.

7.4.4 Divider circuit

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

7.4.5 When no subsystem clock is used

If a 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 VSS
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 suboscillation 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.

7.5 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), suboscillation 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 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.

(c) Two standby modes, STOP and HALT, can be used with the main system clock selected. In a system where no subsystem clock is used, setting bit 1 (FRC) of the SCKM so that the on-chip feedback resistor cannot be used reduces current consumption in STOP mode. In a system where a 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 current 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 the watch timer and LCD controller/driver. The watch timer and 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).

7.6 Changing Setting of System Clock and CPU Clock

7.6.1 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, and the old clock is used for the duration of several instructions after that (see **Table 7-2**).

Table 7-2. 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	x
0	0			4 clocks		$2f_X/f_{XT}$ clocks (306 clocks)	
	1	2 clocks				$f_X/2f_{XT}$ clocks (76 clocks)	
1	x	2 clocks		2 clocks			

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

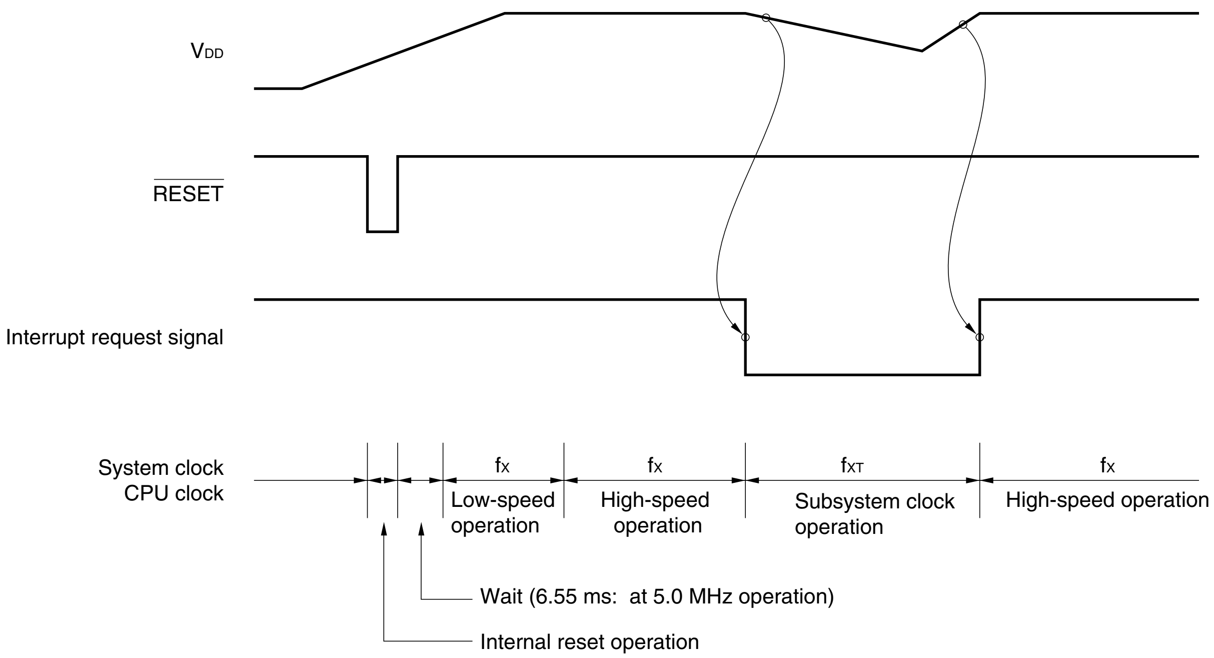
2. The parenthesized values apply to operation at f_X = 5.0 MHz or f_{XT} = 32.768 kHz.

3. x: don't care

7.6.2 Switching between system clock and CPU clock

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

Figure 7-8. 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 ($2^{15}/f_X$) 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.

8.1 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 (RC) oscillator**
  This circuit oscillates at 2.0 to 4.0 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).

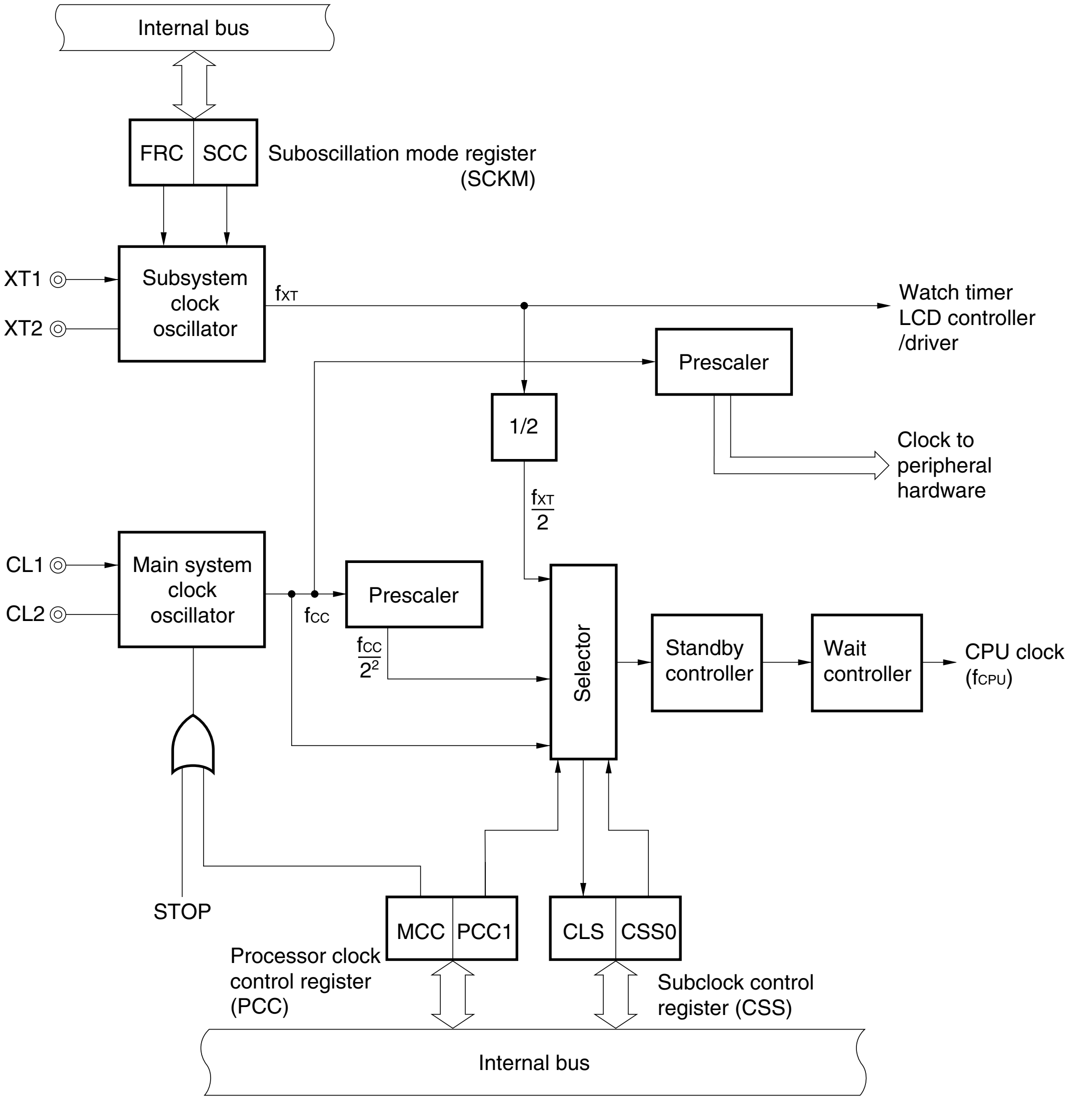
8.2 Clock Generator Configuration

The clock generator includes the following hardware.

Table 8-1. Configuration of Clock Generator

Item	Configuration
Control registers	Processor clock control register (PCC) Suboscillation mode register (SCKM) Subclock control register (CSS)
Oscillators	Main system clock oscillator Subsystem clock oscillator

Figure 8-1. Block Diagram of Clock Generator



8.3 Registers Controlling Clock Generator

The clock generator is controlled by the following registers.

- Processor clock control register (PCC)
- Suboscillation mode register (SCKM)
- Subclock control register (CSS)

(1) Processor clock control register (PCC)

PCC sets CPU clock selection and the division ratio.
PCC is set with a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets PCC to 02H.

Figure 8-2. 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	Control of main system clock oscillator operation
0	Operation enabled
1	Operation disabled

CSS0	PCC1	Selection of CPU clock (f_{CPU})Note	Minimum instruction execution time: $2/f_{CPU}$
			f_{CC} = 4.0 MHz or f_{XT} = 32.768 kHz operation
0	0	f_{CC}	0.5 μs
0	1	$f_{CC}/2^2$	2.0 μs
1	0	$f_{XT}/2$	122 μs
1	1		

Note The CPU clock is selected according to a combination of the PCC1 flag in the processor clock control register (PCC) and the CSS0 flag in the subclock control register (CSS) (Refer to **8.3 (3) Subclock control register (CSS)**).

Cautions 1. Bits 0 and 2 to 6 must be set to 0.

2. The MCC can be set only when the subsystem clock has been selected as the CPU clock.

Remarks 1. f_{CC}: Main system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

(2) Suboscillation 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 clears SCKM to 00H.

Figure 8-3. Format of Suboscillation 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. Only when 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 clears CSS to 00H.

Figure 8-4. 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.

8.4 System Clock Oscillators

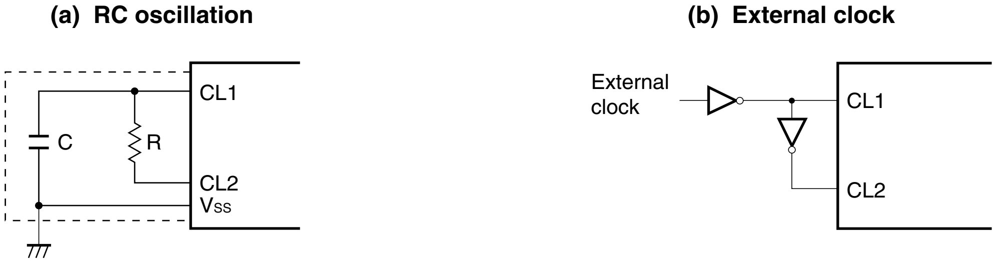
8.4.1 Main system clock oscillator

The main system clock oscillator is oscillated by the resistor (R) and capacitor (C) (4.0 MHz: TYP.) connected across the CL1 and CL2 pins.

An external clock can also be input to the circuit. In this case, input the clock signal to the CL1 pin, and input the inverted signal to the CL2 pin.

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

Figure 8-5. External Circuit of Main System Clock Oscillator



Caution When using the main system or subsystem clock oscillator, wire as follows in the area enclosed by the broken lines in Figures 8-5 and 8-6 to avoid an adverse effect from wiring capacitance.

- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines. 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 VSS. Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

8.4.2 Subsystem clock oscillator

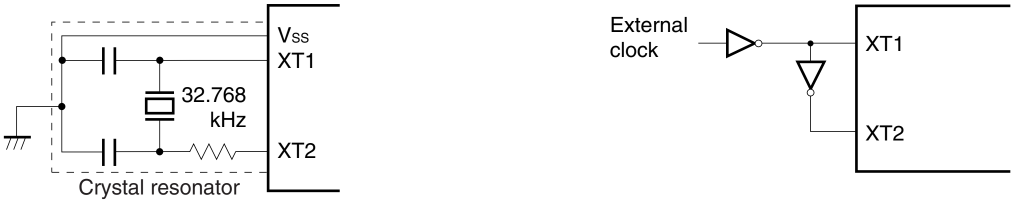
The subsystem clock oscillator is oscillated by the crystal resonator (32.768 kHz TYP.) connected across the XT1 and XT2 pins.

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 8-6 shows the external circuit of the subsystem clock oscillator.

Figure 8-6. 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 8-5 and 8-6 to avoid an adverse effect from wiring capacitance.

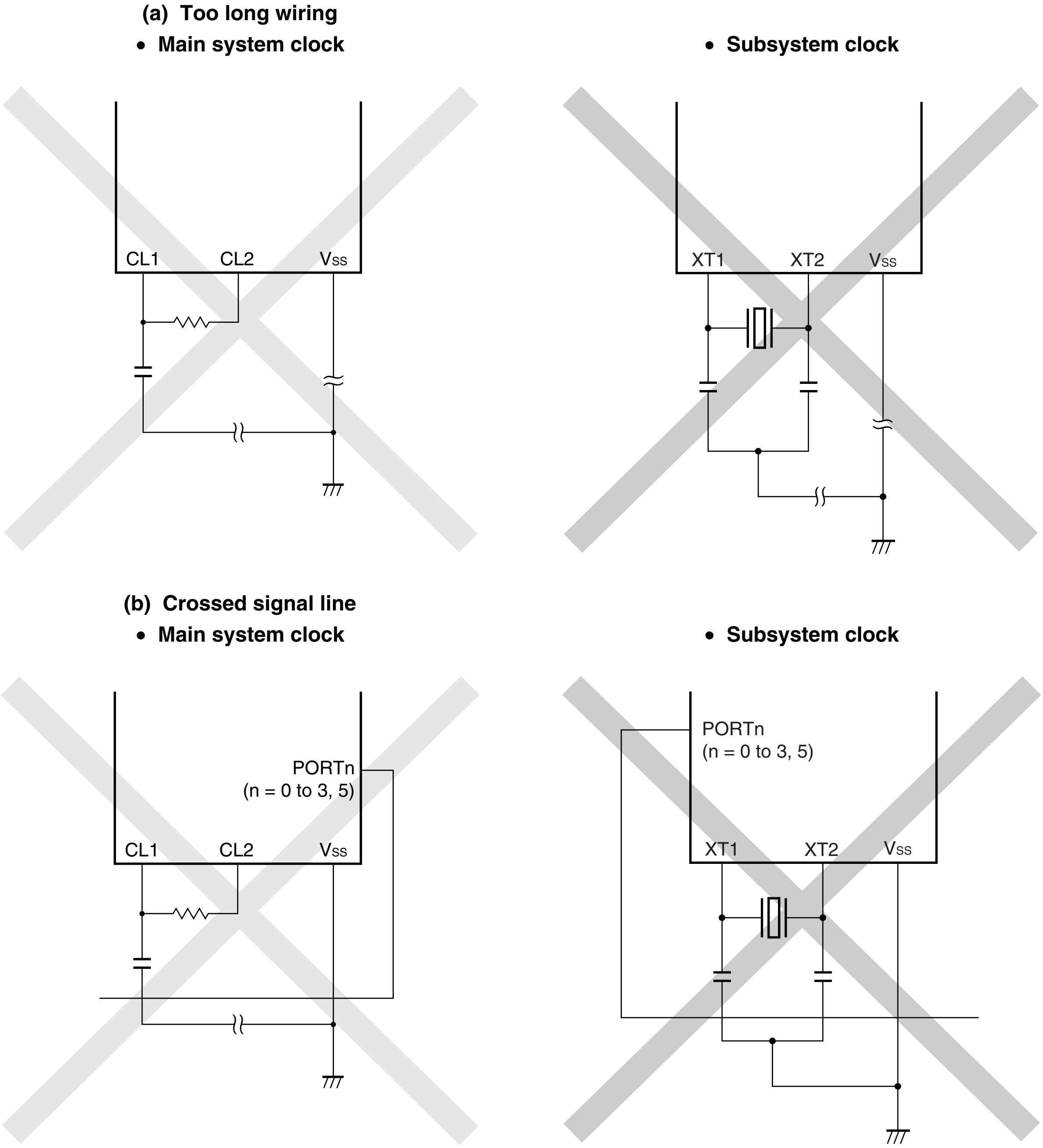
- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as V_{SS}. Do not ground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

8.4.3 Examples of incorrect resonator connection

Figure 8-7 shows examples of incorrect resonator connection.

Figure 8-7. Examples of Incorrect Resonator Connection (1/3)



(a) Too long wiring

- **Main system clock**
- **Subsystem clock**

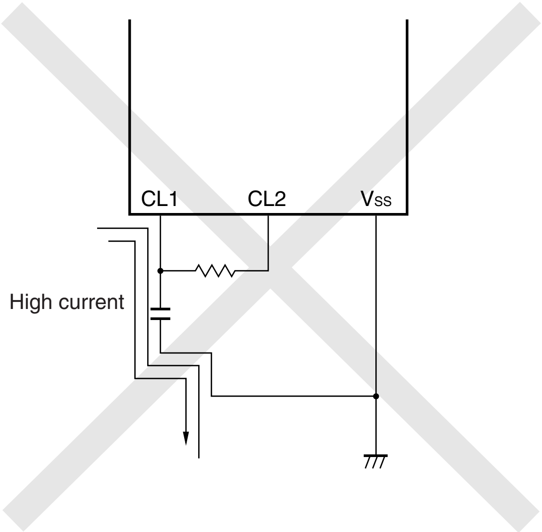
(b) Crossed signal line

- **Main system clock**
- **Subsystem clock**

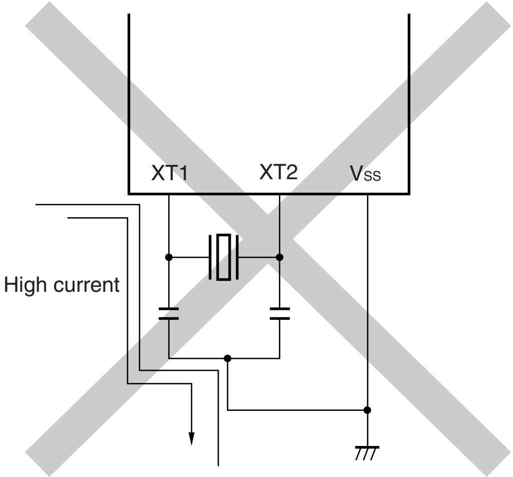
Figure 8-7. Examples of Incorrect Resonator Connection (2/3)

(c) Wiring near high fluctuating current

- **Main system clock**

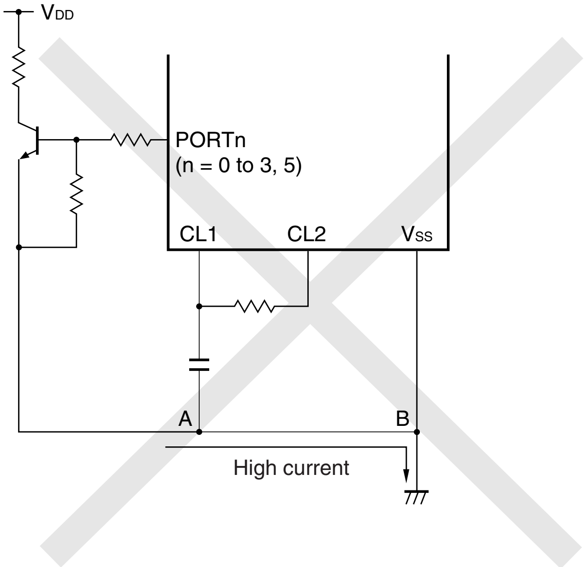


- **Subsystem clock**



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

- **Main system clock**



- **Subsystem clock**

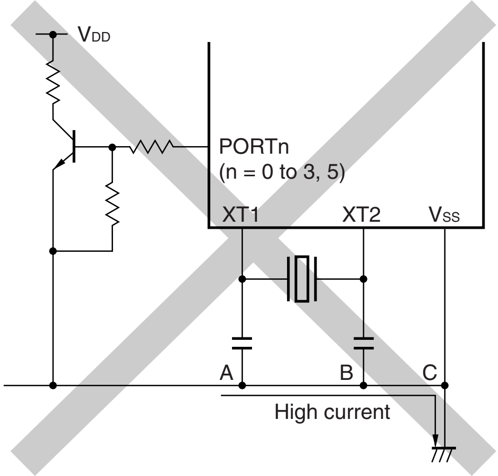
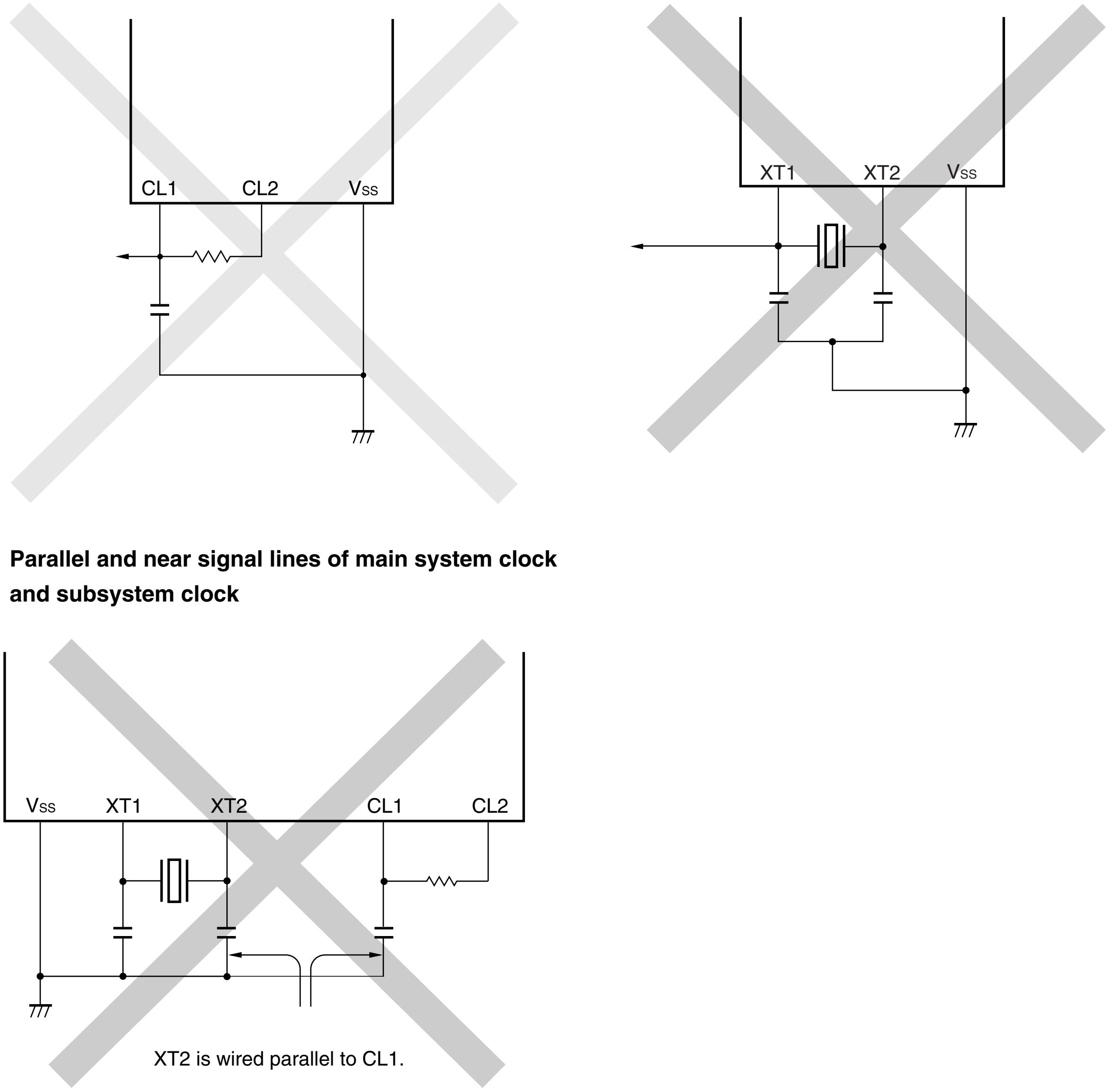


Figure 8-7. Examples of Incorrect Resonator Connection (3/3)



(e) Signal is fetched

- Main system clock
- Subsystem clock

(f) Parallel and near signal lines of main system clock and subsystem clock

XT2 is wired parallel to CL1.

8.4.4 Divider circuit

The divider circuit divides the output of the main system clock oscillator (f_{CC}) to generate various clocks.

8.4.5 When no subsystem clock is used

If a 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 suboscillation 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.

8.5 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_{CC}
- 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), suboscillation mode register (SCKM), and subclock control register (CSS), as follows.

(a) The low-speed mode (2.0 μs: at 4.0 MHz operation) of the main system clock is selected when the $\overline{\text{RESET}}$ signal is generated (PCC = 02H). While a low level is input to the $\overline{\text{RESET}}$ pin, oscillation of the main system clock is stopped.

(b) Three types of Minimum instruction execution time (0.5 μs and 2.0 μs: main system clock (at 4.0 MHz operation), 122 μs: subsystem clock (at 32.768 kHz operation)) can be selected by the PCC, SCKM, and CSS settings.

(c) Two standby modes, STOP and HALT, can be used with the main system clock selected. In a system where no subsystem clock is used, setting bit 1 (FRC) of the SCKM so that the on-chip feedback resistor cannot be used reduces current consumption in STOP mode. In a system where a 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 current 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 the watch timer and LCD controller/driver. The watch timer and 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).

8.6 Changing Setting of System Clock and CPU Clock

8.6.1 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, and the old clock is used for the duration of several instructions after that (see **Table 8-2**).

Table 8-2. 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	x
0	0			4 clocks		$2f_{CC}/f_{XT}$ clocks (244 clocks)	
	1	2 clocks				$f_{CC}/2f_{XT}$ clocks (61 clocks)	
1	x	2 clocks		2 clocks			

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

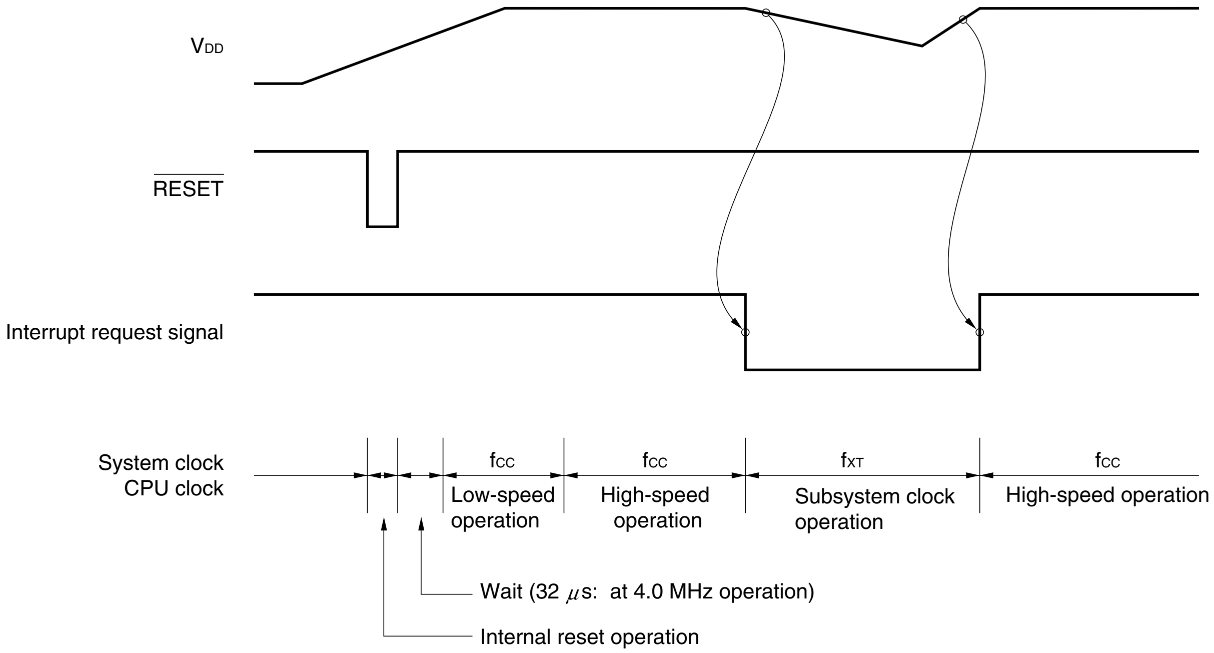
2. The parenthesized values apply to operation at f_{CC} = 4.0 MHz or f_{XT} = 32.768 kHz.

3. x: don't care

8.6.2 Switching between system clock and CPU clock

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

Figure 8-8. Switching Between System Clock and CPU Clock



<1> The CPU is reset when the $\overline{RESET}$ pin is made low on power application. The effect of resetting is released when the $\overline{RESET}$ pin is later made high, and the main system clock starts oscillating. At this time, the oscillation stabilization time ($2^7/f_{CC}$) is automatically secured.
After that, the CPU starts instruction execution at the slow speed of the main system clock (2.0 μs: at 4.0 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 9 16-BIT TIMER 20

16-bit timer 20 references the free running counter and provides the functions such as timer interrupt and timer output. In addition, the count value can be captured by a capture trigger pin.

9.1 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 matches.

(2) Timer output

Timer output control is possible when a count value and compare value matches.

(3) Count value capture

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

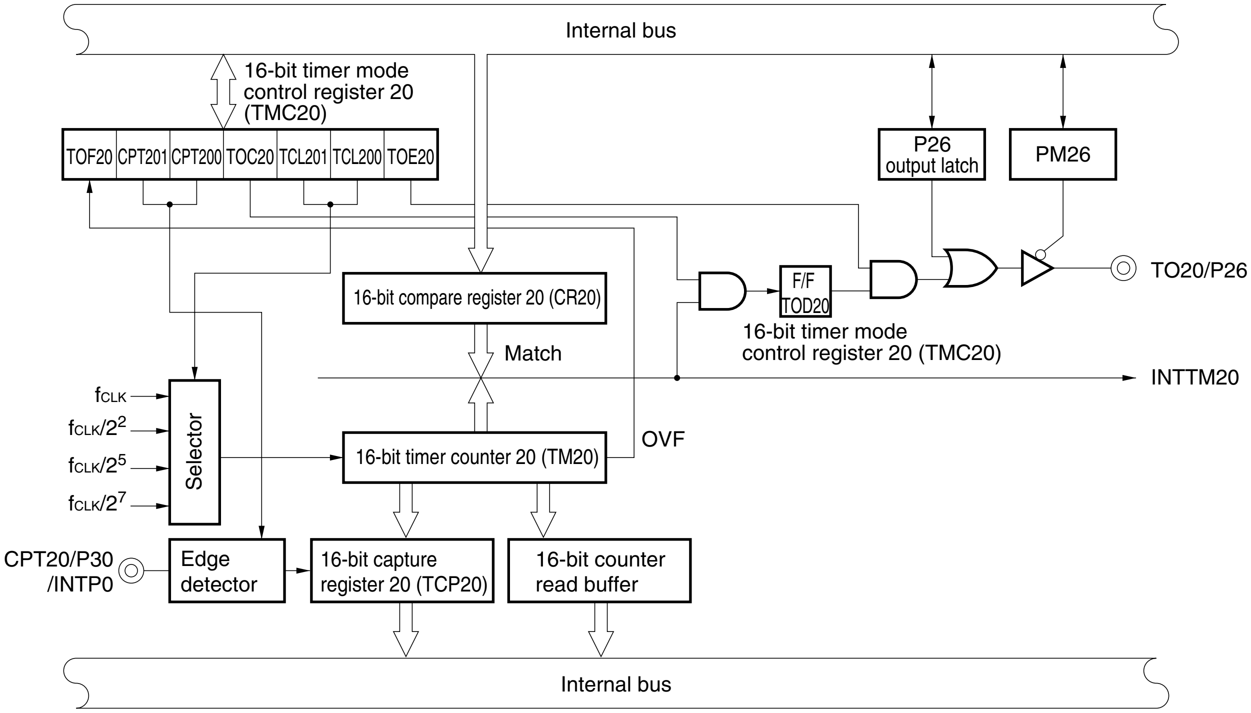
9.2 16-Bit Timer 20 Configuration

16-bit timer 20 includes in the following hardware.

Table 9-1. Configuration of 16-Bit Timer 20

Item	Configuration
Timer counter	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 2 (PM2) Port mode register 3 (PM3) Port 2 (P2)

Figure 9-1. Block Diagram of 16-Bit Timer 20


Internal bus
16-bit timer mode control register 20 (TMC20)
TOF20
CPT201
CPT200
TOC20
TCL201
TCL200
TOE20
P26 output latch
PM26
TO20/P26
16-bit compare register 20 (CR20)
F/F
TOD20
16-bit timer mode control register 20 (TMC20)
Match
INTTM20
fCLK
fCLK/2^2
fCLK/2^5
fCLK/2^7
Selector
16-bit timer counter 20 (TM20)
OVF
CPT20/P30 /INTP0
Edge detector
16-bit capture register 20 (TCP20)
16-bit counter read buffer
Internal bus

Remark f_{CLK}: f_X or f_{CC}

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

This register compares the value set to CR20 with the count value of 16-bit timer counter 20 (TM20), and when they match, generates an interrupt request (INTTM20).

CR20 is set with a 16-bit memory manipulation instruction. The values 0000H to FFFFH can be set. RESET input sets CR20 to FFFFH.

Cautions 1. This register is manipulated with a 16-bit memory manipulation instruction, however an 8-bit memory manipulation instruction can also be used. When manipulated with an 8-bit memory manipulation instruction, the accessing method should be direct addressing.

2. When rewriting CR20 during a count operation, preset CR20 to interrupt disabled using interrupt mask flag register 0 (MK0). Also set the timer output data to inversion disabled using 16-bit timer mode control register 20 (TMC20).

If CR20 is rewritten while interrupts are enabled, an interrupt request may be generated at that time.

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

This is a 16-bit register that counts count pulses.

TM20 is read with a 16-bit memory manipulation instruction.

This register is in free running mode during count clock input.

RESET input sets TM20 to 0000H and then to free running mode again.

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

2. This register is manipulated with a 16-bit memory manipulation instruction, however an 8-bit memory manipulation instruction can also be used. When manipulated with an 8-bit memory manipulation instruction, the accessing method should be direct addressing.

3. When manipulated with an 8-bit memory manipulation instruction, readout should be performed in the order from lower byte to higher byte and must be in pairs.

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

This is a 16-bit register that captures the contents of 16-bit timer counter 20 (TM20).

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

RESET input sets TCP20 undefined.

Caution This register is manipulated with a 16-bit memory manipulation instruction, however an 8-bit memory manipulation instruction can also be used. When manipulated with an 8-bit memory manipulation instruction, the accessing method should be direct addressing.

(4) 16-bit counter read buffer

This buffer latches a counter value and retains the count value of 16-bit timer counter 20 (TM20).

9.3 Registers Controlling 16-Bit Timer 20

The following four registers control 16-bit timer 20.

- 16-bit timer mode control register 20 (TMC20)
- Port mode register 2 (PM2)
- Port mode register 3 (PM3)
- Port 2 (P2)

(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 clears TMC20 to 00H.

Figure 9-2. 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]

TOD20	Timer output data
0	Timer output data is 0
1	Timer output data is 1

TOF20	Overflow flag set
0	Cleared by reset and software
1	Set by overflow of 16-bit timer

CPT201	CPT200	Capture edge selection
0	0	Capture operation disabled
0	1	Rising edge of CPT20
1	0	Falling edge of CPT20
1	1	Both edges of CPT20

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

TCL201	TCL200	16-bit timer counter 20 count clock selection	
		During f_X = 5.0 MHz operation	During f_{CC} = 4.0 MHz operation
0	0	f_X (5.0 MHz)	f_{CC} (4.0 MHz)
0	1	$f_X/2^2$ (1.25 MHz)	$f_{CC}/2^2$ (1.0 MHz)
1	0	$f_X/2^5$ (156.3 kHz)	$f_{CC}/2^5$ (125 kHz)
1	1	$f_X/2^7$ (39.1 kHz)	$f_{CC}/2^7$ (31.3 kHz)

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

Note Bit 7 is read-only.

Remarks 1. f_X: Main system clock oscillation frequency (ceramic/crystal oscillation)

2. f_{CC}: Main system clock oscillation frequency (RC oscillation)

(2) Port mode register 2, 3 (PM2, PM3)

This register sets the input/output of port 2, 3 in 1-bit units.

To use the P26/TO20 pin for timer output, set the output latch of PM26 and P26 to 0.

To use the P30/INTP0/CPT20 pin for capture input, set the PM30 to 1.

PM2, PM3 is set with a 1-bit or 8-bit memory manipulation instruction.

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

Figure 9-3. Format of Port Mode Register 2, 3

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM2	1	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FF22H	FFH	R/W
PM3	1	1	1	1	PM33	PM32	PM31	PM30	FF23H	FFH	R/W

PMmn	Pmn pin input/output mode selection (mn = 20-26, 30-33)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

9.4 16-Bit Timer 20 Operation

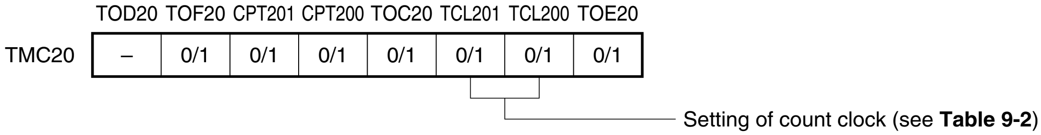
9.4.1 Operation as timer interrupt

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 to CR20
- Set 16-bit timer mode control register 20 (TMC20) as shown in Figure 9-4.

Figure 9-4. Settings of 16-Bit Timer Mode Control Register 20 at Timer Interrupt Operation



	TOD20	TOF20	CPT201	CPT200	TOC20	TCL201	TCL200	TOE20
TMC20	–	0/1	0/1	0/1	0/1	0/1	0/1	0/1

Setting of count clock (see **Table 9-2**)

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

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

Table 9-2 shows the interval time, and Figure 9-5 shows the timing of the timer interrupt operation.

Caution Process as follows when rewriting CR20 during a count operation.

<1> Set the interrupt to disabled (TMMK20 (bit 1of interrupt mask flag register 1(MK1) = 1)

<2> Set the inversion control of timer output data to disabled (TOC20 = 0)

If CR20 is rewritten while interrupts are enabled, an interrupt request may be generated at that time.

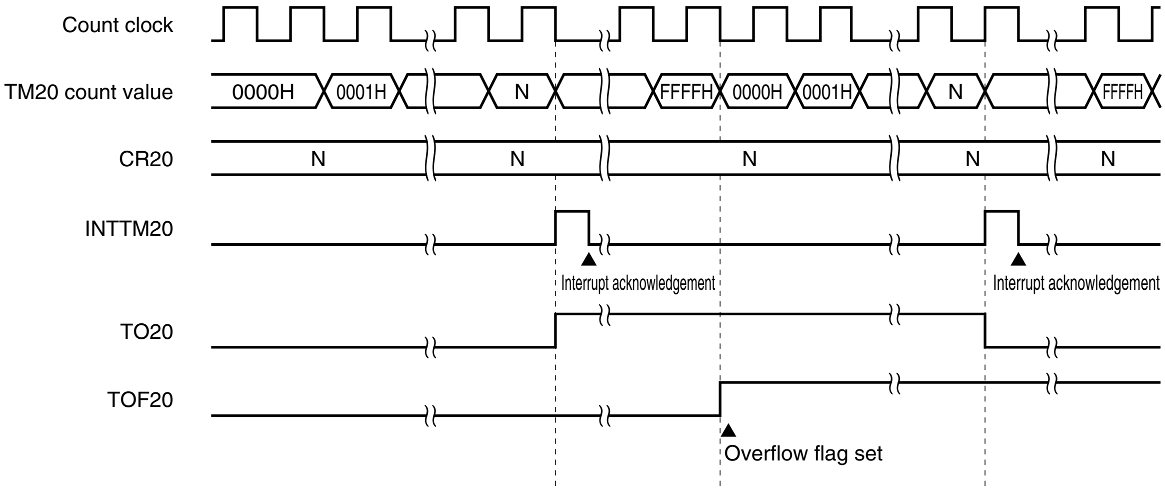
Table 9-2. Interval Time of 16-Bit Timer 20

TCL201	TCL200	Count Clock		Interval Time	
		During fx = 5.0 MHz Operation	During fcc = 4.0 MHz Operation	During fx = 5.0 MHz Operation	During fcc = 4.0 MHz Operation
0	0	1/fx (0.2 μs)	1/fcc (0.25 μs)	2^{16}/fx (13.1 ms)	2^{16}/fcc (16.4 ms)
0	1	2^{2}/fx (0.8 μs)	2^{2}/fcc (1.0 μs)	2^{18}/fx (52.4 ms)	2^{18}/fcc (65.5 ms)
1	0	2^{5}/fx (6.4 μs)	2^{5}/fcc (8.0 μs)	2^{21}/fx (419.4 ms)	2^{21}/fcc (524.2 ms)
1	1	2^{7}/fx (25.6 μs)	2^{7}/fcc (32 μs)	2^{23}/fx (1.68 s)	2^{23}/fcc (2.10 ms)

Remarks 1. fx: Main system clock oscillation frequency (ceramic/crystal oscillation)

2. fcc: Main system clock oscillation frequency (RC oscillation)

Figure 9-5. Timing of Timer Interrupt Operation



Remark N = 0000H to FFFFH

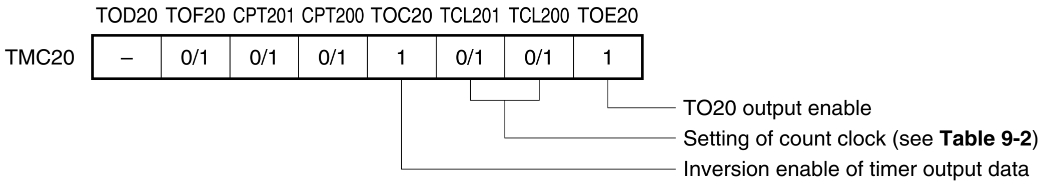
9.4.2 Operation as timer output

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 P26 to output mode (PM26 = 0)
- Set the output latch of P26 to 0
- Set the count value to CR20
- Set 16-bit timer mode control register 20 (TMC20) as shown in Figure 9-6

Figure 9-6. Settings of 16-Bit Timer Mode Control Register 20 at Timer Output Operation

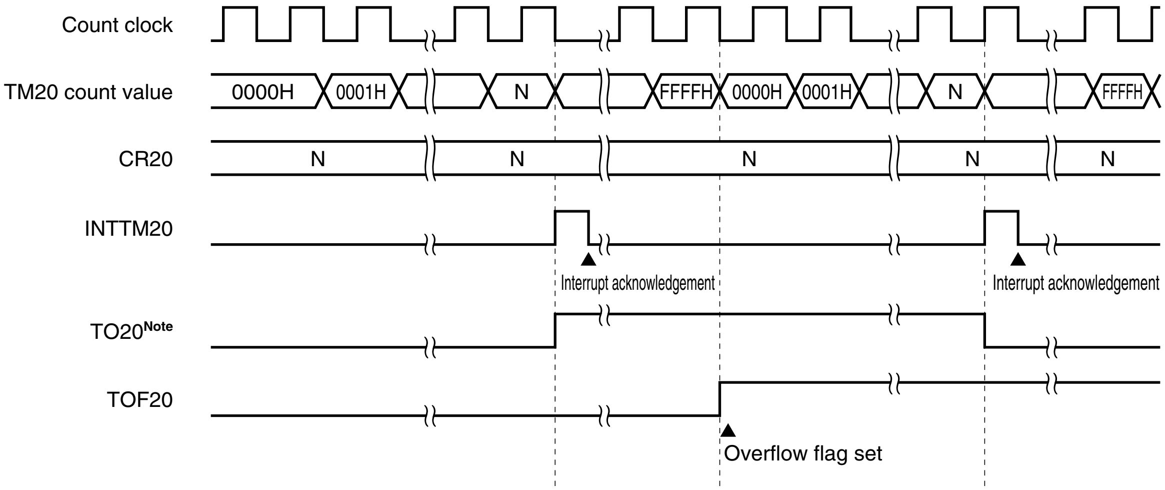


Caution **If both the CPT201 flag and CPT200 flag are set to 0, the capture edge becomes operation disabled.**

When the count value of 16-bit timer counter 20 (TM20) matches the value set in CR20, the output status of the TO20/P26 pin is inverted. This enables timer output. At that time, TM20 continues counting and an interrupt request signal (INTTM20) is generated.

Figure 9-7 shows the timing of timer output (see **Table 9-2** for the interval time of the 16-bit timer 20).

Figure 9-7. Timer Output Timing



Note The TO20 initial value becomes low level during output enable (TOE20 = 1).

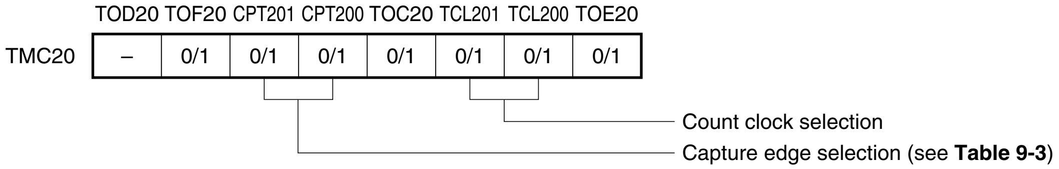
Remark N = 0000H to FFFFH

9.4.3 Capture operation

The capture operation functions to capture and latch the count value of 16-bit timer counter 20 (TM20) to the capture register in synchronization with a capture trigger.

Set as shown in Figure 9-8 to allow 16-bit timer 20 to start the capture operation.

Figure 9-8. Settings of 16-Bit Timer Mode Control Register 20 at Capture Operation



16-bit capture register 20 (TCP20) starts a capture operation after the CPT20 capture trigger edge has been detected, and latches and retains the count value of 16-bit timer counter 20 (TM20). TCP20 fetches count value within 2 clocks and retains the count value until the next capture edge detection.

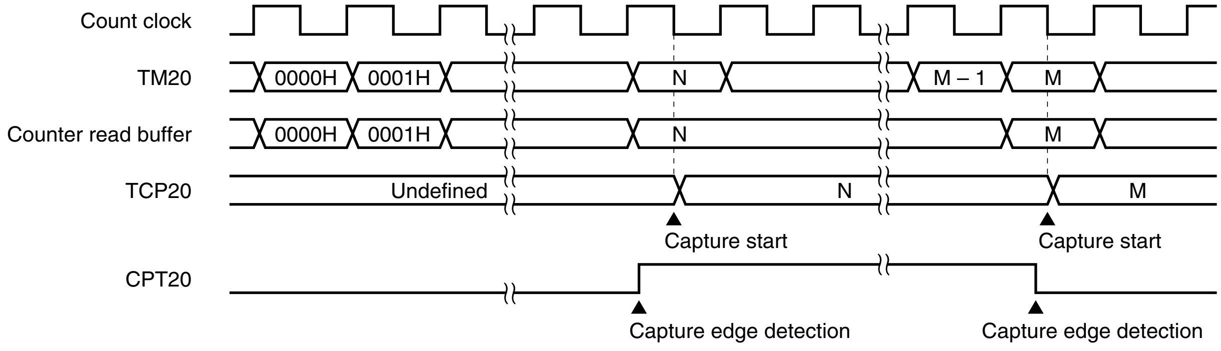
Table 9-3 and Figure 9-9 show the setting contents of the capture edge and capture operation timing, respectively.

Table 9-3. Setting Contents of Capture Edge

CPT201	CPT200	Capture Edge Selection
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

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

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



9.4.4 16-bit timer counter 20 readout

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

TM20 readout is performed through a counter read buffer. The counter read buffer latches the TM20 count value, and 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 counter read buffer value in the retention state can be read out as the count value.

Cancellation of pending 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 starts it free running.

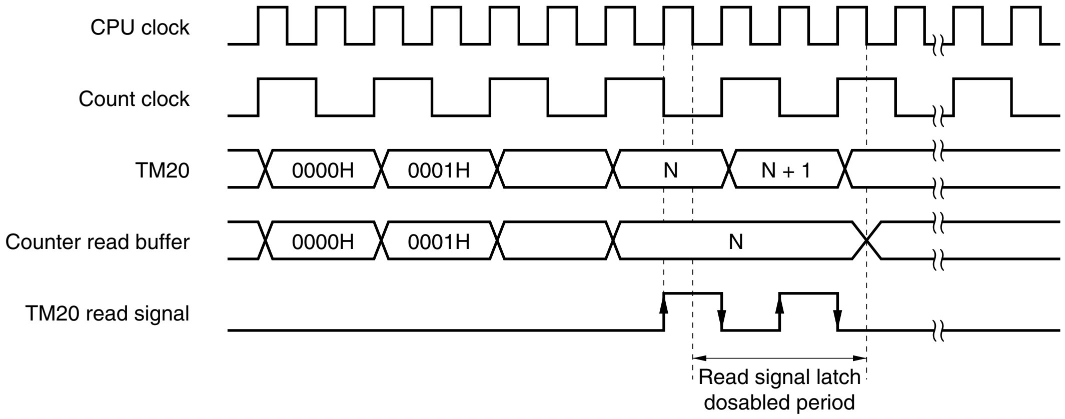
Figure 9-10 shows the timing of 16-bit timer counter 20 readout.

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

2. Although TM20 is a register dedicated for a 16-bit transfer instruction, an 8-bit transfer instruction can also be used.
When using an 8-bit transfer instruction, execute it using 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 9-10. 16-Bit Timer Counter 20 Readout Timing



Remark N = 0000H to FFFFH

9.5 Cautions on Using 16-Bit Timer 20

9.5.1 Restrictions when rewriting 16-bit compare register 20

(1) Disable interrupts (TMMK20 = 1) and the inversion control of timer output (TOC20 = 0) before rewriting the compare register (CR20).
If CR20 is rewritten with interrupts enabled, an interrupt request may be generated immediately.

(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 A or B.

<Countermeasure A> When rewriting using 8-bit access

<1> Disable interrupts (TMMK20 = 1) and the inversion control of timer output (TOC20 = 0).
<2> First rewrite the higher 1 byte of CR20 (16 bits).
<3> Then rewrite the lower 1 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 beginning of the interrupt.

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

```
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 higher byte (6 clocks)
          MOV  !0FF17H,A   ;  Rewrite CR20 higher byte (8 clocks)
          MOV  A,#yyH      ;  Set the rewrite value of lower byte (6 clocks)
          MOV  !0FF16H,A   ;  Rewrite CR20 lower byte (8 clocks)
          CLR1 TMIF20      ;  Clear 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 the 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 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 rewriting CR20 (MOVW CR20, AX).

CHAPTER 10 8-BIT TIMER 30, 40

10.1 8-Bit Timer 30, 40 Functions

An 8-bit timer (one channel, timer 30) and an 8-bit timer/event counter (one channel, timer 40) are incorporated in the µPD789306, 789316 Subseries. The operation modes listed in the following table can be set via mode register settings.

Table 10-1. Operation Modes

Mode \ Channel	Timer 30	Timer 40
8-bit timer counter mode (Discrete mode)	Available	Available
16-bit timer counter mode (Cascade connection mode)	Available	
Carrier generator mode	Available	
PWM output mode	Not available	Available

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

The following functions can be used in this mode.

- Interval timer with 8-bit resolution
- External event counter with 8-bit resolution (timer 40 only)
- Square wave output with 8-bit resolution

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

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

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

(3) Carrier generator mode

The carrier clock generated by timer 40 is output in cycles set by timer 30.

(4) PWM output mode (timer 40 only)

Pulses are output using any duty factor set by timer 40.

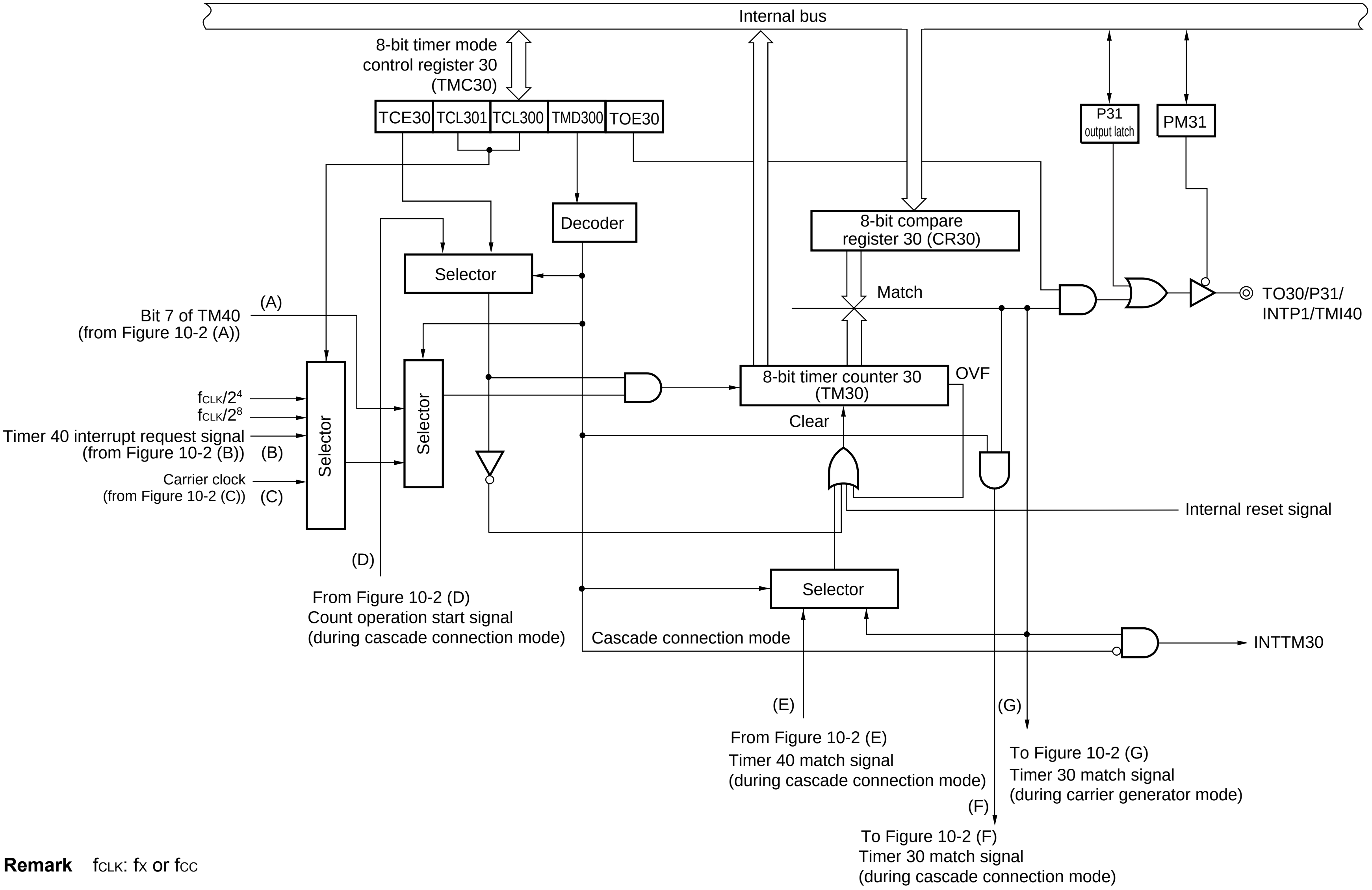
10.2 8-Bit Timer 30, 40 Configuration

The 8-bit timer 30, 40 includes the following hardware.

Table 10-2. 8-Bit Timer 30, 40 Configuration

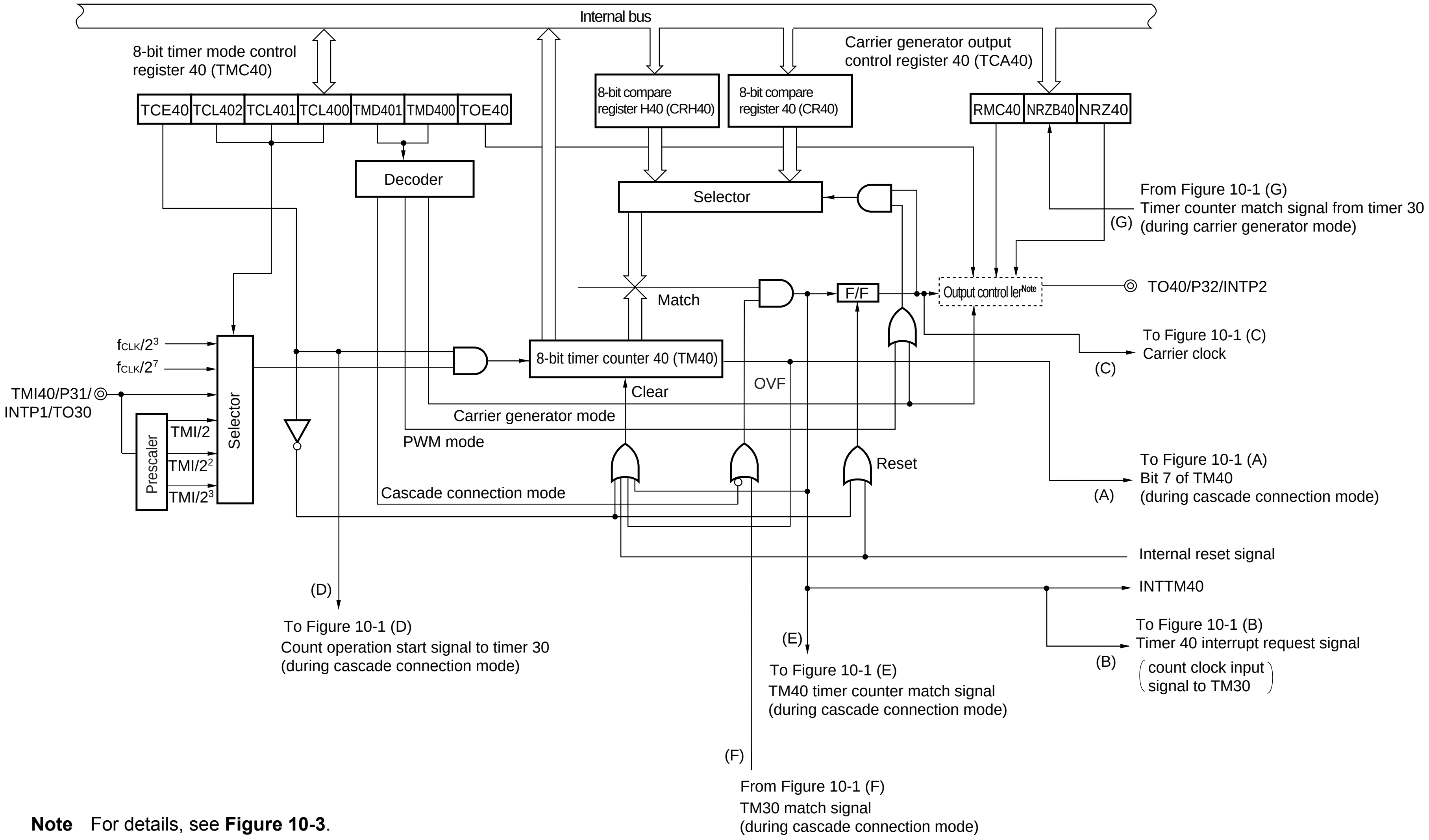
Item	Configuration
Timer counters	8 bits × 2 (TM30, TM40)
Registers	Compare registers: 8 bits × 3 (CR30, CR40, CRH40)
Timer outputs	2 (TO30, TO40)
Control registers	8-bit timer mode control register 30 (TMC30) 8-bit timer mode control register 40 (TMC40) Carrier generator output control register 40 (TCA40) Port mode register 3 (PM3) Port 3 (P3)

Figure 10-1. Block Diagram of Timer 30



Remark f_{CLK}: f_X or f_{CC}

Figure 10-2. Block Diagram of Timer 40

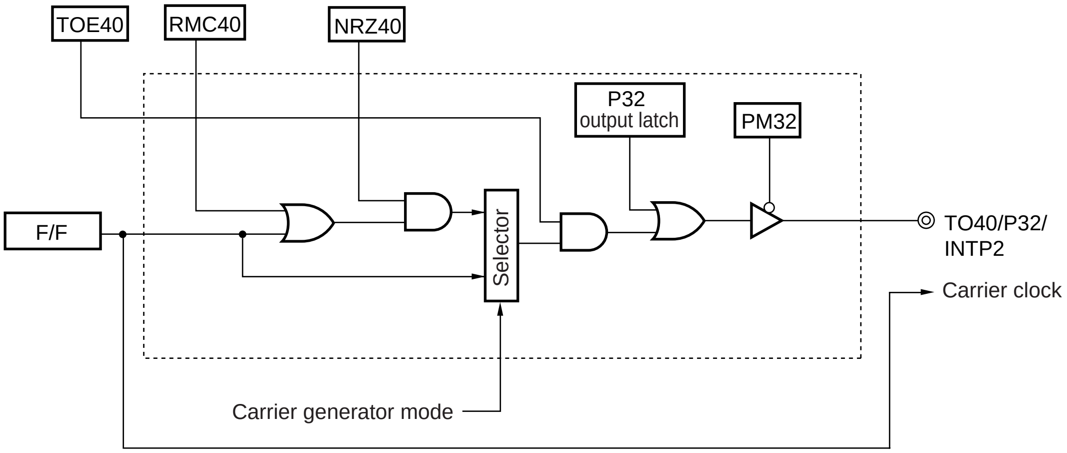


Internal bus
8-bit timer mode control register 40 (TMC40)
TCE40
TCL402
TCL401
TCL400
TMD401
TMD400
TOE40
Decoder
8-bit compare register H40 (CRH40)
8-bit compare register 40 (CR40)
Carrier generator output control register 40 (TCA40)
RMC40
NRZB40
NRZ40
Selector
From Figure 10-1 (G)
Timer counter match signal from timer 30
(G)
(during carrier generator mode)
Match
F/F
Output control ler^{Note}
TO40/P32/INTP2
To Figure 10-1 (C)
Carrier clock
(C)
fCLK/2³
fCLK/2⁷
TMI40/P31/
INTP1/TO30
Prescaler
TMI/2
TMI/2²
TMI/2³
Selector
8-bit timer counter 40 (TM40)
Clear
OVF
Carrier generator mode
PWM mode
Cascade connection mode
Reset
To Figure 10-1 (A)
Bit 7 of TM40
(A)
(during cascade connection mode)
Internal reset signal
INTTM40
(D)
To Figure 10-1 (D)
Count operation start signal to timer 30
(during cascade connection mode)
(E)
To Figure 10-1 (E)
TM40 timer counter match signal
(during cascade connection mode)
To Figure 10-1 (B)
Timer 40 interrupt request signal
(B)
(count clock input signal to TM30)
(F)
From Figure 10-1 (F)
TM30 match signal
(during cascade connection mode)

Note For details, see **Figure 10-3**.

Remark f_{CLK}: f_X or f_{CC}

Figure 10-3. Block Diagram of Output Controller (Timer 40)



(1) 8-bit compare register 30 (CR30)

This 8-bit register is used to continually compare the value set to CR30 with the count value in 8-bit timer counter 30 (TM30) and to generate an interrupt request (INTTM30) when a match occurs.

CR30 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes CR30 undefined.

Caution CR30 cannot be used during PWM output mode.

(2) 8-bit compare register 40 (CR40)

This 8-bit register is used to continually compare the value set to CR40 with the count value in 8-bit timer counter 40 (TM40) and to generate an interrupt request (INTTM40) when a match occurs. When connected to TM30 via a cascade connection and used as a 16-bit timer/event counter, the interrupt request (INTTM40) occurs only when matches occur simultaneously between CR30 and TM30 and between CR40 and TM40 (INTTM30 does not occur).

In carrier generator mode and PWM output mode, this registers is used for low-level width setting.

CR40 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes CR40 undefined.

(3) 8-bit compare register H40 (CRH40)

During carrier generator mode or PWM output mode, the high-level width of timer output is set by writing a value to CRH40.

The value set in CRH40 is constantly compared with TM40 count value, and an interrupt request (INTTM40) is generated if they match.

CRH40 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes CRH40 undefined.

(4) 8-bit timer counters 30 and 40 (TM30 and TM40)

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

TM30 and TM40 are read with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears TM30 and TM40 to 00H.

TM30 and TM40 are cleared to 00H under the following conditions.

(a) Discrete mode

(i) TM30

- After reset
- When TCE30 (bit 7 of 8-bit timer mode control register 30 (TMC30)) is cleared to 0
- When a match occurs between TM30 and CR30
- When the TM30 count value overflows

(ii) TM40

- After reset
- When TCE40 (bit 7 of 8-bit timer mode control register 40 (TMC40)) is cleared to 0
- When a match occurs between TM40 and CR40
- When the TM40 count value overflows

(b) Cascade connection mode (TM30 and TM40 are simultaneously cleared to 00H)

- After reset
- When the TCE40 flag is cleared to 0
- When matches occur simultaneously between TM30 and CR30 and between TM40 and CR40
- When the TM30 and TM40 count values overflow simultaneously

(c) Carrier generator mode/PWM output mode (TM40 only)

- After reset
- When the TCE40 flag is cleared to 0
- When a match occurs between TM40 and CR40
- When a match occurs between TM40 and CRH40
- When the TM40 count value overflows

10.3 Registers Controlling 8-Bit Timer 30, 40

The 8-bit timer 30, 40 is controlled by the following five registers.

- 8-bit timer mode control register 30 (TMC30)
- 8-bit timer mode control register 40 (TMC40)
- Carrier generator output control register 40 (TCA40)
- Port mode register 3 (PM3)
- Port 3 (P3)

(1) 8-bit timer mode control register 30 (TMC30)

8-bit timer mode control register 30 (TMC30) is used to control the timer 30 count clock setting and the operation mode setting.

TMC30 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears TMC30 to 00H.

Figure 10-4. Format of 8-Bit Timer Mode Control Register 30

Symbol	<7>	6	5	4	3	2	1	<0>	Address	After reset	R/W
TMC30	TCE30	0	0	TCL301	TCL300	0	TMD300	TOE30	FF4DH	00H	R/W

TCE30	Control of TM30 count operation[Note 1]
0	Clears TM30 count value and stops operation
1	Starts count operation

TCL301	TCL300	Selection of timer 30 count clock	
		During f_X = 5.0 MHz operation	During f_{CC} = 4.0 MHz operation
0	0	$f_X/2^4$ (312.5 kHz)	$f_{CC}/2^4$ (250 kHz)
0	1	$f_X/2^8$ (19.5 kHz)	$f_{CC}/2^8$ (15.6 kHz)
1	0	Timer 40 match signal	
1	1	Carrier clock (during carrier generator mode) or timer 40 output signal (during a mode other than carrier generator mode)	

TMD300	TMD401	TMD400	Selection of operation mode for timer 30 and timer 40[Note 2]
0	0	0	8-bit timer counter mode (discrete mode)
1	0	1	16-bit timer counter mode (cascade connection mode)
0	1	1	Carrier generator mode
0	1	0	Timer 40: PWM output mode Timer 30: 8-bit timer counter mode
Other than above			Setting prohibited

TOE30	Control of timer output
0	Output disabled (port mode)
1	Output enabled

Notes 1. Since the count operation is controlled by TCE40 (bit 7 of TMC40) in cascade connection mode, any setting for TCE30 is ignored.

2. The operation mode selection is set to both the TMC30 register and TMC40 register.

Caution In cascade connection mode, the timer 40 output signal is forcibly selected for the count clock.

Remarks 1. f_X: Main system clock oscillation frequency (ceramic/crystal oscillation)

2. f_{CC}: Main system clock oscillation frequency (RC oscillation)

(2) 8-bit timer mode control register 40 (TMC40)

8-bit timer mode control register 40 (TMC40) is used to control the timer 40 count clock setting and the operation mode setting.

TMC40 is set with a 1-bit or 8-bit memory manipulation instruction.

RESET input clears TMC40 to 00H.

Figure 10-5. Format of 8-Bit Timer Mode Control Register 40

Symbol	<7>	6	5	4	3	2	1	<0>	Address	After reset	R/W
TMC40	TCE40	0	TCL402	TCL401	TCL400	TMD401	TMD400	TOE40	FF4EH	00H	R/W

TCE40	Control of TM40 count operation[Note 1]
0	Clears TM40 count value and stops operation (the count value is also cleared for TM30 during cascade connection mode)
1	Starts count operation (the count operation is also started for TM30 during cascade connection mode)

TCL402	TCL401	TCL400	Selection of timer 40 count clock	
			During f_X = 5.0 MHz operation	During f_{CC} = 4.0 MHz operation
0	0	0	$f_X/2^3$ (625 kHz)	$f_{CC}/2^3$ (500 kHz)
0	0	1	$f_X/2^7$ (39.1 kHz)	$f_{CC}/2^7$ (31.3 kHz)
0	1	0	f_{TMI}	
0	1	1	$f_{TMI}/2$	
1	0	0	$f_{TMI}/2^2$	
1	0	1	$f_{TMI}/2^3$	

TMD300	TMD401	TMD400	Selection of operation mode for timer 30 and timer 40[Note 2]
0	0	0	8-bit timer counter mode (discrete mode)
1	0	1	16-bit timer counter mode (cascade connection mode)
0	1	1	Carrier generator mode
0	1	0	Timer 40: PWM output mode Timer 30: 8-bit timer counter mode
Other than above			Setting prohibited

TOE40	Control of timer output
0	Output disabled (port mode)
1	Output enabled

Notes 1. Since the count operation is controlled by TCE40 (bit 7 of TMC40) in cascade connection mode, any setting for TCE30 is ignored.

2. The operation mode selection is set to both the TMC30 register and TMC40 register.

Remarks 1. f_X: Main system clock oscillation frequency (ceramic/crystal oscillation)

2. f_{CC}: Main system clock oscillation frequency (RC oscillation)

3. f_{TMI}: External clock input from the TMI40 pin

(3) Carrier generator output control register 40 (TCA40)

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

TCA40 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears TCA40 to 00H.

Figure 10-6. Format of Carrier Generator Output Control Register 40

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TCA40	0	0	0	0	0	RMC40	NRZB40	NRZ40	FF4FH	00H	W

RMC40	Control of remote control output
0	When NRZ40 = 1, a carrier pulse is output to TO40/INTP2/P32 pin
1	When NRZ40 = 1, high-level signal is output to TO40/INTP2/P32 pin

NRZB40	This is the bit that stores the next data to be output to NRZ40. Data is transferred to NRZ40 at the rising edge of the timer 30 match signal.

NRZ40	No return zero data
0	Outputs low-level signal (carrier clock is stopped)
1	Outputs carrier pulse or high-level signal

Cautions 1. Bits 3 to 7 must be set to 0.

2. TCA40 cannot be set with a 1-bit memory manipulation instruction. Be sure to use an 8-bit memory manipulation instruction to set TCA40.

3. The NRZ40 flag can be written only when carrier generator output is stopped (TOE40 = 0). The data cannot be overwritten when TOE40 = 1.

4. When the carrier generator is stopped once and then started again, NRZB40 does not hold the previous data. Re-set data to NRZB40. At this time, a 1-bit memory manipulation instruction must not be used. Be sure to use an 8-bit memory manipulation instruction.

5. To enable operation in the carrier generator mode, set a value to the compare registers (CR30, CR40, and CRH40), and input the necessary value to the NRZB40 and NRZ40 flags in advance. Otherwise, the signal of the timer match circuit will become unstable and the NRZ40 flag will be undefined.

6. While INTTM30 (interrupt generated by the match signal of timer 30) is being output, accessing TCA40 is prohibited.

Accessing TCA40 is prohibited while 8-bit timer counter 30 (TM30) is 00H.

To access TCA40 while TM30 = 00H, wait for more than half a period of the TM30 count clock and then rewrite TCA40.

(4) 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 P31/TO30/INTP1/TMI40 pin as a timer output, set the PM31 and P31 output latch to 0.

When using the P31/TO30/INTP1/TMI40 pin as a timer input, set the PM31 to 1.

When using the P32/TO40/INTP2 pin as a timer output, set the PM32 and P32 output latch to 0.

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

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

Figure 10-7. Format of Port Mode Register 3

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

PM3n	I/O mode of P3n pin (n = 0 to 3)
0	Output mode (output buffer is ON)
1	Input mode (output buffer is OFF)

10.4 8-Bit Timer 30, 40 Operation

10.4.1 Operation as 8-bit timer counter

Timer 30 and timer 40 can independently be used as an 8-bit timer counter.
The following modes can be used for the 8-bit timer counter.

- Interval timer with 8-bit resolution
- External event counter with 8-bit resolution (timer 40 only)
- Square wave output with 8-bit resolution

(1) Operation as interval timer with 8-bit resolution

The interval timer with 8-bit resolution repeatedly generates an interrupt at a time interval specified by the count value preset in 8-bit compare register n0 (CRn0).
To operate 8-bit timer n0 as an interval timer, settings must be made in the following sequence.

<1> Disable operation of 8-bit timer counter n0 (TMn0) (TCEn0 = 0).
<2> Disable timer output of TOn0 (TOEn0 = 0).
<3> Set a count value in CRn0.
<4> Set the operation mode of timer n0 to 8-bit timer counter mode (see Figures 10-4 and 10-5).
<5> Set the count clock for timer n0 (see Tables 10-3 to 10-6).
<6> Enable the operation of TMn0 (TCEn0 = 1).

When the count value of 8-bit timer counter n0 (TMn0) matches the value set in CRn0, TMn0 is cleared to 00H and continues counting. At the same time, an interrupt request signal (INTTMn0) is generated.
Tables 10-3 to 10-6 show interval time, and Figures 10-8 to 10-12 show the timing of the interval timer operation.

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

Remark n = 3, 4

Table 10-3. Interval Time of Timer 30 (During f_X = 5.0 MHz Operation)

TCL301	TCL300	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	$2^4/f_X$ (3.2 μs)	$2^{12}/f_X$ (0.82 ms)	$2^4/f_X$ (3.2 μs)
0	1	$2^8/f_X$ (51.2 μs)	$2^{16}/f_X$ (13.1 ms)	$2^8/f_X$ (51.2 μs)
1	0	Input cycle of timer 40 match signal	Input cycle of timer 40 match signal × 2^8	Input cycle of timer 40 match signal
1	1	Input cycle of timer 40 output	Input cycle of timer 40 output × 2^8	Input cycle of timer 40

Remark f_X: Main system clock oscillation frequency (ceramic/crystal oscillation)

Table 10-4. Interval Time of Timer 30 (During f_{CC} = 4.0 MHz Operation)

TCL301	TCL300	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	$2^4/f_{CC}$ (4.0 μs)	$2^{12}/f_{CC}$ (1.02 ms)	$2^4/f_{CC}$ (4.0 μs)
0	1	$2^8/f_{CC}$ (64 μs)	$2^{16}/f_{CC}$ (16.4 ms)	$2^8/f_{CC}$ (64 μs)
1	0	Input cycle of timer 40 match signal	Input cycle of timer 40 match signal × 2^8	Input cycle of timer 40 match signal
1	1	Input cycle of timer 40 output	Input cycle of timer 40 output × 2^8	Input cycle of timer 40

Remark f_{CC}: Main system clock oscillation frequency (RC oscillation)

Table 10-5. Interval Time of Timer 40 (During f_X = 5.0 MHz Operation)

TCL402	TCL401	TCL400	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	0	$2^3/f_X$ (1.6 μs)	$2^{11}/f_X$ (0.41 ms)	$2^3/f_X$ (1.6 μs)
0	0	1	$2^7/f_X$ (25.6 μs)	$2^{15}/f_X$ (6.55 ms)	$2^7/f_X$ (25.6 μs)
0	1	0	f_{TMI} input cycle	f_{TMI} input cycle × 2^8	f_{TMI} input cycle
0	1	1	$f_{TMI}/2$ input cycle	$f_{TMI}/2$ input cycle × 2^8	$f_{TMI}/2$ input cycle
1	0	0	$f_{TMI}/2^2$ input cycle	$f_{TMI}/2^2$ input cycle × 2^8	$f_{TMI}/2^2$ input cycle
1	0	1	$f_{TMI}/2^3$ input cycle	$f_{TMI}/2^3$ input cycle × 2^8	$f_{TMI}/2^3$ input cycle

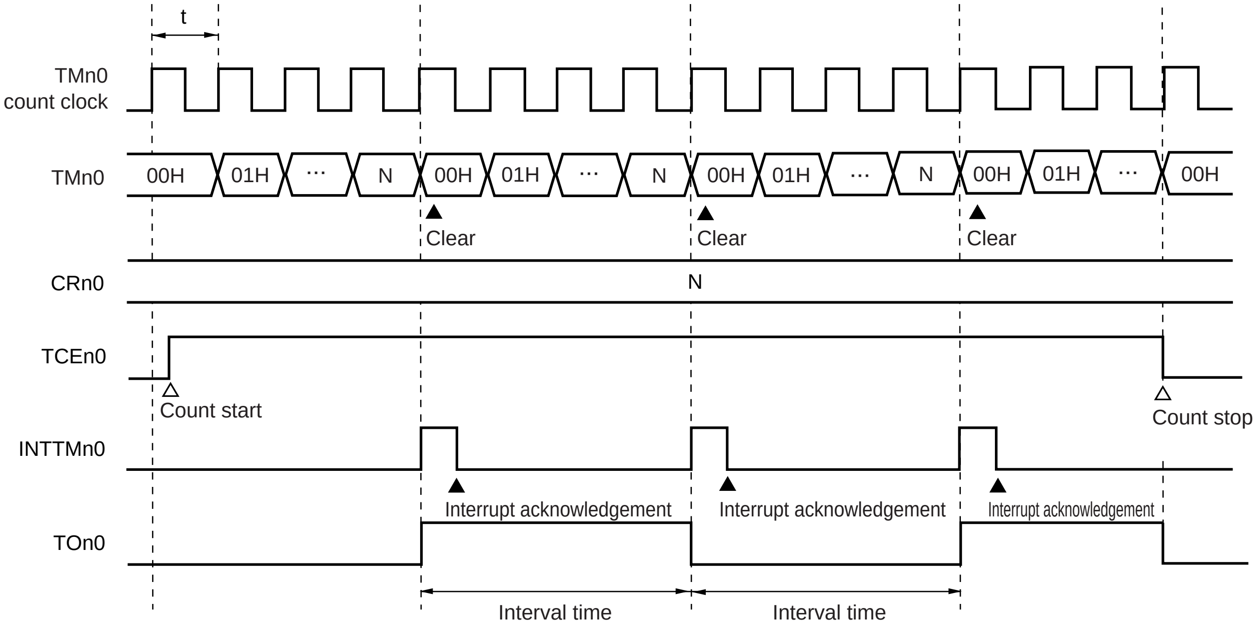
Remark f_X: Main system clock oscillation frequency (ceramic/crystal oscillation)

Table 10-6. Interval Time of Timer 40 (During f_{CC} = 4.0 MHz Operation)

TCL402	TCL401	TCL400	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	0	$2^3/f_{CC}$ (2.0 μs)	$2^{11}/f_{CC}$ (0.51 ms)	$2^3/f_{CC}$ (2.0 μs)
0	0	1	$2^7/f_{CC}$ (32 μs)	$2^{15}/f_{CC}$ (8.19 ms)	$2^7/f_{CC}$ (32 μs)
0	1	0	f_{TMI} input cycle	f_{TMI} input cycle × 2^8	f_{TMI} input cycle
0	1	1	f_{TM}I/2 input cycle	$f_{TMI}/2$ input cycle × 2^8	$f_{TMI}/2$ input cycle
1	0	0	$f_{TMI}/2^2$ input cycle	$f_{TMI}/2^2$ input cycle × 2^8	$f_{TMI}/2^2$ input cycle
1	0	1	$f_{TMI}/2^3$ input cycle	$f_{TMI}/2^3$ input cycle × 2^8	$f_{TMI}/2^3$ input cycle

Remark f_{CC}: Main system clock oscillation frequency (RC oscillation)

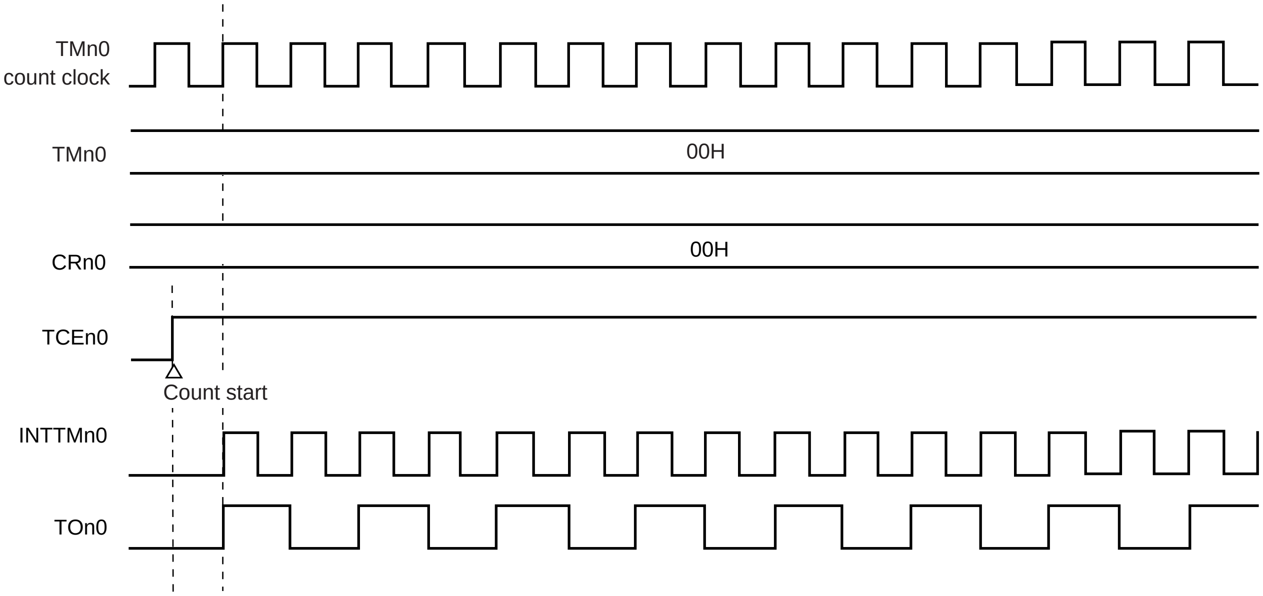
Figure 10-8. Timing of Interval Timer Operation with 8-Bit Resolution (Basic Operation)



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

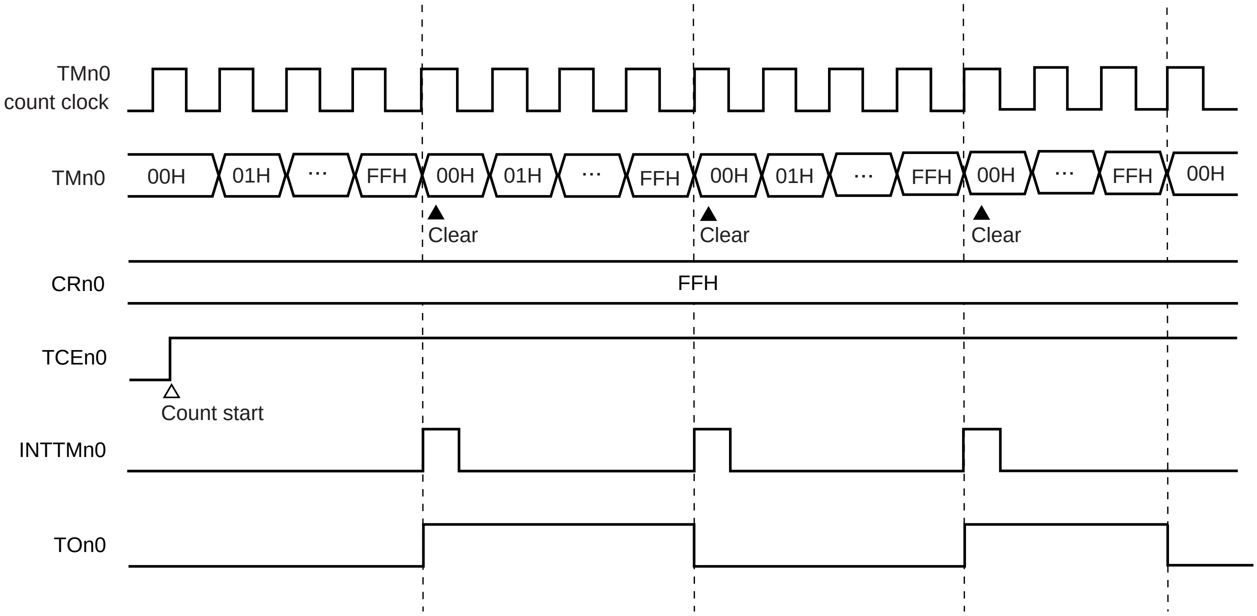
2. n = 3, 4

Figure 10-9. Timing of Interval Timer Operation with 8-Bit Resolution (When CRn0 Is Cleared to 00H)



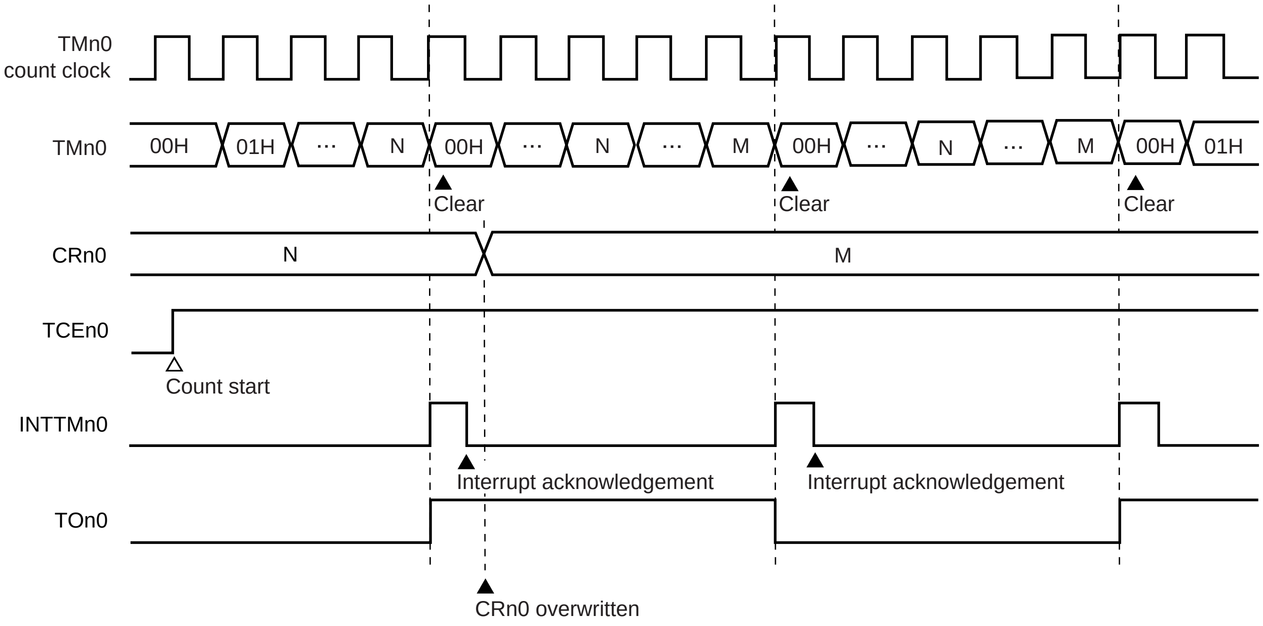
Remark n = 3, 4

Figure 10-10. Timing of Interval Timer Operation with 8-Bit Resolution (When CRn0 Is Set to FFH)



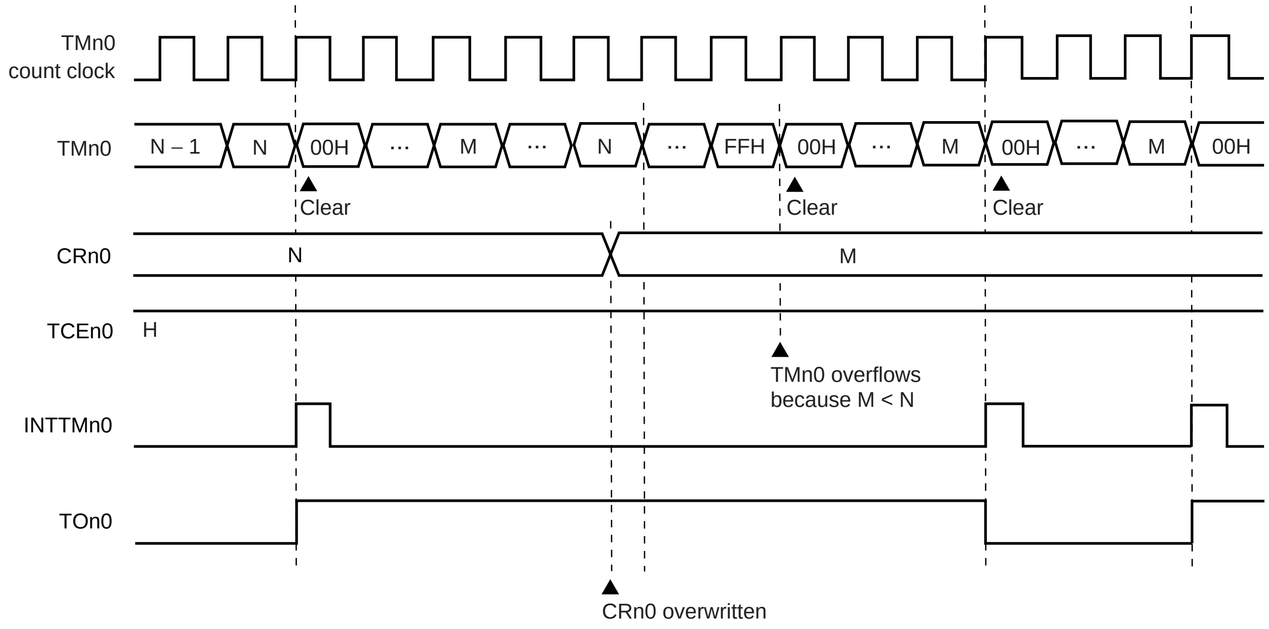
Remark n = 3, 4

Figure 10-11. Timing of Interval Timer Operation with 8-Bit Resolution (When CRn0 Changes from N to M (N < M))



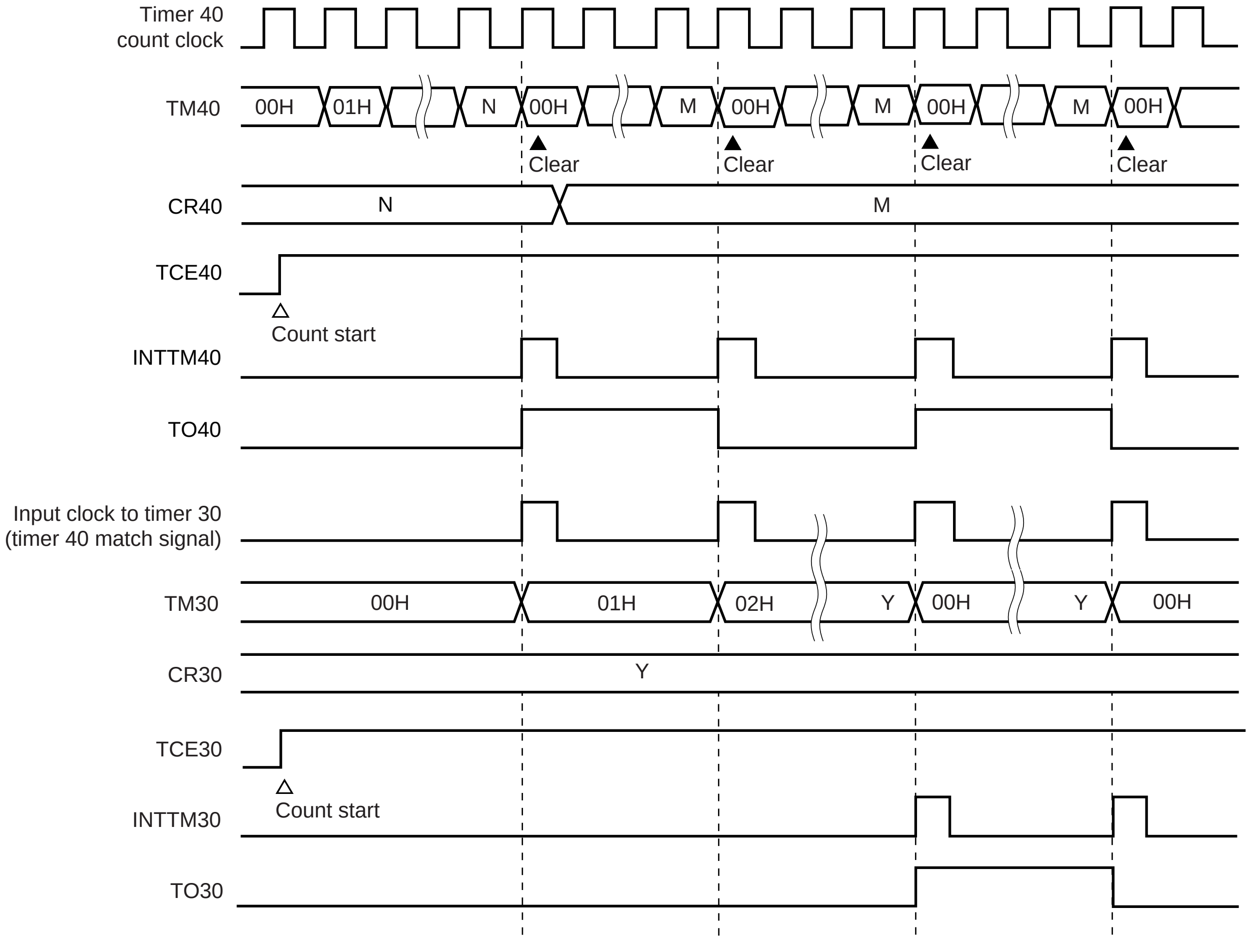
Remark n = 3, 4

Figure 10-12. Timing of Interval Timer Operation with 8-Bit Resolution (When CRn0 Changes from N to M (N > M))



Remark n = 3, 4

Figure 10-13. Timing of Interval Timer Operation with 8-Bit Resolution (When Timer 40 Match Signal Is Selected for Timer 30 Count Clock)



(2) Operation as external event counter with 8-bit resolution (timer 40 only)

The external event counter counts the number of external clock pulses input to the TMI40/P31/INTP1/TO30 pin by using 8-bit timer counter 40 (TM40).

To operate timer 40 as an external event counter, settings must be made in the following sequence.

<1> Disable operation of 8-bit timer counter 40 (TM40) (TCE40 = 0).
<2> Disable timer output of TO40 (TOE40 = 0).
<3> Set P31 to input mode (PM31 = 1).
<4> Select the external input clock for timer 40 (see Tables 10-5 and 10-6).
<5> Set the operation mode of timer 40 to 8-bit timer counter mode (see Figure 10-5).
<6> Set a count value in CR40.
<7> Enable the operation of TM40 (TCE40 = 1).

Note This operation only applies to timer 40.

Each time the valid edge is input, the value of TM40 is incremented.

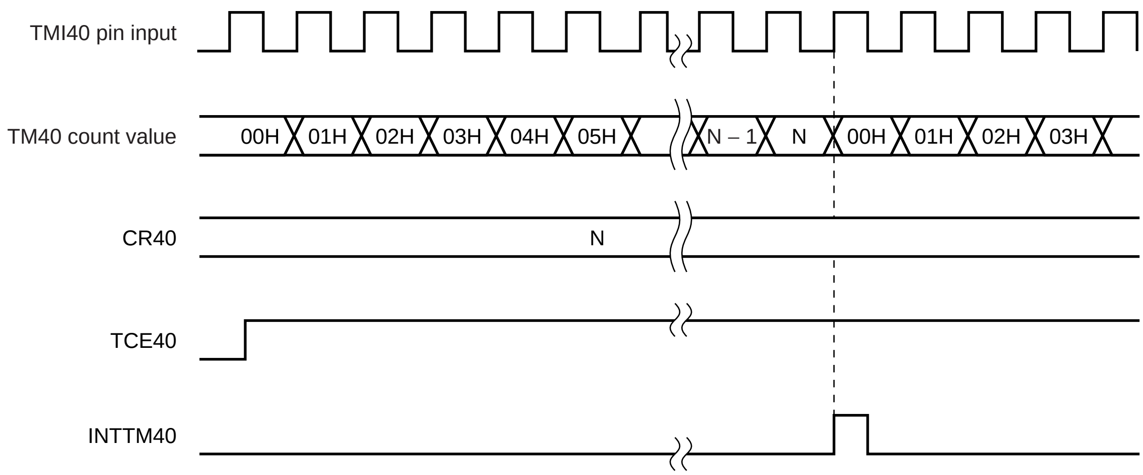
When the count value of TM40 matches the value set in CR40, TM40 is cleared to 00H and continues counting. At the same time, an interrupt request signal (INTTM40) is generated.

Figure 10-14 shows the timing of the external event counter operation.

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

Figure 10-14. Timing of Operation of External Event Counter with 8-Bit Resolution

★



Remark N = 00H to FFH

(3) Operation as square-wave output with 8-bit resolution

Square waves of any frequency can be output at an interval specified by the value preset in 8-bit compare register n0 (CRn0).

To operate timer n0 for square-wave output, settings must be made in the following sequence.

<1> When using timer 30, set P31 to output mode (PM31 = 0).
When using timer 40, set P32 to output mode (PM32 = 0).
<2> Set the output latches of P31 and P32 to 0.
<3> Disable operation of timer counter n0 (TMn0) (TCEn0 = 0).
<4> Set a count clock for timer n0 and enable output of TOn0 (TOEn0 = 1).
<5> Set a count value in CRn0.
<6> Enable the operation of TMn0 (TCEn0 = 1).

When the count value of TMn0 matches the value set in CRn0, the TOn0 pin output will be inverted. Through application of this mechanism, square waves of any frequency can be output. As soon as a match occurs, TMn0 is cleared to 00H and continues counting. At the same time, an interrupt request signal (INTTMn0) is generated.

The square-wave output is cleared to 0 by setting TCEn0 to 0.

Tables 10-7 to 10-10 show the square-wave output range, and Figure 10-15 shows the timing of square-wave output.

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

Remark n = 3, 4

Table 10-7. Square-Wave Output Range of Timer 30 (During f_X = 5.0 MHz Operation)

TCL301	TCL300	Minimum Pulse Width	Maximum Pulse Width	Resolution
0	0	$2^{4}/f_X$ (3.2 μs)	$2^{12}/f_X$ (0.82 ms)	$2^{4}/f_X$ (3.2 μs)
0	1	$2^{8}/f_X$ (51.2 μs)	$2^{16}/f_X$ (13.1 ms)	$2^{8}/f_X$ (51.2 μs)

Remark f_X: Main system clock oscillation frequency (ceramic/crystal oscillation)

Table 10-8. Square-Wave Output Range of Timer 30 (During f_{CC} = 4.0 MHz Operation)

TCL301	TCL300	Minimum Pulse Width	Maximum Pulse Width	Resolution
0	0	$2^{4}/f_{CC}$ (4.0 μs)	$2^{12}/f_{CC}$ (1.02 ms)	$2^{4}/f_{CC}$ (4.0 μs)
0	1	$2^{8}/f_{CC}$ (64 μs)	$2^{16}/f_{CC}$ (16.4 ms)	$2^{8}/f_{CC}$ (64 μs)

Remark f_{CC}: Main system clock oscillation frequency (RC oscillation)

Table 10-9. Square-Wave Output Range of Timer 40 (During f_X = 5.0 MHz Operation)

TCL402	TCL401	TCL400	Minimum Pulse Width	Maximum Pulse Width	Resolution
0	0	0	$2^{3}/f_X$ (1.6 μs)	$2^{11}/f_X$ (0.41 ms)	$2^{3}/f_X$ (1.6 μs)
0	0	1	$2^{7}/f_X$ (25.6 μs)	$2^{15}/f_X$ (6.55 ms)	$2^{7}/f_X$ (25.6 μs)

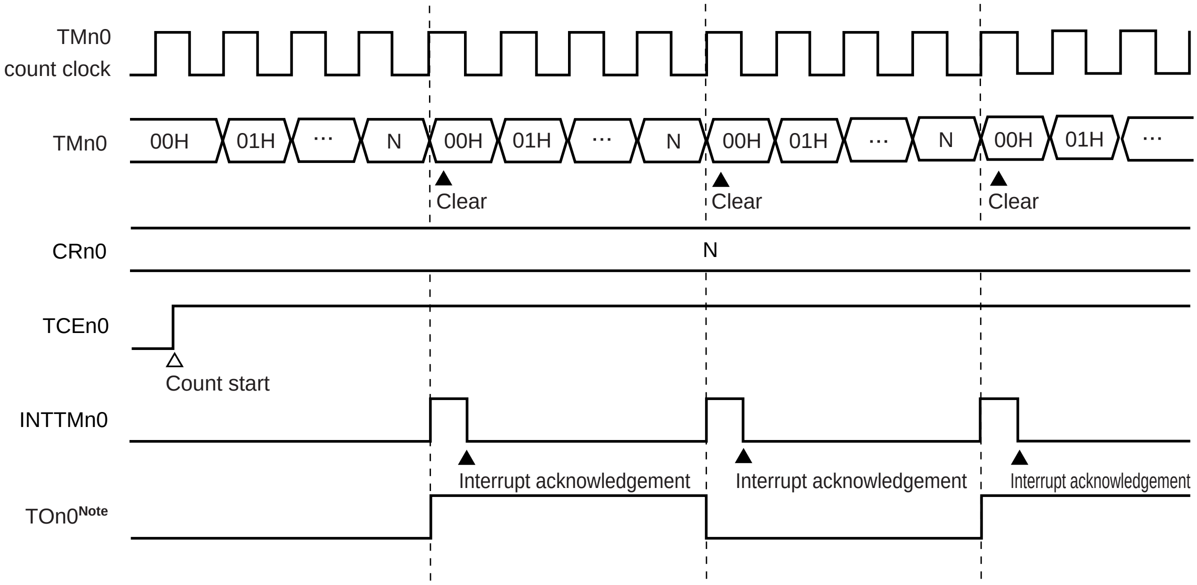
Remark f_X: Main system clock oscillation frequency (ceramic/crystal oscillation)

Table 10-10. Square-Wave Output Range of Timer 40 (During fCC = 4.0 MHz Operation)

TCL402	TCL401	TCL400	Minimum Pulse Width	Maximum Pulse Width	Resolution
0	0	0	2^3/fCC (2.0 μs)	2^{11}/fCC (0.51 ms)	2^3/fCC (2.0 μs)
0	0	1	2^7/fCC (32 μs)	2^{15}/fCC (8.19 ms)	2^7/fCC (32 μs)

Remark fCC: Main system clock oscillation frequency (RC oscillation)

Figure 10-15. Timing of Square-Wave Output with 8-Bit Resolution



Note The initial value of TOn0 is low level when output is enabled (TOEn0 = 1).

Remark n = 3, 4

10.4.2 Operation as 16-bit timer counter

Timer 30 and timer 40 can be used as a 16-bit timer counter using cascade connection. In this case, 8-bit timer counter 30 (TM30) is the higher 8 bits and 8-bit timer counter 40 (TM40) is the lower 8 bits. 8-bit timer 40 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 30 (CR30) and 8-bit compare register 40 (CR40).

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 30 (TM30) and 8-bit timer counter 40 (TM40) (TCE30 = 0, TCE40 = 0).

<2> Disable timer output of TO40 (TOE40 = 0).

<3> Set the count clock for timer 40 (see Tables 10-5 and 10-6).

<4> Set the operation mode of timer 30 and 8-bit timer 40 to 16-bit timer counter mode (see Figures 10-4 and 10-5).

<5> Set a count value in CR30 and CR40.

<6> Enable the operation of TM30 and TM40 (TCE40 = 1[Note]).

Note Start and clear of the timer in the 16-bit timer counter mode are controlled by TCE40 (the value of TCE30 is invalid).

When the count values of TM30 and TM40 match the values set in CR30 and CR40 respectively, both TM30 and TM40 are simultaneously cleared to 00H and counting continues. At the same time, an interrupt request signal (INTTM40) is generated (INTTM30 is not generated).

Tables 10-11 and 10-12 show interval time, and Figure 10-16 shows the timing of the interval timer operation.

Cautions 1. Be sure to stop the timer operation before overwriting the count clock with different data.

2. In the 16-bit timer counter mode, TO30 cannot be used. Be sure to set TOE30 = 0 to disable TO30 output.

Table 10-11. Interval Time with 16-Bit Resolution (During fX = 5.0 MHz Operation)

TCL402	TCL401	TCL400	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	0	$2^3/f_X$ (1.6 μs)	$2^{19}/f_X$ (0.10 s)	$2^3/f_X$ (1.6 μs)
0	0	1	$2^7/f_X$ (25.6 μs)	$2^{23}/f_X$ (1.68 s)	$2^7/f_X$ (25.6 μs)
0	1	0	f_{TMI} input cycle	f_{TMI} input cycle × 2^{16}	f_{TMI} input cycle
0	1	1	$f_{TMI}/2$ input cycle	$f_{TMI}/2$ input cycle × 2^{16}	$f_{TMI}/2$ input cycle
1	0	0	$f_{TMI}/2^2$ input cycle	$f_{TMI}/2^2$ input cycle × 2^{16}	$f_{TMI}/2^2$ input cycle
1	0	1	$f_{TMI}/2^3$ input cycle	$f_{TMI}/2^3$ input cycle × 2^{16}	$f_{TMI}/2^3$ input cycle

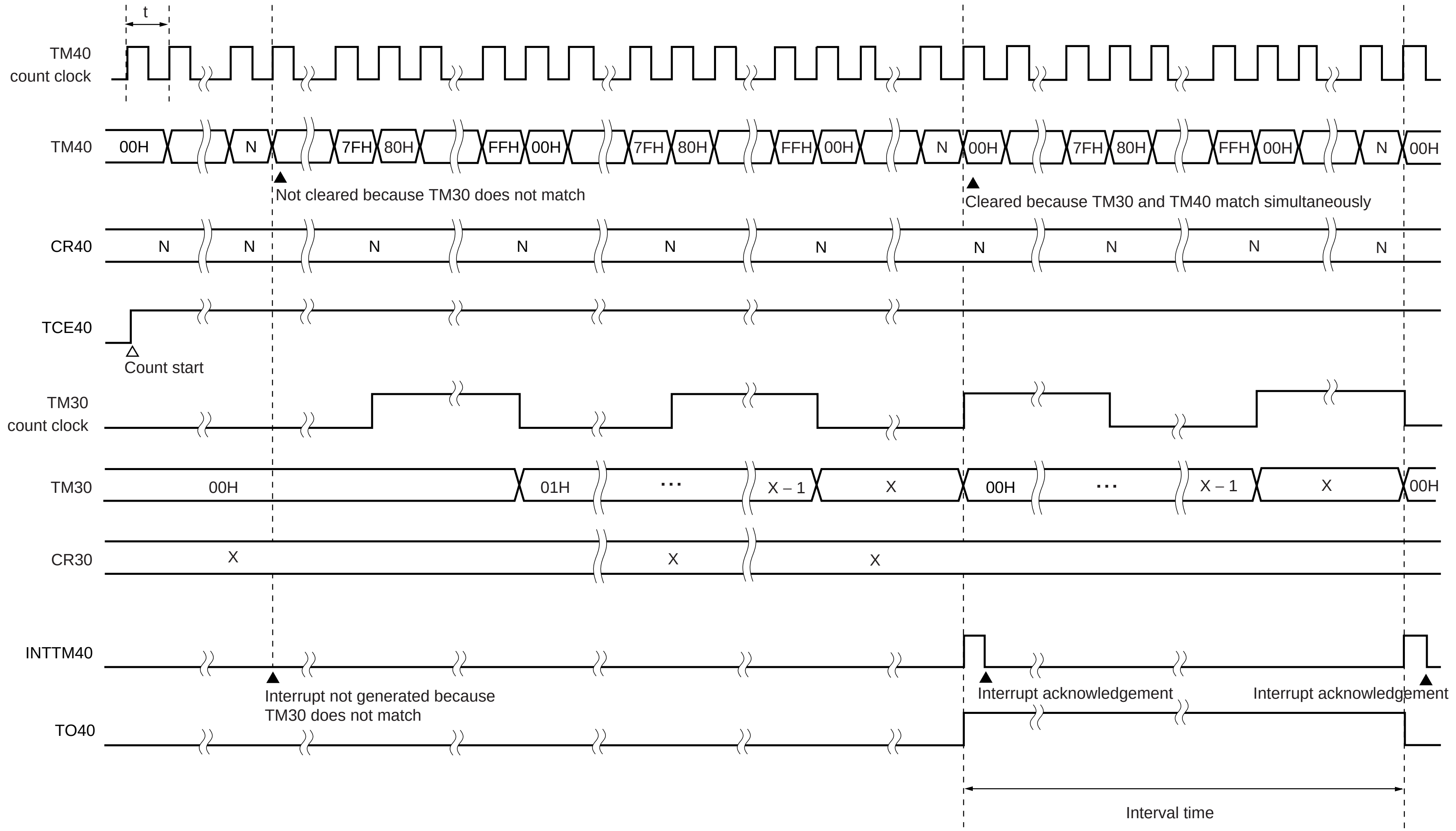
Remark f_X: Main system clock oscillation frequency (ceramic/crystal oscillation)

Table 10-12. Interval Time with 16-Bit Resolution (During fCC = 4.0 MHz Operation)

TCL402	TCL401	TCL400	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	0	$2^3/f_{CC}$ (2.0 μs)	$2^{19}/f_{CC}$ (0.13 s)	$2^3/f_{CC}$ (2.0 μs)
0	0	1	$2^7/f_{CC}$ (32 μs)	$2^{23}/f_{CC}$ (2.10 s)	$2^7/f_{CC}$ (32 μs)
0	1	0	f_{TMI} input cycle	f_{TMI} input cycle × 2^{16}	f_{TMI} input cycle
0	1	1	$f_{TMI}/2$ input cycle	$f_{TMI}/2$ input cycle × 2^{16}	$f_{TMI}/2$ input cycle
1	0	0	$f_{TMI}/2^2$ input cycle	$f_{TMI}/2^2$ input cycle × 2^{16}	$f_{TMI}/2^2$ input cycle
1	0	1	$f_{TMI}/2^3$ input cycle	$f_{TMI}/2^3$ input cycle × 2^{16}	$f_{TMI}/2^3$ input cycle

Remark f_{CC}: Main system clock oscillation frequency (RC oscillation)

Figure 10-16. Timing of Interval Timer Operation with 16-Bit Resolution


t
TM40
count clock
TM40
00H
N
7FH
80H
FFH
00H
Not cleared because TM30 does not match
Cleared because TM30 and TM40 match simultaneously
CR40
N
TCE40
Count start
TM30
count clock
TM30
00H
01H
···
X – 1
X
CR30
X
INTTM40
Interrupt not generated because
TM30 does not match
Interrupt acknowledgement
Interrupt acknowledgement
TO40
Interval time

Remark Interval time = $(256X + N + 1) \times t$: 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 TMI40/P31/INTP1/TO30 pin by TM30 and TM40.

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

<1> Disable operation of TM30 and TM40 (TCE30 = 0, TCE40 = 0).
<2> Disable timer output of TO40 (TOE40 = 0).
<3> Set P31 to input mode (PM31 = 1).
<4> Select the external input clock for timer 40 (see Tables 10-5 and 10-6).
<5> Set the operation mode of timer 30 and 8-bit timer 40 to 16-bit timer counter mode (see Figures 10-4 and 10-5).
<6> Set a count value in CR30 and CR40.
<7> Enable the operation of TM30 and TM40 (TCE40 = 1**Note**).

Note Start and clear of the timer in the 16-bit timer counter mode are controlled by TCE40 (the value of TCE30 is invalid).

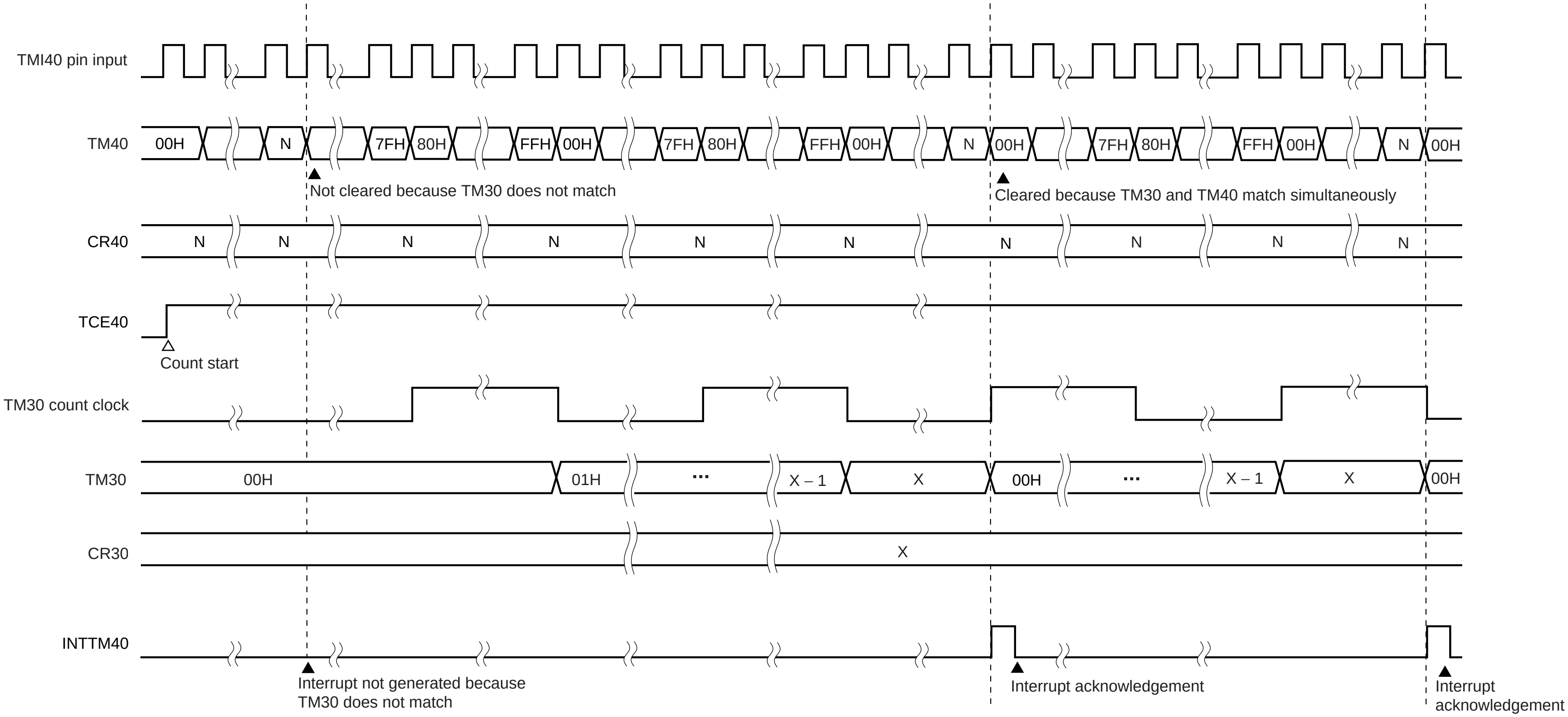
Each time the valid edge is input, the values of TM30 and TM40 are incremented.

When the count values of TM30 and TM40 simultaneously match the values set in CR30 and CR40 respectively, both TM30 and TM40 are cleared to 00H and counting continues. At the same time, an interrupt request signal (INTTM40) is generated (INTTM30 is not generated).

Figure 10-17 shows the timing of the external event counter operation.

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

Figure 10-17. 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 CR30 and CR40.

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

<1> Disable operation of TM30 and TM40 (TCE30 = 0, TCE40 = 0).
<2> Disable output of TO30 and TO40 (TOE30 = 0, TOE40 = 0).
<3> Set a count clock for timer 40.
<4> Set P32 to output mode (PM32 = 0) and P32 output latch to 0 and enable TO40 output (TOE40 = 1) (TO30 cannot be used).
<5> Set count values in CR30 and CR40.
<6> Enable the operation of TM40 (TCE40 = 1[Note]).

Note Start and clear of the timer in the 16-bit timer counter mode are controlled by TCE40 (the value of TCE30 is invalid).

When the count values of TM30 and TM40 simultaneously match the values set in CR30 and CR40 respectively, the TO40 pin output will be inverted. Through application of this mechanism, square waves of any frequency can be output. As soon as a match occurs, TM30 and TM40 are cleared to 00H and counting continues. At the same time, an interrupt request signal (INTTM40) is generated (INTTM30 is not generated).

The square-wave output is cleared to 0 by setting TCE40 to 0.

Tables 10-13 and 10-14 show the square wave output range, and Figure 10-18 shows timing of square wave output.

Cautions 1. Be sure to stop the timer operation before overwriting the count clock with different data.

2. In the 16-bit timer counter mode, TO30 cannot be used. Be sure to set TOE30 = 0 to disable TO30 output.

Table 10-13. Square-Wave Output Range with 16-Bit Resolution (During fx = 5.0 MHz Operation)

TCL402	TCL401	TCL400	Minimum Pulse Width	Maximum Pulse Width	Resolution
0	0	0	2^3/fx (1.6 μs)	2^{19}/fx (0.10 s)	2^3/fx (1.6 μs)
0	0	1	2^7/fx (25.6 μs)	2^{23}/fx (1.68 s)	2^7/fx (25.6 μs)

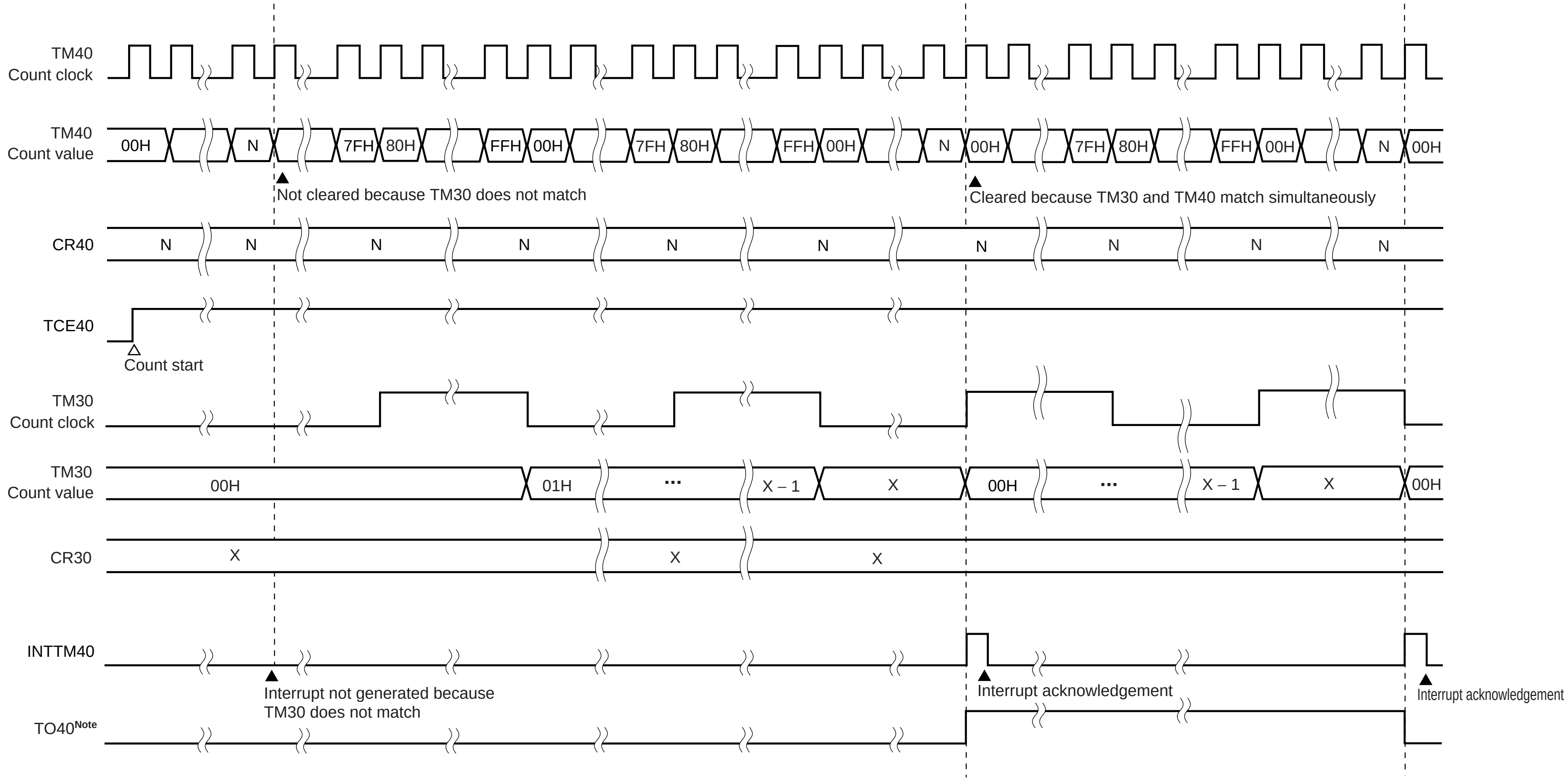
Remark fx: Main system clock oscillation frequency (ceramic/crystal oscillation)

Table 10-14. Square-Wave Output Range with 16-Bit Resolution (During fcc = 4.0 MHz Operation)

TCL402	TCL401	TCL400	Minimum Pulse Width	Maximum Pulse Width	Resolution
0	0	0	2^3/fcc (2.0 μs)	2^{19}/fcc (0.13 s)	2^3/fcc (2.0 μs)
0	0	1	2^7/fcc (32 μs)	2^{23}/fcc (2.10 s)	2^7/fcc (32 μs)

Remark fcc: Main system clock oscillation frequency (RC oscillation)

Figure 10-18. Timing of Square-Wave Output with 16-Bit Resolution


TM40
Count clock
TM40
Count value
00H
N
7FH
80H
FFH
00H
N
00H
Not cleared because TM30 does not match
Cleared because TM30 and TM40 match simultaneously
CR40
N
TCE40
Count start
TM30
Count clock
TM30
Count value
00H
01H
...
X – 1
X
CR30
X
INTTM40
Interrupt not generated because
TM30 does not match
Interrupt acknowledgement
Interrupt acknowledgement
TO40Note

Note The initial value of TO40 is low level when output is enabled (TOE40 = 1).

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

10.4.3 Operation as carrier generator

An arbitrary carrier clock generated by TM40 can be output in the cycle set in TM30.
To operate timer 30 and timer 40 as carrier generators, settings must be made in the following sequence.

<1> Disable operation of TM30 and TM40 (TCE30 = 0, TCE40 = 0).
<2> Disable timer output of TO30 and TO40 (TOE30 = 0, TOE40 = 0).
<3> Set count values in CR30, CR40, and CRH40.
<4> Set the operation mode of timer 30 and timer 40 to carrier generator mode (see Figures 10-4 and 10-5).
<5> Set the count clock for timer 30 and timer 40.
<6> Set remote control output to carrier pulse (RMC40 (bit 2 of carrier generator output control register 40 (TCA40)) = 0).
Input the required value to NRZB40 (bit 1 of TCA40) by program.
Input a value to NRZ40 (bit 0 of TCA40) before it is reloaded from NRZB40.
<7> Set P32 to output mode (PM32 = 0) and the P32 output latch to 0 and enable TO40 output by setting TOE40 to 1.
<8> Enable the operation of TM30 and TM40 (TCE30 = 1, TCE40 = 1).
<9> Save the value of NRZB40 to a general-purpose register.
<10> When INTTM30 rises, the value of NRZB40 is transferred to NRZ40. After that, rewrite TCA40 with an 8-bit memory manipulation instruction. Input the value to be transferred to NRZ40 next time to NRZB40, and input the value saved in <9> to NRZ40.
<11> Generate the desired carrier signal by repeating <9> and <10>.

The operation of the carrier generator is as follows.

<1> When the count value of TM40 matches the value set in CR40, an interrupt request signal (INTTM40) is generated and output of timer 40 is inverted, which makes the compare register switch from CR40 to CRH40.
<2> After that, when the count value of TM40 matches the value set in CRH40, an interrupt request signal (INTTM40) is generated and output of timer 40 is inverted again, which makes the compare register switch from CRH40 to CR40.
<3> The carrier clock is generated by repeating <1> and <2> above.
<4> When the count value of TM30 matches the value set in CR30, an interrupt request signal (INTTM30) is generated. The rising edge of INTTM30 is the data reload signal of NRZB40 and is transferred to NRZ40.
<5> When NRZ40 is 1, a carrier clock is output from TO40 pin.

Cautions 1. TCA40 cannot be set with a 1-bit memory manipulation instruction. Be sure to use an 8-bit memory manipulation instruction.

2. The NRZ40 flag can be rewritten only when the carrier generator output is stopped (TOE40 = 0). The data of the flag is not changed even if a write instruction is executed while TOE40 = 1.

3. When setting the carrier generator operation again after stopping it once, reset NRZB40 because the previous value is not retained. In this case also a 1-bit memory manipulation instruction cannot be used. Be sure to use an 8-bit memory manipulation instruction.

Cautions 4. To enable operation in the carrier generator mode, set a value to the compare registers (CR30, CR40, and CRH40), and input the necessary value to the NRZB40 and NRZ40 flags in advance. Otherwise, the signal of the timer match circuit will become unstable and the NRZ40 flag will be undefined.

5. While INTTM30 (interrupt generated by the match signal of timer 30) is being output, accessing TCA40 is prohibited.

Accessing TCA40 is prohibited while 8-bit timer counter 30 (TM30) is 00H.

To access TCA40 while TM30 = 00H, wait for more than half a period of the TM30 count clock and then rewrite TCA40.

Figures 10-19 to 10-21 show the operation timing of the carrier generator.

Figure 10-19. Timing of Carrier Generator Operation (When CR40 = N, CRH40 = M (M > N))

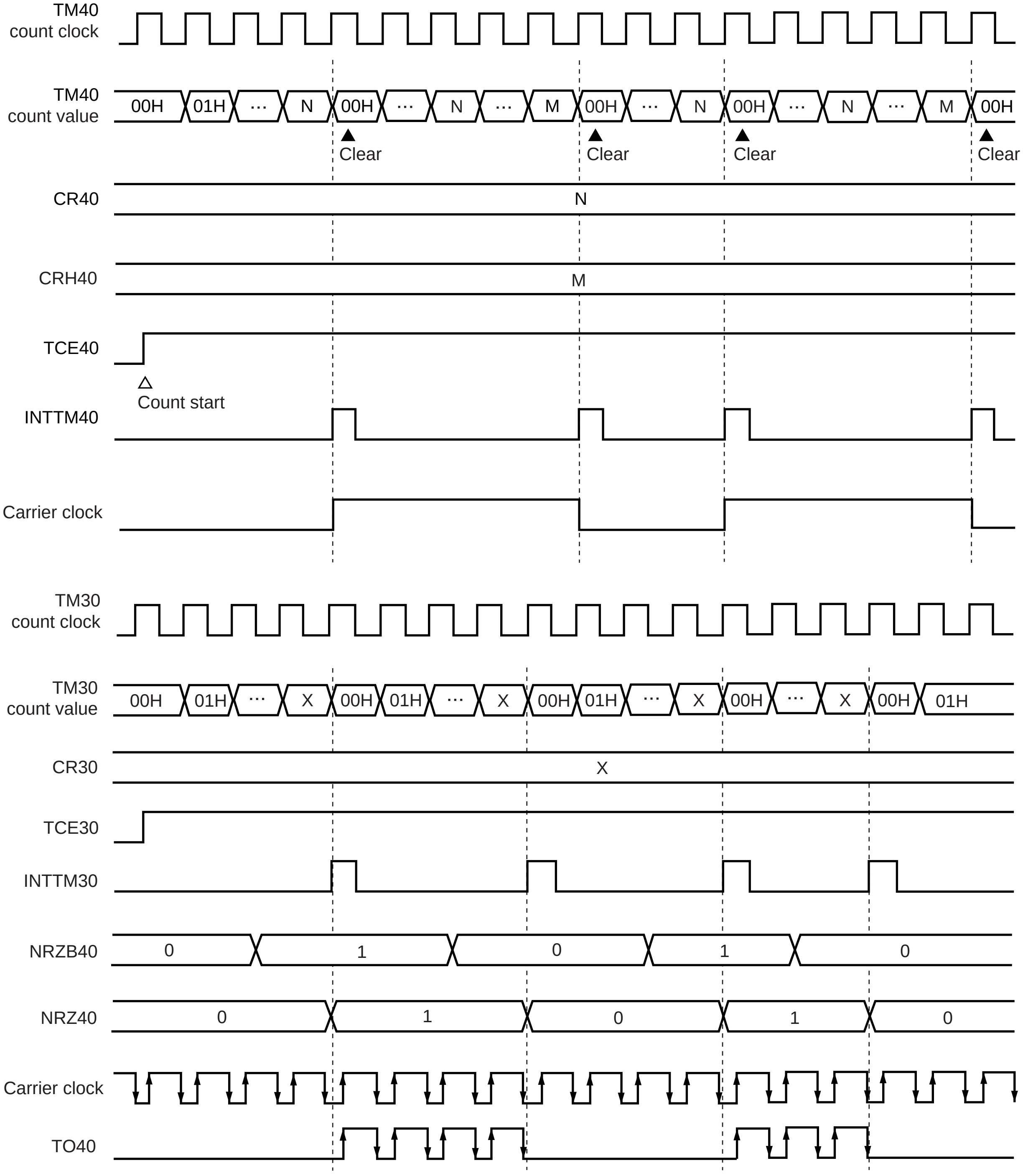
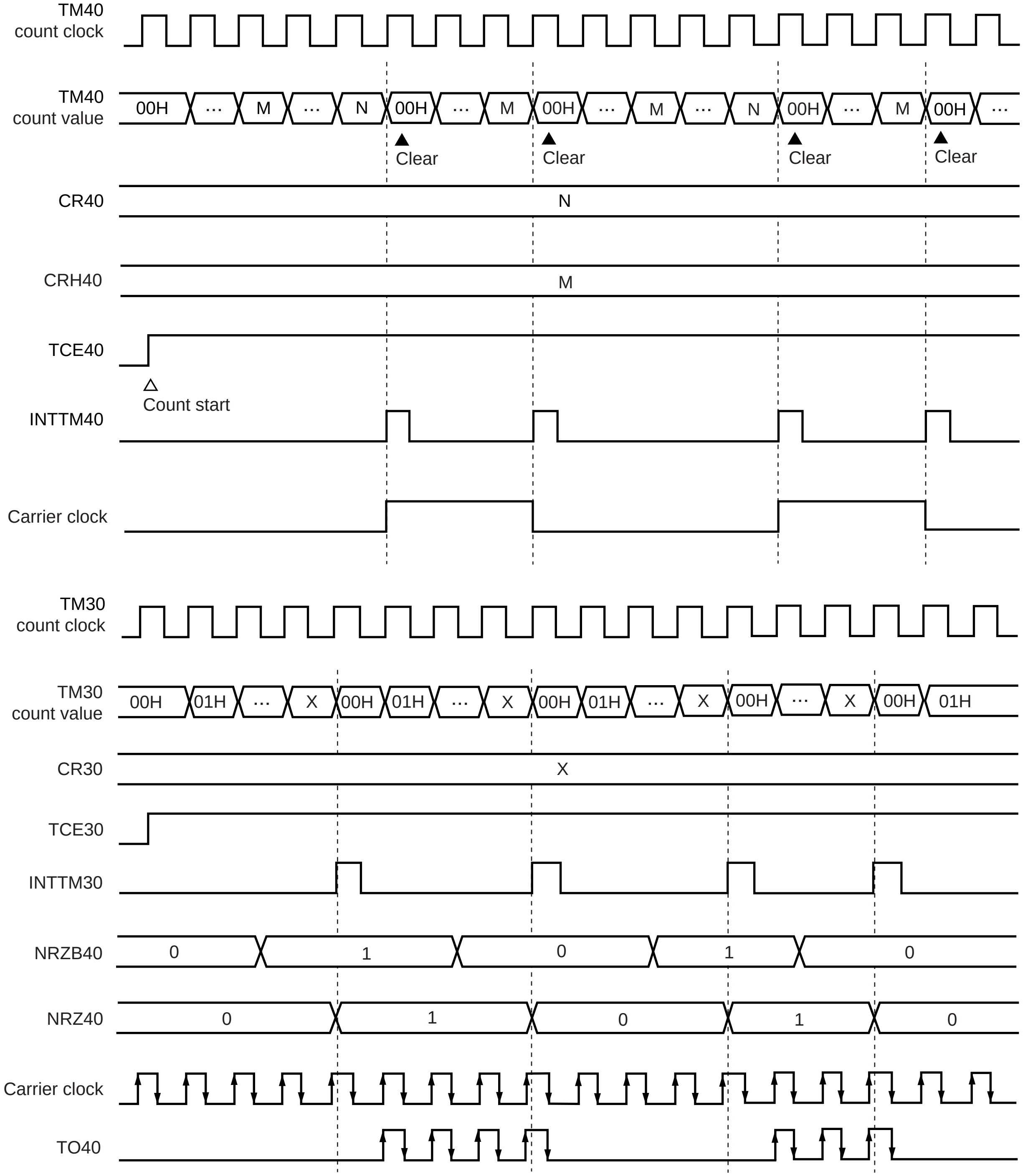
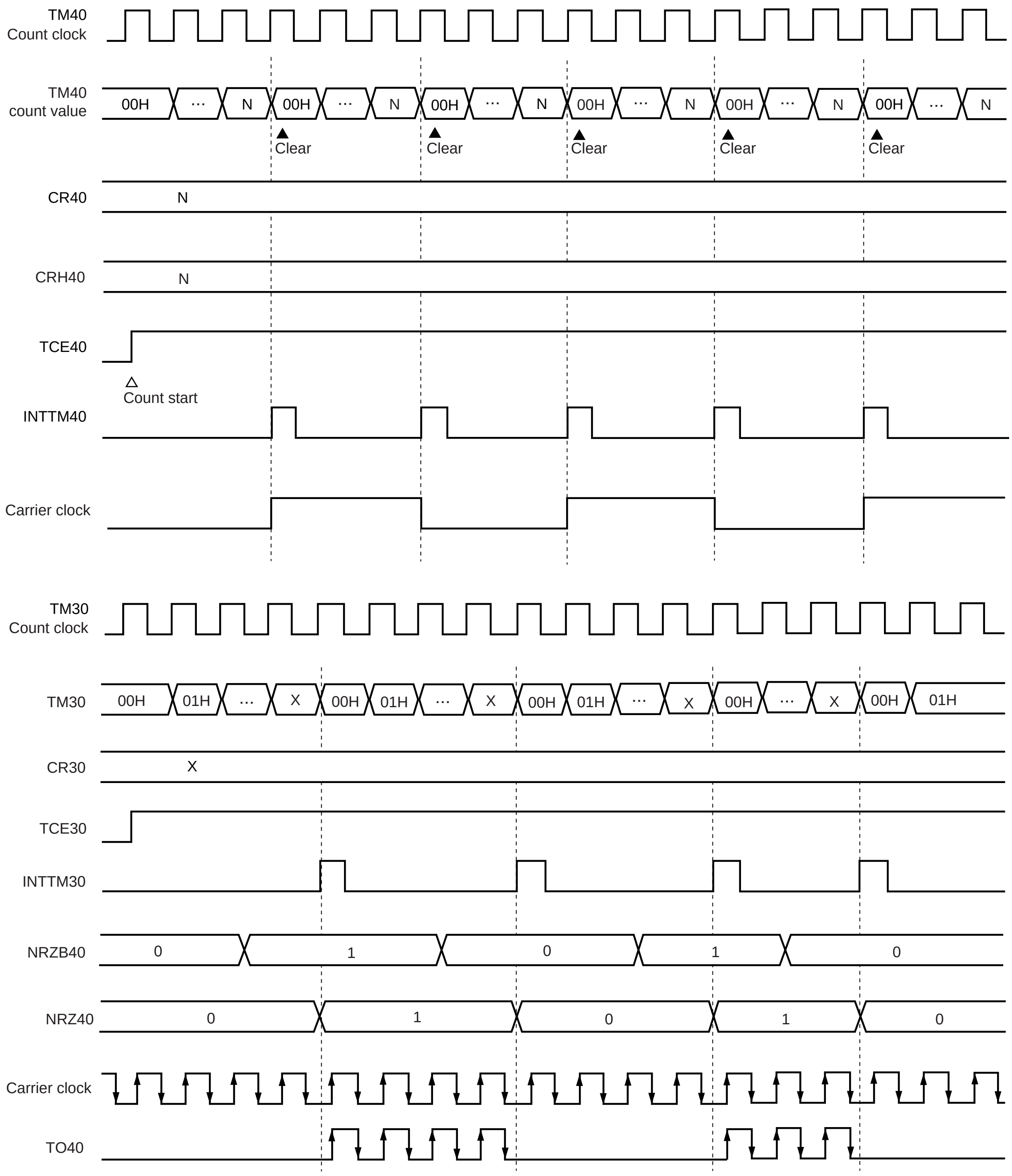


Figure 10-20. Timing of Carrier Generator Operation (When CR40 = N, CRH40 = M (M < N))



Remark This timing chart shows an example in which the value of NRZ40 is changed while the carrier clock is high.

Figure 10-21. Timing of Carrier Generator Operation (When CR40 = CRH40 = N)



10.4.4 Operation as PWM output (timer 40 only)

In the PWM output mode, a pulse of any duty ratio can be output by setting a low-level width using CR40 and a high-level width using CRH40.

To operate timer 40 in PWM output mode, settings must be made in the following sequence.

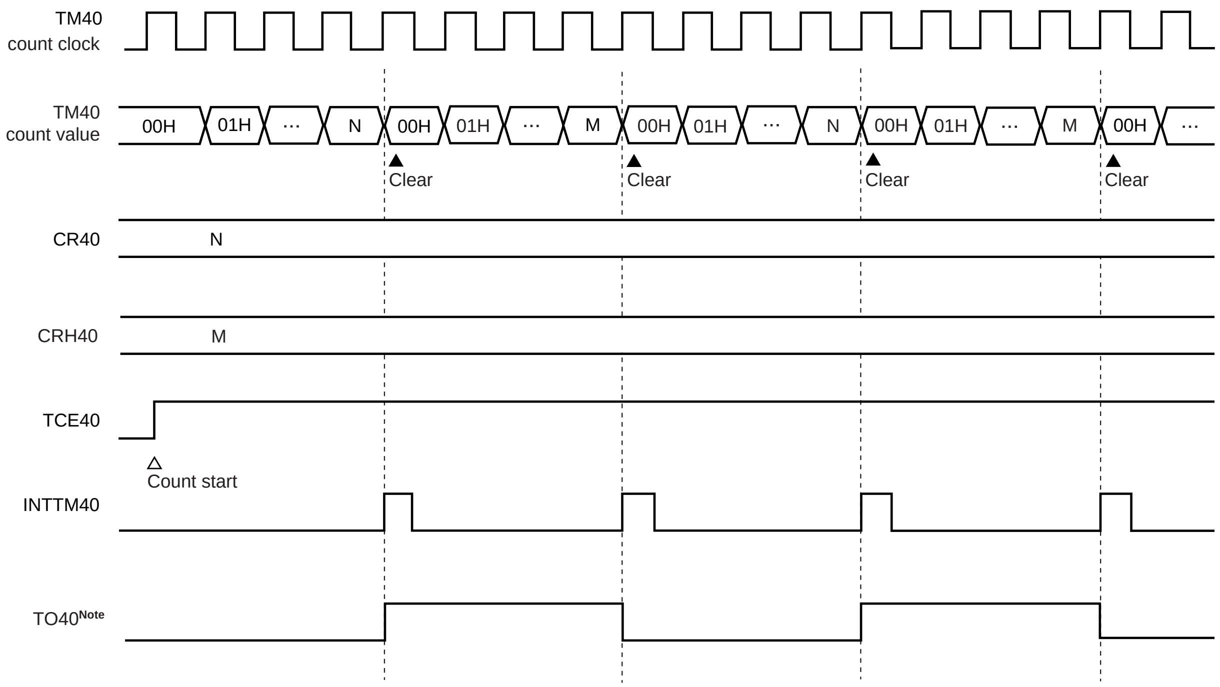
<1> Disable operation of TM40 (TCE40 = 0).
<2> Disable timer output of TO40 (TOE40 = 0).
<3> Set count values in CR40 and CRH40.
<4> Set the operation mode of timer 40 to carrier generator mode (see Figure 10-5).
<5> Set the count clock for timer 40.
<6> Set P32 to output mode (PM32 = 0) and the P32 output latch to 0 and enable timer output of TO40 (TOE40 = 1).
<7> Enable the operation of TM40 (TCE40 = 1).

The operation in the PWM output mode is as follows.

<1> When the count value of TM40 matches the value set in CR40, an interrupt request signal (INTTM40) is generated and output of timer 40 is inverted, which makes the compare register switch from CR40 to CRH40.
<2> A match between TM40 and CR40 clears the TM40 value to 00H and then counting starts again.
<3> After that, when the count value of TM40 matches the value set in CRH40, an interrupt request signal (INTTM40) is generated and output of timer 40 is inverted again, which makes the compare register switch from CRH40 to CR40.
<4> A match between TM40 and CRH40 clears the TM40 value to 00H and then counting starts again.

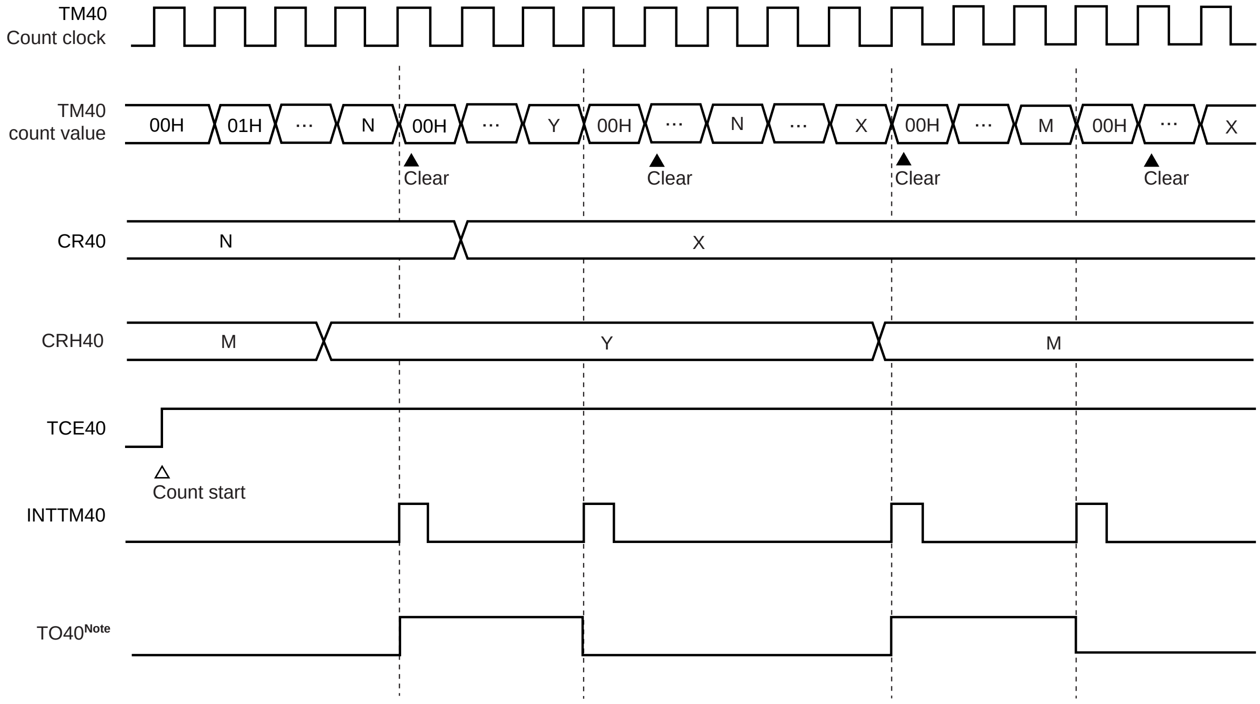
A pulse of any duty ratio is output by repeating <1> to <4> above. Figures 10-22 and 10-23 show the operation timing in the PWM output mode.

Figure 10-22. PWM Output Mode Timing (Basic Operation)



Note The initial value of TO40 is low level when output is enabled (TOE40 = 1).

Figure 10-23. PWM Output Mode Timing (When CR40 and CRH40 Are Overwritten)



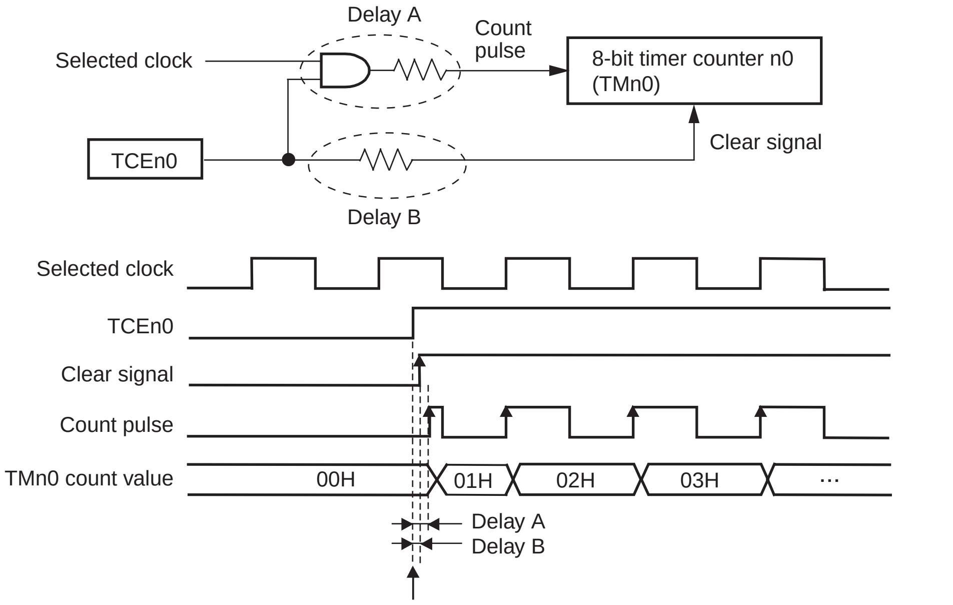
Note The initial value of TO40 is low level when output is enabled (TOE40 = 1).

10.5 Notes on Using 8-Bit Timer 30, 40

(1) Error on starting timer

An error of up to 1.5 clocks is included in the time between when the timer is started and a match signal is generated. This is because the counter may be incremented by detecting a rising edge at the timing at which the timer starts while the count clock is high level (see **Figure 10-24**).

Figure 10-24. Case in Which Error of 1.5 Clocks (Max.) Occurs



If delay A > delay B when the timer starts while the selected clock is high level, an error of 1.5 clocks (max.) occurs.

Remark n = 3, 4

(2) Count value if external clock input from TMI40 pin is selected

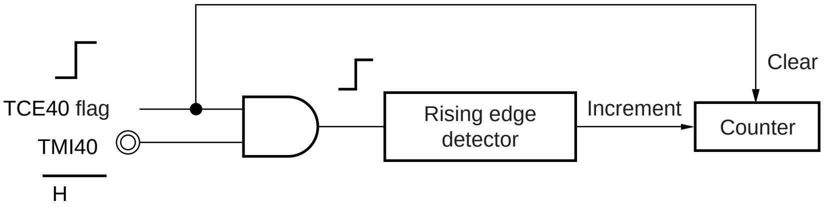
When the external clock signal input from the TMI40 pin is selected as the count clock, the count value may start from 01H if the timer is enabled (TCE40 = 0 → 1) while the TMI40 pin is high. This is because the input signal of the TMI40 pin is internally ANDed with the TCE40 signal. Consequently, the counter is incremented because the rising edge of the count clock is input to the timer immediately when the TCE40 pin is set. Depending on the delay timing, the count value is incremented by one if the rising edge is input after the counter is cleared. Counting is not affected if the rising edge is input before the counter is cleared (the counter operates normally).

Use the timer being aware that it has an error of one count, or take either of the following actions A or B.

<Action A> Always start the timer when the TMI40 pin is low.

<Action B> Save the count value to a control register when the timer is started, subtract the count value from the count value saved to the control register when reading the count value, and take the result as the true count value.

Figure 10-25. Counting Operation if Timer Is Started When TMI40 Is High



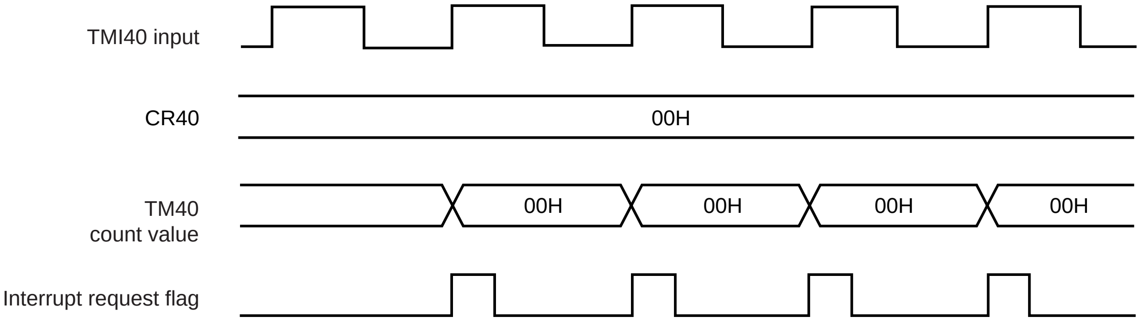
Remark n = 0, 1

(3) Setting of 8-bit compare register n0

8-bit compare register n0 (CRn0) can be cleared to 00H.

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

Figure 10-26. Timing of Operation as External Event Counter (8-Bit Resolution)



Remark n = 3, 4

CHAPTER 11 WATCH TIMER

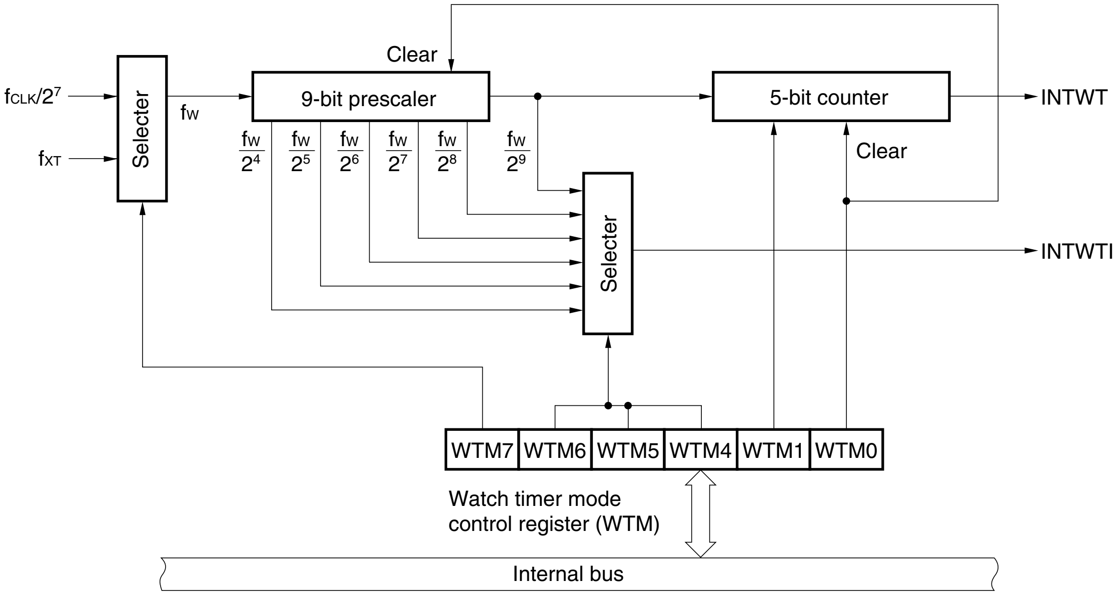
11.1 Watch Timer Functions

The watch timer has the following functions.

- Watch timer
- Interval timer

The watch and interval timers can be used at the same time.
Figure 11-1 is a block diagram of the watch timer.

Figure 11-1. Watch Timer Block Diagram



Remark fCLK: fX or fCC

(1) Watch timer

The 4.19 MHz main system clock or 32.768 kHz subsystem clock is used to generate an interrupt request (INTWT) at 0.5-second intervals.

Caution When the main system clock is operating at 5.0 MHz (ceramic/crystal oscillation) or 4.0 MHz (RC oscillation), it cannot be used to generate a 0.5-second interval. In this case, the subsystem clock, which operates at 32.768 kHz, should be used instead.

(2) Interval timer

The interval timer is used to generate an interrupt request (INTWT) at specified intervals.

Table 11-1. Interval Time of Interval Timer (Ceramic/Crystal Oscillation)

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

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

2. f_X: Main system clock oscillation frequency (ceramic/crystal oscillation)

3. f_{XT}: Subsystem clock oscillation frequency

Table 11-2. Interval Time of Interval Timer (RC Oscillation)

Interval	During f_{CC} = 4.0 MHz Operation	During f_{XT} = 32.768 kHz Operation
$2^4 \times 1/f_W$	512 μs	488 μs
$2^5 \times 1/f_W$	1.02 ms	977 μs
$2^6 \times 1/f_W$	2.05 ms	1.95 ms
$2^7 \times 1/f_W$	4.10 ms	3.91 ms
$2^8 \times 1/f_W$	8.19 ms	7.81 ms
$2^9 \times 1/f_W$	16.4 ms	15.6 ms

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

2. f_{CC}: Main system clock oscillation frequency (RC oscillation)

3. f_{XT}: Subsystem clock oscillation frequency

11.2 Watch Timer Configuration

The watch timer includes the following hardware.

Table 11-3. Watch Timer Configuration

Item	Configuration
Counter	5 bits × 1
Prescaler	9 bits × 1
Control register	Watch timer mode control register (WTM)

11.3 Register Controlling Watch Timer

The watch timer mode control register (WTM) is used to control the watch timer.

- Watch timer mode control register (WTM)
  WTM selects a count clock for the watch timer and specifies whether to enable clocking of the timer. It also specifies the prescaler interval and how the 5-bit counter is controlled.
  WTM is set with a 1-bit or 8-bit memory manipulation instruction.
  $\overline{\text{RESET}}$ input clears WTM to 00H.

Figure 11-2. 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	Watch timer count clock selection	
	During f_X = 5.0 MHz, f_{XT} = 32.768 kHz operation	During f_{CC} = 4.0 MHz, f_{XT} = 32.768 kHz operation
0	$f_X/2^7$ (39.1 kHz)	$f_{CC}/2^7$ (31.3 kHz)
1	f_{XT} (32.768 kHz)	

WTM6	WTM5	WTM4	Prescaler interval selection
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 stop
1	Started

WTM0	Watch timer operation
0	Operation stopped (both prescaler and timer cleared)
1	Operation enabled

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

2. f_X: Main system clock oscillation frequency (ceramic/crystal oscillation)

3. f_{CC}: Main system clock oscillation frequency (RC oscillation)

4. f_{XT}: Subsystem clock oscillation frequency

11.4 Watch Timer Operation

11.4.1 Operation as watch timer

The main system clock (4.19 MHz) or subsystem clock (32.768 kHz) is used as a watch timer which generates 0.5-second intervals.

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.

When the interval timer also operates at the same time, only the watch timer can be started from 0 seconds by setting WTM1 to 0. However, an error of up to $2^9 \times 1/f_W$ seconds may occur for the first overflow of the watch timer (INTWT) after a 0-second start because the 9-bit prescaler is not cleared in this case.

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

Table 11-4. Interval Time of Interval Timer (Ceramic/Crystal Oscillation)

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

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

2. f_X: Main system clock oscillation frequency (ceramic/crystal oscillation)

3. f_{XT}: Subsystem clock oscillation frequency

Table 11-5. Interval Time of Interval Timer (RC Oscillation)

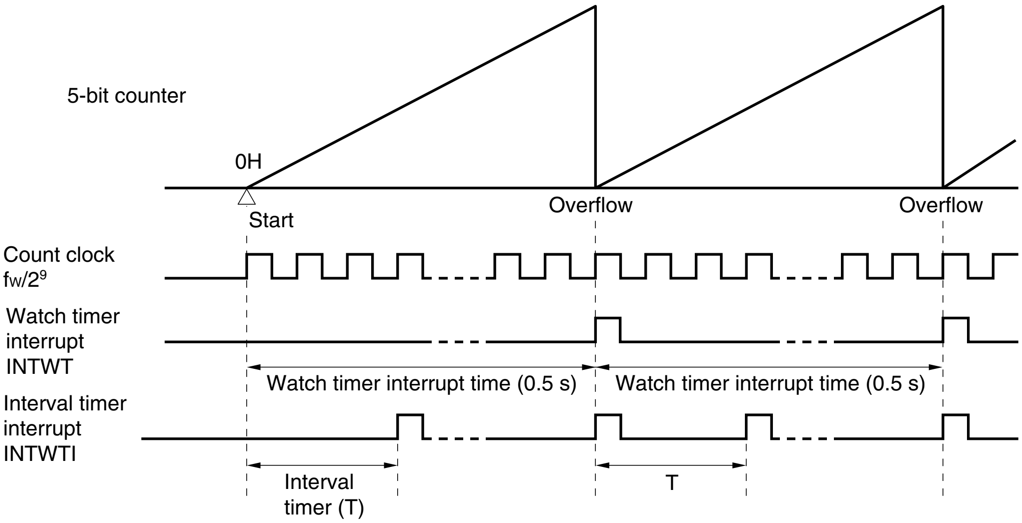
Interval	During f_{CC} = 4.0 MHz Operation	During f_{XT} = 32.768 kHz Operation
$2^4 \times 1/f_W$	512 μs	488 μs
$2^5 \times 1/f_W$	1.02 ms	977 μs
$2^6 \times 1/f_W$	2.05 ms	1.95 ms
$2^7 \times 1/f_W$	4.10 ms	3.91 ms
$2^8 \times 1/f_W$	8.19 ms	7.81 ms
$2^9 \times 1/f_W$	16.4 ms	15.6 ms

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

2. f_{CC}: Main system clock oscillation frequency (RC oscillation)

3. f_{XT}: Subsystem clock oscillation frequency

Figure 11-3. Watch Timer/Interval Timer Operation Timing



Caution **When operation of the watch timer and 5-bit counter has been enabled by setting the watch timer mode control register (WTM) (setting WTM0 (bit 0 of WTM) to 1), the time until the first interrupt request after this setting will not be exactly the same as the time set by watch timer interrupt time (0.5 s). This is because the 5-bit counter starts counting one cycle after the output of the 9-bit prescaler. The INTWT signal will be generated at the set time from its second generation.**

Remarks **1.** fw: Watch timer clock frequency

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

CHAPTER 12 WATCHDOG TIMER

12.1 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 program runaway. When the runaway is detected, a non-maskable interrupt or the $\overline{\text{RESET}}$ signal can be generated.

Table 12-1. Runaway Detection Time of Watchdog Timer

Runaway Detection Time	During fx = 5.0 MHz Operation	During fCC = 4.0 MHz Operation
$2^{11} \times 1/f_{CLK}$	$2^{11}/f_X$ (410 μs)	$2^{11}/f_{CC}$ (512 μs)
$2^{13} \times 1/f_{CLK}$	$2^{13}/f_X$ (1.64 ms)	$2^{13}/f_{CC}$ (2.05 ms)
$2^{15} \times 1/f_{CLK}$	$2^{15}/f_X$ (6.55 ms)	$2^{15}/f_{CC}$ (8.19 ms)
$2^{17} \times 1/f_{CLK}$	$2^{17}/f_X$ (26.2 ms)	$2^{17}/f_{CC}$ (32.8 ms)

Remarks 1. f_{CLK}: f_X or f_{CC}

2. f_X: Main system clock oscillation frequency (ceramic/crystal oscillation)

3. f_{CC}: Main system clock oscillation frequency (RC oscillation)

(2) Interval timer

The interval timer generates an interrupt at any preset intervals.

Table 12-2. Interval Time of Watchdog Timer

Interval Time	During fx = 5.0 MHz Operation	During fCC = 4.0 MHz Operation
$2^{11} \times 1/f_{CLK}$	$2^{11}/f_X$ (410 μs)	$2^{11}/f_{CC}$ (512 μs)
$2^{13} \times 1/f_{CLK}$	$2^{13}/f_X$ (1.64 ms)	$2^{13}/f_{CC}$ (2.05 ms)
$2^{15} \times 1/f_{CLK}$	$2^{15}/f_X$ (6.55 ms)	$2^{15}/f_{CC}$ (8.19 ms)
$2^{17} \times 1/f_{CLK}$	$2^{17}/f_X$ (26.2 ms)	$2^{17}/f_{CC}$ (32.8 ms)

Remarks 1. f_{CLK}: f_X or f_{CC}

2. f_X: Main system clock oscillation frequency (ceramic/crystal oscillation)

3. f_{CC}: Main system clock oscillation frequency (RC oscillation)

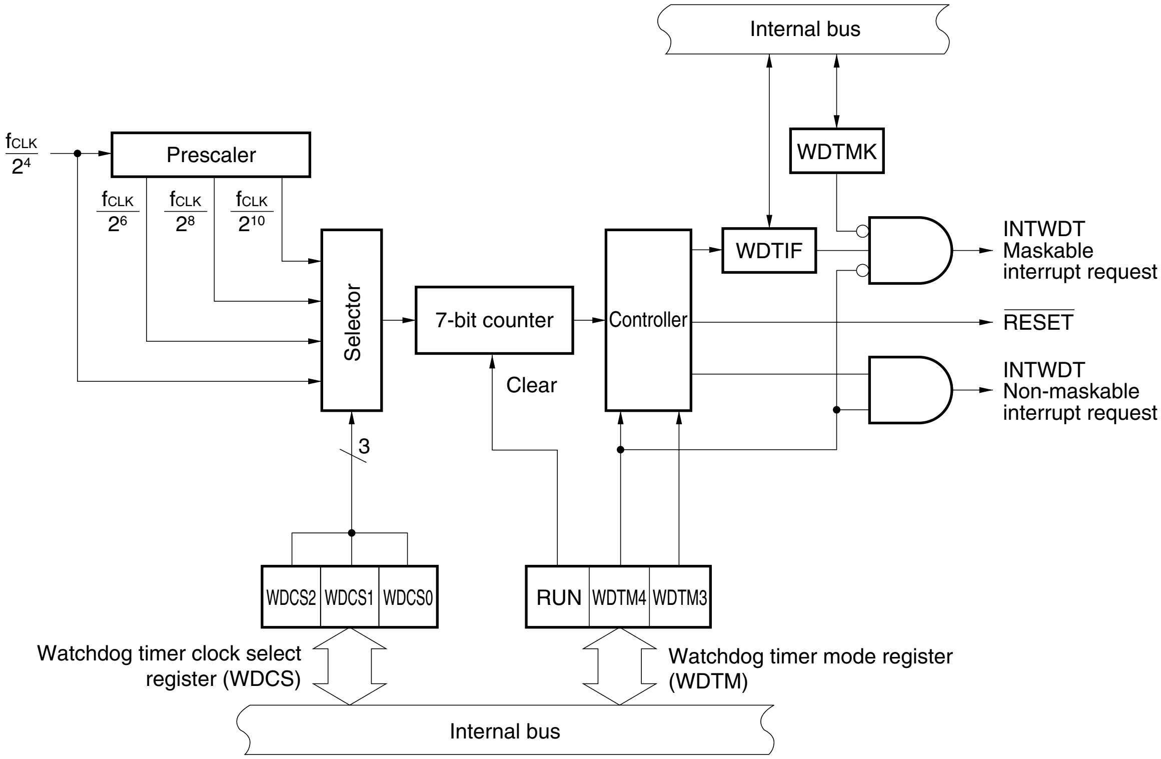
12.2 Watchdog Timer Configuration

The watchdog timer includes the following hardware.

Table 12-3. Watchdog Timer Configuration

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

Figure 12-1. Watchdog Timer Block Diagram



Remark f_{CLK}: f_X or f_{CC}

12.3 Registers Controlling Watchdog Timer

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

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

(1) Watchdog timer clock select register (WDCS)

This register sets the watchdog timer count clock.

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

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

Figure 12-2. Format of Watchdog Timer Clock Select Register

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

WDCS2	WDCS1	Count clock selection	
		During f_X = 5.0 MHz operation	During f_{CC} = 4.0 MHz operation
0	0	$f_X/2^4$ (313 kHz)	$f_{CC}/2^4$ (250 kHz)
0	1	$f_X/2^6$ (78.1 kHz)	$f_{CC}/2^6$ (62.5 kHz)
1	0	$f_X/2^8$ (19.5 kHz)	$f_{CC}/2^8$ (15.6 kHz)
1	1	$f_X/2^{10}$ (4.88 kHz)	$f_{CC}/2^{10}$ (3.91 kHz)
Other than above		Setting prohibited	

Remarks 1. f_X: Main system clock oscillation frequency (ceramic/crystal oscillation)

2. f_{CC}: Main system clock oscillation frequency (RC oscillation)

(2) Watchdog timer mode register (WDTM)

This register sets an 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.

RESET input clears WDTM to 00H.

Figure 12-3. 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	Selection of operation of watchdog timer[Note 1]
0	Stops counting
1	Clears counter and starts counting

WDTM4	WDTM3	Selection of operation mode of watchdog timer[Note 2]
0	0	Operation stopped
0	1	Interval timer mode (when overflow occurs, a maskable interrupt occur)[Note 3]
1	0	Watchdog timer mode 1 (when overflow occurs, a non-maskable interrupt occurs)
1	1	Watchdog timer mode 2 (when overflow occurs, reset operation starts)

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 RESET input.

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

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

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

2. In watchdog timer mode 1 or 2, set WDTM4 to 1 after confirming that the WDTIF (bit 0 of interrupt request flag register 0 (IF0)) is set to 0. While WDTIF is 1, a non-maskable interrupt is generated upon write completion if watchdog timer mode 1 or 2 is selected.

12.4 Watchdog Timer Operation

12.4.1 Operation as watchdog timer

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

The count clock (runaway detection time interval) of the watchdog timer can be selected by bits 0 to 2 (WDCS0 to WDCS2) of the watchdog timer clock select register (WDCS). By setting bit 7 (RUN) of WDTM to 1, the watchdog timer is started. Set RUN to 1 within the set runaway 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 runaway detection time is exceeded, the system is reset or a non-maskable interrupt is generated by the value of bit 3 (WDTM3) of WDTM.

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

Cautions 1. The actual runaway 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 stops counting. In this case, therefore, the watchdog timer stops operation even though the main system clock is oscillating.

Table 12-4. Runaway Detection Time of Watchdog Timer

WDCS2	WDCS1	During fx = 5.0 MHz Operation	During fcc = 4.0 MHz Operation
0	0	2^{11}/fx (410 μs)	2^{11}/fcc (512 μs)
0	1	2^{13}/fx (1.64 ms)	2^{13}/fcc (2.05 ms)
1	0	2^{15} /fx (6.55 ms)	2^{15}/fcc (8.19 ms)
1	1	2^{17}/fx (26.2 ms)	2^{17}/fcc (32.8 ms)
Other than above		Setting prohibited	

Remarks 1. fx: Main system clock oscillation frequency (ceramic/crystal oscillation)

2. fcc: Main system clock oscillation frequency (RC oscillation)

12.4.2 Operation as interval timer

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

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

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

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

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

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

Table 12-5. Interval Time of Watchdog Timer

WDCS2	WDCS1	During fx = 5.0 MHz Operation	During fcc = 4.0 MHz Operation
0	0	2^{11}/fx (410 μs)	2^{11}/fcc (512 μs)
0	1	2^{13}/fx (1.64 ms)	2^{13}/fcc (2.05 ms)
1	0	2^{15} /fx (6.55 ms)	2^{15}/fcc (8.19 ms)
1	1	2^{17}/fx (26.2 ms)	2^{17}/fcc (32.8 ms)
Other than above		Setting prohibited	

Remarks 1. fx: Main system clock oscillation frequency (ceramic/crystal oscillation)

2. fcc: Main system clock oscillation frequency (RC oscillation)

CHAPTER 13 SERIAL INTERFACE 10

13.1 Serial Interface 10 Functions

Serial interface 10 has the following two modes.

- Operation stop mode
- 3-wire serial I/O mode

(1) Operation stop mode

This mode is used when serial transfer is not carried out. It enables a reduction in power consumption.

(2) 3-wire serial I/O mode (MSB/LSB-first switchable)

In this mode, 8-bit data transfer is carried out-first with three lines, one for the serial clock ($\overline{\text{SCK10}}$) and two for serial data (SI10 and SO10).

The 3-wire serial I/O mode supports simultaneous transmit and receive operations, reducing data transfer processing time.

It is possible to switch the start bit of 8-bit data to be transmitted between the MSB and the LSB, thus allowing connection to devices with either start bit.

The 3-wire serial I/O mode is effective for connecting display controllers and peripheral I/O such as the 75XL Series, 78K Series, and 17K Series, which have internal conventional clocked serial interfaces.

13.2 Serial Interface 10 Configuration

Serial interface 10 includes the following hardware.

Table 13-1. Configuration of Serial Interface 10

Item	Configuration
Register	Serial shift register 10 (SIO10)
Control register	Serial operation mode register 10 (CSIM10) Port mode register 2 (PM2) Port 2 (P2)

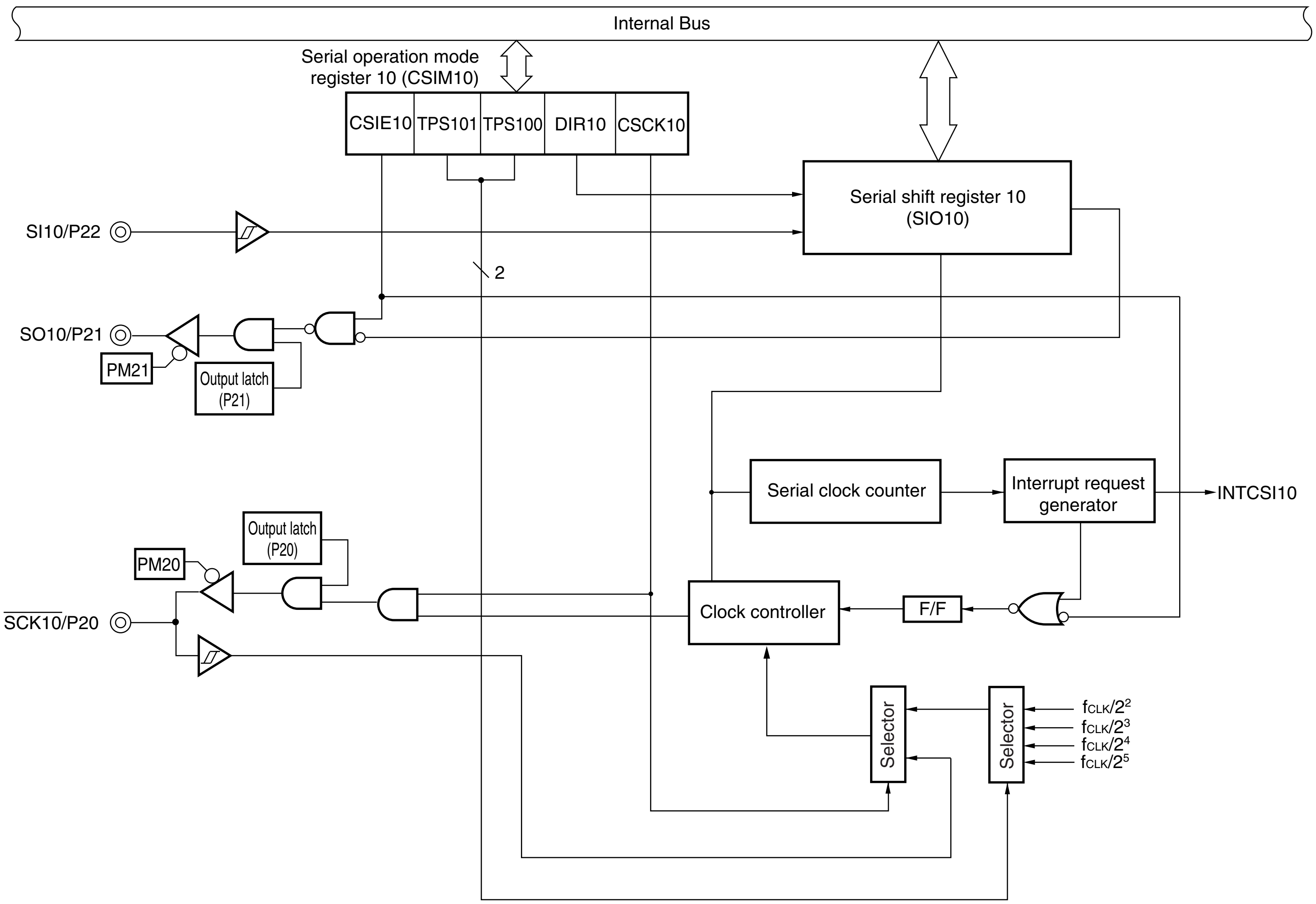
(1) Serial shift register 10 (SIO10)

This is an 8-bit register used for parallel-to-serial conversion and to perform serial data transmission/reception in synchronization with serial clocks.

SIO10 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes SIO10 undefined.

Figure 13-1. Block Diagram of Serial Interface 10



Remark f_{CLK}: f_X or f_{CC}

13.3 Register Controlling Serial Interface 10

The following three registers are used to control serial interface 10.

- Serial operation mode register 10 (CSIM10)
- Port mode register 2 (PM2)
- Port 2 (P2)

(1) Serial operation mode register 10 (CSIM10)

This register is used to control serial interface 10 and set the serial clock and start bit.
CSIM10 is set with a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears CSIM10 to 00H.

Figure 13-2. Format of Serial Operation Mode Register 10

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
CSIM10	CSIE10	0	TPS101	TPS100	0	DIR10	CSCK10	0	FF78H	00H	R/W

CSIE10	Operation control in 3-wire serial I/O mode
0	Operation stopped
1	Operation enabled

TPS101	TPS100	Count clock selection when internal clock is selected	
		During f_X = 5.0 MHz operation	During f_{CC} = 4.0 MHz operation
0	0	$f_X/2^2$ (1.25 MHz)	$f_{CC}/2^2$ (1.0 MHz)
0	1	$f_X/2^3$ (625 kHz)	$f_{CC}/2^3$ (500 kHz)
1	0	$f_X/2^4$ (313 kHz)	$f_{CC}/2^4$ (250 kHz)
1	1	$f_X/2^5$ (156 kHz)	$f_{CC}/2^5$ (125 kHz)

DIR10	Start bit specification
0	MSB
1	LSB

CSCK10	SIO10 clock selection
0	Input clock to $\overline{\text{SCK10}}$ pin from external
1	Internal clock selected by TPS100, TPS101

Cautions 1. Bits 0, 3, and 6 must be set to 0.
2. Switch operation mode after stopping the serial transmit/receive operation.

Remarks 1. f_X: Main system clock oscillation frequency (ceramic/crystal oscillation)
2. f_{CC}: Main system clock oscillation frequency (RC oscillation)

Table 13-2. Settings of Serial Interface 10 Operating Mode

(1) Operation stop mode

CSIM10			PM22	P22	PM21	P21	PM20	P20	Start	Shift	P22/SI10	P21/SO10	P20/$\overline{\text{SCK10}}$
CSIE10	DIR10	CSCK10							Bit	Clock	Pin Function	Pin Function	Pin Function
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

CSIM10			PM22	P22	PM21	P21	PM20	P20	Start	Shift	P22/SI10	P21/SO10	P20/$\overline{\text{SCK10}}$
CSIE10	DIR10	CSCK10							Bit	Clock	Pin Function	Pin Function	Pin Function
1	0	0	1[Note 2]	×[Note 2]	0	1	1	×	MSB	External clock	SI10[Note 2]	SO10 (CMOS output)	$\overline{\text{SCK10}}$ input
		1					0	1		Internal clock			$\overline{\text{SCK10}}$ output
1	1	0					1	×	LSB	External clock			$\overline{\text{SCK10}}$ input
		1					0	1		Internal clock			$\overline{\text{SCK10}}$ output
Other than above									Setting prohibited				

Notes 1. Can be used as port function.

2. If used only for transmission, can be used as P22 (CMOS I/O).

Remark ×: don't care

13.4 Serial Interface 10 Operation

Serial interface 10 provides the following two types of modes.

- Operation stop mode
- 3-wire serial I/O mode

13.4.1 Operation stop mode

In the operation stop mode, serial transfer is not executed, therefore enabling a reduction in the power consumption.

The P20/$\overline{\text{SCK10}}$, P21/SO10, and P22/SI10 pins can be used as normal I/O ports.

(1) Register setting

Operation stop mode is set by serial operation mode register 10 (CSIM10).

Serial operation mode register 10 (CSIM10)

CSIM10 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM10 to 00H.

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
CSIM10	CSIE10	0	TPS101	TPS100	0	DIR10	CSCK10	0	FF78H	00H	R/W

CSIE10	Operation control in 3-wire serial I/O mode
0	Operation stopped
1	Operation enabled

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

13.4.2 3-wire serial I/O mode

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

Communication is performed using three lines: a serial clock line ($\overline{\text{SCK10}}$), serial output line (SO10), and serial input line (SI10).

(1) Register setting

3-wire serial I/O mode settings are performed using serial operation mode register 10 (CSIM10), port mode register 2 (PM2), and port 2 (P2).

(a) Serial operation mode register 10 (CSIM10)

CSIM10 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM10 to 00H.

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
CSIM10	CSIE10	0	TPS101	TPS100	0	DIR10	CSCK10	0	FF78H	00H	R/W

CSIE10	Operation control in 3-wire serial I/O mode
0	Operation stopped
1	Operation enabled

TPS101	TPS100	Count clock selection when internal clock is selected	
		During f_X = 5.0 MHz operation	During f_{CC} = 4.0 MHz operation
0	0	$f_X/2^2$ (1.25 MHz)	$f_{CC}/2^2$ (1.0 MHz)
0	1	$f_X/2^3$ (625 kHz)	$f_{CC}/2^3$ (500 kHz)
1	0	$f_X/2^4$ (313 kHz)	$f_{CC}/2^4$ (250 kHz)
1	1	$f_X/2^5$ (156 kHz)	$f_{CC}/2^5$ (125 kHz)

DIR10	Start bit specification
0	MSB
1	LSB

CSCK10	SIO10 clock selection
0	Input clock to $\overline{\text{SCK10}}$ pin from external
1	Internal clock selected by TPS100, TPS101

Cautions 1. Bits 0, 3, and 6 must be set to 0.

2. Switch operation mode after stopping the serial transmit/receive operation.

Remarks 1. f_X: Main system clock oscillation frequency (ceramic/crystal oscillation)

2. f_{CC}: Main system clock oscillation frequency (RC oscillation)

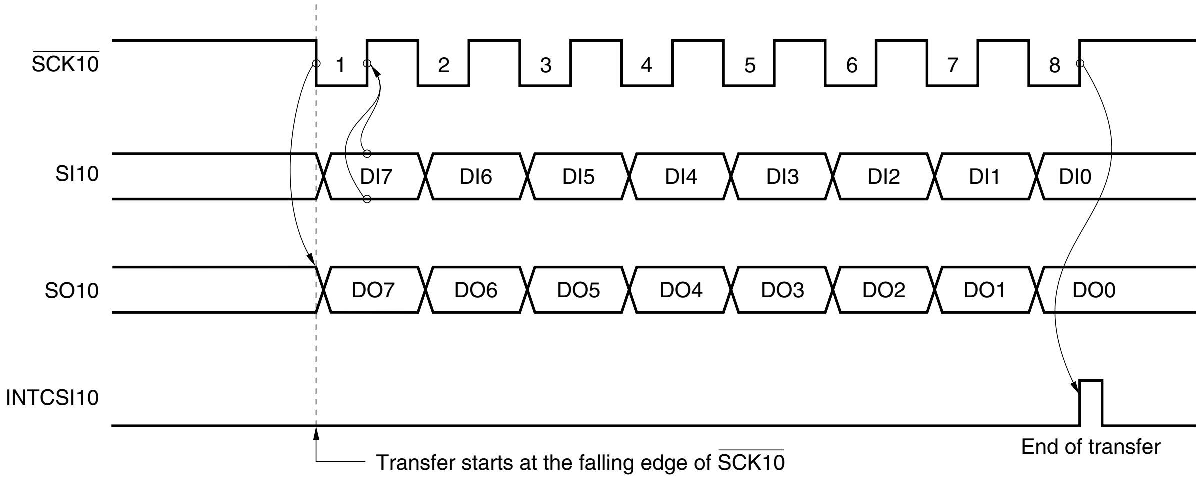
(2) Communication operation

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

Transmit shift register 10 (SIO10) shift operations are performed in synchronization with the fall of the serial clock ($\overline{\text{SCK10}}$). Transmit data is then held in the SO10 latch and output from the SO10 pin. Also, receive data input to the SI10 pin is latched in the input bits of SIO10 on the rise of $\overline{\text{SCK10}}$.

At the end of an 8-bit transfer, the operation of SIO10 stops automatically, and the interrupt request signal (INTCSI10) is generated.

Figure 13-3. 3-Wire Serial I/O Mode Timing


$\overline{\text{SCK10}}$
1
2
3
4
5
6
7
8
SI10
DI7
DI6
DI5
DI4
DI3
DI2
DI1
DI0
SO10
DO7
DO6
DO5
DO4
DO3
DO2
DO1
DO0
INTCSI10
End of transfer
Transfer starts at the falling edge of $\overline{\text{SCK10}}$

Cautions 1. When data is written to SIO10 in the serial operation disabled status (CSIE10 = 0), the data cannot be transmitted or received.

2. When data is written to SIO10 in the serial operation disabled status (CSIE10 = 0) and then serial operation is enabled (CSIE10 = 1), the data cannot be transmitted or received.

3. Once data has been written to SIO10 with the serial clock selected (CSCK10 = 0), overwriting the data does not update the contents of SIO10.

4. When CSIM10 is operated during data transmission/reception, data cannot be transmitted or received normally.

5. When SIO10 is operated during data transmission/reception, the data cannot be transmitted or received normally.

(3) Transfer start

Serial transfer is started by setting transfer data to the transmit shift register 10 (SIO10) when the following two conditions are satisfied.

- Bit 7 (CSIE10) of serial operation mode register 10 (CSIM10) = 1
- Internal serial clock is stopped or $\overline{\text{SCK10}}$ is a high level after 8-bit serial transfer.

Termination of 8-bit transfer stops the serial transfer automatically and generates the interrupt request signal (INTCSI10).

CHAPTER 14 SERIAL INTERFACE 20

14.1 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 carried out. It can reduce power consumption.

(2) Asynchronous serial interface (UART) mode

In this mode, one byte of data following the start bit is transmitted/received, and full-duplex operation is possible.

A dedicated UART baud rate generator is incorporated, allowing communication over a wide range of baud rates. In addition, the baud rate can be defined by scaling the input clock to the ASCK20 pin.

Caution Use the main system clock with ceramic/crystal oscillation in the UART mode. With RC oscillation, the frequency varies so much that transmission and reception may be affected when the internal clock is selected for the source clock of the baud rate generator.

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

In this mode, 8-bit data transfer is carried out with three lines, one for the serial clock ($\overline{\text{SCK20}}$) and two for serial data (SI20, SO20).

The 3-wire serial I/O mode supports simultaneous transmit and receive operations, reducing data transfer processing time.

It is possible to switch the start bit of 8-bit data to be transmitted between the MSB and the LSB, thus allowing connection to devices with either start bit.

The 3-wire serial I/O mode is effective for connecting display controllers and peripheral I/O such as the 75XL Series, 78K Series, and 17K Series, which have internal conventional clocked serial interfaces.

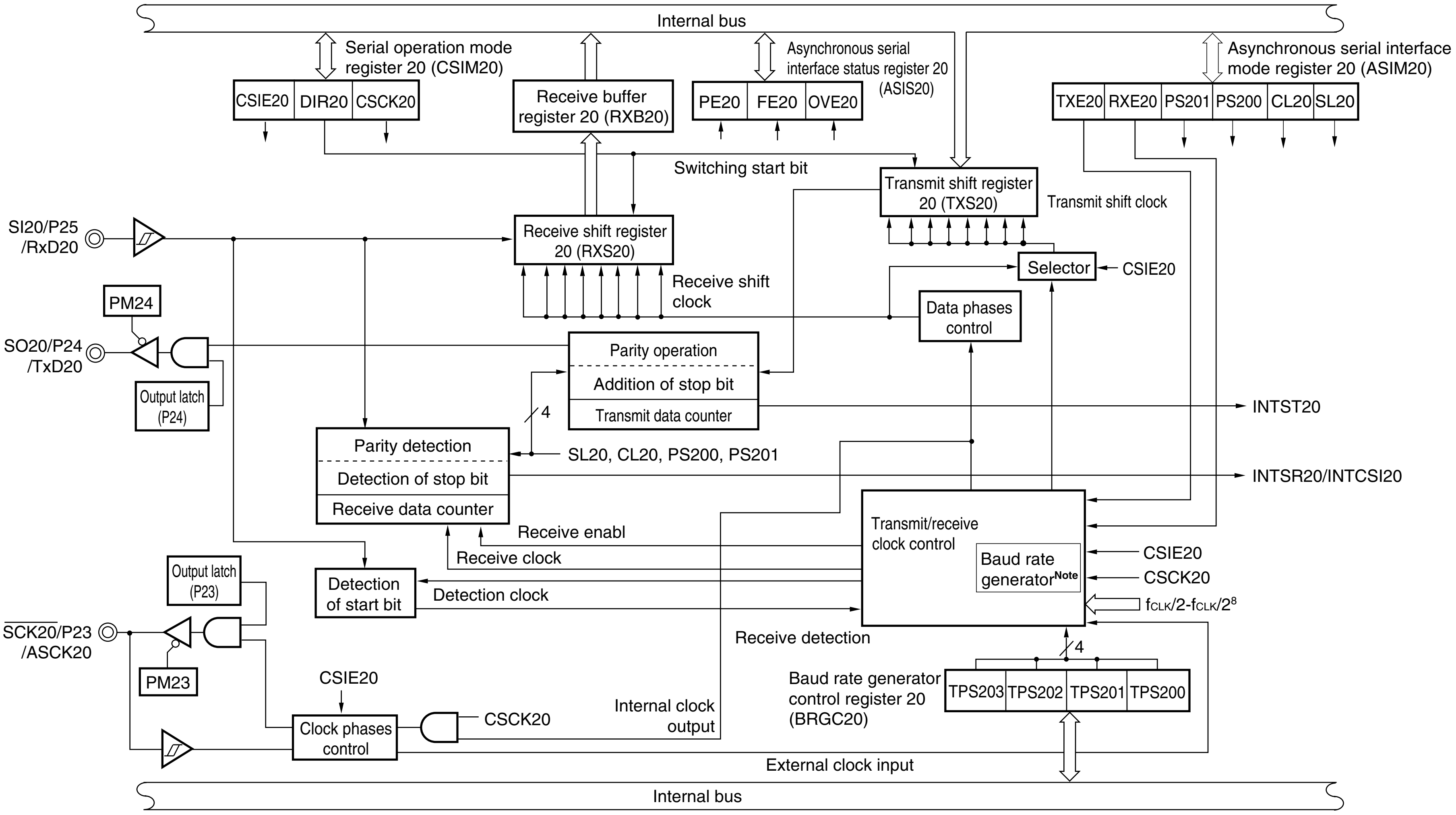
14.2 Serial Interface 20 Configuration

Serial interface 20 includes the following hardware configuration.

Table 14-1. 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)

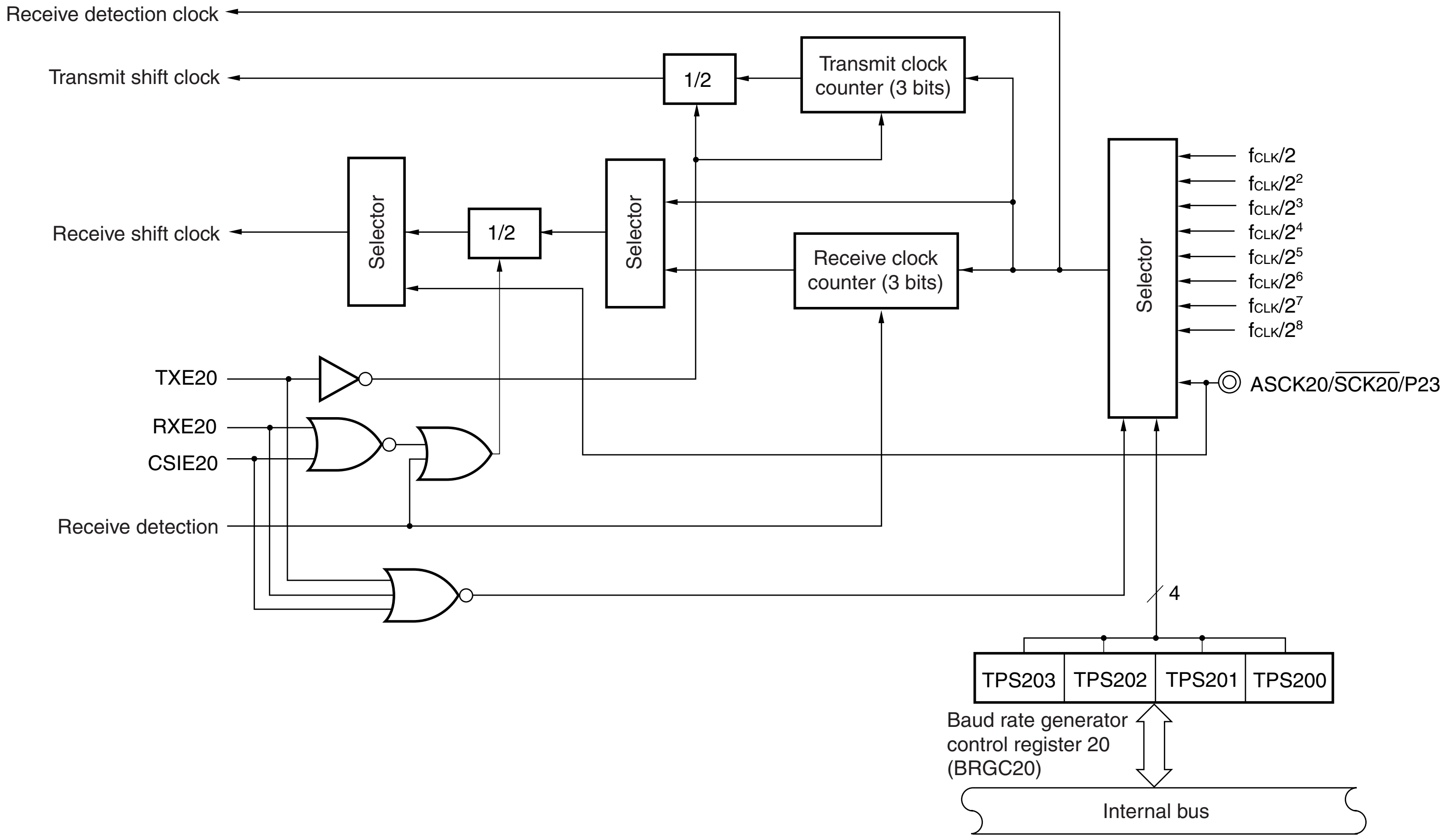
Figure 14-1. Block Diagram of Serial Interface 20



Note For the baud rate generator configuration, see Figure 14-2.

Remark fCLK: fX or fCC

Figure 14-2. Block Diagram of Baud Rate Generator



Remark f_{CLK}: f_X or f_{CC}

(1) Transmit shift register 20 (TXS20)

This register is used to specify data to be transmitted. Data written to TXS20 is transmitted as serial data.

If the data length is specified as 7 bits, bits 0 to 6 of the data written to TXS20 are transferred as the transmit data. The transmit operation is started by writing data to TXS20.

TXS20 is written to with an 8-bit memory manipulation instruction. It cannot be read.

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

Caution During a transmit operation, do not write to TXS20.
TXS20 and receive buffer register 20 (RXB20) are allocated to the same address, so when reading is performed, RXB20 values are read.

(2) Receive shift register 20 (RXS20)

This register is used to convert serial data input to the RxD20 pin into parallel data. Each time one byte of data is received, it is transferred to receive buffer register 20 (RXB20).

The RXS20 cannot be manipulated directly by program.

(3) Receive buffer register 20 (RXB20)

This register is used to hold received data. Each time one byte of data is received, a new byte of data is transferred from receive shift register 20 (RXS20).

If the data length is specified as 7 bits, receive data is transferred to bits 0 to 6 of RXB20, and the MSB of RXB20 always becomes 0.

RXB20 can be read with an 8-bit memory manipulation instruction. It cannot be written to.

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

Caution RXB20 and transmit shift register 20 (TXS20) are allocated to the same address, so when writing is performed, the values are written to TXS20.

(4) Transmit controller

This circuit controls transmit operations by adding a start bit, parity bit, and stop bit to data written to transmit shift register 20 (TXS20), according to the data set to asynchronous serial interface mode register 20 (ASIM20).

(5) Receive controller

This circuit controls receive operations according to the data set to asynchronous serial interface mode register 20 (ASIM20). It performs also parity error check, etc., during receive operations, and when an error is detected, it sets the value to asynchronous serial interface status register 20 (ASIS20) depending on the nature of the error.

14.3 Registers Controlling Serial Interface 20

The following six registers are used to control serial interface 20.

- 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)

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

Figure 14-3. 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	Operation control in 3-wire serial I/O mode
0	Operation stopped
1	Operation enabled

DIR20	Start bit specification
0	MSB
1	LSB

CSCK20	Clock selection in 3-wire serial I/O mode
0	Input clock to $\overline{\text{SCK20}}$ pin from external
1	Dedicated baud rate generator output

Cautions 1. Bits 0, and 3 to 6 must be set to 0.

2. Clear CSIM20 to 00H in the UART mode.

3. When the external input clock is selected in 3-wire serial I/O mode, set input mode by setting bit 3 of port mode register 2 (PM2) to 1.

4. Switching operation modes must be performed after the serial transmit/receive operation is stopped.

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

This register is set when using the serial interface 20 in the asynchronous serial interface mode.

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

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

Figure 14-4. 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	Character length specification of transmit data
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. Clear ASIM20 to 00H in the 3-wire serial I/O mode.

3. Switching operation modes must be performed after the serial transmit/receive operation is stopped.

Table 14-2. Settings of Serial Interface 20 Operating Mode

(1) Operation stop mode

ASIM20		CSIM20			PM25	P25	PM24	P24	PM23	P23	Start	Shift	P25/SI20/RxD20	P24/SO20/TxD20	P23/$\overline{\text{SCK20}}$/ASCK20
TXE20	RXE20	CSIE20	DIR20	CSCK20							Bit	Clock	Pin Function	Pin Function	Pin Function
0	0	0	x	x	x[Note 1]	x[Note 1]	x[Note 1]	x[Note 1]	x[Note 1]	x[Note 1]	—	—	P25	P24	P23
Other than above											Setting prohibited				

(2) 3-wire serial I/O mode

ASIM20		CSIM20			PM25	P25	PM24	P24	PM23	P23	Start	Shift	P25/SI20/RxD20	P24/SO20/TxD20	P23/$\overline{\text{SCK20}}$/ASCK20
TXE20	RXE20	CSIE20	DIR20	CSCK20							Bit	Clock	Pin Function	Pin Function	Pin Function
0	0	1	0	0	1[Note 2]	x[Note 2]	0	1	1	x	MSB	External clock	SI20[Note 2]	SO20 (CMOS output)	$\overline{\text{SCK20}}$ input
				1					0	1		Internal clock			$\overline{\text{SCK20}}$ output
		1	1	0					1	x	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			PM25	P25	PM24	P24	PM23	P23	Start	Shift	P25/SI20/RxD20	P24/SO20/TxD20	P23/$\overline{\text{SCK20}}$/ASCK20
TXE20	RXE20	CSIE20	DIR20	CSCK20							Bit	Clock	Pin Function	Pin Function	Pin Function
1	0	0	0	0	x[Note 1]	x[Note 1]	0	1	1	x	LSB	External clock	P25	TxD20 (CMOS output)	ASCK20 input
									x[Note 1]	x[Note 1]		Internal clock			P23
0	1	0	0	0	1	x	x[Note 1]	x[Note 1]	1	x		External clock	RxD20	P24	ASCK20 input
									x[Note 1]	x[Note 1]		Internal clock			P23
1	1	0	0	0	1	x	0	1	1	x		External clock		TxD20 (CMOS output)	ASCK20 input
									x[Note 1]	x[Note 1]		Internal clock			P23
Other than above											Setting prohibited				

Notes 1. Can be used as port function.

2. If used only for transmission, can be used as P25 (CMOS I/O).

Remark x: don't care

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

This register indicates types of error when a reception error is generated in the asynchronous interface mode.

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

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

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

Figure 14-5. 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	Parity error not generated
1	Parity error generated (when the transmit parity and receive parity did not match)

FE20	Flaming error flag
0	Flaming error not generated
1	Flaming error generated[Note 1] (when stop bit is not detected.)

OVE20	Overrun error flag
0	Overrun error not generated
1	Overrun error generated[Note 2] (when the next receive operation is completed before the data is read from the 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 in the case of reception is performed with 1 bit.

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

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

This register is used to set the serial clock of serial interface 20.

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

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

Figure 14-6. 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
				During f_X = 5.0 MHz operation	During f_{CC} = 4.0 MHz operation	
0	0	0	0	$f_X/2$ (2.5 MHz)	$f_{CC}/2$ (2.0 MHz)	1
0	0	0	1	$f_X/2^2$ (1.25 MHz)	$f_{CC}/2^2$ (1.0 MHz)	2
0	0	1	0	$f_X/2^3$ (625 kHz)	$f_{CC}/2^3$ (500 kHz)	3
0	0	1	1	$f_X/2^4$ (313 kHz)	$f_{CC}/2^4$ (250 kHz)	4
0	1	0	0	$f_X/2^5$ (156 kHz)	$f_{CC}/2^5$ (125 kHz)	5
0	1	0	1	$f_X/2^6$ (78.1 kHz)	$f_{CC}/2^6$ (62.5 kHz)	6
0	1	1	0	$f_X/2^7$ (39.1 kHz)	$f_{CC}/2^7$ (31.3 kHz)	7
0	1	1	1	$f_X/2^8$ (19.5 kHz)	$f_{CC}/2^8$ (15.6 kHz)	8
1	0	0	0	Input clock from external to ASCK20 pin[Note]		–
Other than above				Setting prohibited		

Note Only used in the UART mode.

Cautions 1. When writing to BRGC20 is performed 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.

2. Use the main system clock with ceramic/crystal oscillation in the UART mode. With RC oscillation, the frequency varies so much that transmission and reception may be affected when the internal clock is selected for the source clock of the baud rate generator.

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

4. When the external input clock is selected in 3-wire serial I/O mode, set input mode by setting bit 3 of port mode register 2 (PM2) to 1.

Remarks 1. f_X: Main system clock oscillation frequency (ceramic/crystal oscillation)

2. f_{CC}: Main system clock oscillation frequency (RC oscillation)

3. n: Value determined in the settings of TPS200 to TPS203 ($1 \leq n \leq 8$)

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

(a) Generation of UART baud rate transmit/receive clock by means of system clock

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

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

f_X: Main system clock oscillation frequency (ceramic/crystal oscillation)
n: Value in Figure 14-6 that is determined by the settings of TPS200 to TPS203 ($2 \leq n \leq 8$)

Table 14-3. Example of Relationship Between System Clock and Baud Rate

Baud Rate (bps)	n	BRGC00 Set Value	Error (%)	
			f_X = 5.0 MHz	f_X = 4.9152 MHz
1200	8	70H	1.73	0
2400	7	60H		
4800	6	50H		
9600	5	40H		
19200	4	30H		
38400	3	20H		
76800	2	10H		

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

(b) Generation of UART baud rate transmit/receive clock by means of external clock from ASCK20 pin

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

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

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

Table 14-4. 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
1200	19.2
2400	38.4
4800	76.8
9600	153.6
19200	307.2
31250	500.0
38400	614.4

(c) Generation of serial clock from system clock in 3-wire serial I/O

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, it is unnecessary to set BRGC20.

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

f_{CLK}: f_X or f_{CC}

f_X: Main system clock oscillation frequency (ceramic/crystal oscillation)

f_{CC}: Main system clock oscillation frequency (RC oscillation)

n: Values in Figure 14-6 determined by the settings of TPS200 to TPS203 ($1 \leq n \leq 8$)

14.4 Serial Interface 20 Operation

Serial interface 20 has the following three modes.

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

14.4.1 Operation stop mode

In the operation stop mode, serial transfer is not executed, therefore enabling a reduction in the power consumption.

The P23/$\overline{\text{SCK20}}$/ASCK20, P24/SO20/TxD20, and P25/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 stopped
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.

RESET input clears ASIM20 to 00H.

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

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

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

Caution Bits 0 and 1 must be set to 0.

14.4.2 Asynchronous serial interface (UART) mode

In this mode, the one-byte data following the start bit is transmitted/received and thus full-duplex communications are possible.

This device incorporates a UART-dedicated baud rate generator that enables communications at a desired transfer rate from many options. In addition, the baud rate can be also 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 which complies with the MIDI standard.

Caution Use the main system clock with ceramic/crystal oscillation in the UART mode. With RC oscillation, the frequency varies so much that transmission and reception may be affected when the internal clock is selected for the source clock of the baud rate generator.

(1) Register setting

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

(a) Serial operation mode register 20 (CSIM20)

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

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

Clear CSIM20 to 00H in the UART mode.

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 stopped
1	Operation enabled

DIR20	Start bit specification
0	MSB
1	LSB

CSCK20	Clock selection in 3-wire serial I/O mode
0	Input clock to $\overline{\text{SCK20}}$ pin from external
1	Dedicated baud rate generator output

Caution 1. Bits 0 and 3 to 6 must be set to 0.

2. Switching operation modes must be performed after the serial transmit/receive operation is stopped.

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

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

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

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

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

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

PS201	PS200	Parity bit specification
0	0	No parity
0	1	Always add 0 parity at transmission Parity check is not performed at reception (No parity error is generated)
1	0	Odd parity
1	1	Even parity

CL20	Character length specification
0	7 bits
1	8 bits

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

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

2. Switching operation modes must be performed after the serial transmit/receive operation is stopped.

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

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

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

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

PE20	Parity error flag
0	Parity error not generated
1	Parity error generated (when the transmit parity and receive parity did not match)

FE20	Flaming error flag
0	Flaming error not generated
1	Flaming error generated (when stop bit is not detected.)[Note 1]

OVE20	Overrun error flag
0	Overrun error not generated
1	Overrun error generated (when the next receive operation is completed before the data is read from the receive buffer register.)[Note 2]

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 in the case of reception is performed with 1 bit.

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

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

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

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

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

TPS203	TPS202	TPS201	TPS200	Selection of baud rate generator source clock	n
0	0	0	0	fx/2 (2.5 MHz)	1
0	0	0	1	fx/2^2 (1.25 MHz)	2
0	0	1	0	fx/2^3 (625 kHz)	3
0	0	1	1	fx/2^4 (313 kHz)	4
0	1	0	0	fx/2^5 (156 kHz)	5
0	1	0	1	fx/2^6 (78.1 kHz)	6
0	1	1	0	fx/2^7 (39.1 kHz)	7
0	1	1	1	fx/2^8 (19.5 kHz)	8
1	0	0	0	Input clock to ASCK20 pin from external[Note]	–
Other than above				Setting prohibited	

Note Only used in the UART mode.

Cautions 1. When writing to BRGC20 is performed 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.

2. Use the main system clock with ceramic/crystal oscillation in the UART mode. With RC oscillation, the frequency varies so much that transmission and reception may be affected when the internal clock is selected for the source clock of the baud rate generator.

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

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

Remarks 1. fx: Main system clock oscillation frequency (ceramic/crystal oscillation)

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

3. n: Value determined in the settings of TPS200 to TPS203 ($1 \leq n \leq 8$)

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

(i) Generation of UART baud rate transmit/receive clock by means of system clock

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

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

f_X: Main system clock oscillation frequency (ceramic/crystal oscillation)

n: Value in the above table that is determined by the settings of TPS200 to TPS203 ($2 \leq n \leq 8$)

Table 14-5. Example of Relationship Between Main System Clock and Baud Rate

Baud Rate (bps)	n	BRGC20 Set Value	Error (%)	
			f_X = 5.0 MHz	f_X = 4.9152 MHz
1200	8	70H	1.73	0
2400	7	60H		
4800	6	50H		
9600	5	40H		
19200	4	30H		
38400	3	20H		
76800	2	10H		

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

(ii) Generation of UART baud rate transmit/receive clock by means of external clock from ASCK20 pin

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

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

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

Table 14-6. 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
1200	19.2
2400	38.4
4800	76.8
9600	153.6
19200	307.2
31250	500.0
38400	614.4

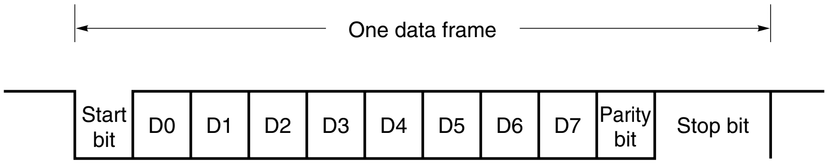
(2) Communication operation

(a) Data format

The transmit/receive data format is as shown in Figure 14-7. One data frame consists of a start bit, character bits, parity bit and stop bit(s).

The specification of character bit length, parity selection, and specification of stop bit length for each data frame is carried out by asynchronous serial interface mode register 20 (ASIM20).

Figure 14-7. Format of Asynchronous Serial Interface Transmit/Receive Data



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

When 7 bits are selected as the number of character bits, only the lower 7 bits (bits 0 to 6) are valid; in transmission the most significant bit (bit 7) is ignored, and in reception the most significant bit (bit 7) is always 0.

The serial transfer rate is selected by means of baud rate generator control register 20 (BRGC20).

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

(b) Parity types and operation

The parity bit is used to detect a bit error in the communication data. Normally, the same kind of parity bit is used on the transmitting side and the receiving side. With even parity and odd parity, a one-bit (odd number) error can be detected. With 0 parity and no parity, an error cannot be detected.

(i) Even parity

- **At transmission**

  The transmission operation is controlled so that the number of bits with a value of 1 in the transmit data including parity bit may be even. The parity bit value should be as follows.

  The number of bits with a value of 1 is an odd number in transmit data: 1
  The number of bits with a value of 1 is an even number in transmit data: 0

- **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 is generated.

(ii) Odd parity

- **At transmission**

  Conversely to the even parity, the transmission operation is controlled so that the number of bits with a value of 1 in the transmit data including parity bit may be odd. The parity bit value should be as follows.

  The number of bits with a value of 1 is an odd number in transmit data: 0
  The number of bits with a value of 1 is an even number in transmit data: 1

- **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 is generated.

(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 is not generated, 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 is not generated.

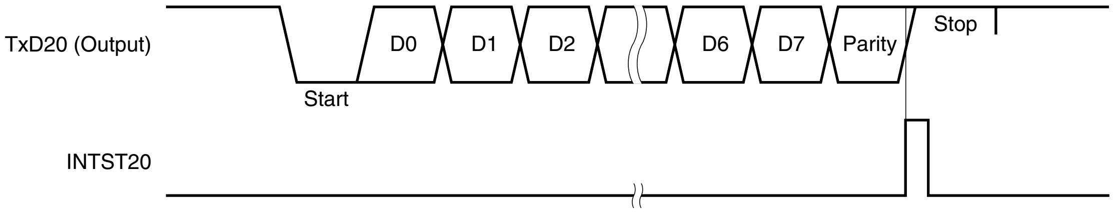
(c) Transmission

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.

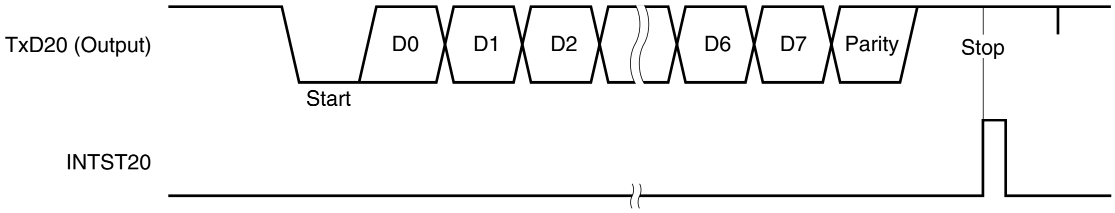
When the transmit operation starts, the data in TXS20 is shifted out, and when TXS20 is empty, a transmission completion interrupt (INTST20) is generated.

Figure 14-8. Asynchronous Serial Interface Transmission Completion Interrupt Timing

(a) Stop bit length: 1



(b) Stop bit length: 2



Caution Do not overwrite asynchronous serial interface mode register 20 (ASIM20) during a transmit operation. If the ASIM20 register is overwritten during transmission, subsequent transmission may not operate (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 the 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 ASIM20.

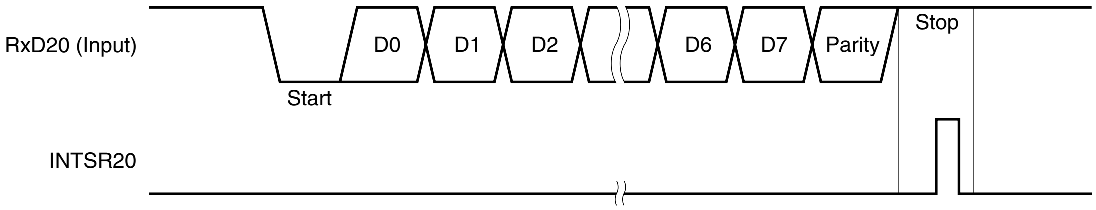
When the RxD20 pin input becomes low, the 3-bit counter starts counting, and at the time when half the time determined by specified baud rate has passed, the data sampling start timing signal is output. If the RxD20 pin input sampled again as a result of this start timing signal is low, it is identified 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 is generated, the receive data in which the error was generated 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 14-9. 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 be occurred when the next data is received, and the receive error state will continue indefinitely.

(e) Receive errors

The following three errors may occur during a receive operation: a parity error, framing error, or overrun error. An error flag is set in asynchronous serial interface status register 20 (ASIS20) as the result of data reception. Receive error causes are shown in Table 14-7.

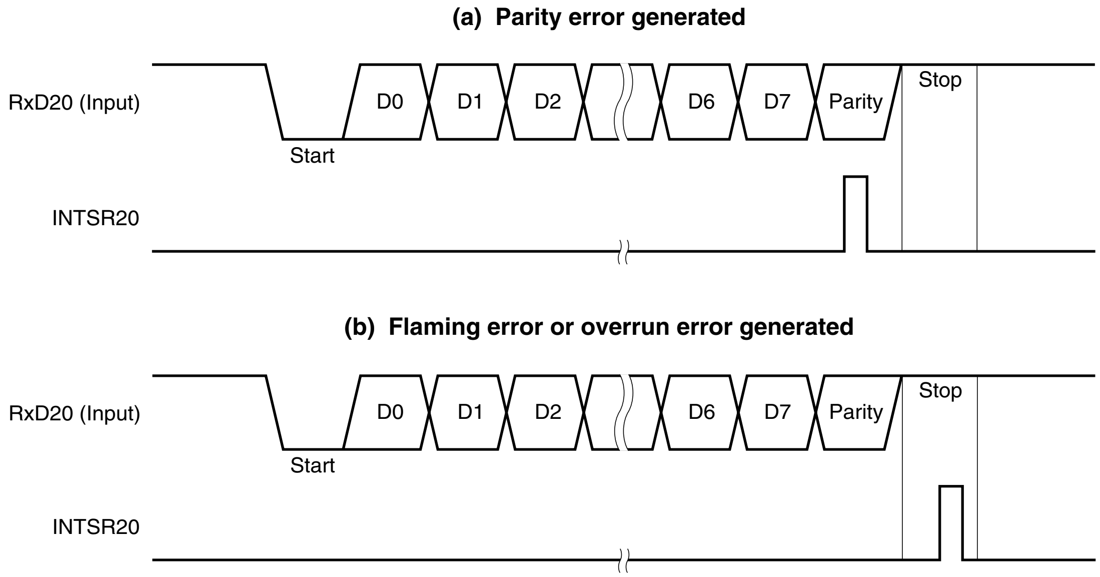
It is possible to determine what kind of error was generated during reception by reading the contents of ASIS20 in the reception error interrupt servicing (see Figures 14-9 and 14-10).

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 14-7. Receive Error Causes

Receive Errors	Cause	Value of ASIS20
Parity error	Parity specified at transmission and reception data parity do not match.	04H
Framing error	Stop bit is not detected.	02H
Overrun error	Reception of next data is completed before data is read from receive buffer register.	01H

Figure 14-10. Receive Error Timing



(a) Parity error generated

(b) Flaming error or overrun error generated

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) even if a receive error occurs. If RXB20 is not read, an overrun error will be occurred when the next data is received, and the receive error state will continue indefinitely.

(f) Reading receive data

When the reception completion interrupt (INTSR20) is generated, 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))

```
INTRXE:                   ;<Reception completion interrupt routine>
      NOP                 ;2 clocks
      CLR1 RXE20          ;Reception stopped
      MOV  A, RXB20       ;Read receive data
```

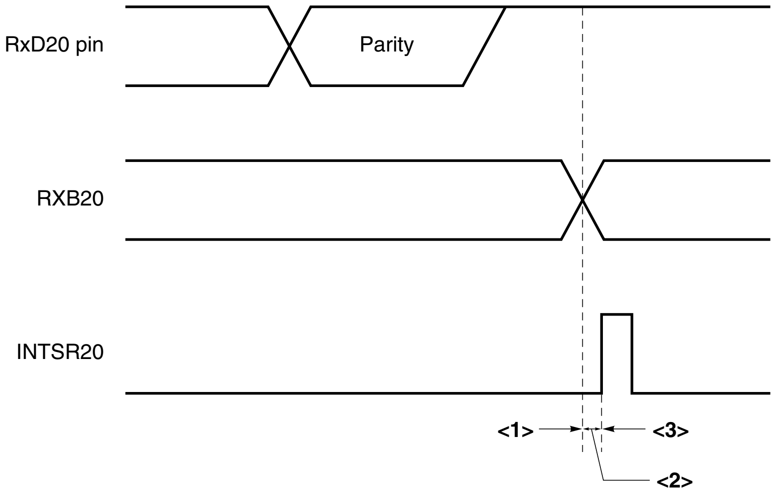
Program example of (b)

```
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) UART mode cautions

(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 a time indicated by **<1>**, RXB20 holds the previous data and does not generate INTSR20.
When RXE20 is set to 0 at a time indicated by **<2>**, RXB20 updates the data and does not generate INTSR20.
When RXE20 is set to 0 at a time indicated by **<3>**, RXB20 updates the data and generates INTSR20.

14.4.3 3-wire serial I/O mode

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

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

(1) Register setting

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

(a) Serial operation mode register 20 (CSIM20)

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

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

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

CSIE20	Operation control in 3-wire serial I/O mode
0	Operation stopped
1	Operation enabled

DIR20	Start bit specification
0	MSB
1	LSB

CSCK20	Clock selection in 3-wire serial I/O mode
0	Input clock to $\overline{\text{SCK20}}$ pin from external
1	Dedicated baud rate generator output

Caution 1. Bits 0 and 3 to 6 must be set to 0.

2. When the external input clock is selected in 3-wire serial I/O mode, set input mode by setting bit 3 of port mode register 2 (PM2) to 1.

3. Switching operation modes must be performed after the serial transmit/receive operation is stopped.

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

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

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

When the 3-wire serial I/O mode is selected, ASIM20 must be cleared 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. Switching operation modes must be performed after the serial transmit/receive operation is stopped.

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

BRGC20 is set with an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears BRGC20 to 00H.

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

TPS203	TPS202	TPS201	TPS200	Selection of baud rate generator source clock		n
				During f_X = 5.0 MHz operation	During f_{CC} = 4.0 MHz operation	
0	0	0	0	$f_X/2$ (2.5 MHz)	$f_{CC}/2$ (2.0 MHz)	1
0	0	0	1	$f_X/2^2$ (1.25 MHz)	$f_{CC}/2^2$ (1.0 MHz)	2
0	0	1	0	$f_X/2^3$ (625 kHz)	$f_{CC}/2^3$ (500 kHz)	3
0	0	1	1	$f_X/2^4$ (313 kHz)	$f_{CC}/2^4$ (250 kHz)	4
0	1	0	0	$f_X/2^5$ (156 kHz)	$f_{CC}/2^5$ (125 kHz)	5
0	1	0	1	$f_X/2^6$ (78.1 kHz)	$f_{CC}/2^6$ (62.5 kHz)	6
0	1	1	0	$f_X/2^7$ (39.1 kHz)	$f_{CC}/2^7$ (31.3 kHz)	7
0	1	1	1	$f_X/2^8$ (19.5 kHz)	$f_{CC}/2^8$ (15.6 kHz)	8
1	0	0	0	Input clock to ASCK20 pin from external[Note]		–
Other than above				Setting prohibited		

Note In 3-wire serial I/O mode, it is setting prohibited.

Caution When writing to BRGC20 is performed 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.

Remarks 1. f_X: Main system clock oscillation frequency (ceramic/crystal oscillation)

2. f_{CC}: Main system clock oscillation frequency (RC oscillation)

3. n: Value in the above table that is determined in the settings of TPS200 to TPS203 ($1 \le n \le 8$)

If the internal clock is used as the serial clock for the 3-wire serial I/O mode, set the TPS200 to TPS203 bits to set the frequency of the serial clock. To obtain the frequency to be set, use the following formula. When the serial clock is input from external, setting BRGC20 is unnecessary.

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

f_{CLK}: f_X or f_{CC}
f_X: Main system clock oscillation frequency (ceramic/crystal oscillation)
f_{CC}: Main system clock oscillation frequency (RC oscillation)
n: Value in the above table that is determined in the settings of TPS200 to TPS203 ($1 \le n \le 8$)

(2) Communication operation

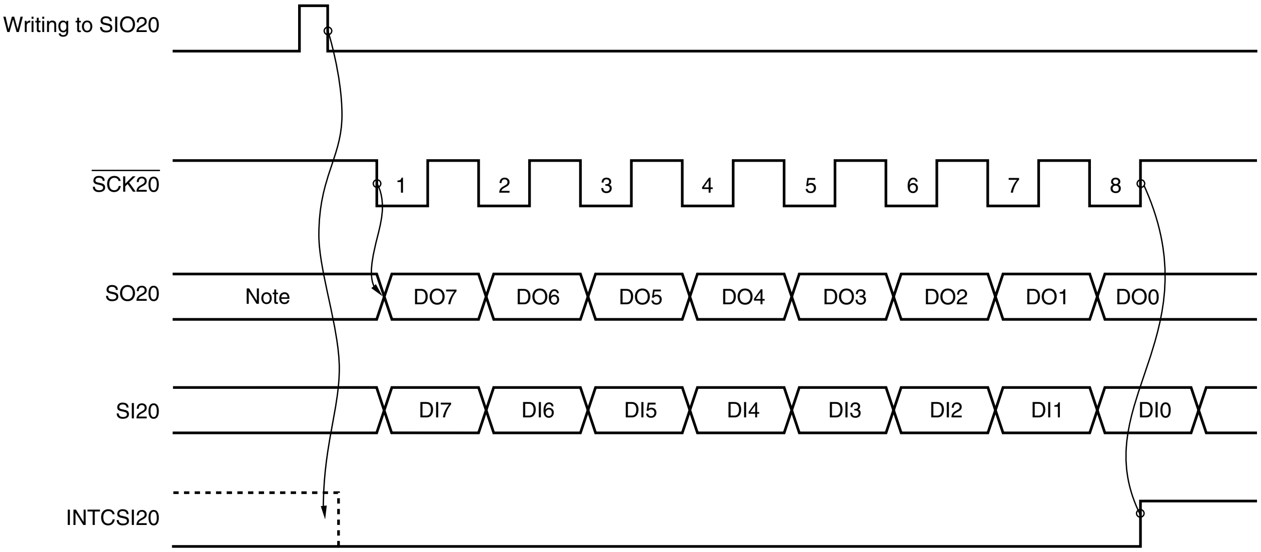
In the 3-wire serial I/O mode, data transmission/reception is performed in 8-bit units. Data is transmitted/received bit by bit in synchronization with the serial clock.

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

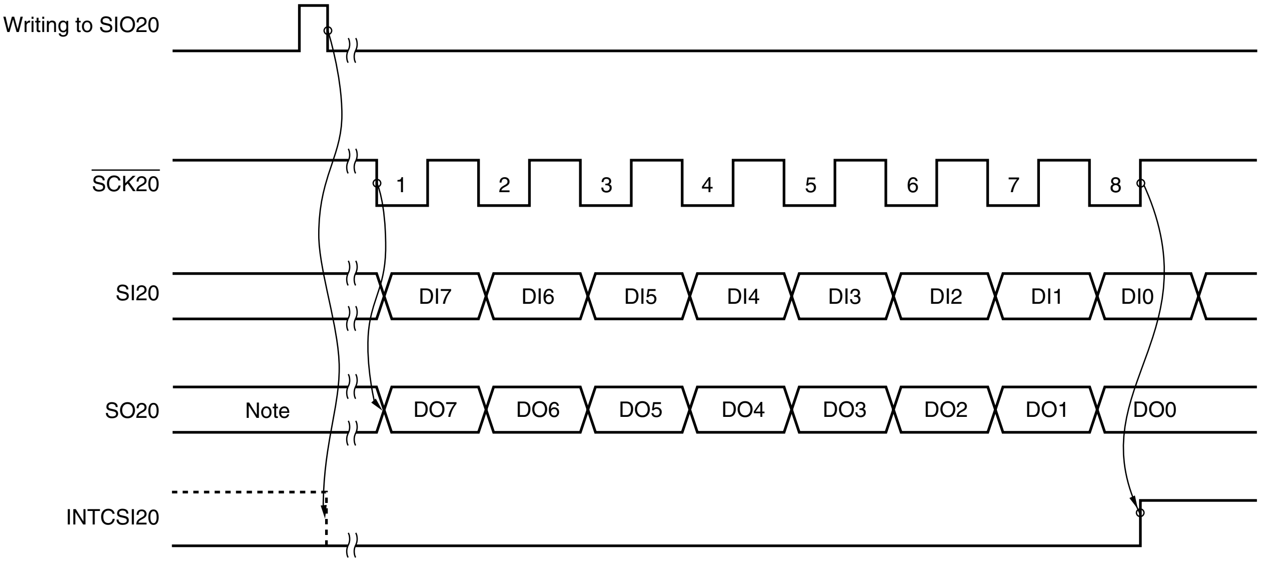
At the end of an 8-bit transfer, the operation of TXS20/SIO20 or RXS20 stops automatically, and the interrupt request signal (INTCSI20) is generated.

Figure 14-11. 3-Wire Serial I/O Mode Timing

(i) Master operation



(ii) Slave operation



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 a high level after 8-bit serial transfer.

Caution If CSIE20 is set to 1 after data write 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 15 LCD CONTROLLER/DRIVER

15.1 LCD Controller/Driver Functions

The functions of the LCD controller/driver of the μPD789306 and μPD789316 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) Up to 24 segment signal outputs (S0 to S23) and four common signal outputs (COM0 to COM3)
(5) Operation with a subsystem clock
(6) On-chip voltage boosting circuit

Table 15-1 lists the maximum number of pixels that can be displayed in each display mode.

Table 15-1. Maximum Number of Pixels

Bias Mode	Number of Time Slices	Common Signals Used	Maximum Number of Pixels
1/3	3	COM0 to COM2	72 (24 segments × 3 commons)Note 1
	4	COM0 to COM3	96 (24 segments × 4 commons)Note 2

Notes 1. 8-digit LCD panel, each digit having a 3-segment 8. configuration.
2. 12-digit LCD panel, each digit having a 2-segment 8. configuration.

15.2 LCD Controller/Driver Configuration

The LCD controller/driver includes the following hardware.

Table 15-2. Configuration of LCD Controller/Driver

Item	Configuration
Display outputs	Segment signals: 24 Common signals: 4
Control registers	LCD display mode register 0 (LCDM0) LCD clock control register 0 (LCDC0) LCD voltage amplification control register 0 (LCDVA0)

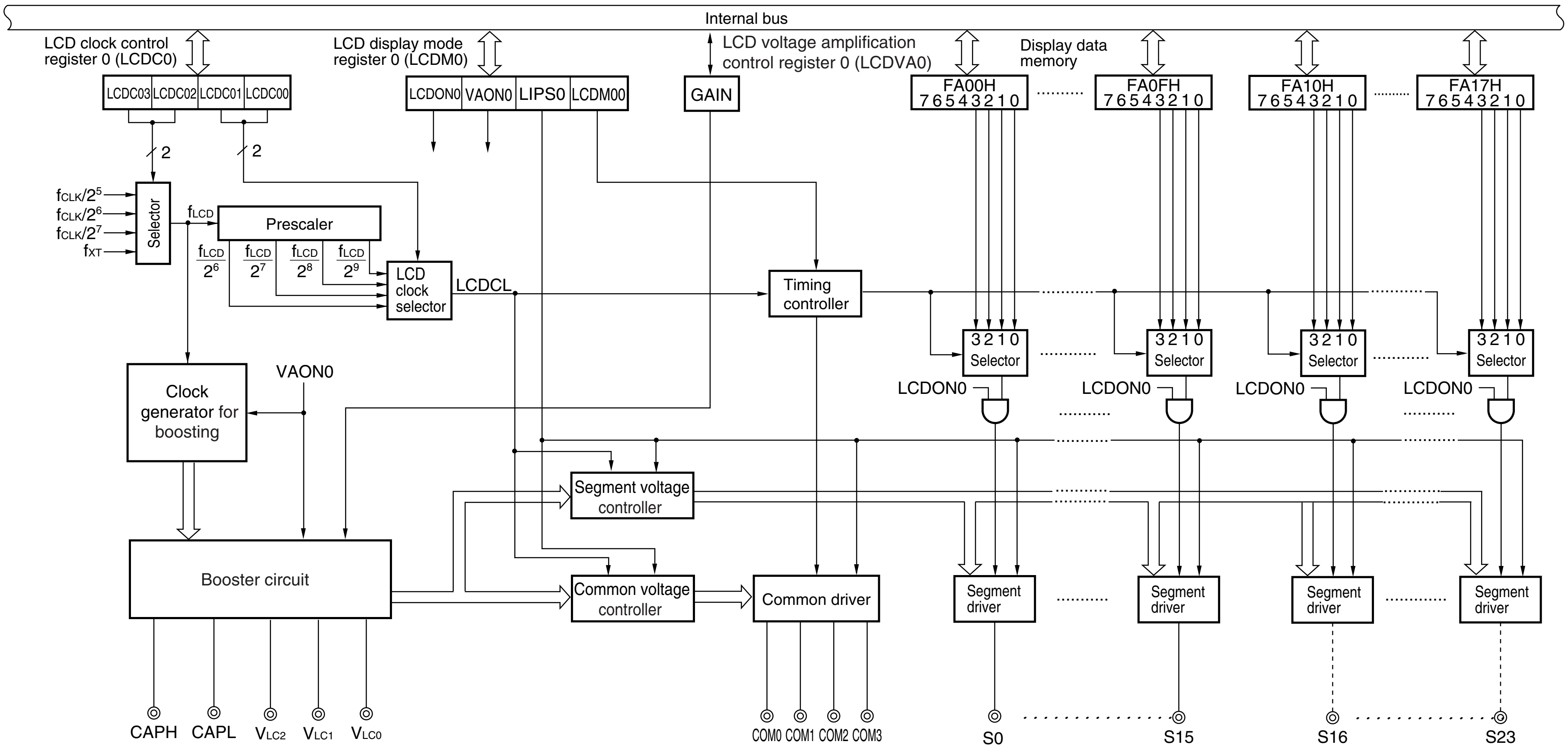
The correspondence with the LCD display RAM is shown in Figure 15-1 below.

Figure 15-1. Correspondence with LCD Display RAM

Address	Bit								Segment
	7	6	5	4	3	2	1	0	
FA17H	0	0	0	0					→ S23
FA16H	0	0	0	0					→ S22
FA15H	0	0	0	0					→ S21
FA14H	0	0	0	0					→ S20
FA13H	0	0	0	0					→ S19
FA12H	0	0	0	0					→ S18
FA11H	0	0	0	0					→ S17
FA10H	0	0	0	0					→ S16
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	

Remark Bit 4 to 7 are fixed to 0.

★ Figure 15-2. Block Diagram of LCD Controller/Driver



Remark f_{CLK}: f_X or f_{CC}

15.3 Registers Controlling LCD Controller/Driver

The following three registers control the LCD controller/driver.

- LCD display mode register 0 (LCDM0)
- LCD clock control register 0 (LCDC0)
- LCD voltage amplification control register 0 (LCDVA0)

(1) LCD display mode register 0 (LCDM0)

LCDM0 is used to control the LCD display enable/disable status, booster circuit operation enable/disable status, segment pin/common pin output, and the display mode.

LCDM0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears LCDM0 to 00H.

Figure 15-3. Format of LCD Display Mode Register 0

Symbol	<7>	<6>	5	<4>	3	2	1	0	Address	After reset	R/W
LCDM0	LCDON0	VAON0	0	LIPS0	0	0	0	LCDM00	FFB0H	00H	R/W

LCDON0	LCD display enable/disable
0	Display off (all segment signals are deselected.)
1	Display on

VAON0	Booster circuit operation enable/disable[Note]
0	Booster circuit stopped
1	Booster circuit enabled

LIPS0	Operation control of segment pin/common pin output[Note]
0	Output ground level to segment/common pin
1	Output deselect 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, the VAON0 and LIPS0 must be set to 0 to reduce power consumption.

Cautions 1. Bits 1 to 3 and 5 must be set to 0.

2. When operating VAON0, follow the procedure described below.

A. To stop voltage amplification after switching display status from on to off:

1) Set to display off status by setting LCDON0 = 0.

2) Disable outputs of all the segment buffers and common buffers by setting LIPS0 = 0.

3) Stop voltage amplification by setting VAON0= 0.

B. To stop voltage amplification during display on status:

Setting prohibited. Be sure to stop voltage amplification after setting display off.

C. To set display on from voltage amplification stop status:

★ **1) Start voltage amplification by setting VAON0 = 1, then wait for voltage boost wait time (t_{VAWAIT}) (See CHAPTER 22 ELECTRICAL SPECIFICATIONS).**

2) Set all the segment buffers and common buffers to non-display output status by setting LIPS0 = 1.

3) Set display on by setting LCDON0 = 1.

(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 the number of time slices.

LCDC0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears LCDC0 to 00H.

Figure 15-4. 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]	
		During f_X = 5.0 MHz or f_{XT} = 32.768 kHz operation	During f_{CC} = 4.0 MHz or f_{XT} = 32.768 kHz operation
0	0	f_{XT} (32.768 kHz)	
0	1	$f_X/2^5$ (156.3 kHz)	$f_{CC}/2^5$ (125 kHz)
1	0	$f_X/2^6$ (78.1 kHz)	$f_{CC}/2^6$ (62.5 kHz)
1	1	$f_X/2^7$ (39.1 kHz)	$f_{CC}/2^7$ (31.3 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.

Remarks 1. f_X: Main system clock oscillation frequency (ceramic/crystal oscillation)

2. f_{CC}: Main system clock oscillation frequency (RC oscillation)

3. f_{XT}: Subsystem clock oscillation frequency

As an example, Table 15-3 lists the frame frequencies used when f_{XT} (32.768 kHz) is supplied as the LCD source clock (f_{LCD}).

Caution Set the frame frequency to 128 Hz or lower.

Table 15-3. Frame Frequencies (Hz)

LCD Clock (LCDCL) 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[Note]
4	16	32	64	128

Note This setting is prohibited because it causes the frame frequency to exceed 128 Hz.

(3) LCD voltage amplification control register 0 (LCDVA0)

LCDVA0 controls the voltage amplification level during the voltage amplifier operation.

LCDVA0 is set using a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears LCDVA0 to 00H.

Figure 15-5. Format of LCD Voltage Boost Control Register 0

Symbol	7	6	5	4	3	2	1	<0>	Address	After reset	R/W
LCDVA0	0	0	0	0	0	0	0	GAIN	FFB3H	00H	R/W

GAIN	Reference voltage (V_{LC2}) level selection[Note]
0	1.5 V (specification of the LCD panel used is 4.5 V.)
1	1.0 V (specification of the LCD panel used is 3 V.)

Note Select the settings according to the specifications of the LCD panel that is used.

Caution Before changing the LCDVA0 setting, be sure to stop voltage boosting (VAON0 = 0).

Remark The TYP. value is indicated as the reference voltage (V_{LC2}) value.

15.4 Setting LCD Controller/Driver

Set the LCD controller/driver using the following procedure.

<1> Set the frame frequency using LCD clock control register 0 (LCDC0).
<2> Set the voltage amplification level using LCD voltage amplification control register 0 (LCDVA0).
GAIN = 0: V_{LC0} = 4.5 V, V_{LC1} = 3 V, V_{LC2} = 1. 5 V
GAIN = 1: V_{LC0} = 3 V, V_{LC1} = 2 V, V_{LC2} = 1 V
<3> Set the time division using LCDM00 (bit 0 of LCD display mode register 0 (LCDM0)).
<4> Enable voltage amplification by setting VAON0 (bit 6 of LCDM0) (VAON0 = 1).
★ **<5>** Wait for voltage boost wait time (t_{VAWAIT}) after setting VAON0 (See **CHAPTER 22 ELECTRICAL SPECIFICATIONS**).
<6> Set LIPS0 (bit 4 of LCDM0) (LIPS0 = 1) and output the deselect potential.
<7> Start output corresponding to each data memory by setting LCDON0 (bit 7 of LCDM0) (LCDON0 =1).

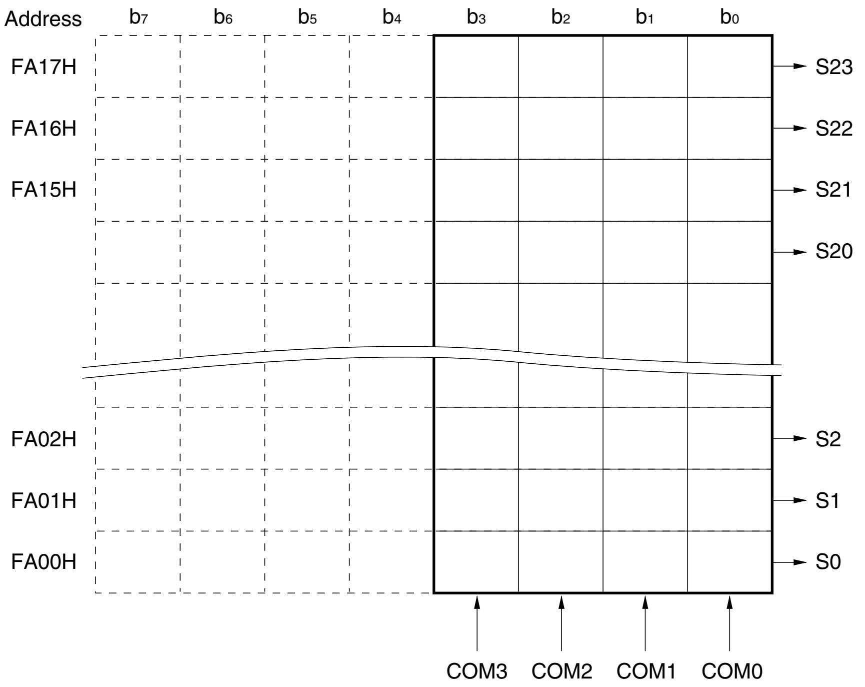
15.5 LCD Display Data Memory

The LCD display data memory is mapped at addresses FA00H to FA17H. Data in the LCD display data memory can be displayed on the LCD panel using the LCD controller/driver.

Figure 15-6 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 15-6. Relationship Between LCD Display Data Memory Contents and Segment/Common Outputs



Caution No memory has been installed as the higher 4 bits of the LCD display data memory. Be sure to set 0 to them.

15.6 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 a specified number of time slots at the timing listed in Table 15-4. In the static display mode, the same signal is output to COM0 to COM3 in common.

In the three-time slot mode, keep the COM3 pin open.

Table 15-4. COM Signals

COM Signal / Number of Time Slots	COM0	COM1	COM2	COM3
Three-time slot mode	↑→	→	→↓	Open
Four-time slot mode	↑→	→	→	→↓

(2) Segment signals

The segment signals correspond to 24 bytes of LCD display data memory (FA00H to FA17H). 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, it 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 (S0 to S23).

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 slot mode. So this bit can be used for purposes other than display.

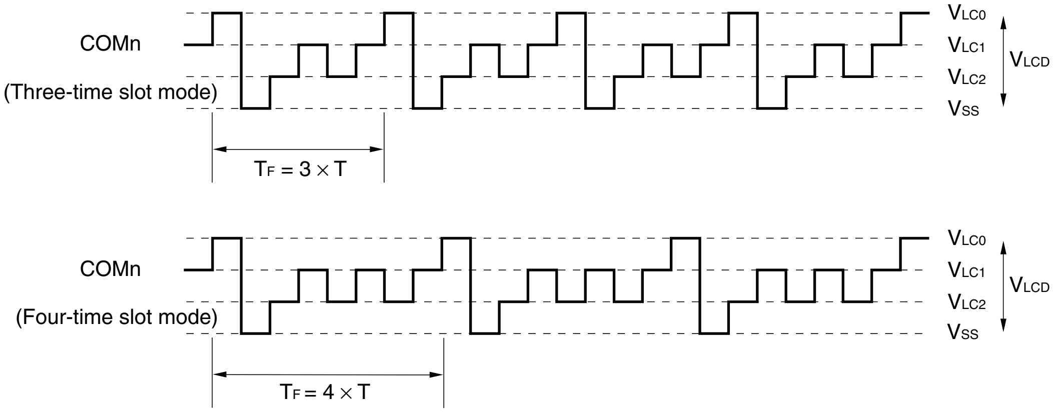
LCD display data memory bits 4 to 7 are fixed to 0.

(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.

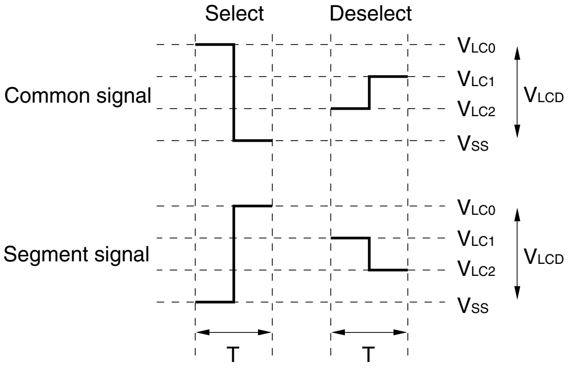
Figure 15-7 shows the common signal waveforms, and Figure 15-8 shows the voltages and phases of the common and segment signals.

Figure 15-7. Common Signal Waveforms



T: One LCD clock period T_F: Frame frequency

Figure 15-8. Voltages and Phases of Common and Segment Signals



T: One LCD clock period

15.7 Display Modes

15.7.1 Three-time slot display example

Figure 15-10 shows how the 8-digit LCD panel having the display pattern shown in Figure 15-9 is connected to the segment signals (S0 to S23) and the common signals (COM0 to COM2) of the μPD789306 or μPD789316 Subseries chip. This example displays data "123456.78" in the LCD panel. The contents of the display data memory (addresses FA00H to FA17H) correspond to this display.

The following description focuses on numeral "6." (6.) displayed in the third digit. To display "6." in the LCD panel, it is necessary to apply the select or deselect voltage to the S6 to S8 pins according to Table 15-5 at the timing of the common signals COM0 to COM2; see Figure 15-9 for the relationship between the segment signals and LCD segments.

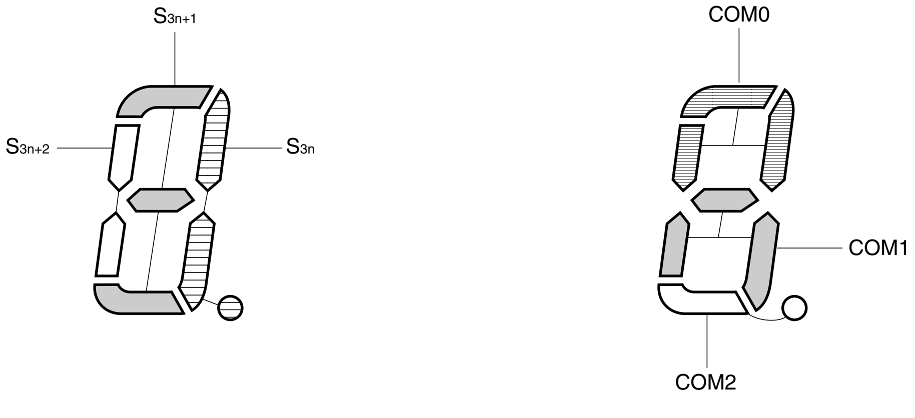
Table 15-5. Select and Deselect Voltages (COM0 to COM2)

Segment Common	S6	S7	S8
COM0	Deselect	Select	Select
COM1	Select	Select	Select
COM2	Select	Select	–

According to Table 15-5, it is determined that the display data memory location (FA06H) that corresponds to S6 must contain x110.

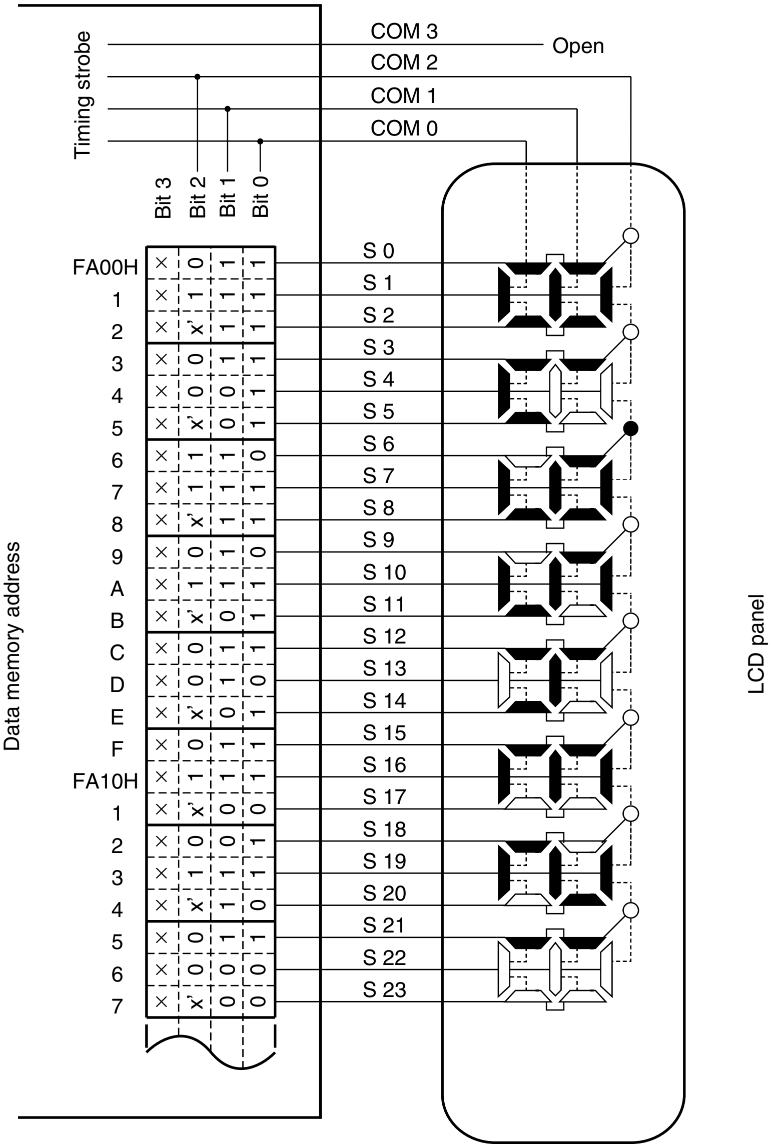
Figure 15-11 shows examples of LCD drive waveforms between the S6 signal and each common signal. When the select voltage is applied to S6 at the timing of COM1 or COM2, an alternate rectangle waveform, $+V_{LCD}/-V_{LCD}$, is generated to turn on the corresponding LCD segment.

Figure 15-9. Three-Time Slot LCD Display Pattern and Electrode Connections



Remark n = 0 to 7

Figure 15-10. Example of Connecting Three-Time Slot LCD Panel

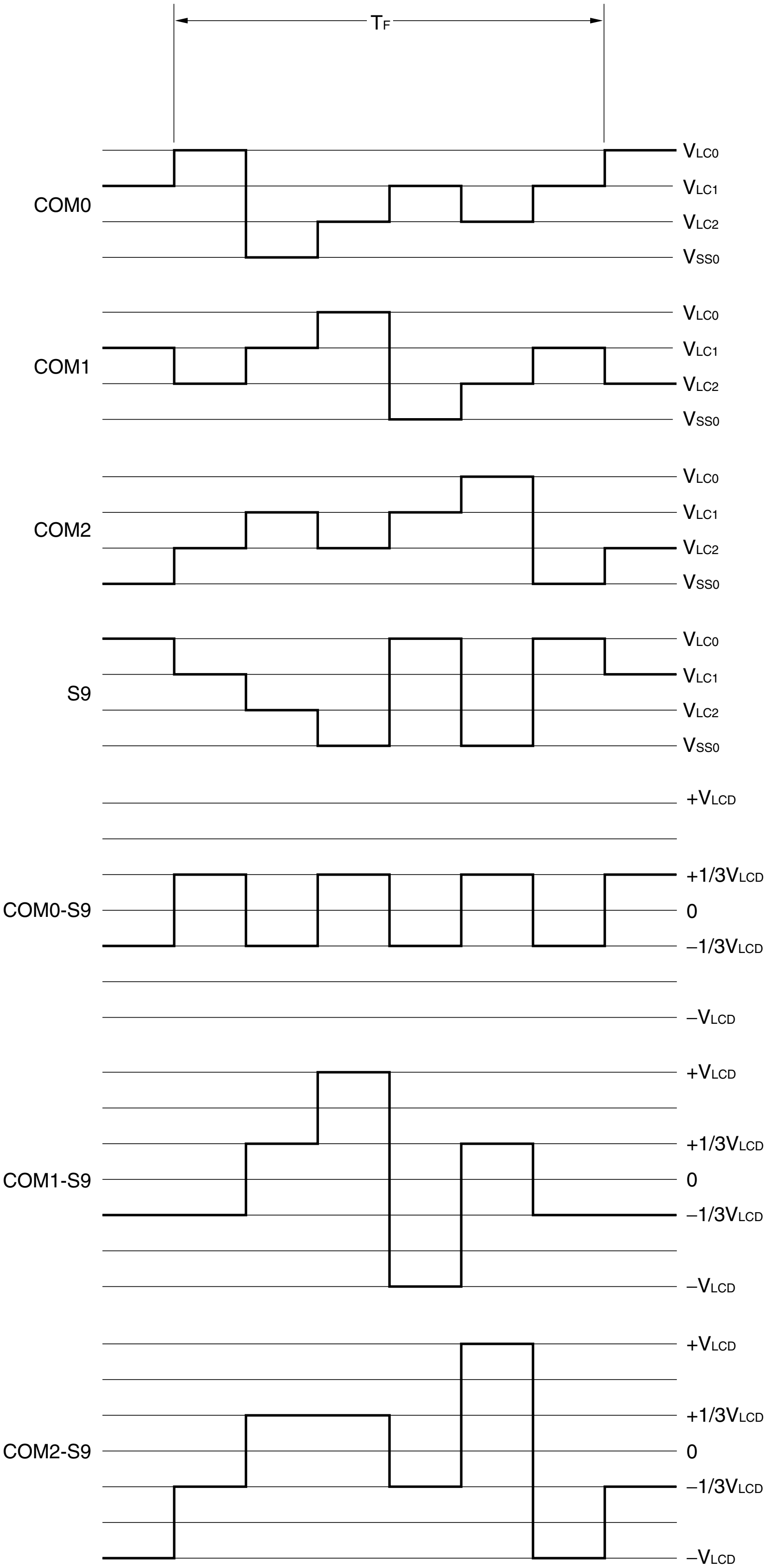


Data memory address	Bit 3	Bit 2	Bit 1	Bit 0	
FA00H	×	0	1	1	S 0
1	×	1	1	1	S 1
2	×	x’	1	1	S 2
3	×	0	1	1	S 3
4	×	0	0	1	S 4
5	×	x’	0	1	S 5
6	×	1	1	0	S 6
7	×	1	1	1	S 7
8	×	x’	1	1	S 8
9	×	0	1	0	S 9
A	×	1	1	1	S 10
B	×	x’	0	1	S 11
C	×	0	1	1	S 12
D	×	0	1	0	S 13
E	×	x’	0	1	S 14
F	×	0	1	1	S 15
FA10H	×	1	1	1	S 16
1	×	x’	0	0	S 17
2	×	0	0	1	S 18
3	×	1	1	1	S 19
4	×	x’	1	0	S 20
5	×	0	1	1	S 21
6	×	0	0	0	S 22
7	×	x’	0	0	S 23

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 of the three-time slot mode being used.

Figure 15-11. Three-Time Slot LCD Drive Waveform Examples



15.7.2 Four-time slot display example

Figure 15-13 shows how the 12-digit LCD panel having the display pattern shown in Figure 15-12 is connected to the segment signals (S0 to S23) and the common signals (COM0 to COM3) of the μPD789306 or μPD789316 Subseries chip. This example displays data “123456.789012” in the LCD panel. The contents of the display data memory (addresses FA00H to FA17H) correspond to this display.

The following description focuses on numeral “6.” (6.) displayed in the seventh digit. To display “6.” in the LCD panel, it is necessary to apply the select or deselect voltage to the S12 and S13 pins according to Table 15-6 at the timing of the common signals COM0 to COM3; see Figure 15-12 for the relationship between the segment signals and LCD segments.

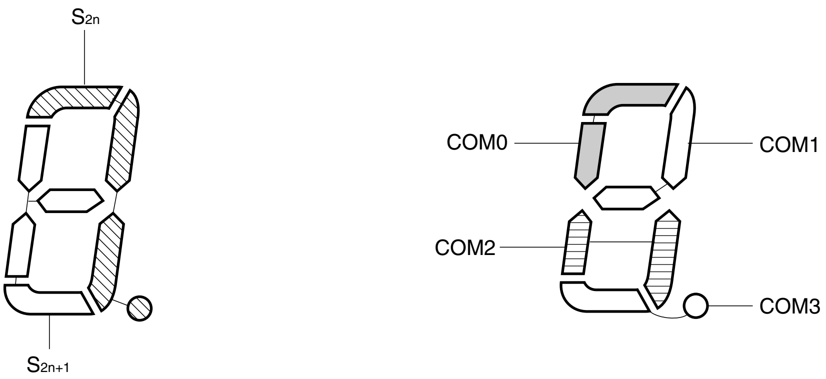
Table 15-6. Select and Deselect Voltages (COM0 to COM3)

Segment / Common	S12	S13
COM0	Select	Select
COM1	Deselect	Select
COM2	Select	Select
COM3	Select	Select

According to Table 15-6, it is determined that the display data memory location (FA0CH) that corresponds to S12 must contain 1101.

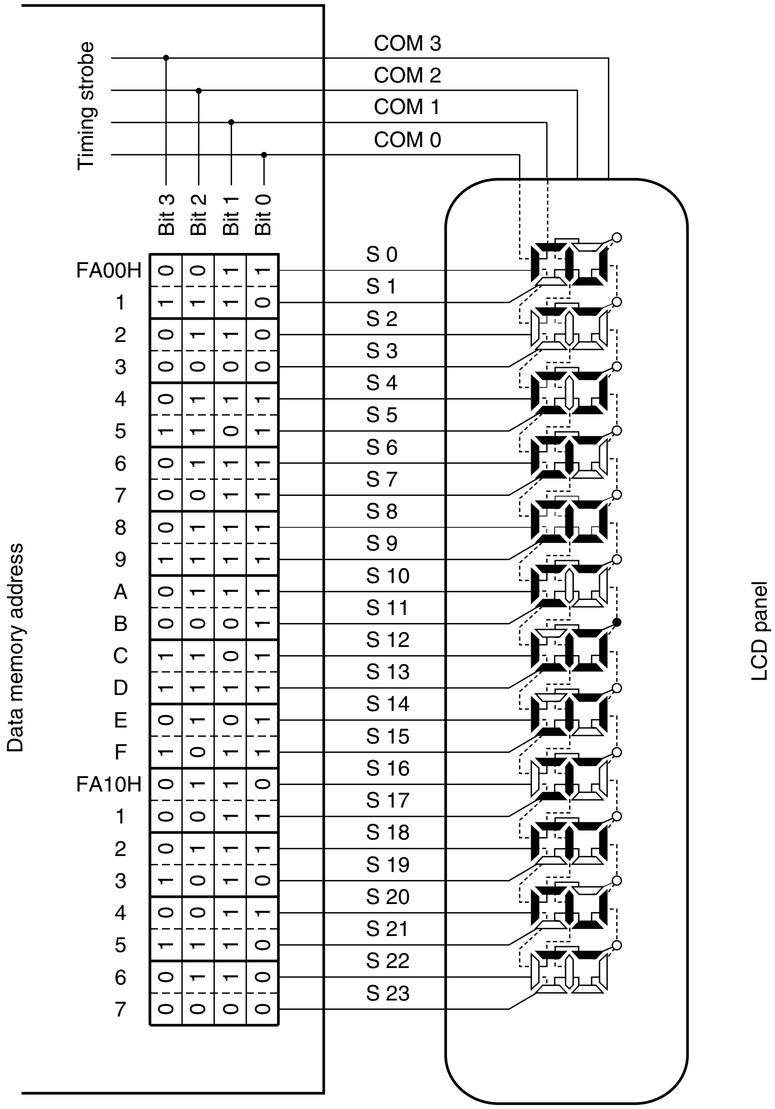
Figure 15-14 shows examples of LCD drive waveforms between the S12 signal and each common signal. When the select voltage is applied to S12 at the timing of COM0, an alternate rectangle waveform, $+V_{LCD}$/$-V_{LCD}$, is generated to turn on the corresponding LCD segment.

Figure 15-12. Four-Time Slot LCD Display Pattern and Electrode Connections



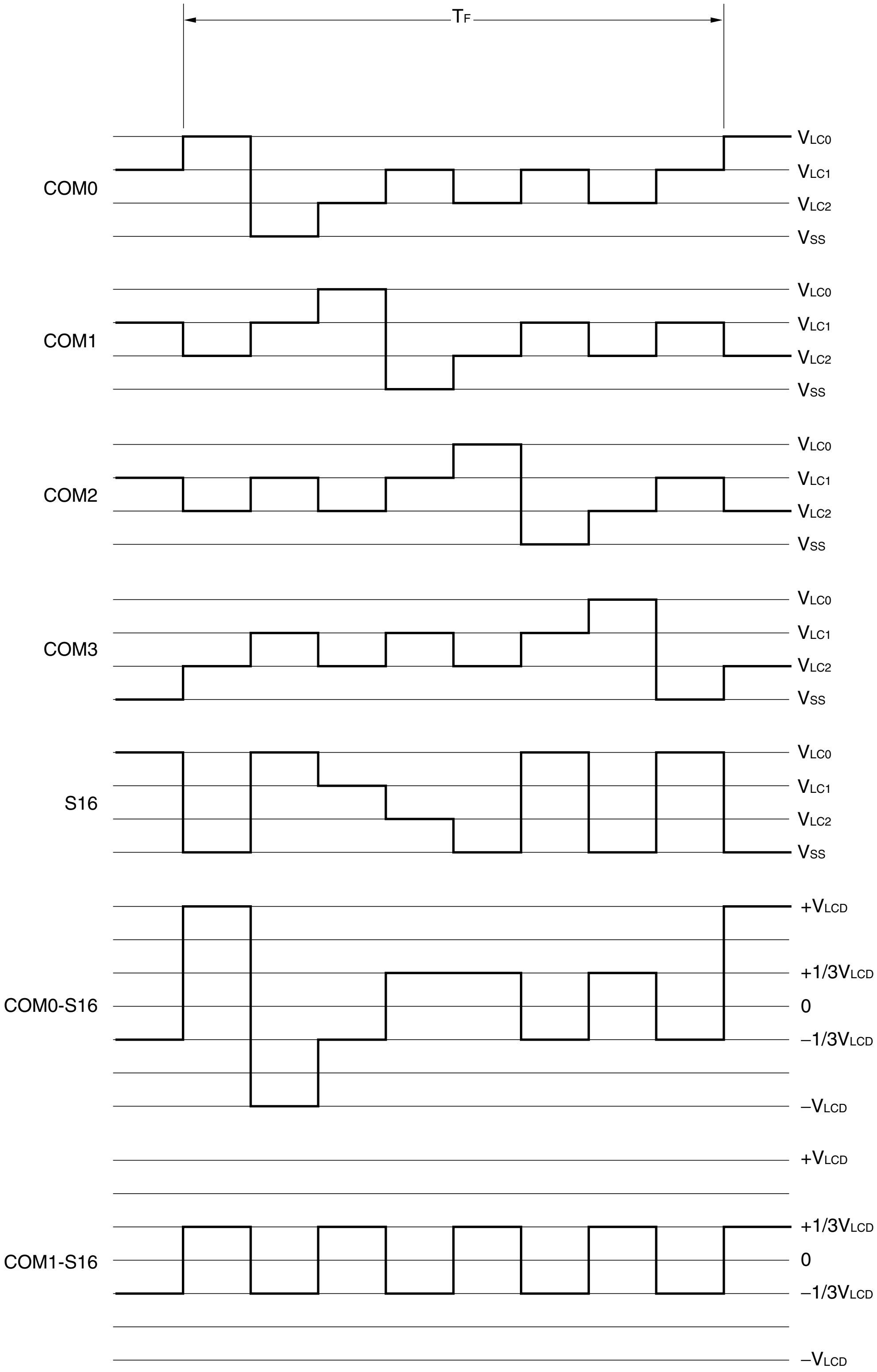
Remark n = 0 to 11

Figure 15-13. Example of Connecting Four-Time Slot LCD Panel



Data memory address	Bit 3	Bit 2	Bit 1	Bit 0
FA00H	0	0	1	1
1	1	1	1	0
2	0	1	1	0
3	0	0	0	0
4	0	1	1	1
5	1	1	0	1
6	0	1	1	1
7	0	0	1	1
8	0	1	1	1
9	1	1	1	1
A	0	1	1	1
B	0	0	0	1
C	1	1	0	1
D	1	1	1	1
E	0	1	0	1
F	1	0	1	1
FA10H	0	1	1	0
1	0	0	1	1
2	0	1	1	1
3	1	0	1	0
4	0	0	1	1
5	1	1	1	0
6	0	1	1	0
7	0	0	0	0

Figure 15-14. Four-Time Slot LCD Drive Waveform Examples



Remark The waveforms for COM2 to S16 and COM3 to S16 are omitted.

15.8 Supplying LCD Drive Voltages V_{LC0}, V_{LC1}, and V_{LC2}

The μPD789306, 789316 Subseries contains a booster circuit (×3 only) to generate a supply voltage to drive the LCD. The internal LCD reference voltage is output from the V_{LC2} pin. A voltage two times higher than that on V_{LC2} is output from the V_{LC1} pin and a voltage three times higher than that on V_{LC2} is output from the V_{LC0} pin.

The LCD reference voltage (V_{LC2}) can be specified by setting LCD boost control register 0 (LCDVA0).

The μPD789306, 789316 Subseries requires an external capacitor (recommended value: 0.47 μF) because it employs a capacitance division method to generate a supply voltage to drive the LCD.

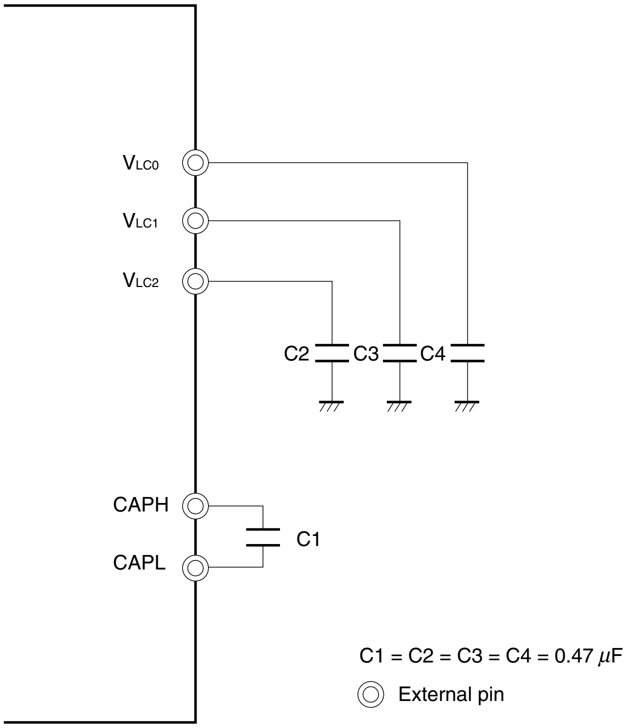
Table 15-7. Output Voltages of V_{LC0} to V_{LC2} Pins

LCDVA0 LCD drive power supply pin	GAIN = 0	GAIN = 1
V_{LC0}	4.5 V	3.0 V
V_{LC1}	3.0 V	2.0 V
V_{LC2} (LCD reference voltage)	1.5 V	1.0 V

Cautions 1. When using the LCD function, do not leave the V_{LC0}, V_{LC1}, and V_{LC2} pins open. Refer to Figure 15-15 for connection.

2. Since the LCD drive voltage is separate from the main power supply, a constant voltage can be supplied regardless of V_{DD} fluctuation.

Figure 15-15. Example of Connecting Pins for LCD Driver



Remark Use a capacitor with as little leakage as possible.
In addition, make C1 a nonpolar capacitor.

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.

5 external and 9 internal interrupt sources are incorporated as maskable interrupts.

16.2 Interrupt Sources and Configuration

A total of 15 non-maskable and maskable interrupts are incorporated as interrupt sources (see **Table 16-1**).

Table 16-1. Interrupt Source List

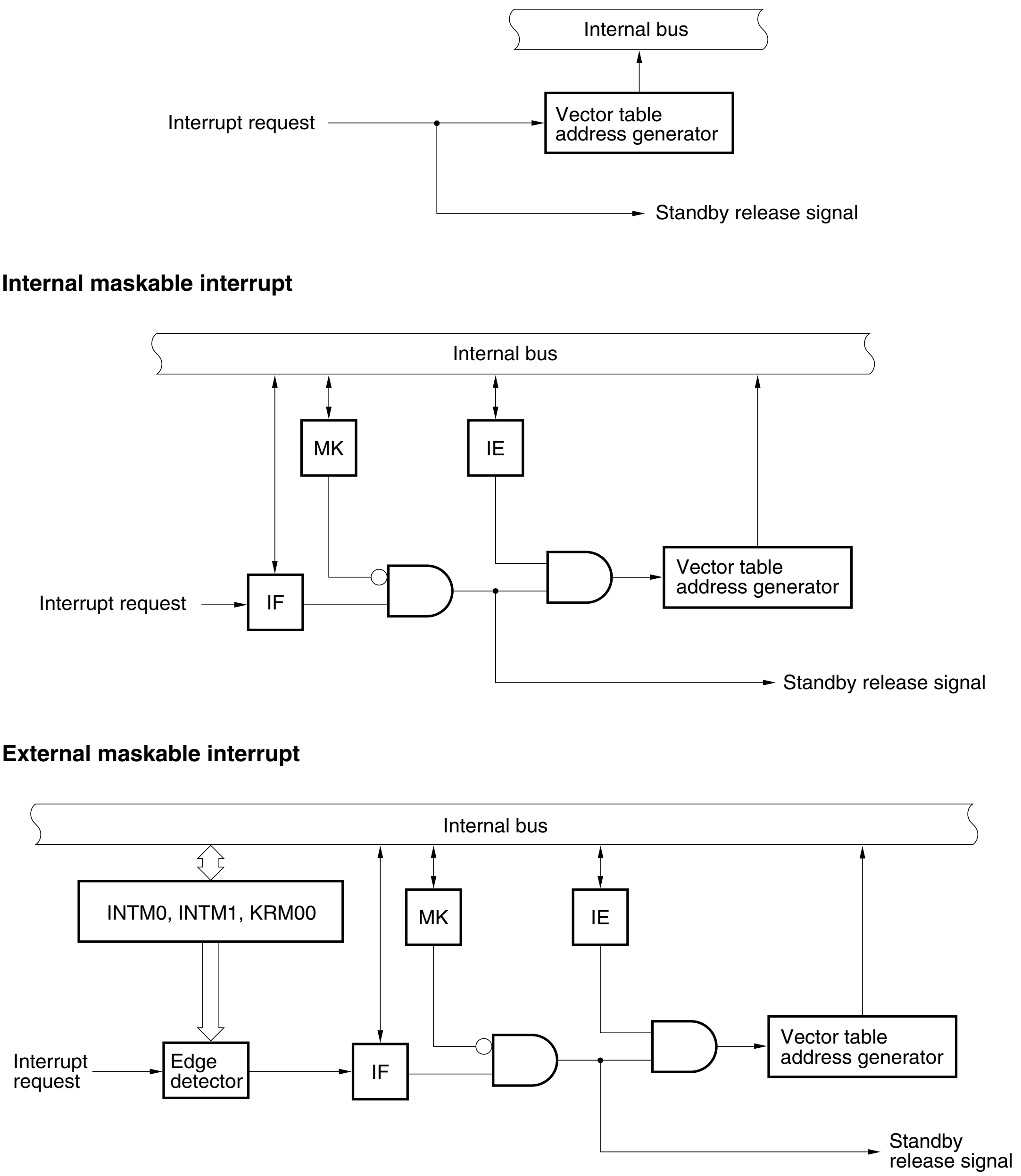
Interrupt Type	Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Non-maskable	–	INTWDT	Watchdog timer overflow (with watchdog timer mode 1 selected)	Internal	0004H	(A)
Maskable	0	INTWDT	Watchdog timer overflow (with interval timer mode selected)			(B)
	1	INTP0	Pin input edge detection	External	0006H	(C)
	2	INTP1			0008H	
	3	INTP2			000AH	
	4	INTP3			000CH	
	5	INTSR20	End of serial interface 20 UART reception	Internal	000EH	(B)
		INTCSI20	End of serial interface 20 3-wire SIO transfer reception			
	6	INTCSI10	End of serial interface 10 3-wire SIO transfer reception		0010H	
	7	INTST20	End of serial interface 20 UART transmission		0012H	
	8	INTWTI	Interval timer interrupt		0014H	
	9	INTTM20	Generation of match signal of 16-bit timer 20		0016H	
	10	INTTM30	Generation of match signal of 8-bit timer 30		0018H	
	11	INTTM40	Generation of match signal of 8-bit timer/event counter 40		001AH	
	12	INTWT	Watch timer interrupt		001EH	
	13	INTKR00	Key return signal detection	External	0020H	(C)

Notes 1. Priority is the priority order when several maskable interrupts are generated at the same time. 0 is the highest order and 13 is the lowest order.

2. Basic configuration types (A) to (C) correspond to (A) to (C) in Figure 16-1.

Remark There are two interrupt sources for the watchdog timer (INTWDT): non-maskable and maskable interrupts (internal). Either one (but not both) should be selected for actual use.

Figure 16-1. Basic Configuration of Interrupt Function



(A) Internal non-maskable interrupt

(B) Internal maskable interrupt

(C) External maskable interrupt

INTP0:	External interrupt mode register 0
INTP1:	External interrupt mode register 1
KRM00:	Key return mode register 00
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 0, 1 (IF0 and IF1)
- Interrupt mask flag registers 0, 1 (MK0 and MK1)
- External interrupt mode registers 0, 1 (INTM0 and INTM1)
- Program status word (PSW)
- Key return mode register 00 (KRM00)

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 Name

Interrupt Request Signal Name	Interrupt Request Flag	Interrupt Mask Flag
INTWDT	WDTIF	WDTMK
INTP0	PIF0	PMK0
INTP1	PIF1	PMK1
INTP2	PIF2	PMK2
INTP3	PIF3	PMK3
INTSR20/INTCSI20	SRIF20	SRMK20
INTCSI10	CSIIF10	CSIMK10
INTST20	STIF20	STMK20
INTWTI	WTIIF	WTIMK
INTTM20	TMIF20	TMMK20
INTTM30	TMIF30	TMMK30
INTTM40	TMIF40	TMMK40
INTWT	WTIF	WTMK
INTKR00	KRIF00	KRMK00

(1) Interrupt request flag registers 0, 1 (IF0 and IF1)

The interrupt request flag is set to 1 when the corresponding interrupt request is generated or an instruction is executed. It is cleared to 0 when an instruction is executed upon acknowledgement of an interrupt request or upon $\overline{\text{RESET}}$ input.

IF0 and IF1 are set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears IF0 and IF1 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	STIF20	CSIIF10	SRIF20	PIF3	PIF2	PIF1	PIF0	WDTIF	FFE0H	00H	R/W

	7	<6>	<5>	4	<3>	<2>	<1>	<0>			
IF1	0	KRIF00	WTIF	0	TMIF40	TMIF30	TMIF20	WTIIF	FFE1H	00H	R/W

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request signal is generated; Interrupt request state

Cautions 1. Bits 4 and 7 of IF1 must be set to 0.

2. The WDTIF flag is R/W enabled only when a watchdog timer is used as an interval timer. If the watchdog timer mode 1 or 2 is used, set the WDTIF flag to 0.

3. Because port 3 has an alternate function as the external interrupt input, when the output level is changed by specifying the output mode of the port function, an interrupt request flag is set. Therefore, the interrupt mask flag should be set to 1 before using the output mode.

4. If an interrupt is acknowledged, the interrupt request flag is automatically cleared before the interrupt routine is entered.

(2) Interrupt mask flag registers 0, 1 (MK0 and MK1)

The interrupt mask flag is used to enable/disable the corresponding maskable interrupt service.

MK0 and MK1 are set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets MK0 and MK1 to FFH.

Figure 16-3. Format of Interrupt Mask Flag Registers

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
MK0	STMK20	CSIMK10	SRMK20	PMK3	PMK2	PMK1	PMK0	WDTMK	FFE4H	FFH	R/W
	7	<6>	<5>	4	<3>	<2>	<1>	<0>			
MK1	1	KRMK00	WTMK	1	TMMK40	TMMK30	TMMK20	WTIMK	FFE5H	FFH	R/W

XXMK	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Cautions 1. Bits 4 and 7 of MK1 must be set to 1.

2. If the WDTMK flag is read when the watchdog timer is used in watchdog timer mode 1 or 2, its value becomes undefined.

3. Because port 3 has an alternate function as the external interrupt input, when the output level is changed by specifying the output mode of the port function, an interrupt request flag is set. Therefore, the interrupt mask flag should be set to 1 before using the output mode.

(3) External interrupt mode register 0 (INTM0)

This register is used to specify a valid edge for INTP0 to INTP2.

INTM0 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears INTM0 to 00H.

Figure 16-4. Format of External Interrupt Mode Register 0

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
INTM0	ES21	ES20	ES11	ES10	ES01	ES00	0	0	FFECH	00H	R/W

ES21	ES20	INTP2 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ES11	ES10	INTP1 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ES01	ES00	INTP0 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

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

2. Before setting the INTM0 register, be sure to set the relevant interrupt mask flag to 1 to disable interrupts.

After that, clear (0) the interrupt request flag, then set the interrupt mask flag to 0 to enable interrupts.

(4) External interrupt mode register 1 (INTM1)

INTM1 is used to specify a valid edge for INTP3.

INTM1 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears INTM1 to 00H.

Figure 16-5. Format of External Interrupt Mode Register 1

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

ES31	ES30	INTP3 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

Cautions 1. Bits 2 to 7 must be set to 0.

2. Before setting INTM1, set PMK3 to 1 to disable interrupts.
After that, clear (0) PIF3, then set PMK3 to 0 to enable interrupts.

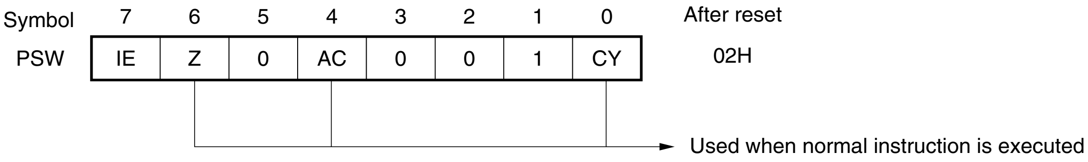
(5) Program status word (PSW)

The program status word is a register used to hold the instruction execution result and the current status for interrupt requests. The IE flag to set maskable interrupt enable/disable is mapped.

Besides 8-bit unit read/write, this register can carry out operations with a bit manipulation instruction and dedicated instructions (EI, DI). When a vectored interrupt is acknowledged, the PSW is automatically saved into a stack, and the IE flag is reset to 0.

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

Figure 16-6. Configuration of Program Status Word



Symbol	7	6	5	4	3	2	1	0	After reset
PSW	IE	Z	0	AC	0	0	1	CY	02H

IE	Interrupt acknowledgement enabled/disabled
0	Disabled
1	Enabled

(6) Key return mode register 00 (KRM00)

This register sets the pin that detects a 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 clears KRM00 to 00H.

Figure 16-7. Format of Key Return Mode Register 00

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
KRM00	0	0	0	0	0	0	0	KRM000	FFF5H	00H	R/W

KRM000	Key return signal detection control
0	No detection
1	Detection (detecting falling edge of port 0)

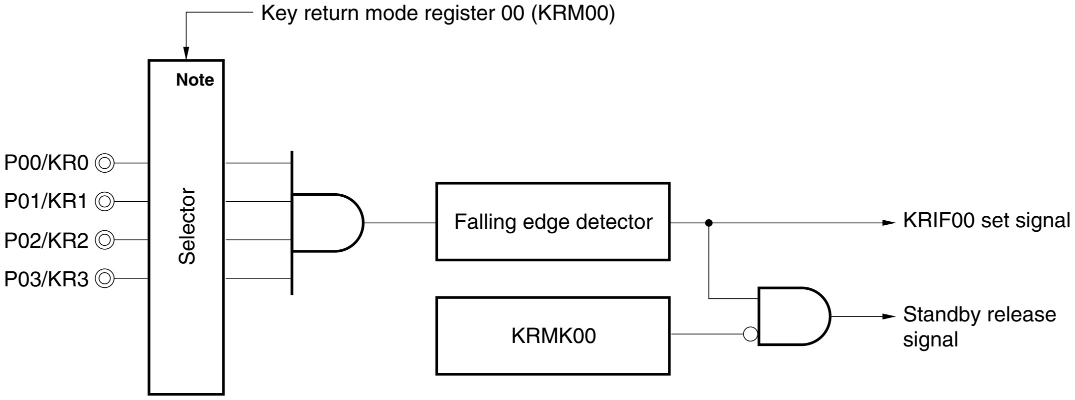
Cautions 1. Bits 1 to 7 must be set to 0.

2. Before setting KRM00, always set bit 6 of MK1 (KRMK00 = 1) to disable interrupts. After setting KRM00, clear KRMK00 after clearing bit 6 of IF1 (KRIF00 = 0) to enable interrupts.

3. When P00 to P03 are in input mode, on-chip pull-up resistors are connected to P00 to P03 by the setting of KRM000. After switching to output mode, the on-chip pull-up resistors are cut off. However, key return signal detection continues.

4. The key return signal cannot be detected while even one of the pins that specify detection of the key return signal is low, even if a falling edge is generated at other key return pins.

Figure 16-8. Block Diagram of Falling Edge Detector



Note Selector that selects the pin used for falling edge input

16.4 Interrupt Servicing Operation

16.4.1 Non-maskable interrupt request acknowledgment operation

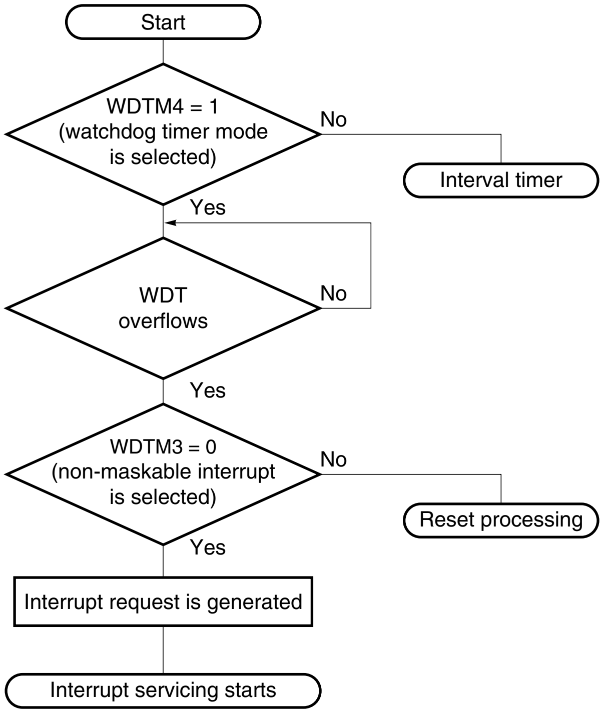
The non-maskable interrupt request is unconditionally acknowledged even when interrupts are disabled. It is not subject to interrupt priority control and takes precedence over all other interrupts.

When the non-maskable interrupt request is acknowledged, PSW and PC are saved to the stack in that order, the IE flag is reset to 0, the contents of the vector table are loaded to the PC, and then program execution branches.

Figure 16-9 shows the flow from non-maskable interrupt request generation to acknowledgement, Figure 16-10 shows the timing of non-maskable interrupt acknowledgement, and Figure 16-11 shows the acknowledgement 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 16-9. Flow from Generation of Non-Maskable Interrupt Request to Acknowledgment



WDTM: Watchdog timer mode register
WDT: Watchdog timer

Figure 16-10. Timing of Non-Maskable Interrupt Request Acknowledgment

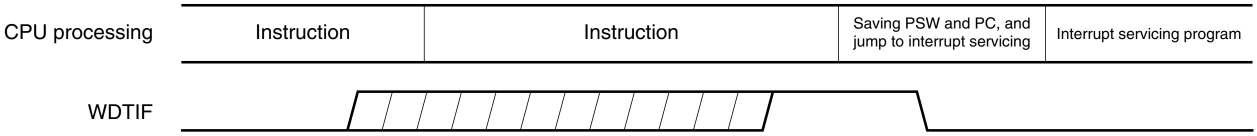
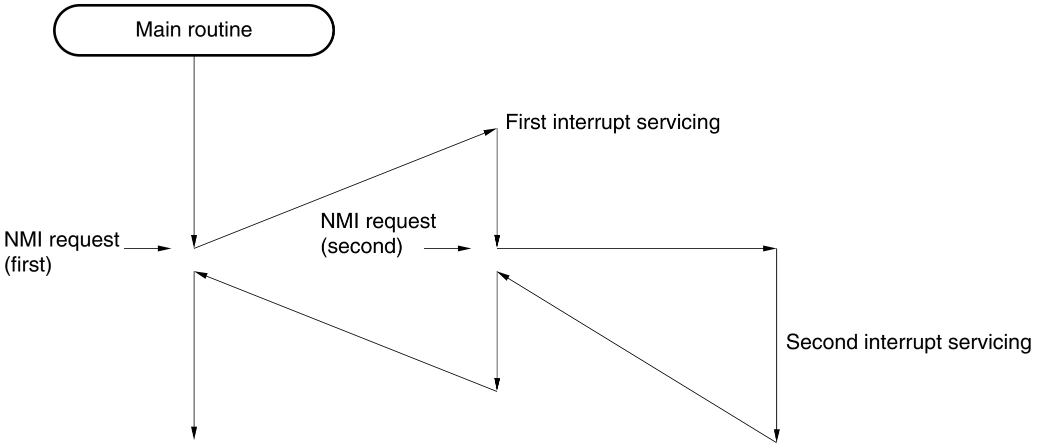


Figure 16-11. Non-Maskable Interrupt Request Acknowledgment



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 16-13 and 16-14 for the timing of interrupt request acknowledgement.

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 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.

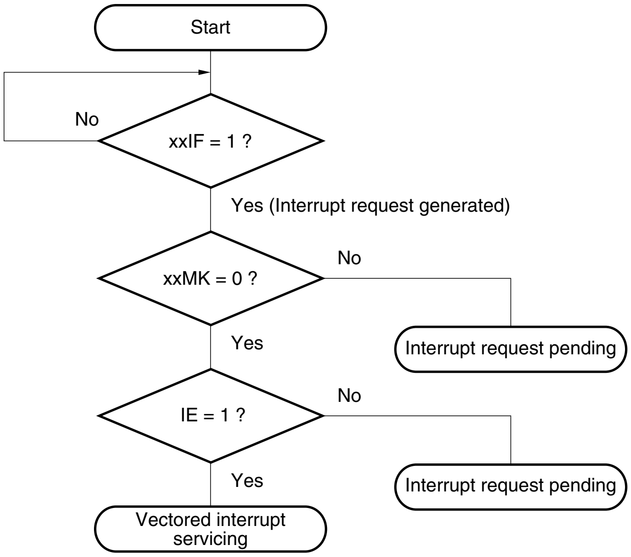
A pending interrupt is acknowledged when the status where it can be acknowledged is set.

Figure 16-12 shows the algorithm of interrupt request acknowledgement.

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, and the data in the vector table determined for each interrupt request is loaded to the PC, and execution branches.

To return from interrupt servicing, use the RETI instruction.

Figure 16-12. Interrupt Request Acknowledgment Program Algorithm

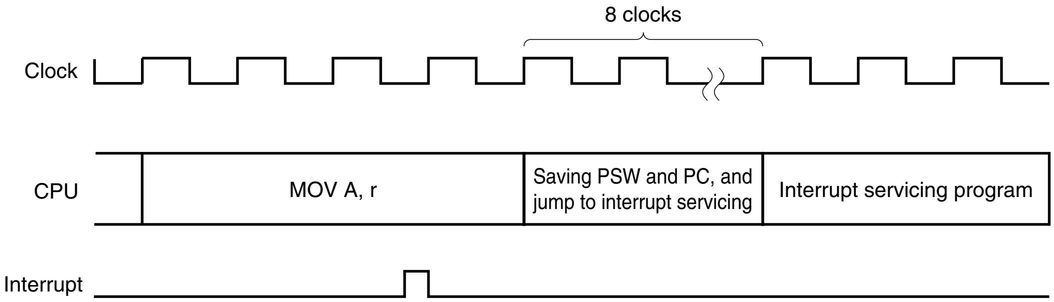

Start
No
xxIF = 1 ?
Yes (Interrupt request generated)
xxMK = 0 ?
No
Yes
Interrupt request pending
IE = 1 ?
No
Yes
Interrupt request pending
Vectored interrupt servicing

xxIF: Interrupt request flag

xxMK: Interrupt mask flag

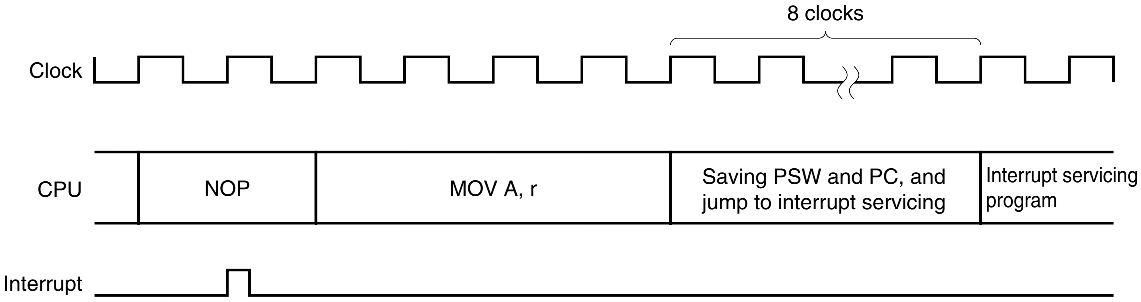
IE: Flag to control maskable interrupt request acknowledgement (1 = enable, 0 = disable)

Figure 16-13. Interrupt Request Acknowledgment Timing (Example: MOV A, r)


8 clocks
Clock
CPU
MOV A, r
Saving PSW and PC, and jump to interrupt servicing
Interrupt servicing program
Interrupt

If the interrupt request has generated an interrupt request flag (XXIF) 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 16-13 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 16-14. Interrupt Request Acknowledgment Timing (When Interrupt Request Flag Is Generated in Final Clock Under Execution)


8 clocks
Clock
CPU
NOP
MOV A, r
Saving PSW and PC, and jump to interrupt servicing
Interrupt servicing program
Interrupt

If the interrupt request flag (XXIF) is generated in the final clock of the instruction, interrupt request acknowledgment processing will begin after execution of the next instruction is complete.

Figure 16-14 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.

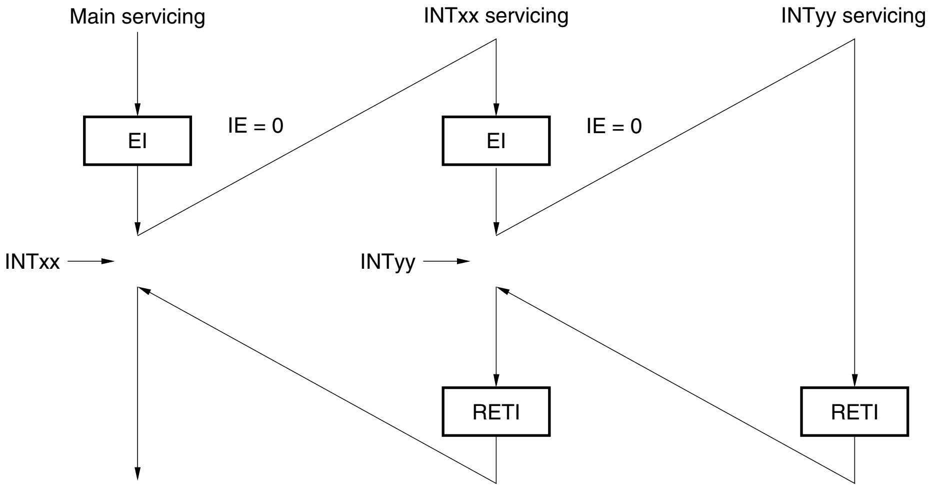
Caution When interrupt request flag registers 0 and 1 (IF0 and IF1), or interrupt mask flag registers 0 and 1 (MK0 and MK1) are being accessed, interrupt requests will be held pending.

16.4.3 Multiple interrupt servicing

Multiple interrupts, in which another interrupt request is acknowledged while an interrupt request being serviced, can be serviced 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**).

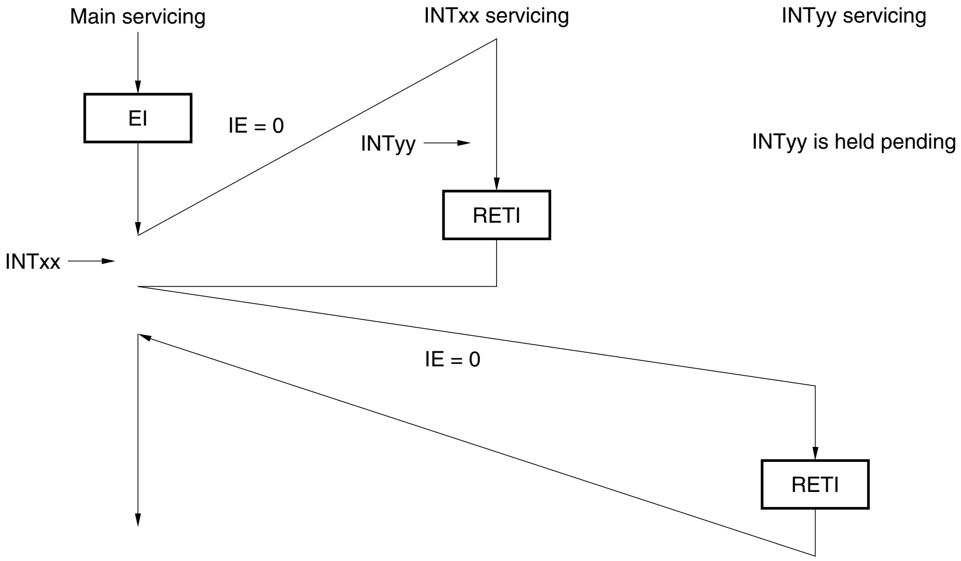
Figure 16-15. Example of Multiple Interrupts

Example 1. Acknowledging multiple interrupts



The interrupt request INTyy is acknowledged during the servicing of interrupt INTxx and multiple interrupts are performed. Before each interrupt request is acknowledged, the EI instruction is issued and the interrupt request is enabled.

Example 2. Multiple interrupts are not performed because interrupts are disabled



Because interrupt requests are disabled (the EI instruction has not been issued) in the interrupt INTxx servicing, the interrupt request INTyy is not acknowledged and multiple interrupts are 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 0, 1 (IF0 and IF1)
- Instructions that manipulate interrupt mask flag registers 0, 1 (MK0 and MK1)

CHAPTER 17 STANDBY FUNCTION

17.1 Standby Function and Configuration

17.1.1 Standby function

The standby function is 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 the low voltage (V_{DD} = 1.8 V). Therefore, this mode is useful for retaining the contents of the data memory at an extremely low current.

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 latch of the I/O ports and output buffer 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 until oscillation stabilizes is controlled with the oscillation stabilization time select register (OSTS)[Note].

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, after $\overline{\text{RESET}}$ input.

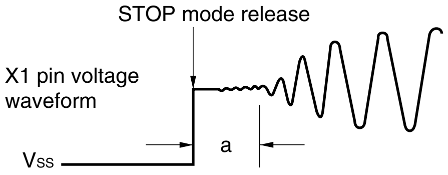
Note The μPD789306 Subseries only.
The μPD789316 Subseries does not have an oscillation stabilization time select register. The oscillation stabilization time for the μPD789316 Subseries is fixed at 2^{7}/fcc.

Figure 17-1. Format of Oscillation Stabilization Time Select 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 (ceramic/crystal oscillation)

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 status in the HALT mode is shown in the following table.

Table 17-1. HALT Mode Operating Status

Item	HALT Mode Operation Status While The Main System Clock Is Running		HALT Mode Operation Status While The Subsystem Clock Is Running	
	While the subsystem clock is running	While the subsystem clock is not running	While the main system clock is running	While the main system clock is not running
Main system clock	Oscillation enabled			Oscillation stopped
CPU	Operation stopped			
Port (output latch)	Remains in the state existing before the selection of HALT mode.			
16-bit timer 20	Operation enabled			Operation stopped
8-bit timer 30	Operation enabled			Operation enabled[Note 1]
8-bit timer 40				Operation enabled[Note 2]
Watch timer	Operation enabled	Operation enabled[Note 3]	Operation enabled	Operation enabled[Note 4]
Watchdog timer	Operation enabled		Operation stopped	
Serial interface 10	Operation enabled			Operation enabled[Note 5]
Serial interface 20				
LCD controller/driver	Operation enabled	Operation enabled[Note 3]	Operation enabled	Operation enabled[Note 4]
External interrupt	Operation enabled[Note 6]			

Notes 1. Operation is enabled only when input signal from timer 40 (timer 40 operation is enabled) is selected as the count clock.

2. Operation is enabled when TMI40 is selected as the count clock.

3. Operation is enabled while the main system clock is selected.

4. Operation is enabled while the subsystem clock is selected.

5. Operation is enabled only when external clock is selected.

6. Maskable interrupt that is not masked

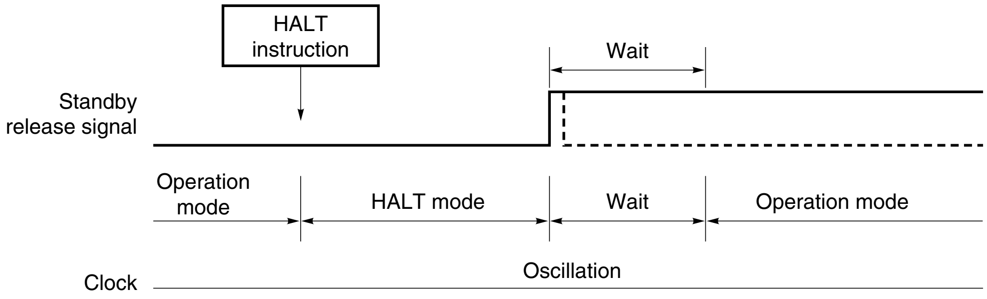
(2) Releasing HALT mode

The HALT mode can be released by the following three types of sources:

(a) Releasing by unmasked interrupt request

The HALT mode is released by an unmasked interrupt request. In this case, if the interrupt is enabled to be acknowledged, vectored interrupt processing is performed. If the interrupt is disabled, the instruction at the next address is executed.

Figure 17-2. Releasing HALT Mode by Interrupt



Remarks 1. The broken line indicates the case where the interrupt request that has released the standby mode is acknowledged.

2. The wait time is as follows:

• When vectored interrupt processing is performed:	9 to 10 clocks
• When vectored interrupt processing is not performed:	1 to 2 clocks

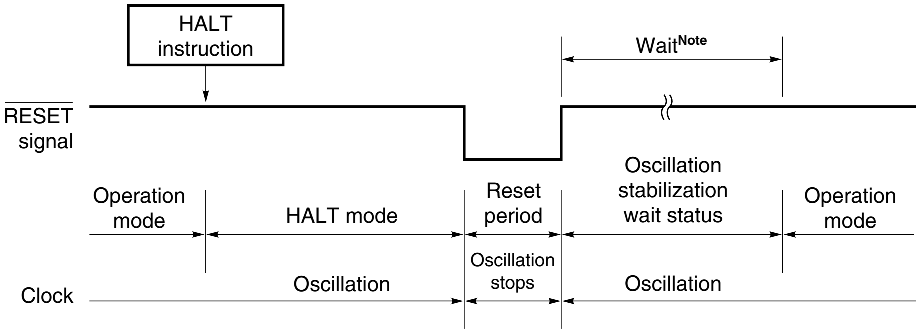
(b) Releasing by non-maskable interrupt request

The HALT mode is released regardless of whether the interrupt is enabled or disabled, and vectored interrupt processing is performed.

(c) Releasing 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



Note For the μPD789306 Subseries, 2^{15}/fx: 6.55 ms (@ fx = 5.0 MHz operation)
For the μPD789316 Subseries, 2^{7}/fcc: 32 μs (@ fcc = 4.0 MHz operation)

Remarks 1. fx: Main system clock oscillation frequency (ceramic/crystal oscillation)
2. fcc: Main system clock oscillation frequency (RC oscillation)

Table 17-2. Operation After Releasing HALT Mode

Releasing Source	MKxx	IE	Operation
Maskable interrupt request	0	0	Executes next address instruction
	0	1	Executes interrupt servicing
	1	x	Retains HALT mode
Non-maskable interrupt request	–	x	Executes interrupt servicing
$\overline{\text{RESET}}$ input	–	–	Reset processing

x: don't care

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 select register (OSTS) elapses, and then an operation mode is set.

The operation status in the STOP mode is shown in the following table.

Table 17-3. STOP Mode Operating Status

Item	STOP Mode Operation Status While The Main System Clock Is Running	
	While the subsystem clock is running	While the subsystem clock is not running
Main system clock	Oscillation stopped	
CPU	Operation stopped	
Port (output latch)	Remains in the state existing before the selection of STOP mode.	
16-bit timer 20	Operation stopped	
8-bit timer 30	Operation enabled[Note 1]	
8-bit timer 40	Operation enabled[Note 2]	
Watch timer	Operation enabled[Note 3]	Operation stopped
Watchdog timer	Operation stopped	
Serial interface 10	Operation enabled[Note 4]	
Serial interface 20		
LCD controller/driver	Operation enabled[Note 3]	Operation stopped
External interrupt	Operation enabled[Note 5]	

Notes 1. Operation is enabled only when input signal from timer 40 (timer 40 operation is enabled) is selected as the count clock.

2. Operation is enabled when TMI40 is selected as the count clock.

3. Operation is enabled while the subsystem clock is selected.

4. Operation is enabled only when external clock is selected.

5. Maskable interrupt that is not masked

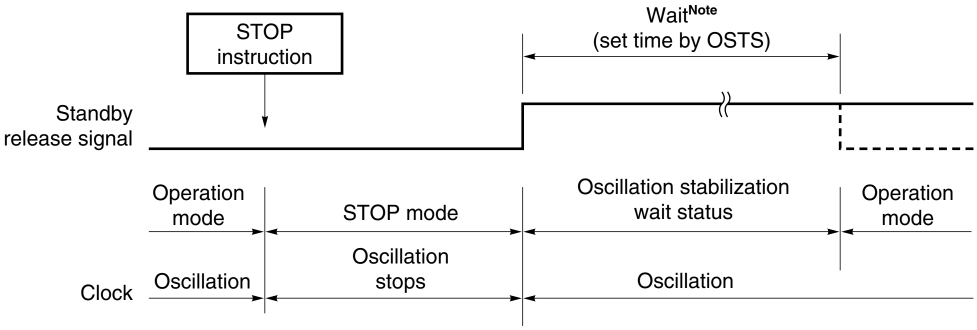
(2) Releasing STOP mode

The STOP mode can be released by the following two types of sources:

(a) Releasing by unmasked interrupt request

The STOP mode can be released by an unmasked interrupt request. In this case, if the interrupt is enabled to be acknowledged, vectored interrupt processing is performed, after the oscillation stabilization time has elapsed. If the interrupt is disabled, the instruction at the next address is executed.

Figure 17-4. Releasing STOP Mode by Interrupt



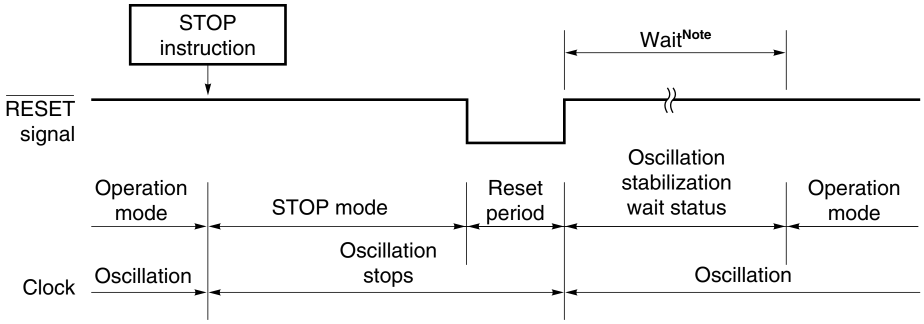
Note The μPD789316 Subseries does not have OSTS and wait is fixed at $2^7/f_{CC}$.

Remark The broken line indicates the case where the interrupt request that has released the standby mode is acknowledged.

(b) Releasing 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



Note For the μPD789306 Subseries, $2^{15}/f_X$: 6.55 ms (@ f_X = 5.0 MHz operation)
For the μPD789316 Subseries, $2^{7}/f_{CC}$: 32 μs (@ f_{CC} = 4.0 MHz operation)

Remarks 1. f_X: Main system clock oscillation frequency (ceramic/crystal oscillation)
2. f_{CC}: Main system clock oscillation frequency (RC oscillation)

Table 17-4. Operation After Releasing STOP Mode

Releasing Source	MKxx	IE	Operation
Maskable interrupt request	0	0	Executes next address instruction
	0	1	Executes interrupt servicing
	1	x	Retains STOP mode
$\overline{\text{RESET}}$ input	–	–	Reset processing

x: 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 runaway 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 has a high impedance during reset input or during oscillation stabilization time just after reset clear.

When a high level is input to the $\overline{\text{RESET}}$ pin, the reset is cleared and program execution is started after the oscillation stabilization time has elapsed. The reset applied by the watchdog timer overflow is automatically cleared after reset, and program execution is started after the oscillation stabilization time has elapsed (see **Figures 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 cleared 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

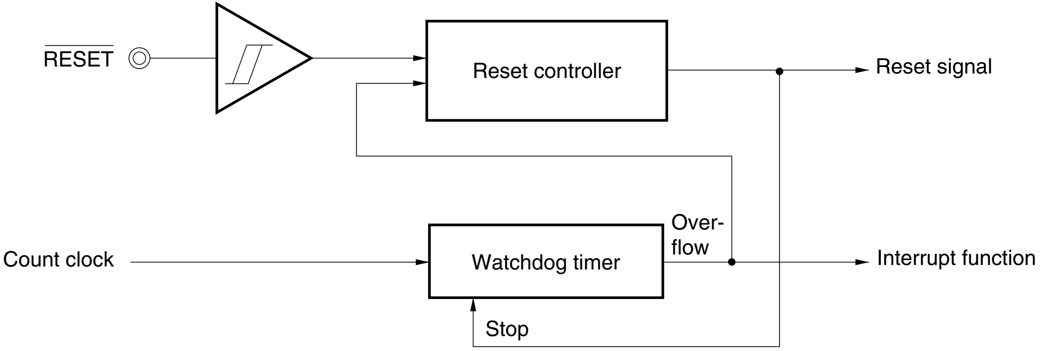


Figure 18-2. Reset Timing by $\overline{\text{RESET}}$ Input

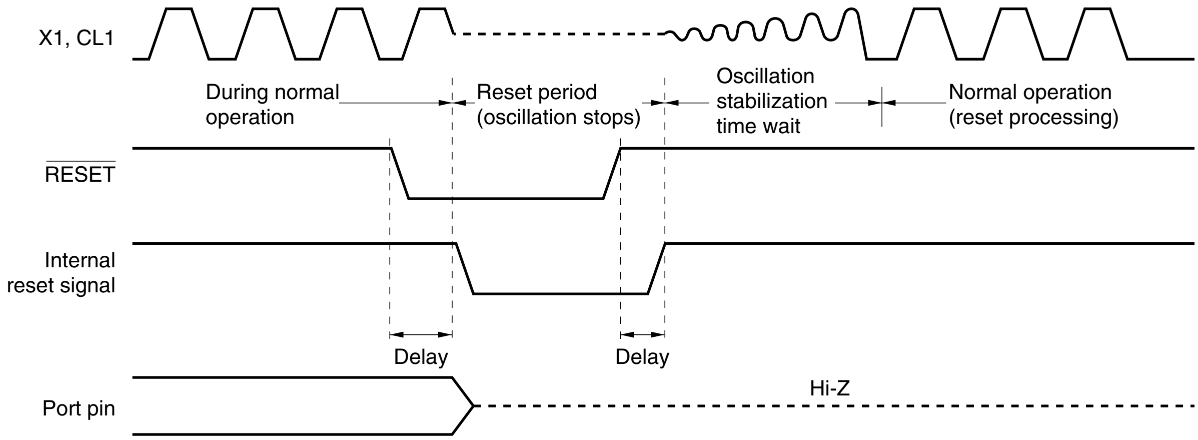

X1, CL1
During normal operation
Reset period (oscillation stops)
Oscillation stabilization time wait
Normal operation (reset processing)
RESET
Internal reset signal
Delay
Delay
Port pin
Hi-Z

Figure 18-3. Reset Timing by Overflow in Watchdog Timer

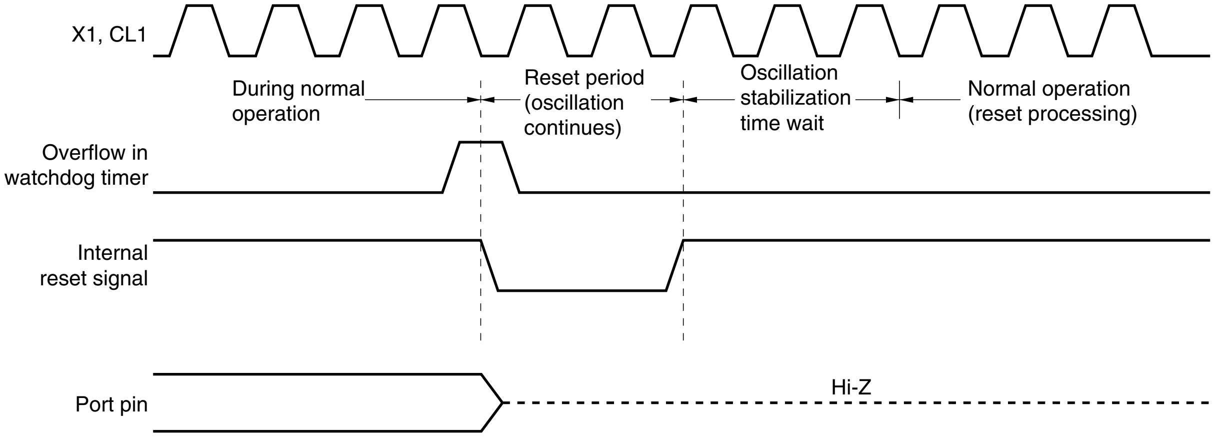

X1, CL1
During normal operation
Reset period (oscillation continues)
Oscillation stabilization time wait
Normal operation (reset processing)
Overflow in watchdog timer
Internal reset signal
Port pin
Hi-Z

Figure 18-4. Reset Timing by $\overline{\text{RESET}}$ Input in STOP Mode

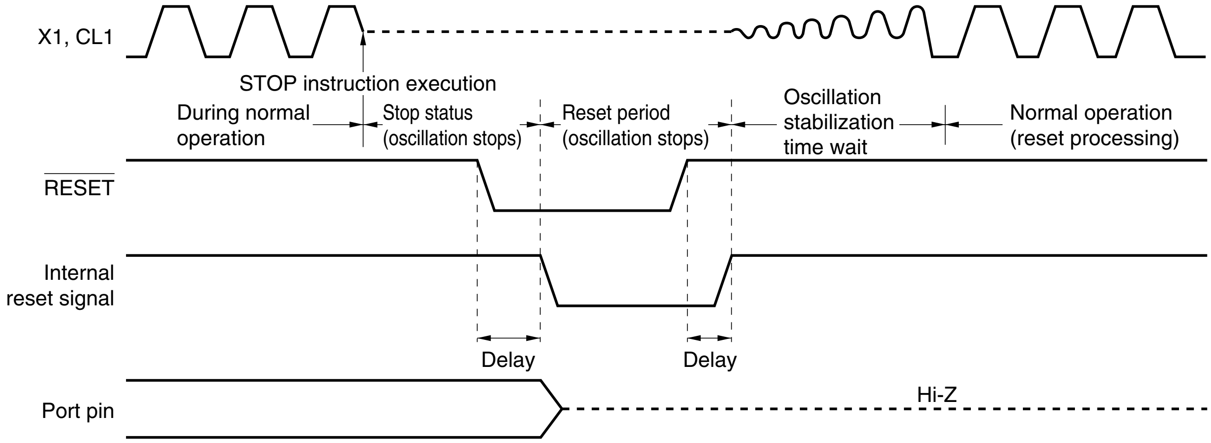

X1, CL1
STOP instruction execution
During normal operation
Stop status (oscillation stops)
Reset period (oscillation stops)
Oscillation stabilization time wait
Normal operation (reset processing)
RESET
Internal reset signal
Delay
Delay
Port pin
Hi-Z

Table 18-1. Hardware Status After Reset

Hardware		Status After Reset
Program counter (PC)[Note 1]		The contents of reset vector tables (0000H and 0001H) are set.
Stack pointer (SP)		Undefined
Program status word (PSW)		02H
RAM	Data memory	Undefined[Note 2]
	General-purpose register	Undefined[Note 2]
Port (P0 to P3, P5) (Output latch)		00H
Port mode register (PM0 to PM3, PM5)		FFH
Pull-up resistor option register (PU0, PUB2, PUB3)		00H
Processor clock control register (PCC)		02H
Suboscillation mode register (SCKM)		00H
Subclock control register (CSS)		00H
Oscillation stabilization time select register (OSTS)[Note 3]		04H
16-bit timer 20	Timer counter (TM20)	0000H
	Compare register (CR20)	FFFFH
	Control register (TMC20)	00H
	Capture register (TCP20)	Undefined
8-bit timer 30, 40	Timer counter (TM30, TM40)	00H
	Compare register (CR30, CR40, CRH40)	Undefined
	Mode control register (TMC30, TMC40)	00H
Watch timer	Mode control register (WTM)	00H
Watchdog timer	Clock select register (WDCS)	00H
	Mode register (WDTM)	00H
Serial interface 20	Serial operation mode register (CSIM10, 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
LCD controller/driver	Display mode register (LCDM0)	00H
	Clock control register (LCDC0)	00H
	Voltage amplification control register (LCDVA0)	00H
Interrupt	Request flag register (IF0, IF1)	00H
	Mask flag register (MK0, MK1)	FFH
	External interrupt mode register (INTM0, INTM1)	00H
	Key return mode register (KRM00)	00H

Notes 1. During reset input and oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware remains unchanged after reset.

2. The post-reset values are retained in the standby mode.

3. μPD789306 Subseries only

CHAPTER 19 FLASH MEMORY

The μPD78F9306, 78F9316 are available as the flash memory versions of the μPD789306, 789316 Subseries.

The μPD78F9306 is a version with the internal ROM of the μPD789304, 789306 replaced with flash memory and the μPD78F9316 is a version with the internal ROM of the μPD789314, 789316 replaced with flash memory. The differences between the μPD78F9306, 78F9316 and the mask ROM versions are shown in Table 19-1.

Table 19-1. Differences Between μPD78F9306, 78F9316 and Mask ROM Versions

	Part Number	Flash Memory Version		Mask ROM Version			
Item		μPD78F9306	μPD78F9316	μPD789304	μPD789306	μPD789314	μPD789316
Internal memory	ROM	16 KB		8 KB	16 KB	8 KB	16 KB
	High-speed RAM	512 bytes					
	LCD display RAM	24 × 4 bits					
System clock		Ceramic/ crystal oscillation	RC oscillation	Ceramic/crystal oscillation		RC oscillation	
IC pin		Not provided		Provided			
V_{PP} pin		Provided		Not provided			
Pull-up resistor		19 (software control: 19)		23 (software control: 19, mask option control: 4)			
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 μPD78F9306, 78F9316 mounted on the target system (on-board). A flash memory program adapter (FA adapter), which is a target board used exclusively for programming, is also provided.

Remark FL-PR3, FL-PR4, and the program adapter are products made by Naito Densei Machida Mfg. Co., Ltd. (TEL +81-45-475-4191).

Programming using flash memory has the following advantages.

- Software can be modified after the microcontroller is solder-mounted on the target system.
- Distinguishing software facilities small-quantity, varied model production
- Easy data adjustment when starting mass production

19.1.1 Programming environment

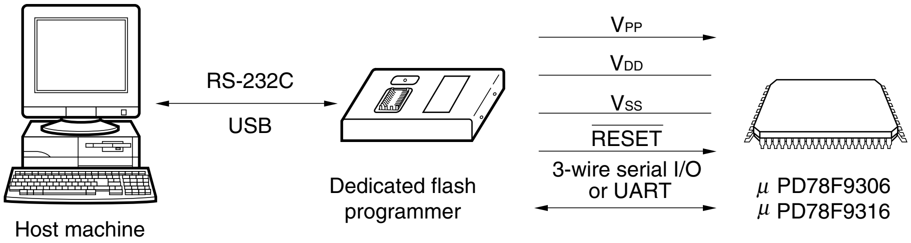
The following shows the environment required for μPD78F9306, 78F9316 flash memory programming.

When Flashpro III (part no. FL-PR3, PG-FP3) or Flashpro IV (part no. FL-PR4, PG-FP4) is used as a dedicated flash programmer, a host machine is required to control the dedicated flash programmer. Communication between the host machine and flash programmer is performed via RS-232C/USB (Rev. 1.1).

For details, refer to the manuals for Flashpro III/Flashpro IV.

Remark USB is supported by Flashpro IV only.

Figure 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 μPD78F9306, 78F9316.

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-wire, sync.)	100 Hz to 1.25 MHz[Note 2]	1, 2, 4, 5 MHz[Notes 2, 3]	1 to 5 MHz[Note 2]	1.0	SI10/P22 SO10/P21 $\overline{\text{SCK10}}$/P20	0
UART	UART ch-0 (Async.)	4,800 to 76,800 bps[Notes 2, 4]	5 MHz[Note 5]	4.91 or 5 MHz[Note 2]	1.0	RxD20/SI20/P25 TxD20/SO20/P24	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. 2 or 4 MHz only for Flashpro III

4. Because signal wave slew also affects UART communication, in addition to the baud rate error, thoroughly evaluate the slew and baud rate error.

5. Only for Flashpro IV. However, when using Flashpro III, be sure to select the clock of the resonator on the board. UART cannot be used with the clock supplied by Flashpro III.

Figure 19-2. Communication Mode Selection Format

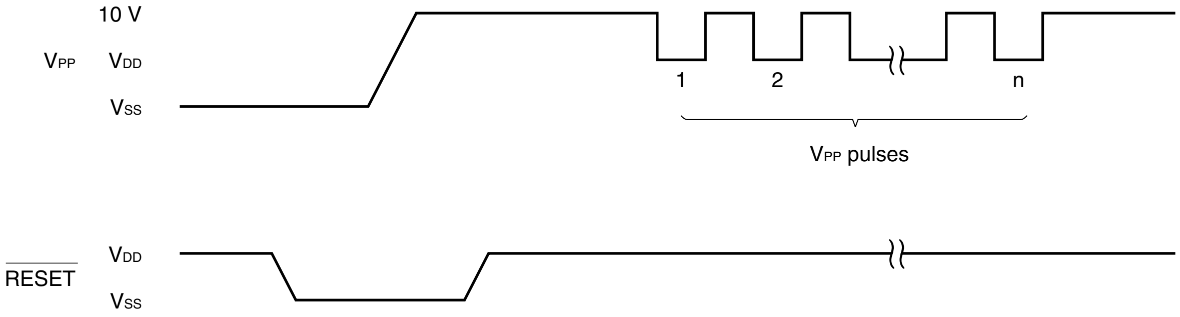
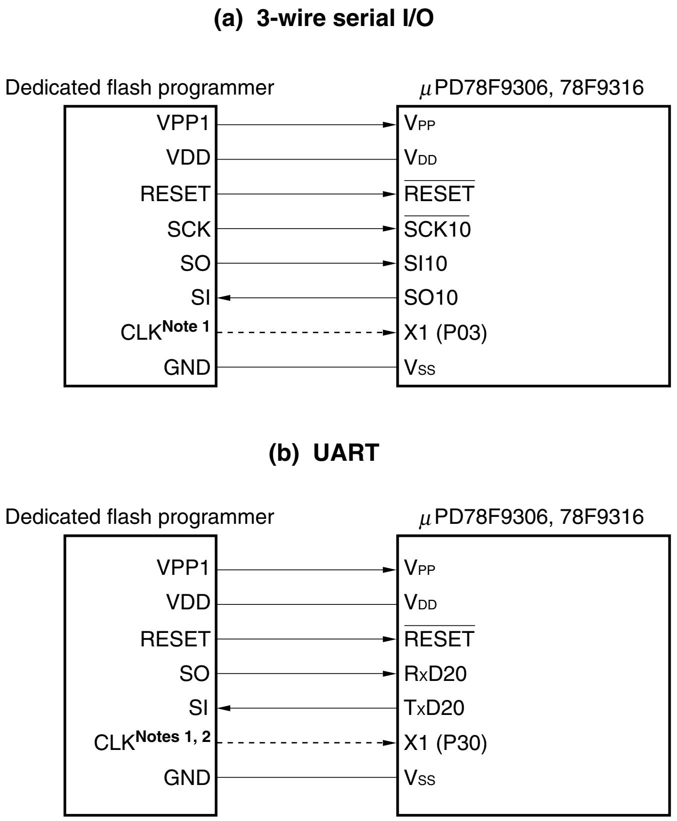


Figure 19-3. Example of Connection with Dedicated Flash Programmer



(a) 3-wire serial I/O

(b) UART

Notes 1. When the system clock is supplied from the dedicated flash programmer to the μ PD78F9306, connect the CLK pin with X1 pin and disconnect the on-board resonator. When using the clock of the on-board resonator, do not connect the CLK pin.

When using the μ PD78F9316, be sure to connect the CLK pin with P03 pin and the system clock is supplied from the dedicated flash programmer.

2. When using UART with Flashpro III, the clock of the resonator connected to the X1 pin must be used, so do not connect to the CLK pin.

Caution The V_{DD} pin, if already connected to the power supply, must be connected to the VDD pin of the dedicated flash programmer. When using the power supply connected to the V_{DD} pin, supply voltage before starting programming.

If Flashpro III/Flashpro IV is used as a dedicated flash programmer, the following signals are generated for the μPD78F9306, 78F9316. 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	UART
VPP1	Output	Write voltage	V_{PP}	◎	◎
VPP2	–	–	–	×	×
VDD	I/O	V_{DD} voltage generation/ voltage monitoring	V_{DD}	◎ Note	◎ Note
GND	–	Ground	V_{SS}	◎	◎
CLK	Output	Clock output	X1 (μPD78F9316)	○	○
			P03 (μPD78F9316)	◎	◎
RESET	Output	Reset signal	$\overline{\text{RESET}}$	◎	◎
SI	Input	Receive signal	SO10/TxD20	◎	◎
SO	Output	Transmit signal	SI10/RxD20	◎	◎
SCK	Output	Transfer clock	$\overline{\text{SCK10}}$	◎	×
HS	Input	Handshake signal	–	×	×

Note V_{DD} voltage must be supplied before programming is started.

Remark ◎**:** Pin must be connected.
○**:** If the signal is supplied on the target board, pin does not need to be connected.
×**:** Pin does not need to be connected.

19.1.3 On-board pin connections

When programming on the target system, provide a connector on the target system to connect to the dedicated flash programmer.

There may be cases in which an on-board function that switches from the normal operation mode to flash memory programming mode is required.

<VPP pin>

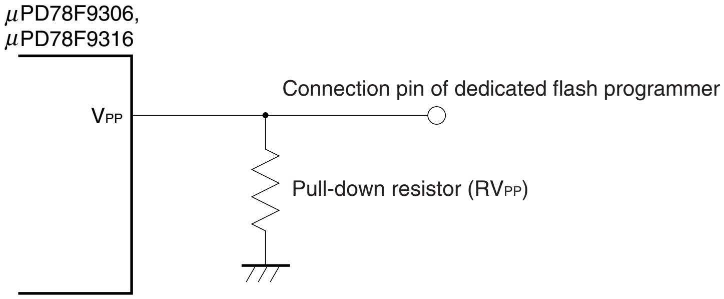
Input 0 V to the VPP pin in the normal operation mode. A write voltage of 10.0 V (TYP.) is supplied to the VPP pin in the flash memory programming mode. Therefore, connect the VPP pin using method (1) or (2) below.

(1) Connect a pull-down resistor of RVPP = 10 kΩ to the VPP pin.

(2) Set the jumper on the board to switch the input of VPP pin to the programmer side or directly to GND.

The following shows an example of VPP pin connection.

Figure 19-4. VPP Pin Connection Example


μPD78F9306,
μPD78F9316
VPP
Connection pin of dedicated flash programmer
Pull-down resistor (RVPP)

<Serial interface pins>

The following shows the pins used by each serial interface.

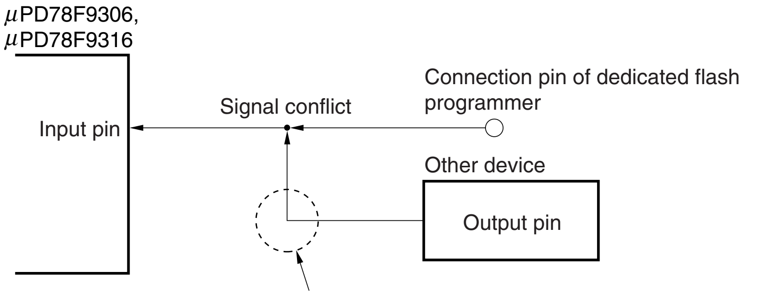
Serial Interface	Pins Used
3-wire serial I/O	SI10, SO10, $\overline{\text{SCK10}}$
UART	RxD20, TxD20

Note that signal conflict or malfunction of other devices may occur when an on-board serial interface pin that is connected to another device is connected to the dedicated flash programmer.

(1) Signal conflict

A signal conflict occurs if the dedicated flash programmer (output) is connected to a serial interface pin (input) connected to another device (output). To prevent this signal conflict, isolate the connection with the other device or put the other device in the output high impedance status.

Figure 19-5. Signal Conflict (Serial Interface Input Pin)

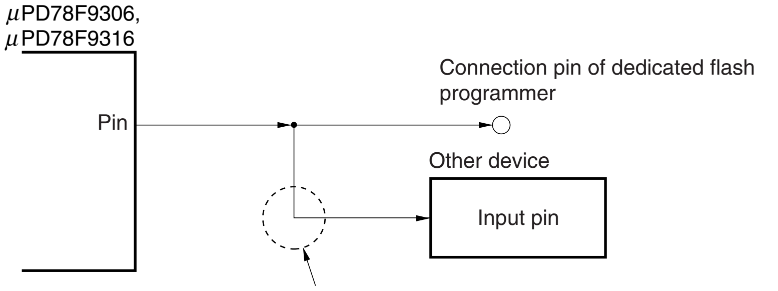


In the flash memory programming mode, the signal output by another device and the signal sent by the dedicated flash programmer conflict. To prevent this, isolate the signal on the device side.

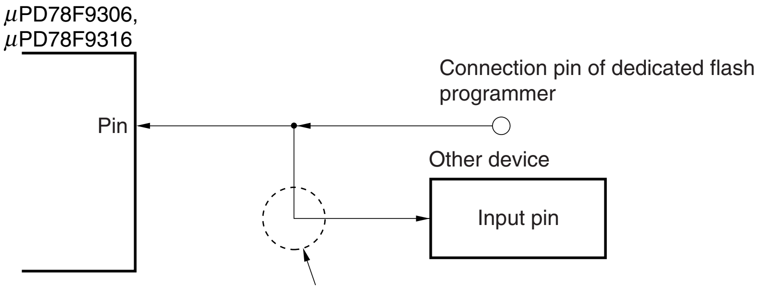
(2) Malfunction of another device

When the dedicated flash programmer (output or input) is connected to a serial interface pin (input or output) connected to another device (input), a signal may be output to the device, causing a malfunction. To prevent such malfunction, isolate the connection with other device or set so that the input signal to the device is ignored.

Figure 19-6. Malfunction of Another Device



If the signal output by the μPD78F9306, 78F9316 affects another device in the flash memory programming mode, isolate the signal on the device side.



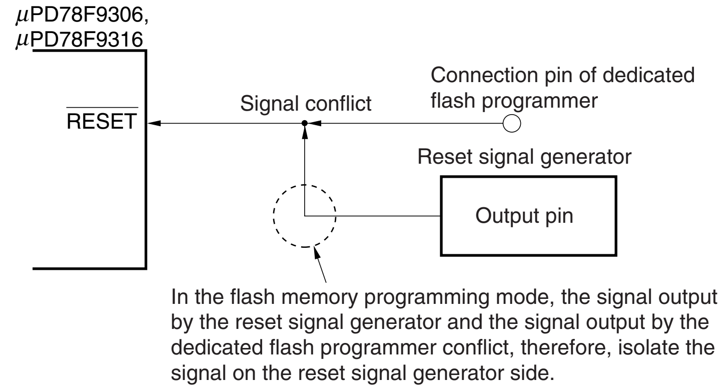
If the signal output by the dedicated flash programmer affects another device, isolate the signal on the device side.

<$\overline{\text{RESET}}$ pin>

When the reset signal of the dedicated flash programmer is connected to the $\overline{\text{RESET}}$ pin connected to the reset signal generator on the board, a signal conflict occurs. To prevent this signal conflict, isolate the connection with the reset signal generator.

If a reset signal is input from the user system in the flash memory programming mode, a normal programming operation will not be performed. Do not input signals other than reset signals from the dedicated flash programmer during this period.

Figure 19-7. Signal Conflict ($\overline{\text{RESET}}$ Pin)


μPD78F9306,
μPD78F9316
$\overline{\text{RESET}}$
Signal conflict
Connection pin of dedicated flash programmer
Reset signal generator
Output pin

In the flash memory programming mode, the signal output by the reset signal generator and the signal output by the dedicated flash programmer conflict, therefore, isolate the signal on the reset signal generator side.

<Port pins>

Shifting to the flash memory programming mode sets all the pins except those used for flash memory programming communication to the status immediately after reset.

Therefore, if the external device does not acknowledge an initial status such as the output high impedance status, connect the external device to V_{DD} or V_{SS} via a resistor.

<Oscillation pins>

- In the case of μPD78F9306

  When using an on-board clock, connection of X1, X2, XT1, and XT2 must conform to the methods in the normal operation mode.

  When using the clock output of the flash programmer, directly connect it to the X1 pin with the on-board main oscillator disconnected, and leave the X2 pin open. For the subclock, connection conforms to that in the normal operation mode.

- In the case of μPD78F9316

  Connect CL1, CL2, XT1, and XT2 as required in the normal operation mode, and connect the clock output of the flash programmer to the P03 pin.

<Power supply>

To use the power output of the flash programmer, connect the V_{DD0} pins to VDD of the flash programmer, and the V_{SS} pins to GND of the flash programmer.

To use the on-board power supply, connection must conform to that in the normal operation mode. However, because the voltage is monitored by the flash programmer, therefore, VDD of the flash programmer must be connected.

<Other pins>

Handle the other pins (S0 to S23, COM0 to COM3, V_{LC0} to V_{LC2}, CAPH, CAPL) in the same way as in the normal operation mode.

19.1.4 Connection when using flash memory writing adapter

The following shows an example of the recommended connection when using the flash memory writing adapter.

Figure 19-8. Example of Flash Memory Writing Adapter Connection When Using 3-Wire Serial I/O Mode (μPD78F9306)

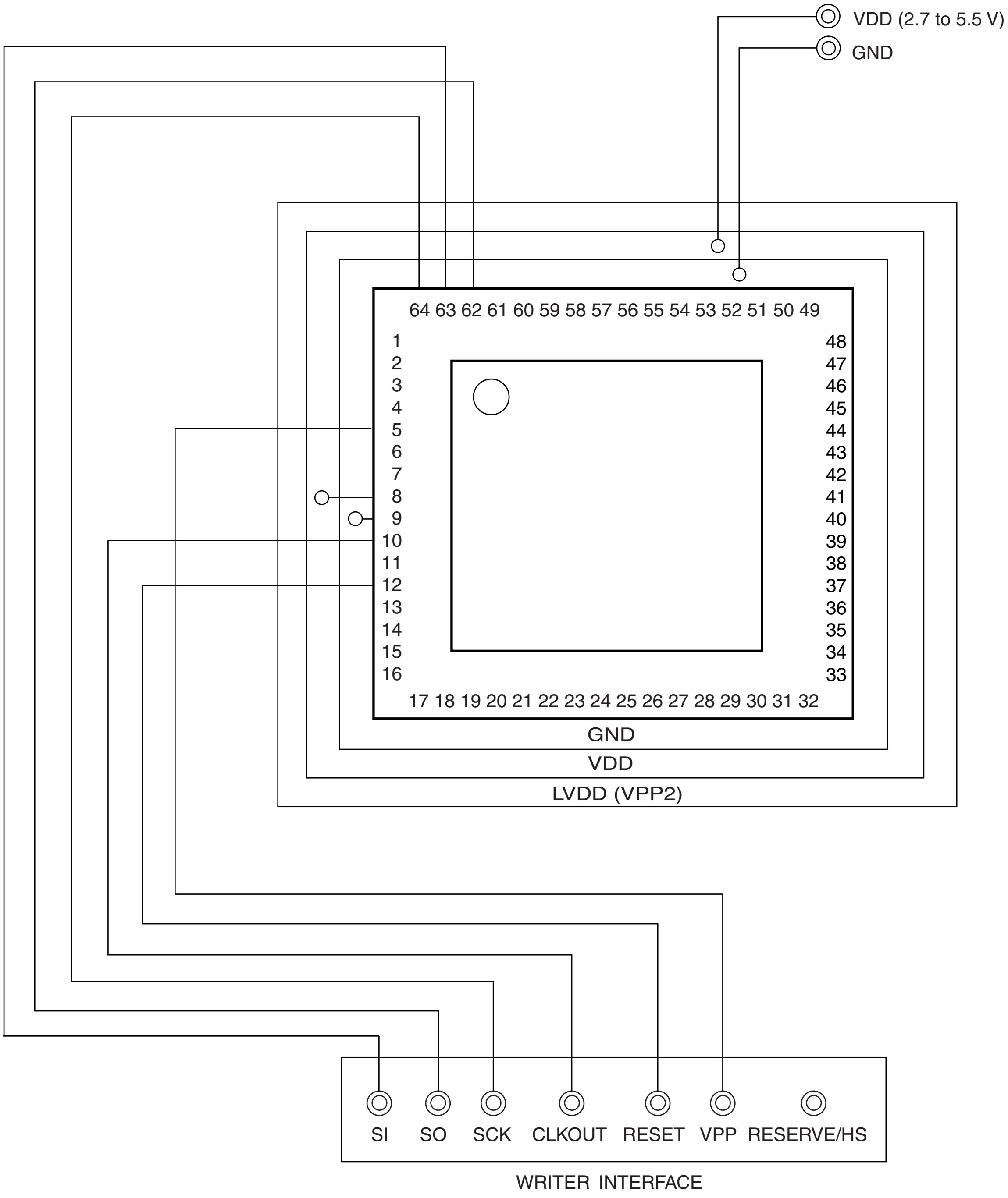


Figure 19-9. Example of Flash Memory Writing Adapter Connection When Using 3-Wire Serial I/O Mode (μPD78F9316)

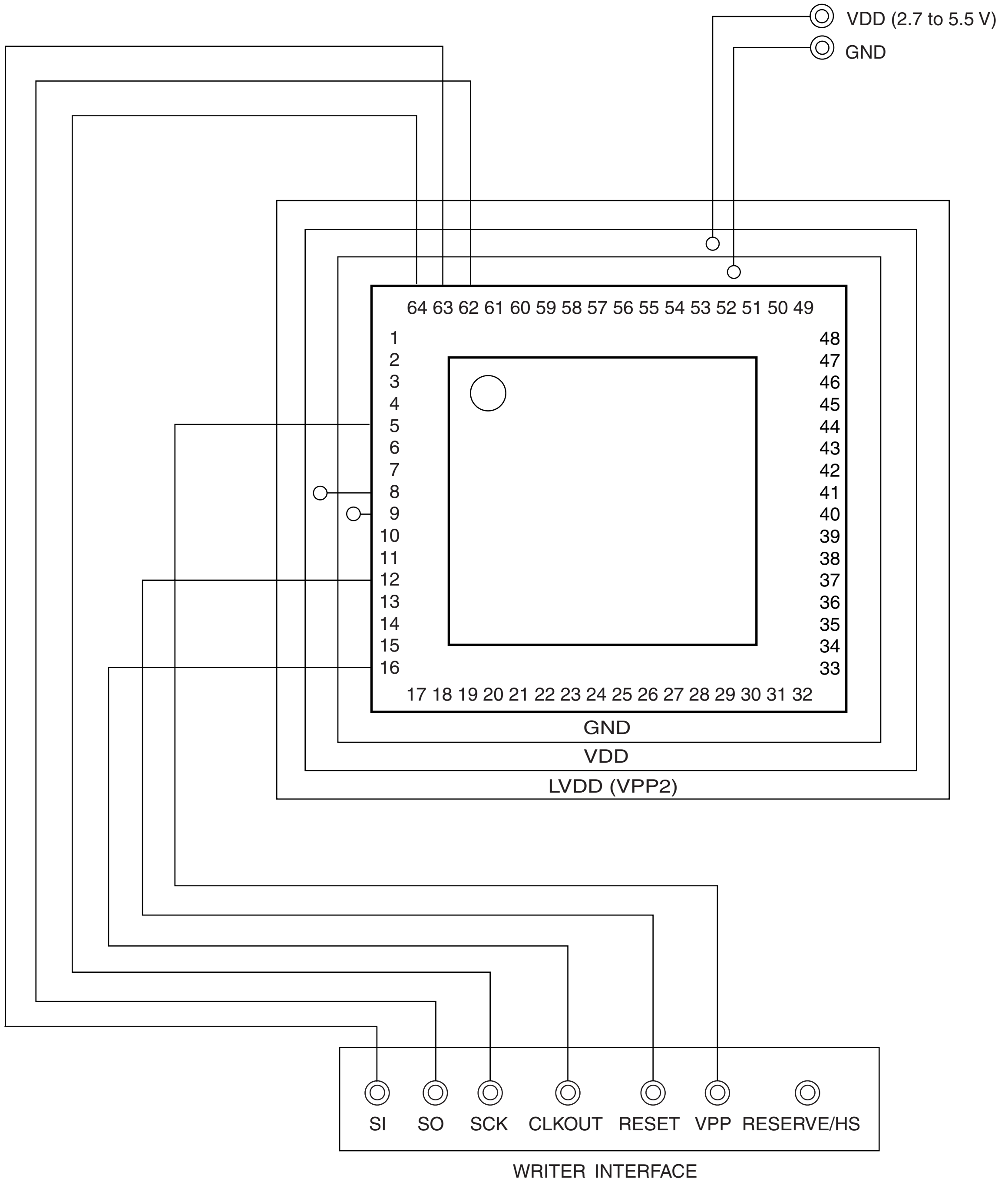


Figure 19-10. Example of Flash Memory Writing Adapter Connection When Using UART Mode (μPD78F9306)

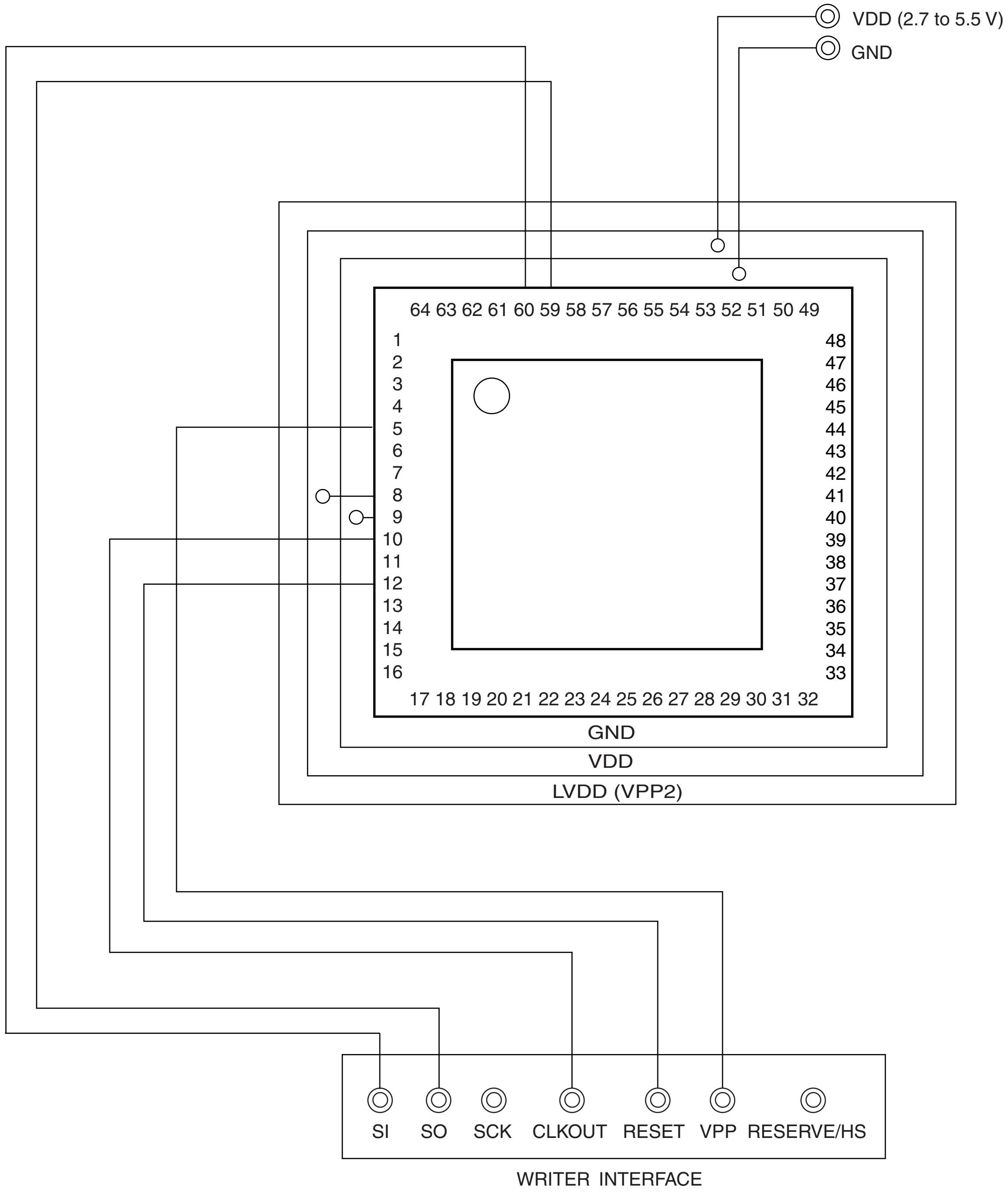
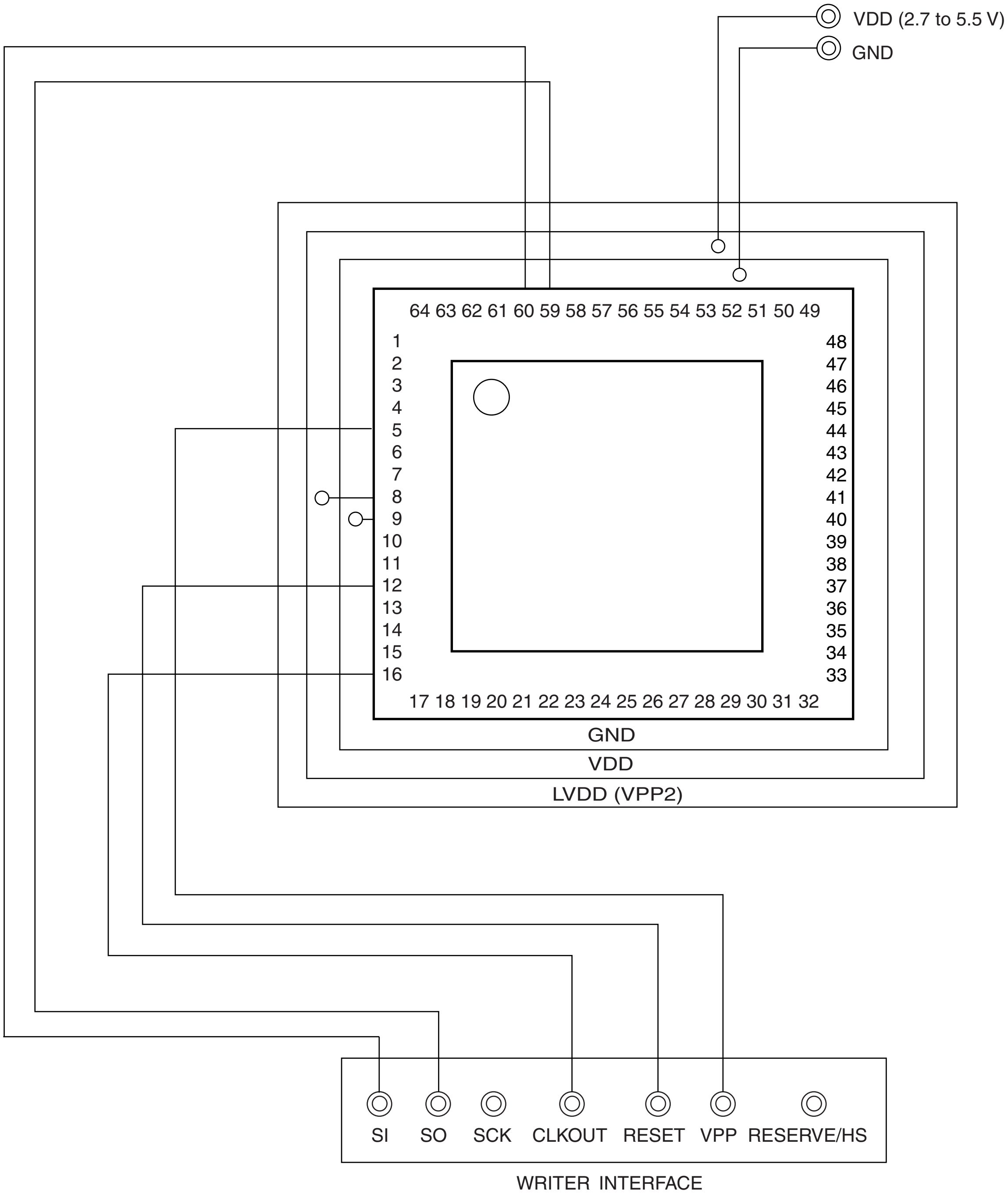


Figure 19-11. Example of Flash Memory Writing Adapter Connection When Using UART Mode (μPD78F9316)



CHAPTER 20 MASK OPTIONS

Table 20-1. Selection of Mask Option for Pins

Pin	Mask Option
P50 to P53	Whether a pull-up resistor is to be incorporated can be specified in 1-bit units.

For P50 to P53 (port 5), a mask option is used to specify whether a pull-up resistor is to be incorporated. The mask option is selectable in 1-bit units.

Caution Flash memory versions do not have a mask option-based on-chip pull-up resistor function.

CHAPTER 21 INSTRUCTION SET

This chapter lists the instruction set of the μPD789306 and μPD789316 Subseries. For the 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 "Operand" column of each instruction in accordance with the description method of the instruction operand identifier (refer to the assembler specifications for detail). When there are two or more description methods, select one of them. Alphabetic letters in capitals and symbols, #, !, $, and [ ] are key words and are described as they are. Each symbol has the following meaning.

- #: Immediate data specification
- !: Absolute address specification
- $: Relative address specification
- [ ]: Indirect address specification

In the case of immediate data, describe an appropriate numeric value or a label. When using a label, be sure to describe the #, !, $ and [ ] symbols.

For operand register identifiers, r and rp, either 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 rp sfr	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7) AX (RP0), BC (RP1), DE (RP2), HL (RP3) Special-function register symbol
saddr saddrp	FE20H to FF1FH Immediate data or labels FE20H to FF1FH Immediate data or labels (even addresses only)
addr16 addr5	0000H to FFFFH Immediate data or labels (only even addresses for 16-bit data transfer instructions) 0040H to 007FH Immediate data or labels (even addresses only)
word byte bit	16-bit immediate data or label 8-bit immediate data or label 3-bit immediate data or label

Remark See **Table 5-3 Special Function Register List** 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
NMIS:	Flag indicating non-maskable interrupt servicing in progress
( ):	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)
$\veebar$:	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	Byte	Clock	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	x	x	x
	A, PSW	2	4	A ← PSW			
	PSW, A	2	4	PSW ← A	x	x	x
	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	Byte	Clock	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	x	x	x
	saddr, #byte	3	6	(saddr), CY ← (saddr) + byte	x	x	x
	A, r	2	4	A, CY ← A + r	x	x	x
	A, saddr	2	4	A, CY ← A + (saddr)	x	x	x
	A, !addr16	3	8	A, CY ← A + (addr16)	x	x	x
	A, [HL]	1	6	A, CY ← A + (HL)	x	x	x
	A, [HL+byte]	2	6	A, CY ← A + (HL + byte)	x	x	x
ADDC	A, #byte	2	4	A, CY ← A + byte + CY	x	x	x
	saddr, #byte	3	6	(saddr), CY ← (saddr) + byte + CY	x	x	x
	A, r	2	4	A, CY ← A + r + CY	x	x	x
	A, saddr	2	4	A, CY ← A + (saddr) + CY	x	x	x
	A, !addr16	3	8	A, CY ← A + (addr16) + CY	x	x	x
	A, [HL]	1	6	A, CY ← A + (HL) + CY	x	x	x
	A, [HL+byte]	2	6	A, CY ← A + (HL + byte) + CY	x	x	x
SUB	A, #byte	2	4	A, CY ← A – byte	x	x	x
	saddr, #byte	3	6	(saddr), CY ← (saddr) – byte	x	x	x
	A, r	2	4	A, CY ← A – r	x	x	x
	A, saddr	2	4	A, CY ← A – (saddr)	x	x	x
	A, !addr16	3	8	A, CY ← A – (addr16)	x	x	x
	A, [HL]	1	6	A, CY ← A – (HL)	x	x	x
	A, [HL+byte]	2	6	A, CY ← A – (HL + byte)	x	x	x

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	Byte	Clock	Operation	Flag		
					Z	AC	CY
SUBC	A, #byte	2	4	A, CY ← A – byte – CY	x	x	x
	saddr, #byte	3	6	(saddr), CY ← (saddr) – byte – CY	x	x	x
	A, r	2	4	A, CY ← A – r – CY	x	x	x
	A, saddr	2	4	A, CY ← A – (saddr) – CY	x	x	x
	A, !addr16	3	8	A, CY ← A – (addr16) – CY	x	x	x
	A, [HL]	1	6	A, CY ← A – (HL) – CY	x	x	x
	A, [HL+byte]	2	6	A, CY ← A – (HL + byte) – CY	x	x	x
AND	A, #byte	2	4	A ← A ∧ byte	x		
	saddr, #byte	3	6	(saddr) ← (saddr) ∧ byte	x		
	A, r	2	4	A ← A ∧ r	x		
	A, saddr	2	4	A ← A ∧ (saddr)	x		
	A, !addr16	3	8	A ← A ∧ (addr16)	x		
	A, [HL]	1	6	A ← A ∧ (HL)	x		
	A, [HL+byte]	2	6	A ← A ∧ (HL + byte)	x		
OR	A, #byte	2	4	A ← A ∨ byte	x		
	saddr, #byte	3	6	(saddr) ← (saddr) ∨ byte	x		
	A, r	2	4	A ← A ∨ r	x		
	A, saddr	2	4	A ← A ∨ (saddr)	x		
	A, !addr16	3	8	A ← A ∨ (addr16)	x		
	A, [HL]	1	6	A ← A ∨ (HL)	x		
	A, [HL+byte]	2	6	A ← A ∨ (HL + byte)	x		
XOR	A, #byte	2	4	A ← A ∀ byte	x		
	saddr, #byte	3	6	(saddr) ← (saddr) ∀ byte	x		
	A, r	2	4	A ← A ∀ r	x		
	A, saddr	2	4	A ← A ∀ (saddr)	x		
	A, !addr16	3	8	A ← A ∀ (addr16)	x		
	A, [HL]	1	6	A ← A ∀ (HL)	x		
	A, [HL+byte]	2	6	A ← A ∀ (HL + byte)	x		

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

Mnemonic	Operands	Byte	Clock	Operation	Flag		
					Z	AC	CY
CMP	A, #byte	2	4	A – byte	x	x	x
	saddr, #byte	3	6	(saddr) – byte	x	x	x
	A, r	2	4	A – r	x	x	x
	A, saddr	2	4	A – (saddr)	x	x	x
	A, !addr16	3	8	A – (addr16)	x	x	x
	A, [HL]	1	6	A – (HL)	x	x	x
	A, [HL+byte]	2	6	A – (HL + byte)	x	x	x
ADDW	AX, #word	3	6	AX, CY ← AX + word	x	x	x
SUBW	AX, #word	3	6	AX, CY ← AX – word	x	x	x
CMPW	AX, #word	3	6	AX – word	x	x	x
INC	r	2	4	r ← r + 1	x	x	
	saddr	2	4	(saddr) ← (saddr) + 1	x	x	
DEC	r	2	4	r ← r – 1	x	x	
	saddr	2	4	(saddr) ← (saddr) – 1	x	x	
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			x
ROL	A, 1	1	2	(CY, A_0 ← A_7, A_{m+1} ← A_m) × 1			x
RORC	A, 1	1	2	(CY ← A_0, A_7 ← CY, A_{m-1} ← A_m) × 1			x
ROLC	A, 1	1	2	(CY ← A_7, A_0 ← CY, A_{m+1} ← A_m) × 1			x
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	x	x	x
	[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	x	x	x
	[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 ← $\overline{CY}$			x

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

Mnemonic	Operands	Byte	Clock	Operation	Flag		
					Z	AC	CY
CALL	!addr16	3	6	(SP – 1) ← (PC + 3)$_H$, (SP – 2) ← (PC + 3)$_L$, PC ← addr16, SP ← SP – 2			
CALLT	[addr5]	1	8	(SP – 1) ← (PC + 1)$_H$, (SP – 2) ← (PC + 1)$_L$, PC$_H$ ← (00000000, addr5 + 1), PC$_L$ ← (00000000, addr5), SP ← SP – 2			
RET		1	6	PC$_H$ ← (SP + 1), PC$_L$ ← (SP), SP ← SP + 2			
RETI		1	8	PC$_H$ ← (SP + 1), PC$_L$ ← (SP), PSW ← (SP + 2), SP ← SP + 3, NMIS ← 0	R	R	R
PUSH	PSW	1	2	(SP – 1) ← PSW, SP ← SP – 1			
	rp	1	4	(SP – 1) ← rp$_H$, (SP – 2) ← rp$_L$, SP ← SP – 2			
POP	PSW	1	4	PSW ← (SP), SP ← SP + 1	R	R	R
	rp	1	6	rp$_H$ ← (SP + 1), rp$_L$ ← (SP), SP ← SP + 2			
MOVW	SP, AX	2	8	SP ← AX			
	AX, SP	2	6	AX ← SP			
BR	!addr16	3	6	PC ← addr16			
	$addr16	2	6	PC ← PC + 2 + jdisp8			
	AX	1	6	PC$_H$ ← A, PC$_L$ ← X			
BC	$saddr16	2	6	PC ← PC + 2 + jdisp8 if CY = 1			
BNC	$saddr16	2	6	PC ← PC + 2 + jdisp8 if CY = 0			
BZ	$saddr16	2	6	PC ← PC + 2 + jdisp8 if Z = 1			
BNZ	$saddr16	2	6	PC ← PC + 2 + jdisp8 if Z = 0			
BT	saddr.bit, $addr16	4	10	PC ← PC + 4 + jdisp8 if (saddr.bit) = 1			
	sfr.bit, $addr16	4	10	PC ← PC + 4 + jdisp8 if sfr.bit = 1			
	A.bit, $addr16	3	8	PC ← PC + 3 + jdisp8 if A.bit = 1			
	PSW.bit, $addr16	4	10	PC ← PC + 4 + jdisp8 if PSW.bit = 1			
BF	saddr.bit, $addr16	4	10	PC ← PC + 4 + jdisp8 if (saddr.bit) = 0			
	sfr.bit, $addr16	4	10	PC ← PC + 4 + jdisp8 if sfr.bit = 0			
	A.bit, $addr16	3	8	PC ← PC + 3 + jdisp8 if A.bit = 0			
	PSW.bit, $addr16	4	10	PC ← PC + 4 + jdisp8 if PSW.bit = 0			
DBNZ	B, $addr16	2	6	B ← B – 1, then PC ← PC + 2 + jdisp8 if B ≠ 0			
	C, $addr16	2	6	C ← C – 1, then PC ← PC + 2 + jdisp8 if C ≠ 0			
	saddr, $addr16	3	8	(saddr) ← (saddr) – 1, then PC ← PC + 3 + jdisp8 if (saddr) ≠ 0			
NOP		1	2	No Operation			
EI		3	6	IE ← 1 (Enable interrupt)			
DI		3	6	IE ← 0 (Disable interrupt)			
HALT		1	2	Set HALT mode			
STOP		1	2	Set STOP mode			

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

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

2nd Operand 1st Operand	#byte	A	r	sfr	saddr	!addr16	PSW	[DE]	[HL]	[HL+byte]	$addr16	1	None
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV[Note] XCH[Note] ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP		ROR ROL RORC ROLC	
r	MOV	MOV											INC DEC
B, C											DBNZ		
sfr	MOV	MOV											
saddr	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV									DBNZ		INC DEC
!addr16		MOV											
PSW	MOV	MOV											PUSH POP
[DE]		MOV											
[HL]		MOV											
[HL+byte]		MOV											

Note Except r = A.

(2) 16-bit instructions

MOVW, XCHW, ADDW, SUBW, CMPW, PUSH, POP, INCW, DECW

2nd Operand 1st Operand	#word	AX	rp[Note]	saddrp	SP	None
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	
rp	MOVW	MOVW[Note]				INCW DECW PUSH POP
saddrp		MOVW				
SP		MOVW				

Note Only when rp = BC, DE, or HL.

(3) Bit manipulation instructions

SET1, CLR1, NOT1, BT, BF

2nd Operand 1st Operand	$addr16	None
A.bit	BT BF	SET1 CLR1
sfr.bit	BT BF	SET1 CLR1
saddr.bit	BT BF	SET1 CLR1
PSW.bit	BT BF	SET1 CLR1
[HL].bit		SET1 CLR1
CY		SET1 CLR1 NOT1

(4) Call instructions/branch instructions

CALL, CALLT, BR, BC, BNC, BZ, BNZ, DBNZ

2nd Operand 1st Operand	AX	!addr16	[addr5]	$addr16
Basic Instructions	BR	CALL BR	CALLT	BR BC BNC BZ BNZ
Compound Instructions				DBNZ

(5) Other instructions

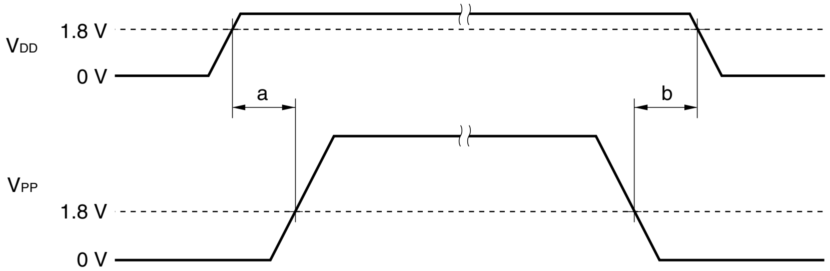
RET, RETI, NOP, EI, DI, HALT, STOP

Absolute Maximum Ratings (T_A = 25°C)

Parameter	Symbol	Conditions		Ratings	Unit
Power supply voltage	V_{DD}			–0.3 to +6.5	V
	V_{PP}	μPD78F9306, 78F9316 only[Note 1]		–0.3 to +10.5	V
Input voltage	V_{I1}	P00 to P03, P10 to P13, P20 to P26, P30 to P33, X1 [CL1], X2 [CL2], XT1, XT2, $\overline{\text{RESET}}$		–0.3 to V_{DD} + 0.3[Note 2]	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 2]	V
Output voltage	V_O			–0.3 to V_{DD} + 0.3[Note 2]	V
Output current, high	I_{OH}	1 pin		–10	mA
		Total for all pins		–30	mA
Output current, low	I_{OL}	1 pin		30	mA
		Total for all pins		160	mA
Operating ambient temperature	T_A	In normal operation		–40 to +85	°C
		During flash memory programming		10 to 40	°C
Storage temperature	T_{stg}	Mask ROM version		–65 to +150	°C
		μPD78F9306, 78F9316		–40 to +125	°C

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. 6.5 V or less

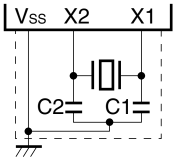
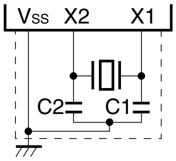
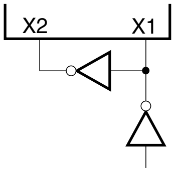
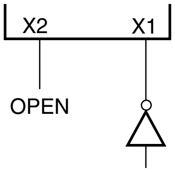
Caution **Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.**

Remarks 1. Pin names enclosed in brackets [ ] apply when using the μPD789316 Subseries.

2. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Main System Clock Oscillator Characteristics

Ceramic/crystal oscillation (μPD789306 Subseries) (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[Note 1]		1.0		5.0	MHz
		Oscillation stabilization time[Note 2]	V_{DD} = 4.5 to 5.5 V			10	ms
			V_{DD} = 1.8 to 5.5 V			30	ms
External clock		X1 input frequency (f_X)[Note 1]		1.0		5.0	MHz
		X1 input high-/low-level width (t_{XH}, t_{XL})		85		500	ns
		X1 input frequency (f_X)[Note 1]	V_{DD} = 2.7 to 5.5 V	1.0		5.0	MHz
		X1 input high-/low-level width (t_{XH}, t_{XL})	V_{DD} = 2.7 to 5.5 V	85		500	ns

Notes 1. Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

2. Time required to stabilize oscillation after reset or STOP mode release. Use a resonator whose oscillation stabilizes within the oscillation stabilization wait time.

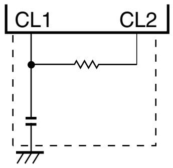
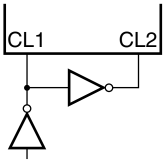
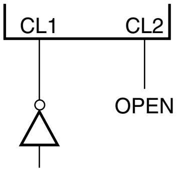
Cautions 1. When using the main system clock oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

- **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 requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

RC oscillation (μPD789316 Subseries) (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
RC resonator		Oscillation frequency (f_{CC})[Note 1]		2.0		4.0	MHz
		Oscillation stabilization time[Note 2]	V_{DD} = 2.7 to 5.5 V	32			μs
			V_{DD} = 1.8 to 5.5 V	128			μs
External clock		CL1 input frequency (f_{CC})[Note 1]		1.0		4.0	MHz
		CL1 input high-/low-level width (t_{XH}, t_{XL})		100		500	ns
		CL1 input frequency (f_{CC})[Note 1]	V_{DD} = 2.7 to 5.5 V	1.0		4.0	MHz
		CL1 input high-/low-level width (t_{XH}, t_{XL})	V_{DD} = 2.7 to 5.5 V	100		500	ns

Notes 1. Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time. The error of capacitor (C) and resistor (R) is not included.

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 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. 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.

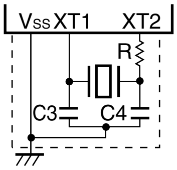
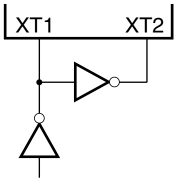
RC Oscillation Frequency Characteristics (T_A = –40 to +85°C)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Oscillation frequency	f_{CC1}	R = 11.0 kΩ, C = 22 pF Target: 2 MHz	V_{DD} = 2.7 to 5.5 V	1.5	2.0	2.5	MHz
	f_{CC2}		V_{DD} = 1.8 to 3.6 V	0.5	2.0	2.5	MHz
	f_{CC3}		V_{DD} = 1.8 to 5.5 V	0.5	2.0	2.5	MHz
	f_{CC4}	R = 6.8 kΩ, C = 22 pF Target: 3 MHz	V_{DD} = 2.7 to 5.5 V	2.5	3.0	3.5	MHz
	f_{CC5}		V_{DD} = 1.8 to 3.6 V	0.75	3.0	3.5	MHz
	f_{CC6}		V_{DD} = 1.8 to 5.5 V	0.75	3.0	3.5	MHz
	f_{CC7}	R = 4.7 kΩ, C = 22 pF Target: 4 MHz	V_{DD} = 2.7 to 5.5 V	3.5	4.0	4.7	MHz
	f_{CC8}		V_{DD} = 1.8 to 3.6 V	1.0	4.0	4.7	MHz
	f_{CC9}		V_{DD} = 1.8 to 5.5 V	1.0	4.0	4.7	MHz

Remarks 1. Set RC to one of the above nine values so that the typical value of the oscillation frequency is within 2.0 to 4.0 MHz.

2. The resistor (R) and capacitor (C) error is not included.

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 current consumption, and is more prone to malfunction due to noise than the main system clock oscillator. Particular care is therefore required with the wiring method when the subsystem clock is used.

Remark For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

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}	1 pin					10	mA
		All pins					80	mA
Output current, high	I_{OH}	1 pin					–1	mA
		All pins					–15	mA
Input voltage, high	V_{IH1}	P10 to P13		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	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 P03, P20 to P26, P30 to P33		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 [CL1], X2 [CL2], 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 to P13		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 P03, P20 to P26, P30 to P33		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 [CL1], X2 [CL2], 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}	I_{OH} = –1 mA		V_{DD} = 4.5 to 5.5 V	V_{DD} – 1.0			V
		I_{OH} = –100 μA		V_{DD} = 1.8 to 5.5 V	V_{DD} – 0.5			V
Output voltage, low	V_{OL1}	P00 to P03, P10 to P13, P20 to P26, P30 to P33		$4.5 \le V_{DD} \le 5.5$ V, I_{OL} = 10 mA			1.0	V
				$1.8 \le V_{DD} < 4.5$ V, I_{OL} = 400 μA			0.5	V
	V_{OL2}	P50 to P53		$4.5 \le V_{DD} < 5.5$ V, I_{OL} = 10 mA			1.0	V
				$1.8 \le V_{DD} < 4.5$ V, I_{OL} = 1.6 mA			0.4	V

Remarks 1. Pin names enclosed in brackets [ ] apply when using the μPD789316 Subseries.

2. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V) (2/6)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	$V_I = V_{DD}$	P00 to P03, P10 to P13, P20 to P26, P30 to P33, $\overline{\text{RESET}}$			3	μA
	I_{LIH2}		X1 [CL1], X2 [CL2], 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 P03, P10 to P13, P20 to P26, P30 to P33, $\overline{\text{RESET}}$			–3	μA
	I_{LIL2}		X1 [CL1], X2 [CL2], XT1, XT2			–20	μA
	I_{LIL3}		P50 to P53 (N-ch open drain)			–3[Note]	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 P03, P10 to P13, P20 to P26, P30 to P33	50	100	200	kΩ
Mask option pullup resistor[Note 2]	R_2	V_I = 0 V	P50 to P53	10	30	60	kΩ

Notes 1. 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 –30 μA flows during only one cycle. At all other times, the maximum leakage current is –3 μA.

2. Mask ROM version only

Remarks 1. Pin names enclosed in brackets [ ] apply when using the μPD789316 Subseries.

2. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V) (3/6)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Power supply current[Note 1] (μPD789304, 789306)	I_{DD1}	5.0 MHz crystal oscillation operation mode (C1 = C2 = 22 pF)		V_{DD} = 5.0 V ±10%[Note 2]		1.8	2.9	mA
				V_{DD} = 3.0 V ±10%[Note 3]		0.36	0.9	mA
				V_{DD} = 2.0 V ±10%[Note 3]		0.16	0.45	mA
	I_{DD2}	5.0 MHz crystal oscillation HALT mode[Note 4] (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 5] (C3 = C4 = 22 pF, R1 = 220 kΩ)		V_{DD} = 5.0 V ±10%		30	58	μA
				V_{DD} = 3.0 V ±10%		9	26	μA
				V_{DD} = 2.0 V ±10%		4	12	μA
	I_{DD4}	32.768 kHz crystal oscillation HALT mode[Note 5] (C3 = C4 = 22 pF, R1 = 220 kΩ)	LCD not operating[Note 4]	V_{DD} = 5.0 V ±10%		25	48	μA
				V_{DD} = 3.0 V ±10%		7	20	μA
				V_{DD} = 2.0 V ±10%		4	10	μA
			LCD operating[Note 7]	V_{DD} = 5.0 V ±10%		28	57	μA
				V_{DD} = 3.0 V ±10%		9.6	27.8	μA
				V_{DD} = 2.0 V ±10%		6	16	μA
	I_{DD5}	STOP mode[Note 6]		V_{DD} = 5.0 V ±10%		0.1	10	μA
				V_{DD} = 3.0 V ±10%		0.05	5.0	μA
				V_{DD} = 2.0 V ±10%		0.05	3.0	μA

Notes
1. The port current (including the current that flows to the on-chip pull-up resistors) is not included.
2. High-speed mode operation (when processor clock control register (PCC) is cleared to 00H)
3. Low-speed mode operation (when PCC is set to 02H)
4. When the LCD is not operating and the booster circuit is operating (LCDON0 = 0, VAON0 = 1, LIPS0 = 1).
5. When the main system clock is stopped
6. When the LCD is not operating (LCDON0 = 0, VAON0 = 0, LIPS0 = 0)
7. Then the LCD is operating (LCDON0 = 1, VAON0 = 1, LIPS0 = 1)

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V) (4/6)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Power supply current[Note 1] (μPD78F9306)	I_{DD1}	5.0 MHz crystal oscillation operation mode (C1 = C2 = 22 pF)		V_{DD} = 5.0 V ±10%[Note 2]		4.5	9	mA
				V_{DD} = 3.0 V ±10%[Note 3]		1	2	mA
				V_{DD} = 2.0 V ±10%[Note 3]		0.65	1.5	mA
	I_{DD2}	5.0 MHz crystal oscillation HALT mode[Note 4] (C1 = C2 = 22 pF)		V_{DD} = 5.0 V ±10%[Note 2]		1.4	2	mA
				V_{DD} = 3.0 V ±10%[Note 3]		0.4	0.8	mA
				V_{DD} = 2.0 V ±10%[Note 3]		0.19	0.42	mA
	I_{DD3}	32.768 kHz crystal oscillation operation mode[Note 5] (C3 = C4 = 22 pF, R1 = 220 kΩ)		V_{DD} = 5.0 V ±10%		100	230	μA
				V_{DD} = 3.0 V ±10%		70	160	μA
				V_{DD} = 2.0 V ±10%		58	120	μA
	I_{DD4}	32.768 kHz crystal oscillation HALT mode[Note 5] (C3 = C4 = 22 pF, R1 = 220 kΩ)	LCD not operating[Note 4]	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 7]	V_{DD} = 5.0 V ±10%		28	70	μA
				V_{DD} = 3.0 V ±10%		9.6	34	μA
				V_{DD} = 2.0 V ±10%		6	25	μA
	I_{DD5}	STOP mode[Note 6]		V_{DD} = 5.0 V ±10%		0.1	17	μA
				V_{DD} = 3.0 V ±10%		0.05	5.5	μA
				V_{DD} = 2.0 V ±10%		0.05	3.5	μA

Notes 1. The port current (including the current that flows to the on-chip pull-up resistors) is not included.

2. High-speed mode operation (when processor clock control register (PCC) is cleared to 00H)

3. Low-speed mode operation (when PCC is set to 02H)

4. When the LCD is not operating and the booster circuit is operating (LCDON0 = 0, VAON0 = 1, LIPS0 = 1).

5. When the main system clock is stopped

6. When the LCD is not operating (LCDON0 = 0, VAON0 = 0, LIPS0 = 0)

7. Then the LCD is operating (LCDON0 = 1, VAON0 = 1, LIPS0 = 1)

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V) (5/6)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Power supply current[Note 1] (μPD789314, 789316)	I_{DD1}	4.0 MHz RC oscillation operation mode (R = 4.7 kΩ, C = 22 pF)		V_{DD} = 5.0 V ±10%[Note 2]		1.65	3.0	mA
				V_{DD} = 3.0 V ±10%[Note 3]		0.65	1.44	mA
				V_{DD} = 2.0 V ±10%[Note 3]		0.38	1.05	mA
	I_{DD2}	4.0 MHz RC oscillation HALT mode[Note 4] (R = 4.7 kΩ, C = 22 pF)		V_{DD} = 5.0 V ±10%[Note 2]		1.1	2.29	mA
				V_{DD} = 3.0 V ±10%[Note 3]		0.6	1.28	mA
				V_{DD} = 2.0 V ±10%[Note 3]		0.35	0.82	mA
	I_{DD3}	32.768 kHz crystal oscillation operation mode[Note 5] (C3 = C4 = 22 pF, R1 = 220 kΩ)		V_{DD} = 5.0 V ±10%		30	58	μA
				V_{DD} = 3.0 V ±10%		9	26	μA
				V_{DD} = 2.0 V ±10%		4	12	μA
	I_{DD4}	32.768 kHz crystal oscillation HALT mode[Note 5] (C3 = C4 = 22 pF, R1 = 220 kΩ)	LCD not operating[Note 4]	V_{DD} = 5.0 V ±10%		25	48	μA
				V_{DD} = 3.0 V ±10%		7	20	μA
				V_{DD} = 2.0 V ±10%		4	10	μA
			LCD operating[Note 7]	V_{DD} = 5.0 V ±10%		28	57	μA
				V_{DD} = 3.0 V ±10%		9.6	27.8	μA
				V_{DD} = 2.0 V ±10%		6	16	μA
	I_{DD5}	STOP mode[Note 6]		V_{DD} = 5.0 V ±10%		0.1	10	μA
				V_{DD} = 3.0 V ±10%		0.05	5.0	μA
				V_{DD} = 2.0 V ±10%		0.05	3.0	μA

Notes 1. The port current (including the current that flows to the on-chip pull-up resistors) is not included.

2. High-speed mode operation (when processor clock control register (PCC) is cleared to 00H)

3. Low-speed mode operation (when PCC is set to 02H)

4. When the LCD is not operating and the booster circuit is operating (LCDON0 = 0, VAON0 = 1, LIPS0 = 1).

5. When the main system clock is stopped

6. When the LCD is not operating (LCDON0 = 0, VAON0 = 0, LIPS0 = 0)

7. Then the LCD is operating (LCDON0 = 1, VAON0 = 1, LIPS0 = 1)

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V) (6/6)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Power supply current[Note 1] (μPD78F9316)	I_{DD1}	4.0 MHz RC oscillation operation mode (R = 4.7 kΩ, C = 22 pF)		V_{DD} = 5.0 V ±10%[Note 2]		6	9	mA
				V_{DD} = 3.0 V ±10%[Note 3]		2.0	2.5	mA
				V_{DD} = 2.0 V ±10%[Note 3]		1.2	1.6	mA
	I_{DD2}	4.0 MHz RC oscillation HALT mode[Note 4] (R = 4.7 kΩ, C = 22 pF)		V_{DD} = 5.0 V ±10%[Note 2]		2.5	3.5	mA
				V_{DD} = 3.0 V ±10%[Note 3]		1.5	2	mA
				V_{DD} = 2.0 V ±10%[Note 3]		0.8	1.5	mA
	I_{DD3}	32.768 kHz crystal oscillation operation mode[Note 5] (C3 = C4 = 22 pF, R1 = 220 kΩ)		V_{DD} = 5.0 V ±10%		100	230	μA
				V_{DD} = 3.0 V ±10%		70	160	μA
				V_{DD} = 2.0 V ±10%		58	120	μA
	I_{DD4}	32.768 kHz crystal oscillation HALT mode[Note 5] (C3 = C4 = 22 pF, R1 = 220 kΩ)	LCD not operating[Note 4]	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 7]	V_{DD} = 5.0 V ±10%		28	70	μA
				V_{DD} = 3.0 V ±10%		9.6	34	μA
				V_{DD} = 2.0 V ±10%		6	25	μA
	I_{DD5}	STOP mode[Note 6]		V_{DD} = 5.0 V ±10%		0.1	17	μA
				V_{DD} = 3.0 V ±10%		0.05	5.5	μA
				V_{DD} = 2.0 V ±10%		0.05	3.5	μA

Notes
1. The port current (including the current that flows to the on-chip pull-up resistors) is not included.
2. High-speed mode operation (when processor clock control register (PCC) is cleared to 00H)
3. Low-speed mode operation (when PCC is set to 02H)
4. When the LCD is not operating and the booster circuit is operating (LCDON0 = 0, VAON0 = 1, LIPS0 = 1).
5. When the main system clock is stopped
6. When the LCD is not operating (LCDON0 = 0, VAON0 = 0, LIPS0 = 0)
7. Then the LCD is operating (LCDON0 = 1, VAON0 = 1, LIPS0 = 1)

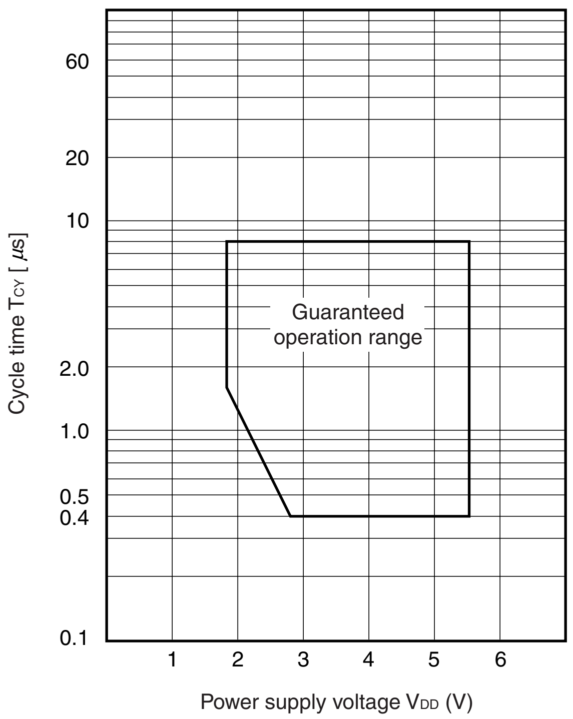
Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

AC Characteristics

(1) Basic operation (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Cycle time (minimum instruction execution time)	T_{CY}	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		114	122	125	µs
CPT20 input high-/low-level width	t_{CPTH}, t_{CPTL}			10			µs
TMI40 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
TMI40 input high-/low-level width	t_{TIH}, t_{TIL}	V_{DD} = 2.7 to 5.5 V		0.1			µs
		V_{DD} = 1.8 to 5.5 V		1.8			µs
Interrupt input high-/low-level width	t_{INTH}, t_{INTL}	INTP0 to INTP3		10			µs
Key return input low-level width	t_{KRL}	KR0 to KR3		10			µs
$\overline{\text{RESET}}$ low-level width	t_{RSL}			10			µs

T_{CY} vs V_{DD} (main system clock)



(2) Serial interface 10, 20 (SIO10, 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
$\overline{\text{SCK}}$n0 cycle time	t_{KCY1}	V_{DD} = 2.7 to 5.5 V		800			ns
		V_{DD} = 1.8 to 5.5 V		3200			ns
$\overline{\text{SCK}}$n0 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
SIn0 setup time (to $\overline{\text{SCK}}$n0↑)	t_{SIK1}	V_{DD} = 2.7 to 5.5 V		150			ns
		V_{DD} = 1.8 to 5.5 V		500			ns
SIn0 hold time (from $\overline{\text{SCK}}$n0↑)	t_{KSI1}	V_{DD} = 2.7 to 5.5 V		400			ns
		V_{DD} = 1.8 to 5.5 V		600			ns
Delay time from $\overline{\text{SCK}}$n0↓ to SOn0 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 SOn0 output lines.

Remark n = 1, 2

(b) 3-wire serial I/O mode (external clock input)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{\text{SCK}}$n0 cycle time	t_{KCY2}	V_{DD} = 2.7 to 5.5 V		800			ns
		V_{DD} = 1.8 to 5.5 V		3200			ns
$\overline{\text{SCK}}$n0 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
SIn0 setup time (to $\overline{\text{SCK}}$n0↑)	t_{SIK2}	V_{DD} = 2.7 to 5.5 V		100			ns
		V_{DD} = 1.8 to 5.5 V		150			ns
SIn0 hold time (from $\overline{\text{SCK}}$n0↑)	t_{KSI2}	V_{DD} = 2.7 to 5.5 V		400			ns
		V_{DD} = 1.8 to 5.5 V		600			ns
Delay time from $\overline{\text{SCK}}$n0↓ to SOn0 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 SOn0 output lines.

Remark n = 1, 2

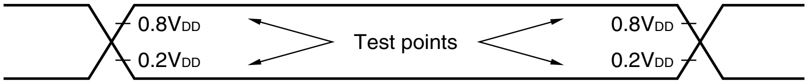
(c) UART mode (SIO20 only) (dedicated baud rate generator output)

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

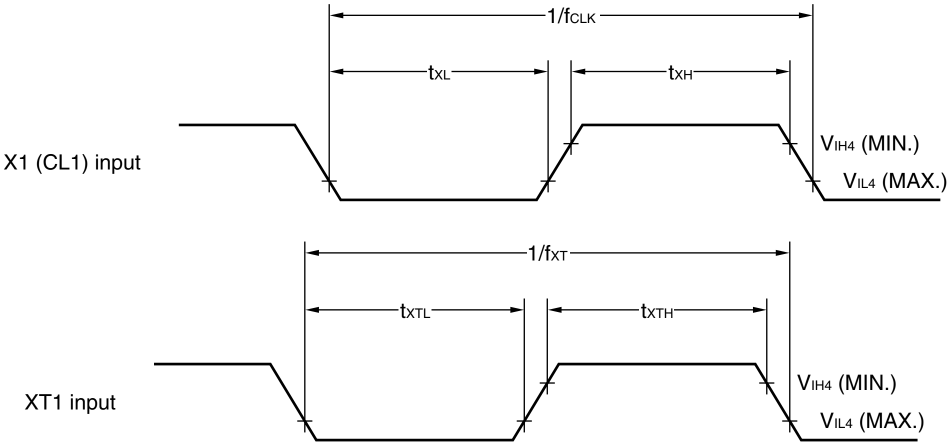
(d) UART mode (SIO20 only) (external clock input)

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

AC Timing Test Points (excluding X1 (CL1) and XT1 inputs)

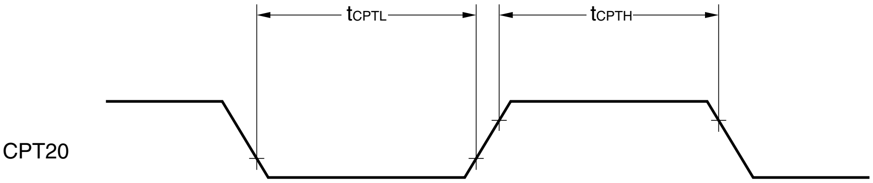


Clock Timing

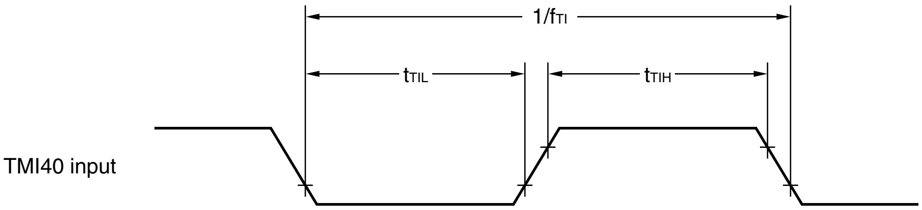


Remark fCLK: fX or fCC

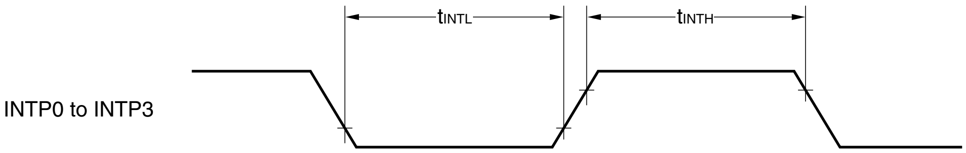
Capture Input Timing



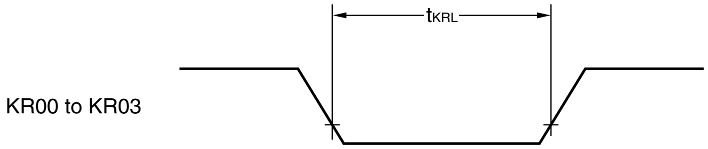
TMI Timing



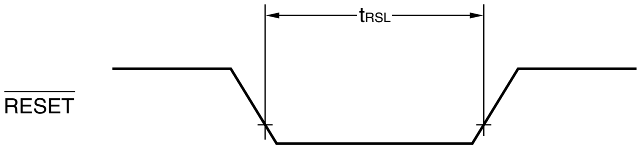
Interrupt Input Timing



Key Return Input Timing

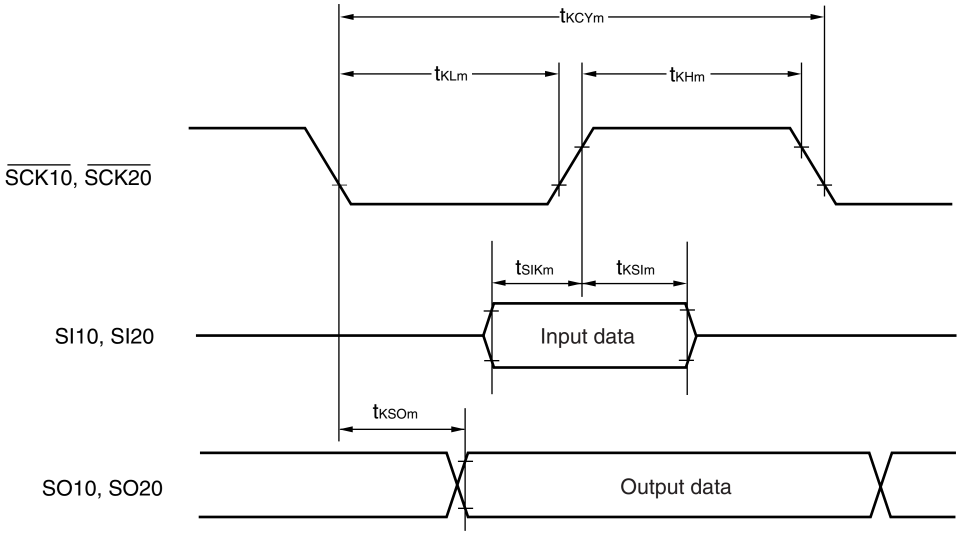


RESET Input Timing



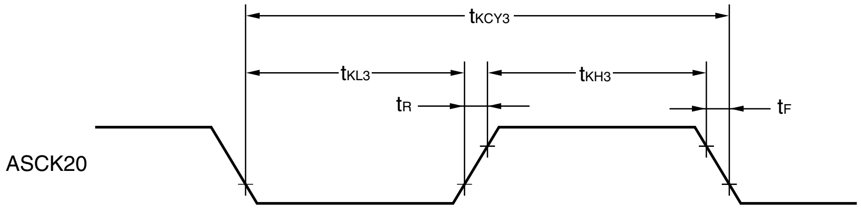
Serial Transfer Timing

3-wire serial I/O mode:



Remark m = 1, 2

UART mode (external clock input):



LCD Characteristics (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
LCD output voltage variation range	V_{LCD2}	C1 to C4[Note 1] = 0.47 μF	GAIN = 1	0.84	1.0	1.165	V
			GAIN = 0	1.26	1.5	1.74	V
Doubler output	V_{LCD1}	C1 to C4[Note 1] = 0.47 μF		2 V_{LCD2}– 0.1	2.0 V_{LCD2}	2.0 V_{LCD2}	V
Tripler output	V_{LCD0}	C1 to C4[Note 1] = 0.47 μF		3 V_{LCD2}– 0.15	3.0 V_{LCD2}	3.0 V_{LCD2}	V
Voltage boost wait time[Note 2]	t_{VAWAIT}	GAIN = 0	1.8 ≤ V_{DD} ≤ 5.5 V	0.5			s
		GAIN = 1	5.0 ≤ V_{DD} ≤ 5.5 V	2.0			s
			4.5 ≤ V_{DD} < 5.0 V	1.0			s
			1.8 ≤ V_{DD} < 4.5 V	0.5			s
LCD output voltage differential[Note 3] (common)	V_{ODC}	I_O = ±5 μA		0		±0.2	V
LCD output voltage differential[Note 3] (segment)	V_{ODS}	I_O = ±1 μA		0		±0.2	V

Notes 1. This is a capacitor that is connected between voltage pins used to drive the LCD.

C1: Capacitor connected between CAPH and CAPL

C2: Capacitor connected between V_{LC0} and V_{SS}

C3: Capacitor connected between V_{LC1} and V_{SS}

C4: Capacitor connected between V_{LC2} and V_{SS}

★ **2.** This is the wait time from when voltage boost is started (VAON0 = 1) until display is enabled (LCDON0 = 1).

3. 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

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] (ceramic/crystal oscillation)	t_{WAIT}	Release by $\overline{\text{RESET}}$		$2^{15}/f_X$		s
		Release by interrupt		**Note 2**		s
Oscillation stabilization wait time (RC oscillation)	t_{WAIT}	Release by $\overline{\text{RESET}}$		$2^{7}/f_{CC}$		s
		Release by interrupt		$2^{7}/f_{CC}$		s

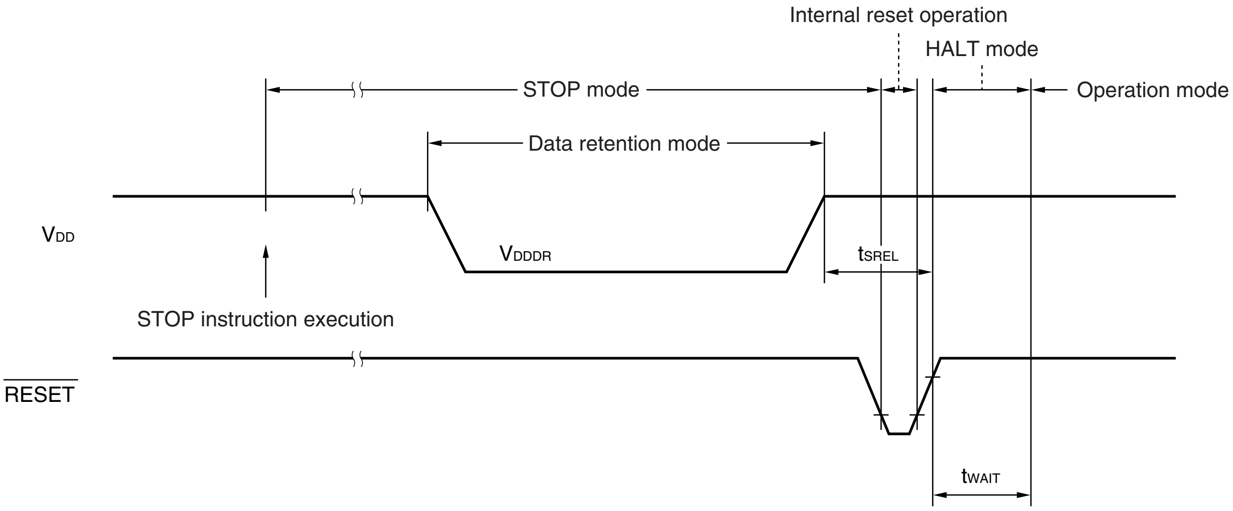
Notes 1. Use a resonator whose oscillation stabilizes within the oscillation stabilization wait time.

2. Selection of $2^{12}/f_X$, $2^{15}/f_X$, or $2^{17}/f_X$ is possible with bits 0 to 2 (OSTS0 to OSTS2) of the oscillation stabilization time select register (OSTS).

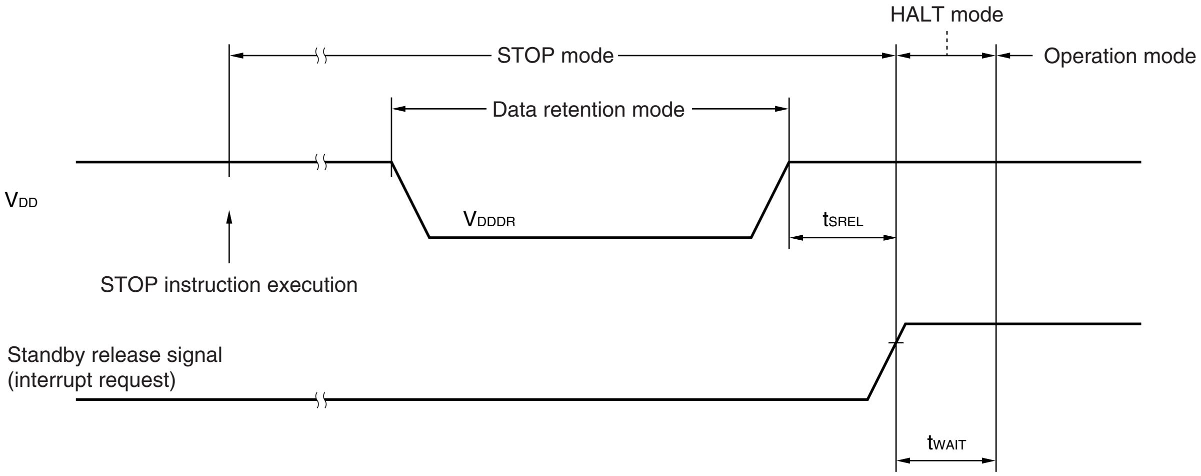
Remarks 1. f_X: Main system clock oscillation frequency (ceramic/crystal oscillation)

2. f_{CC}: Main system clock oscillation frequency (RC oscillation)

Data Retention Timing (STOP Mode Release by $\overline{\text{RESET}}$)



Data Retention Timing (Standby Release Signal: STOP Mode Release by Interrupt Signal)



Flash Memory Write/Erase Characteristics (T_A = 10 to 40°C, V_{DD} = 1.8 to 5.5 V) (μPD78F9306, 78F9316)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Operating frequency	f_X, f_{CC}	V_{DD} = 2.7 to 5.5 V		1.0		5	MHz
		V_{DD} = 1.8 to 5.5 V		1.0		1.25	MHz
Write current[Note 1] (V_{DD} pin)	I_{DDW}	When V_{PP} supply voltage = V_{PP1}	Ceramic oscillation During f_X = 5.0 MHz operation			7	mA
			RC oscillation During f_{CC} = 4.0 MHz operation[Note 2]			9	mA
Write current[Note 1] (V_{PP} pin)	I_{PPW}	When V_{PP} supply voltage = V_{PP1}				12	mA
Erase current[Note 1] (V_{DD} pin)	I_{DDE}	When V_{PP} supply voltage = V_{PP1}	Ceramic oscillation During f_X = 5.0 MHz operation			7	mA
			RC oscillation During f_{CC} = 4.0 MHz operation[Note 2]			9	mA
Erase current[Note 1] (V_{PP} pin)	I_{PPE}	When V_{PP} supply voltage = V_{PP1}				100	mA
Unit erase time	t_{er}			0.5	1	1	s
Total erase time	t_{era}					20	s
Write count		Erase/write are regarded as 1 cycle				20	Times
V_{PP} supply voltage	V_{PP0}	In normal operation		0		0.2V_{DD}	V
	V_{PP1}	During flash memory programming		9.7	10.0	10.3	V

Notes 1. The port current (including the current that flows to the on-chip pull-up resistors) is not included.

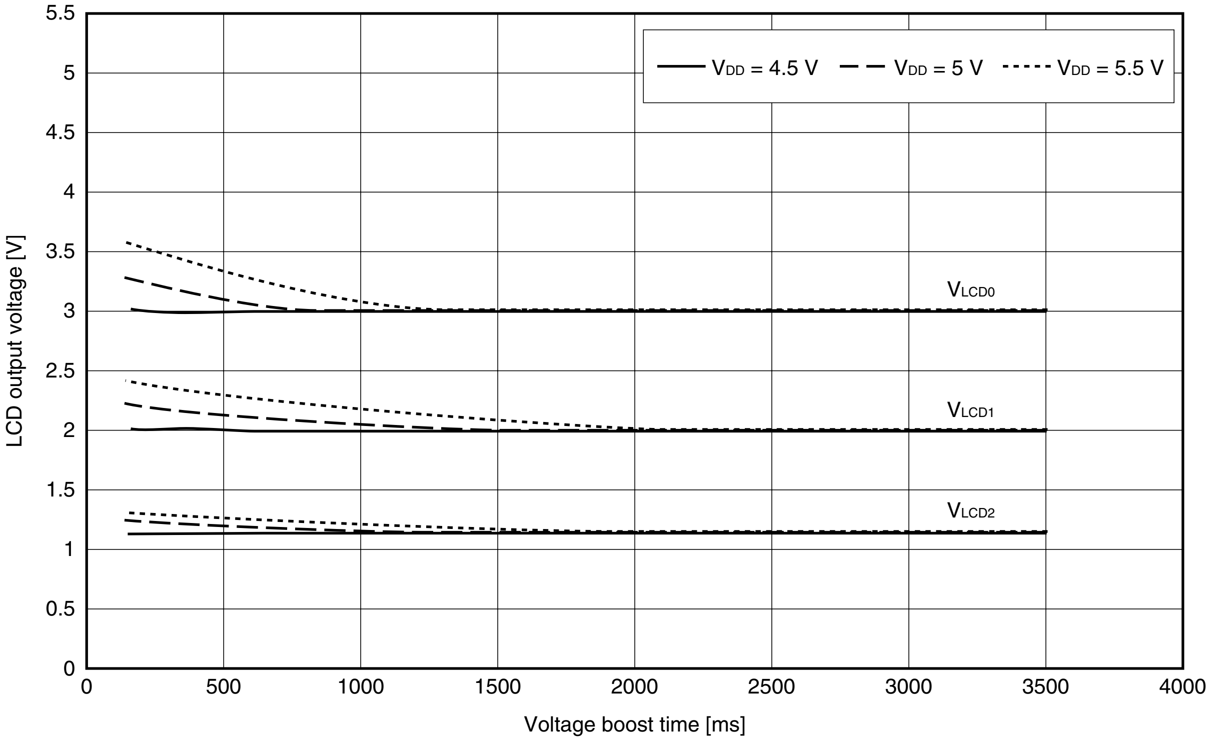
2. When an external clock is input

CHAPTER 23 CHARACTERISTICS CURVES OF LCD CONTROLLER/DRIVER (REFERENCE VALUES)

(1) Characteristics curves of voltage boost stabilization time

The following shows the characteristics curves of the time from the start of voltage boost (VAON0 = 1) and the changes in the LCD output voltage (when GAIN is set as 1 (using the 3 V display panel)).

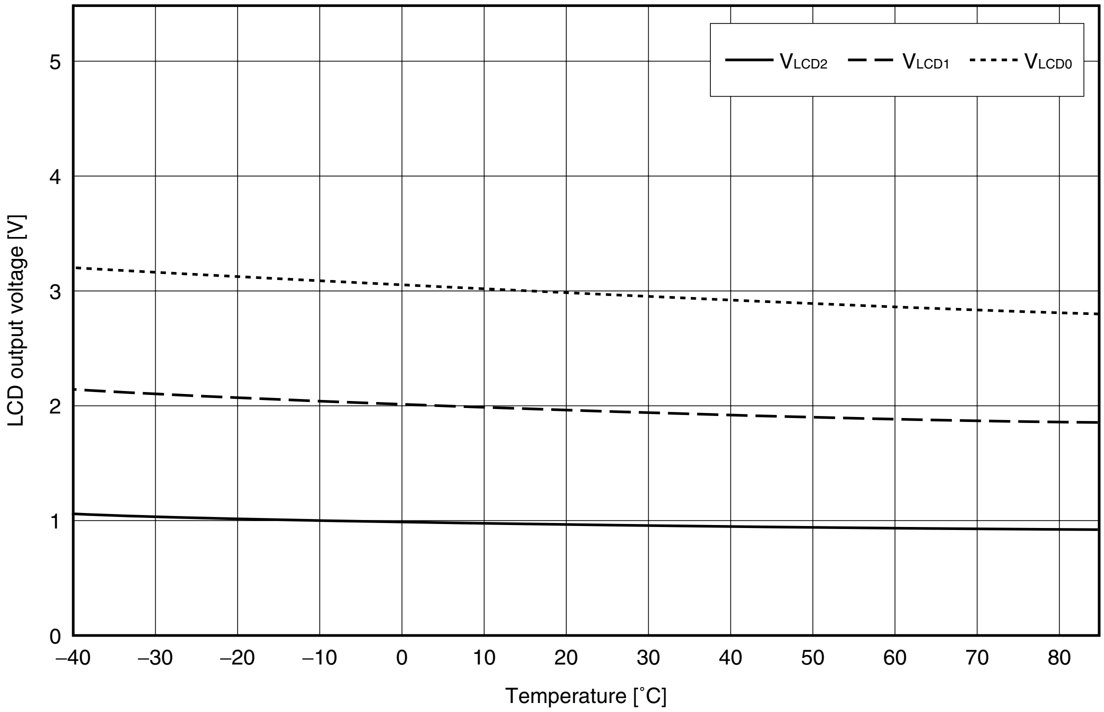
LCD Output Voltage/Voltage Boost Time



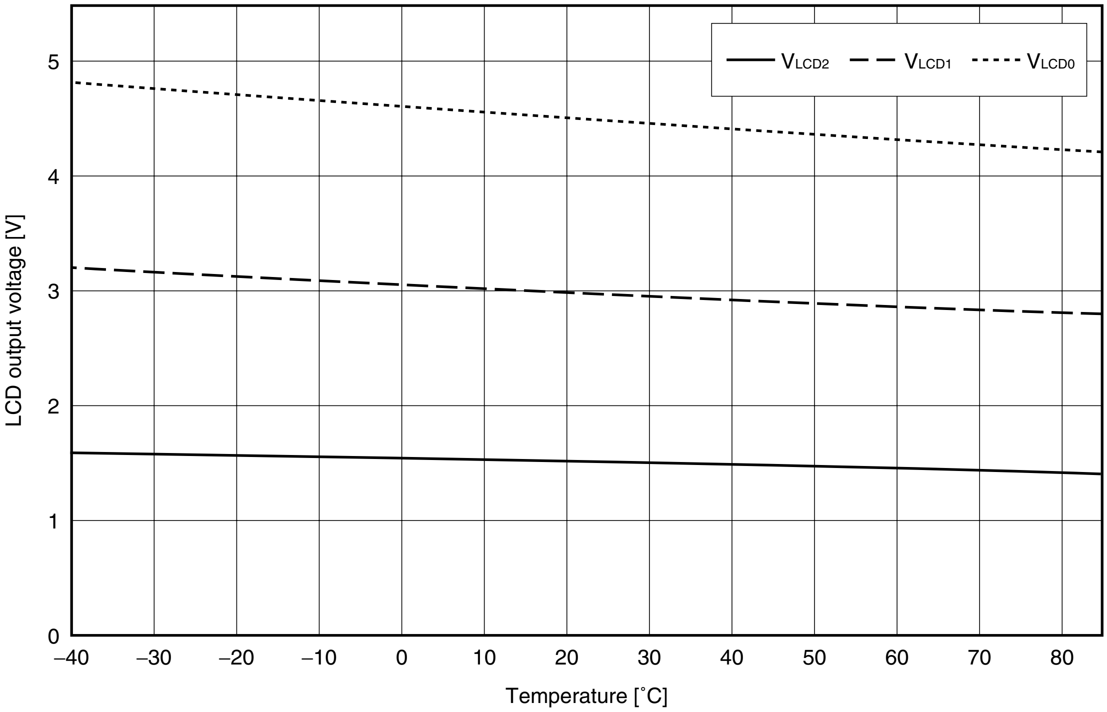
(2) Temperature characteristics of LCD output voltage

The following shows the temperature characteristics curves of LCD output voltage.

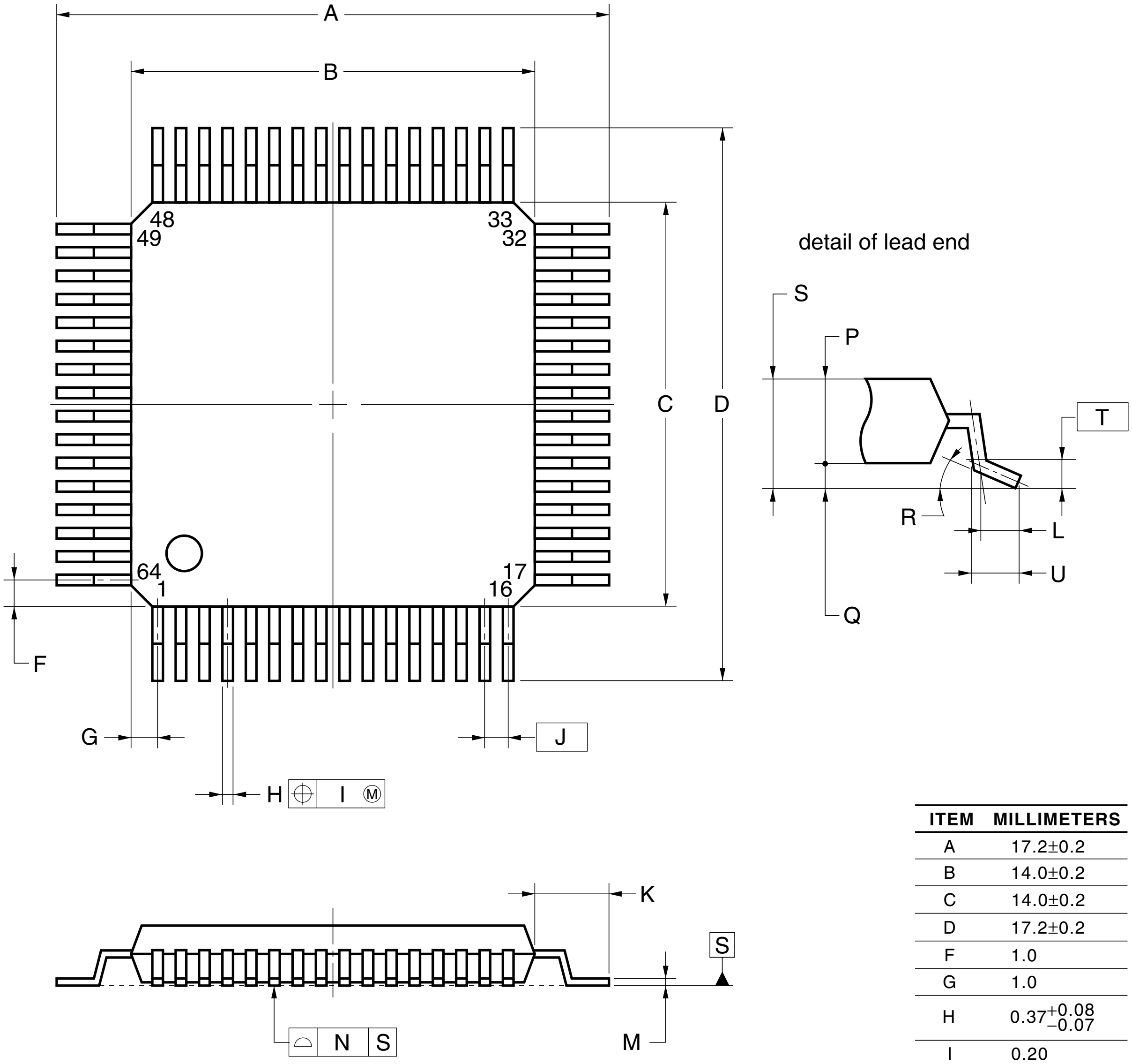
LCD Output Voltage/Temperature (When GAIN = 1)



LCD Output Voltage/Temperature (When GAIN = 0)



★ 64-PIN PLASTIC LQFP (14x14)



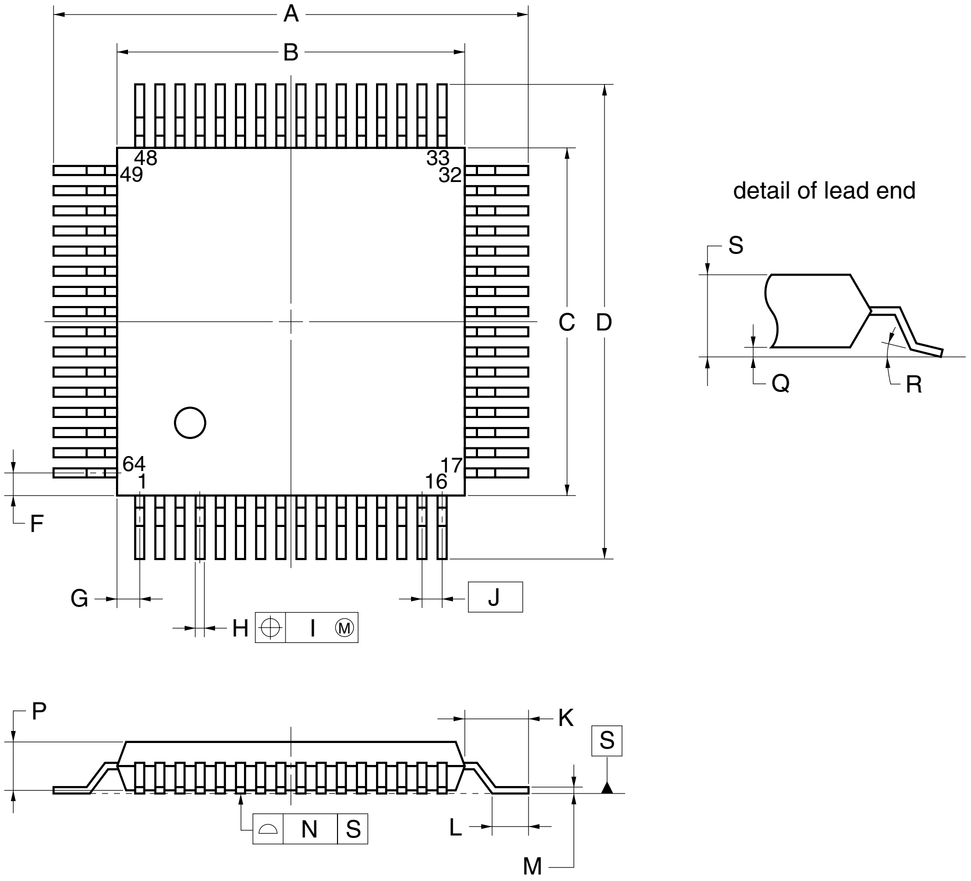
ITEM	MILLIMETERS
A	17.2±0.2
B	14.0±0.2
C	14.0±0.2
D	17.2±0.2
F	1.0
G	1.0
H	$0.37^{+0.08}_{-0.07}$
I	0.20
J	0.8 (T.P.)
K	1.6±0.2
L	0.8
M	$0.17^{+0.03}_{-0.06}$
N	0.10
P	1.4±0.1
Q	0.127±0.075
R	$3°^{+4°}_{-3°}$
S	1.7 MAX.
T	0.25
U	0.886±0.15

P64GC-80-8BS

NOTE

Each lead centerline is located within 0.20 mm of its true position (T.P.) at maximum material condition.

64-PIN PLASTIC QFP (14x14)



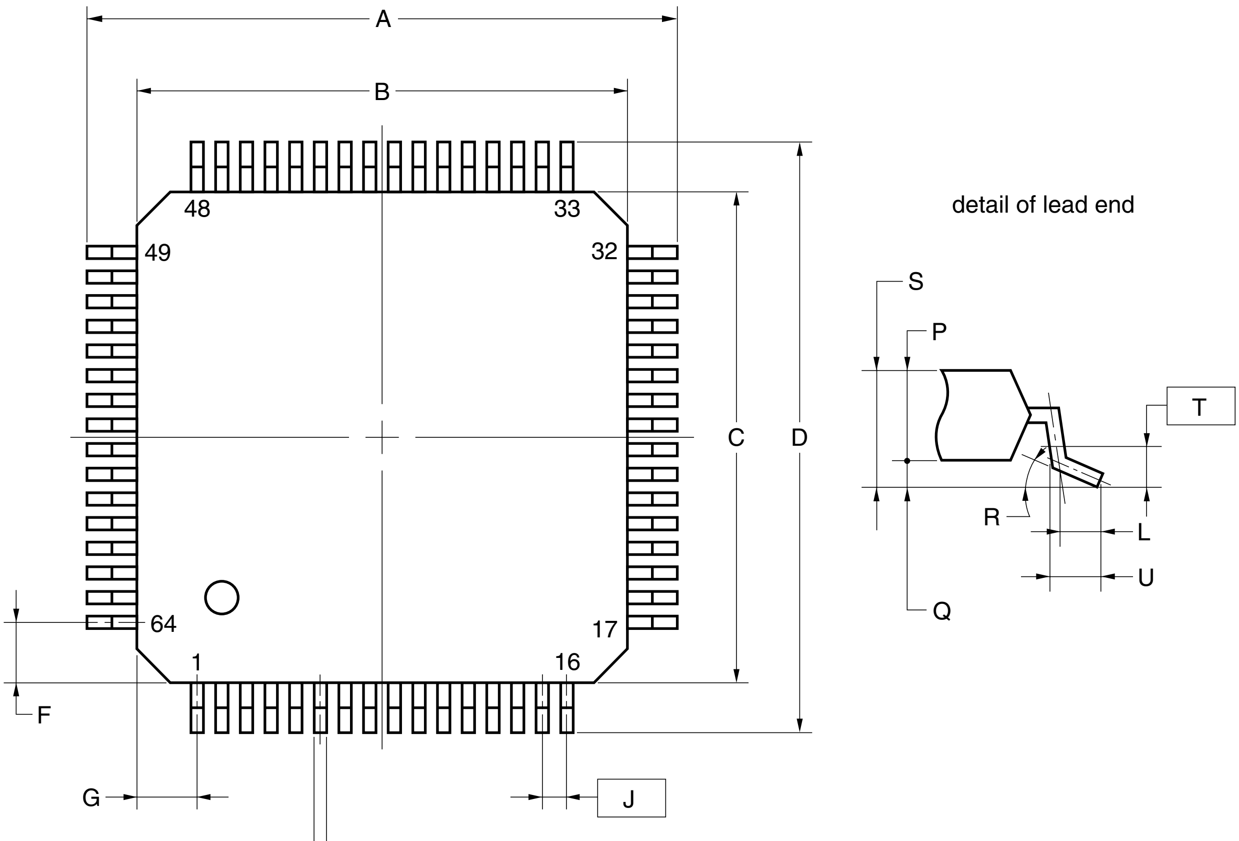
NOTE

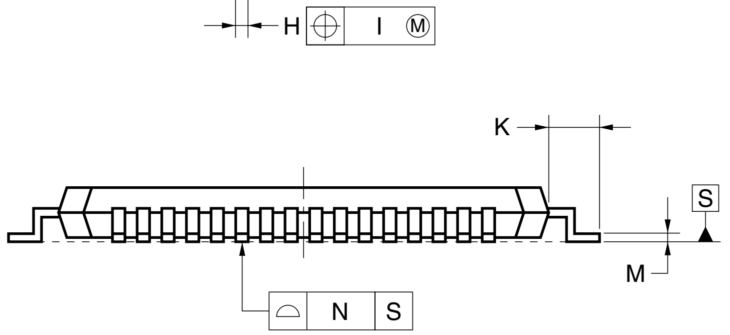
Each lead centerline is located within 0.15 mm of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS
A	17.6±0.4
B	14.0±0.2
C	14.0±0.2
D	17.6±0.4
F	1.0
G	1.0
H	$0.37^{+0.08}_{-0.07}$
I	0.15
J	0.8 (T.P.)
K	1.8±0.2
L	0.8±0.2
M	$0.17^{+0.08}_{-0.07}$
N	0.10
P	2.55±0.1
Q	0.1±0.1
R	5°±5°
S	2.85 MAX.

P64GC-80-AB8-5

64-PIN PLASTIC TQFP (12x12)





NOTE

Each lead centerline is located within 0.13 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	1.125
G	1.125
H	$0.32^{+0.06}_{-0.10}$
I	0.13
J	0.65 (T.P.)
K	1.0±0.2
L	0.5
M	$0.17^{+0.03}_{-0.07}$
N	0.10
P	1.0
Q	0.1±0.05
R	$3°^{+4°}_{-3°}$
S	1.1±0.1
T	0.25
U	0.6±0.15

P64GK-65-9ET-3

CHAPTER 25 RECOMMENDED SOLDERING CONDITIONS

The μPD789306 and μPD789316 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 25-1. Surface Mounting Type Soldering Conditions (1/3)

μPD789304GC-×××-AB8:	64-pin plastic QFP (14 × 14)
μPD789306GC-×××-AB8:	64-pin plastic QFP (14 × 14)
μPD789314GC-×××-AB8:	64-pin plastic QFP (14 × 14)
μPD789316GC-×××-AB8:	64-pin plastic QFP (14 × 14)
μPD78F9306GC-AB8:	64-pin plastic QFP (14 × 14)
μPD78F9316GC-AB8:	64-pin plastic QFP (14 × 14)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 235°C, Time: 30 seconds max. (at 210°C or higher), Count: three times or less	IR35-00-3
VPS	Package peak temperature: 215°C, Time: 40 seconds max. (at 200°C or higher), Count: three times or less	VP15-00-3
Wave soldering	Soldering bath temperature: 260°C max., Time: 10 seconds max., Count: 1, 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).

Table 25-1. Surface Mounting Type Soldering Conditions (2/3)

μPD789304GK-×××-9ET:	64-pin plastic TQFP (fine pitch) (12 × 12)
μPD789306GK-×××-9ET:	64-pin plastic TQFP (fine pitch) (12 × 12)
μPD789314GK-×××-9ET:	64-pin plastic TQFP (fine pitch) (12 × 12)
μPD789316GK-×××-9ET:	64-pin plastic TQFP (fine pitch) (12 × 12)
μPD78F9306GK-9ET:	64-pin plastic TQFP (fine pitch) (12 × 12)
μPD78F9316GK-9ET:	64-pin plastic TQFP (fine pitch) (12 × 12)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 235°C, Time: 30 seconds max. (at 210°C or higher), Count: Twice or less, Exposure limit: 7 days[Note] (after that, prebake at 125°C for 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	Soldering bath temperature: 260°C max., Time: 10 seconds max., Count: 1, 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 pack, store it at 25°C or less and 65% RH or less for the allowable storage period.

Caution Do not use different soldering methods together (except for partial heating).

★

Table 25-1. Surface Mounting Type Soldering Conditions (3/3)

μPD789304GK-×××-9ET-A:	64-pin plastic TQFP (fine pitch) (12 × 12)
μPD789306GK-×××-9ET-A:	64-pin plastic TQFP (fine pitch) (12 × 12)
μPD789314GK-×××-9ET-A:	64-pin plastic TQFP (fine pitch) (12 × 12)
μPD789316GK-×××-9ET-A:	64-pin plastic TQFP (fine pitch) (12 × 12)
μPD78F9306GK-9ET-A:	64-pin plastic TQFP (fine pitch) (12 × 12)
μPD78F9316GK-9ET-A:	64-pin plastic TQFP (fine pitch) (12 × 12)

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 μPD789306 and μPD789316 Subseries.

Figure A-1 shows development tools.

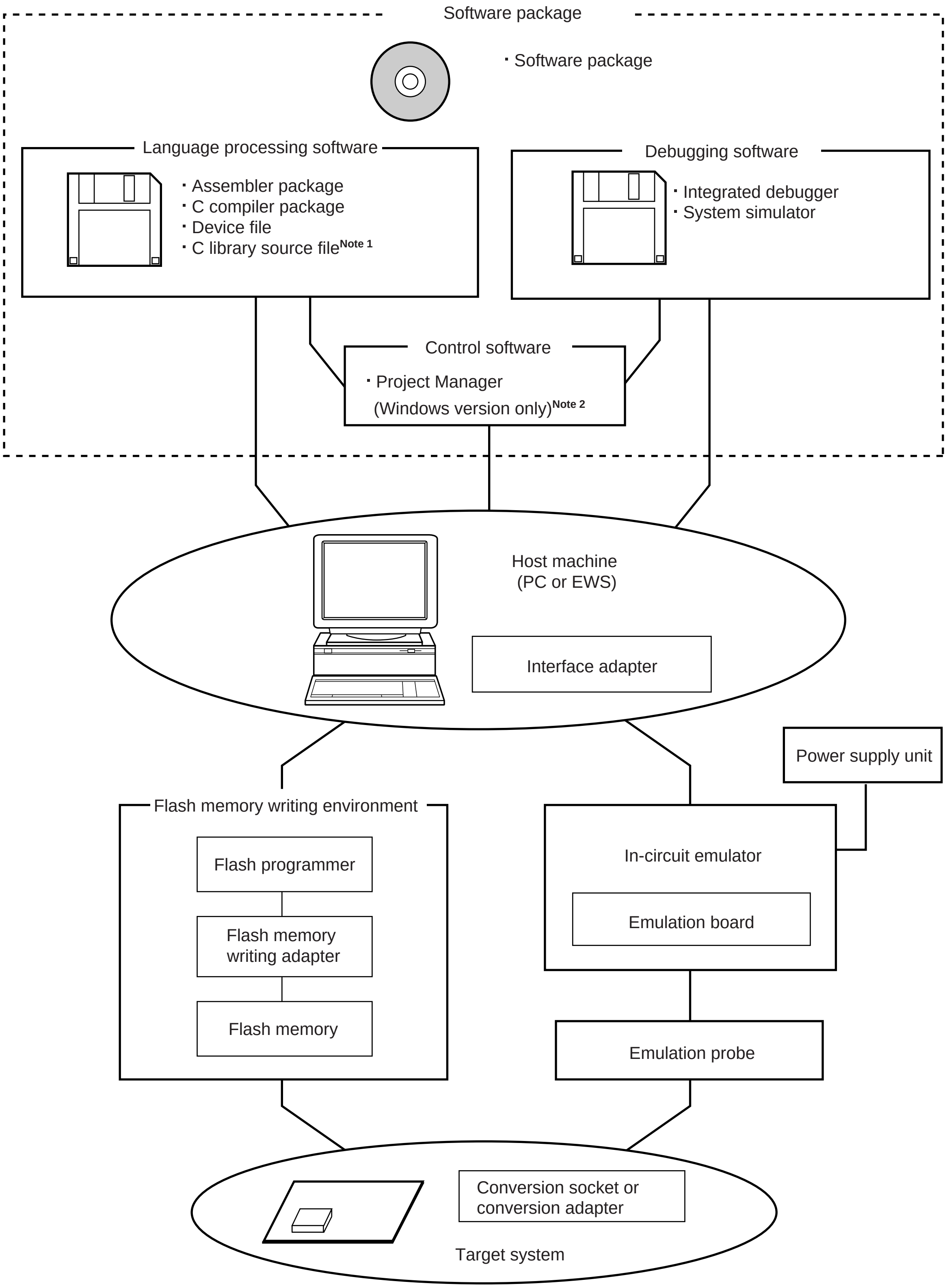
- Support of PC98-NX series

  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. C library source file is not included in the software package.

2. Project Manager is included in the assembler package.
Project Manager is used only in the Windows environment.

A.1 Software Package

SP78K0S Software package	Software tools for development of the 78K/0S Series 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 operating system to be 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 microcontroller. In addition, automatic functions to generate a symbol table and optimize branch instructions are also provided. Used in combination with a device file (DF789306) (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 microcontroller. Used in combination with an assembler package (RA78K0S) and device file (DF789306) (both sold separately). **<Caution when used in PC environment>** The C compiler package is a DOS-based application but may be used in the Windows environment by using the Project Manager of Windows (included in the assembler package).
	Part number: μS××××CC78K0S
★ DF789306[Note 1] Device file	File containing the information inherent to the device. Used in combination with the RA78K0S, CC78K0S, ID78K0S-NS, and SM78K0S (all sold separately).
	Part number: μS××××DF789306
CC78K0S-L[Note 2] C library source file	Source file of functions for generating object library included in C compiler package. Necessary for changing object library included in C compiler package according to customer's specifications. Since this is a source file, its working environment does not depend on any particular operating system.
	Part number: μS××××CC78K0S-L

Notes 1. DF789306 is a common file that can be used with RA78K0S, CC78K0S, ID78K0S-NS, and SM78K0S.

2. CC78K0S-L is not included in the software package (SP78K0S).

Remark ×××× in the part number differs depending on the host machine and operating system to be used.

*μ*S××××RA78K0S
*μ*S××××CC78K0S

××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT compatibles	Japanese Windows	3.5-inch 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)	

★ *μ*S××××DF789306
*μ*S××××CC78K0S-L

××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT compatibles	Japanese Windows	3.5-inch 2HD FD
BB13		English Windows	
3P16	HP9000 series 700	HP-UX (Rel. 10.10)	DAT
3K13	SPARCstation	SunOS (Rel. 4.1.4), Solaris (Rel. 2.5.1)	3.5-inch 2HD FD
3K15			1/4-inch CGMT

A.3 Control Software

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 Project Manager. <Caution> The Project Manager is included in the assembler package (RA78K0S). The Project Manager 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-64GC FA-64GK-9ET Flash memory writing adapter	Adapter for writing to flash memory and connected to Flashpro III or Flashpro IV. • FA-64GC: For 64-pin plastic QFP (GC-AB8 type) • FA-64GK-9ET: For 64-pin plastic TQFP (GK-9ET type)

Remark The FL-PR3, FL-PR4, FA-64GC, and FA-64GK-9ET are products made by Naito Densei Machida Mfg. Co., Ltd. (TEL +81-45-475-4191).

A.5 Debugging Tools (Hardware)

IE-78K0S-NS In-circuit emulator		In-circuit emulator for debugging a hardware and software of application system using the 78K/0S Series. Supports an integrated debugger (ID78K0S-NS). Used in combination with an AC adapter, emulation probe, and interface adapter for connecting the host machine.
IE-78K0S-NS-A In-circuit emulator		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		Adapter necessary when using an IBM PC/AT compatible as the host machine of the (ISA bus supported)
IE-70000-PCI-IF-A Interface adapter		Adapter necessary when using a personal computer incorporating the PCI bus as the host machine
IE-789306-NS-EM1 Emulation board		Board for emulating the peripheral hardware specific to the device. Used in combination with an in-circuit emulator.
NP-64GC Emulation probe		Cable to connect an in-circuit emulator to the target system. Used in combination with the EV-9200G-64.
	EV-9200G-64 Conversion socket	Conversion socket to connect the NP-64GC to a target system board on which an 64-pin plastic QFP (GC-AB8 type) can be mounted.
NP-64GC-TQ NP-H64GC-TQ Emulation probe		Cable to connect an in-circuit emulator to the target system. Used in combination with the TGB-064SAP.
	TGB-064SAP Conversion adapter	Conversion adapter to connect the NP-64GC-TQ or NP-H64GC-TQ to a target system board on which an 64-pin plastic QFP (GC-AB8 type) can be mounted.
NP-64GK NP-H64GK-TQ Emulation probe		Cable to connect an in-circuit emulator to the target system. Used in combination with the TGK-064SBW.
	TGK-064SBW Conversion adapter	Conversion adapter to connect the NP-64GK or NP-H64GK-TQ to a target system board on which an 64-pin plastic TQFP (fine pitch) (GK-9ET type) can be mounted.

Remarks 1. The NP-64GC, NP-64GC-TQ, NP-H64GC-TQ, NP-64GK, and NP-H64GK-TQ are products made by Naito Densei Machida Mfg. Co., Ltd. (TEL +81-45-475-4191).

2. The TGC-064SAP and TGK-064SBW are products made by TOKYO ELETECH CORPORATION.
For further information, contact: Daimaru Kogyo, Ltd.
Tokyo Electronics Department (TEL +81-3-3820-7112)
Osaka Electronics Department (TEL +81-6-6244-6672)

A.6 Debugging Tools (Software)

ID78K0S-NS Integrated debugger	This debugger supports the in-circuit emulators IE-78K0S-NS and IE-78K0S-NS-A for the 78K/0S Series. The ID78K0S-NS is Windows-based software. It has improved C-compatible debugging functions and can display the results of tracing with the source program using an integrating window function that associates the source program, disassemble display, and memory display with the trace result. Used in combination with a device file (DF789306) (sold separately).
	Part number: μS××××ID78K0S-NS
SM78K0S System simulator	This is a system simulator for the 78K/0S Series. The SM78K0S is Windows-based software. It can be used to debug the target system at C source level 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 (DF789306) (sold separately).
	Part number: μS××××SM78K0S
★ DF789306[Note] Device file	File containing the information inherent to the device. Used in combination with the RA78K0S, CC78K0S, ID78K0S-NS, and SM78K0S (all sold separately).
	Part number: μS××××DF789306

Note DF789306 is a common file that can be used with RA78K0S, CC78K0S, ID78K0S-NS, and SM78K0S.

Remark ×××× in the part number differs depending on the operating system 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-inch 2HD FD
BB13		English Windows	
AB17		Japanese Windows	CD-ROM
BB17		English Windows	

APPENDIX B CAUTIONS ON DESIGNING TARGET SYSTEM

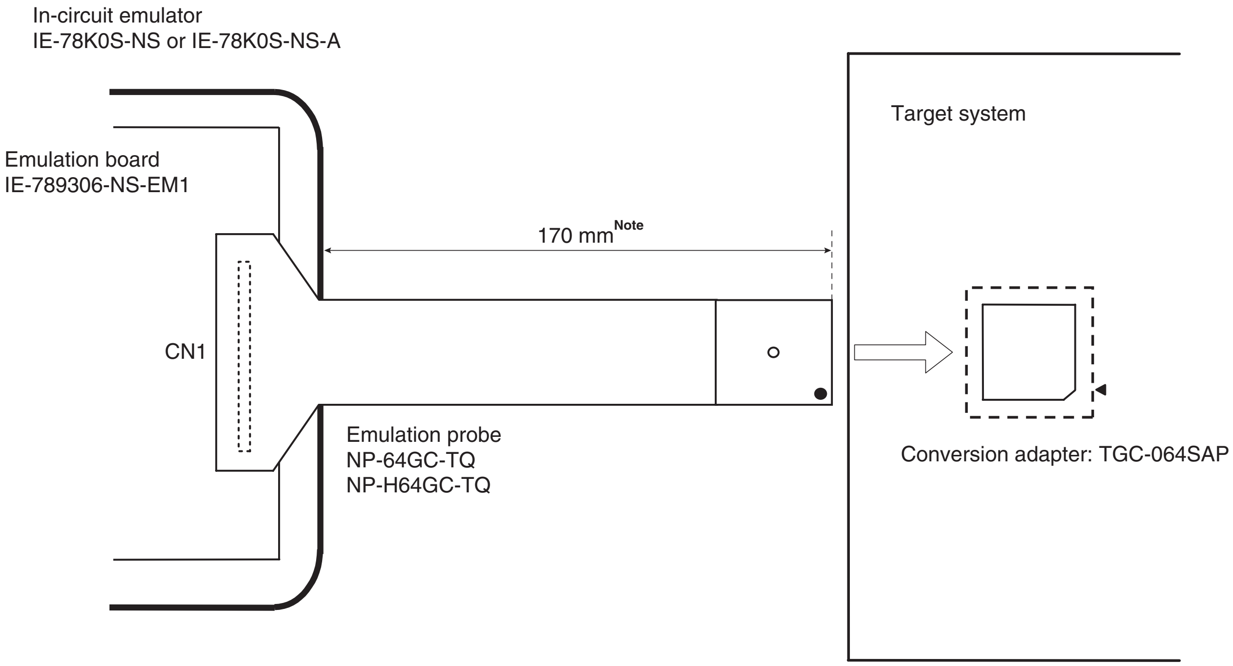
The following shows the conditions when connecting the emulation probe to the conversion adapter. Follow the configuration below and consider the shape of parts to be mounted on the target system when designing a system.

Among the products described in this appendix, NP-64GC-TQ, NP-H64GC-TQ, NP-64GK, and NP-H64GK-TQ are products of Naito Densei Machida Mfg. Co., Ltd, and TGC-064SAP and TGK-064SBW are products of TOKYO ELETECH CORPORATION.

Table B-1. Distance Between IE System and Conversion Adapter

Emulation Probe	Conversion Adapter	Distance Between IE System and Conversion Adapter
NP-64GC-TQ	TGC-064SAP	170 mm
NP-H64GC-TQ		370 mm
NP-64GK	TGK-064SBW	170 mm
NP-H64GK-TQ		370 mm

Figure B-1. Distance Between In-Circuit Emulator and Conversion Adapter (When 64GC Is Used)



Note Distance when NP-64GC-TQ is used. When NP-H64GC-TQ is used, the distance is 370 mm.

Figure B-2. Connection Conditions of Target System (When NP-64GC-TQ Is Used)

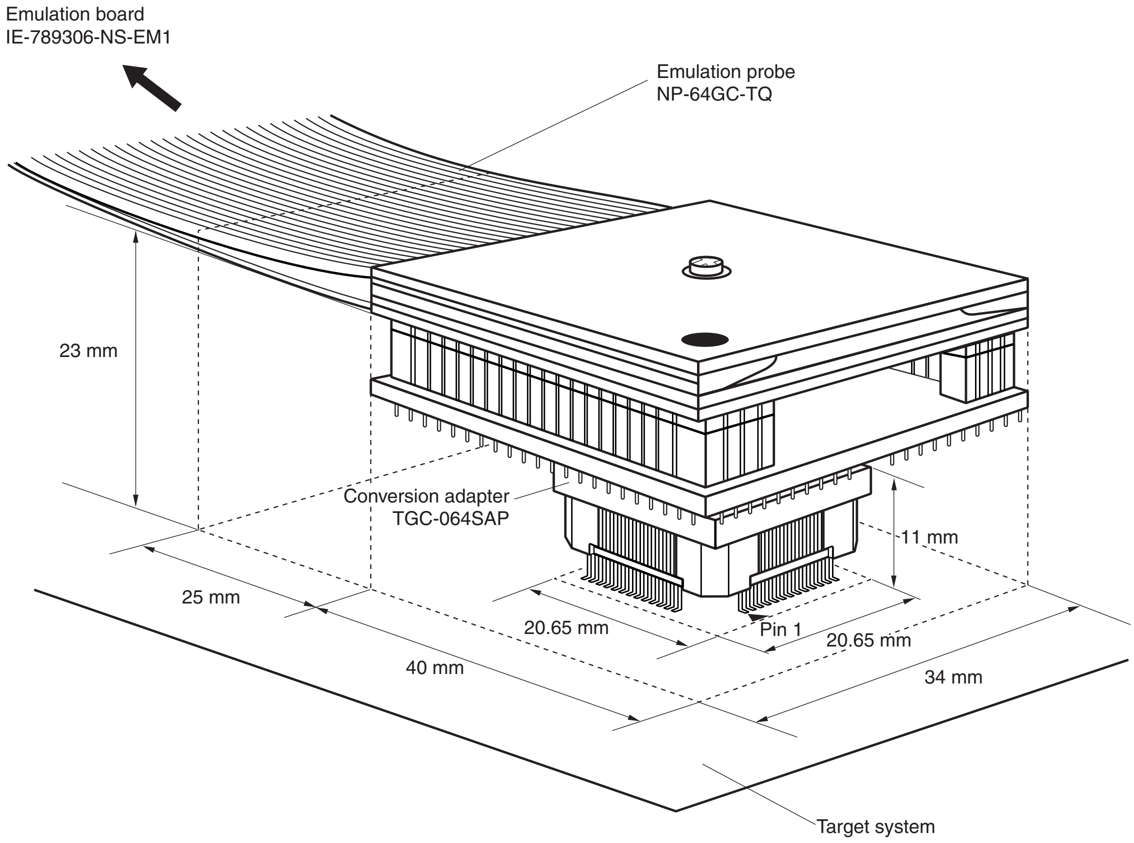


Figure B-3. Connection Conditions of Target System (When NP-H64GC-TQ Is Used)

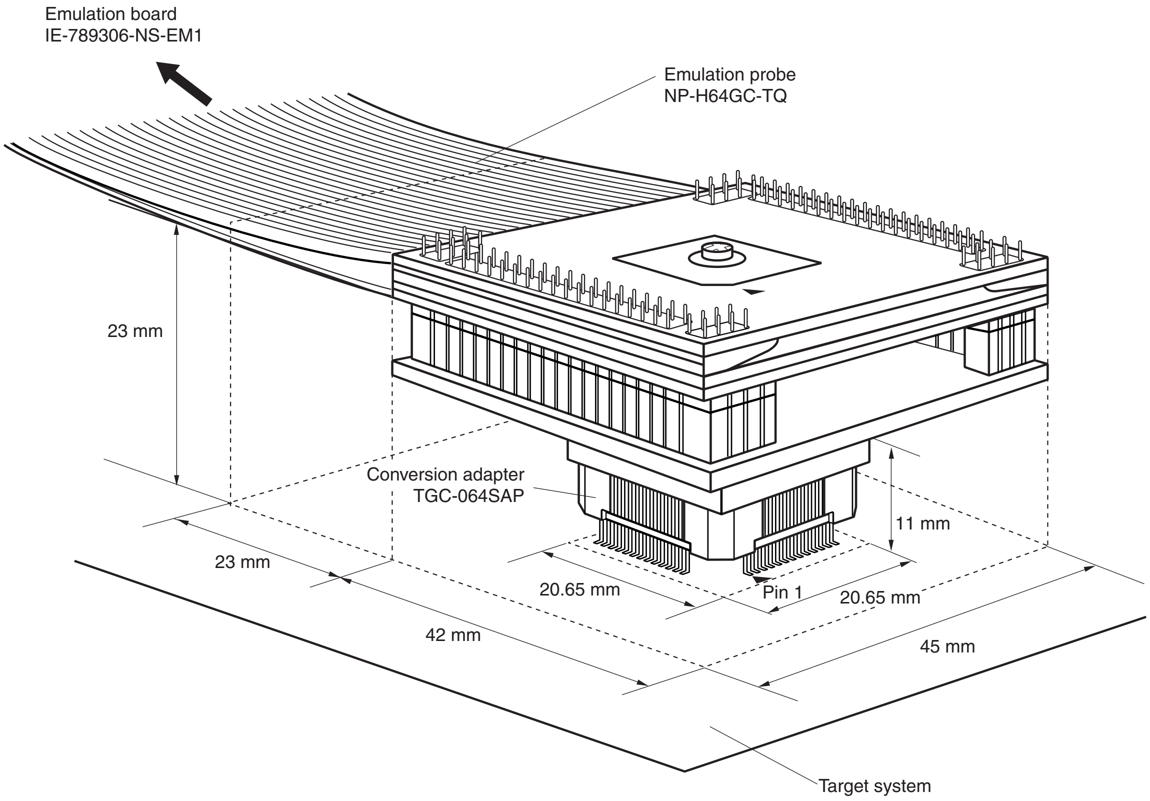
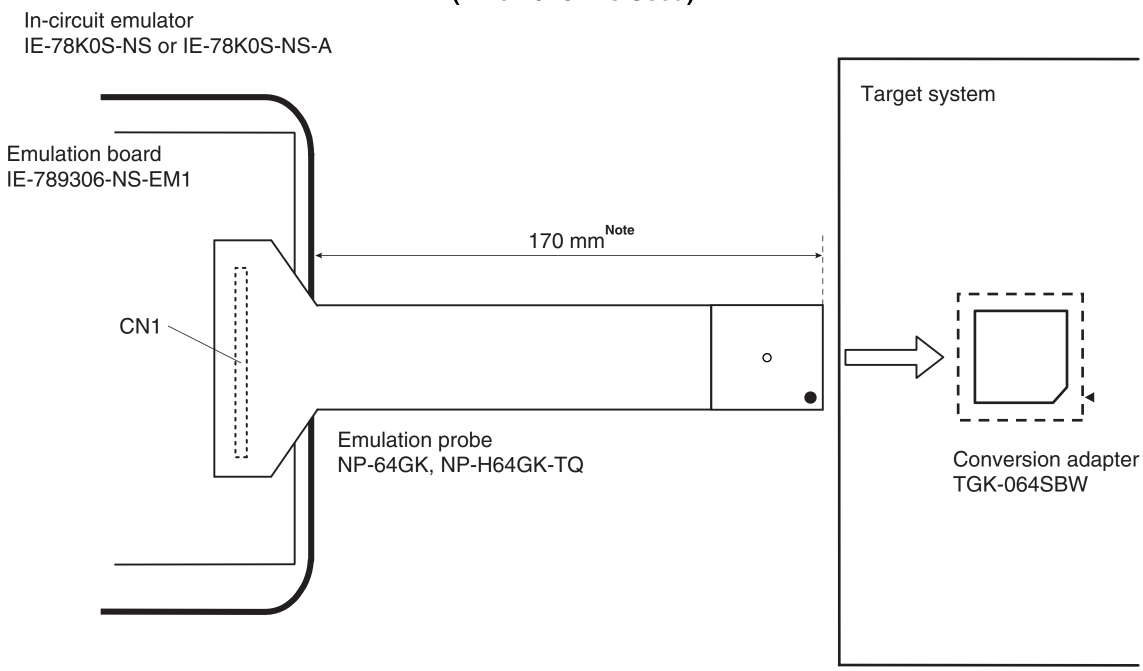


Figure B-4. Distance Between In-Circuit Emulator and Conversion Adapter (When 64GK Is Used)



Note Distance when NP-64GK is used. When NP-H64GK-TQ is used, the distance is 370 mm.

Figure B-5. Connection Conditions of Target System (When NP-64GK Is Used)

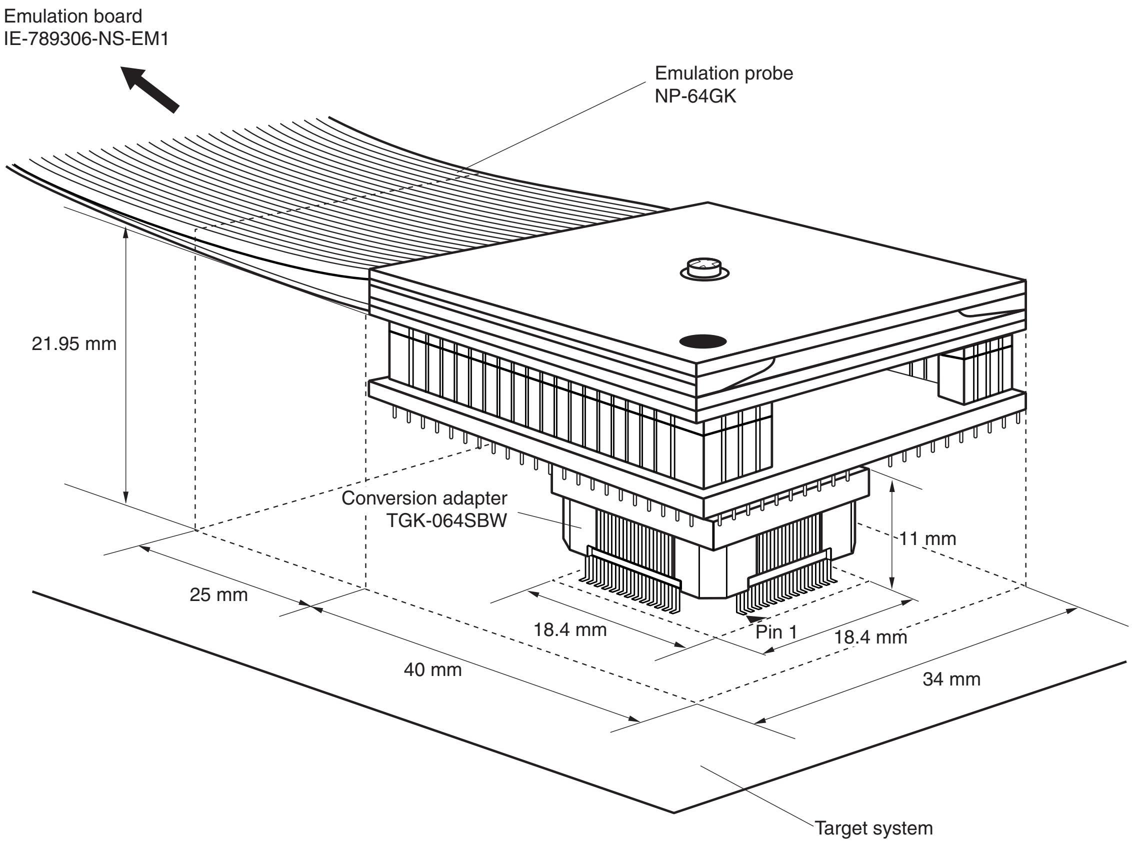
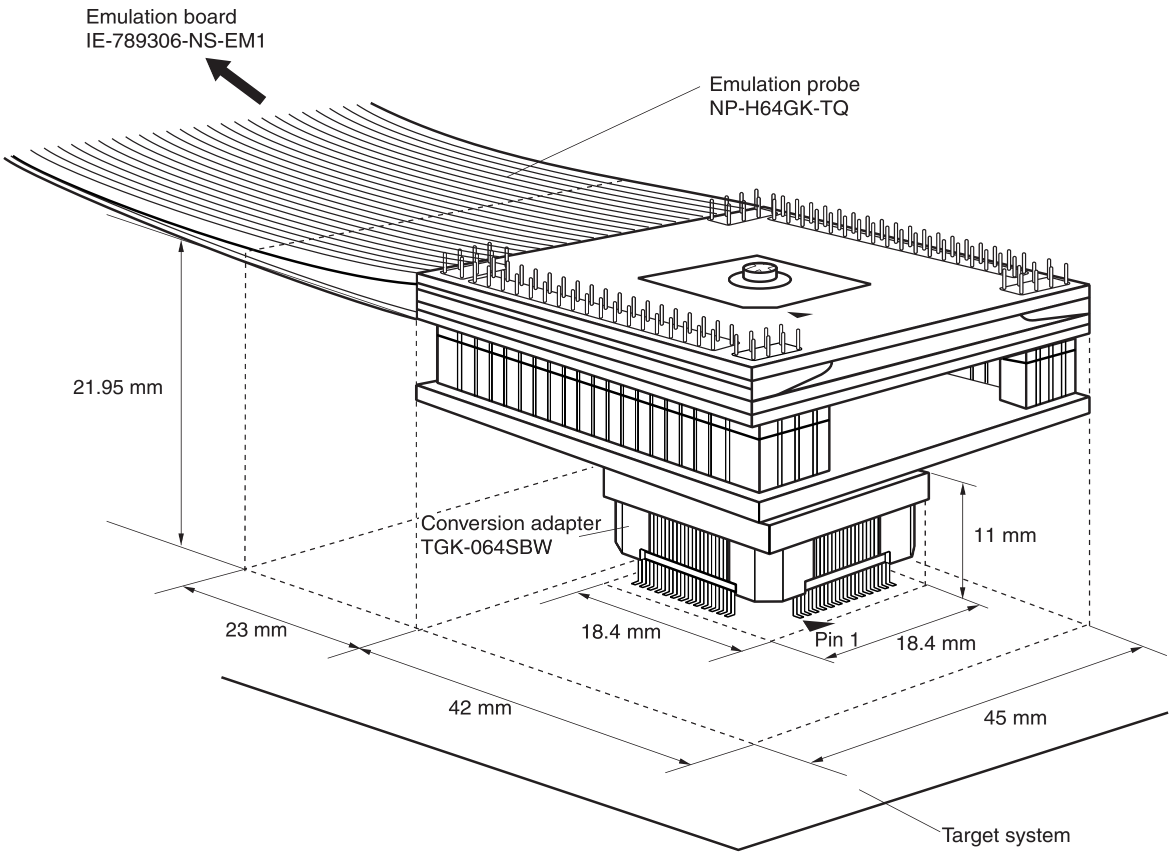


Figure B-6. Connection Conditions of Target System (When NP-H64GK-TQ Is Used)



APPENDIX C REGISTER INDEX

C.1 Register Index (Alphabetic Order of Register Name)

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

APPENDIX D REVISION HISTORY

D.1 Major Revisions in This Edition

Page	Description
pp.17, 18, 20-22, 24	**CHAPTER 1 GENERAL (μPD789306 SUBSERIES)** • Addition of lead-free products μPD789304GK-×××-9ET-A, μPD789304GC-×××-8BS-A, μPD789306GK-×××-9ET-A, μPD789306GC-×××-8BS-A, μPD78F9306GK-9ET-A, μPD78F9306GC-8BS-A • Update of **1.5 78K/0S Series Lineup** to latest version • Addition of package (GC-8BS) in **1.7 Overview of Functions**
pp.27, 28, 30-32, 34	**CHAPTER 2 GENERAL (μPD789316 SUBSERIES)** • Addition of lead-free products μPD789314GK-×××-9ET-A, μPD789314GC-×××-8BS-A, μPD789316GK-×××-9ET-A, μPD789316GC-×××-8BS-A, μPD78F9316GK-9ET-A, μPD78F9316GC-8BS-A • Update of **2.5 78K/0S Series Lineup** to latest version • Addition of package (GC-8BS) in **2.7 Overview of Functions**
p.149	**CHAPTER 10 8-BIT TIMER 30, 40** • Modification of **Figure 10-14. Timing of Operation of External Event Counter with 8-Bit Resolution**
p.222, 224, 227	**CHAPTER 15 LCD CONTROLLER/DRIVER** • Modification of **Figure 15-2. Block Diagram of LCD Controller/Driver** • Modification of **Caution** in **Figure 15-3. Format of LCD Display Mode Register 0** • Modification of **15.4 Setting LCD Controller/Driver**
p.302	**CHAPTER 22 ELECTRICAL SPECIFICATIONS** • Correction of **note** in **LCD Characteristics**
p.307	**CHAPTER 24 PACKAGE DRAWINGS** • Addition of package drawing of 64-pin plastic LQFP (GC-8BS)
p.312	**CHAPTER 25 RECOMMENDED SOLDERING CONDITIONS** • Addition of soldering conditions of lead-free products in **Table 25-1. Surface Mounting Type Soldering Conditions**
pp. 315, 316, 318	**APPENDIX A DEVELOPMENT TOOLS** • Correction of device file name from DF789316 to DF789306

D.2 Revision History of Previous Editions

The revision history is described below. The “Applied to:” column indicates the chapterin each edition.

(1/2)

Edition	Major Revision from Previous Edition	Applied to:
2nd	Modification of pin handling in **3.2.15 VPP (μPD78F9306 only)** and **4.2.15 VPP (μPD78F9316 only)**	**CHAPTER 3 PIN FUNCTIONS (μPD789306 SUBSERIES)**
	Modification of **Table 3-1** and **Table 4-1 Types of Pin Input/Output Circuits**	**CHAPTER 3 PIN FUNCTIONS (μPD789306 SUBSERIES)** **CHAPTER 4 PIN FUNCTIONS (μPD789316 SUBSERIES)**
	Correction of interrupt request neme in **Table 5-2. Vector Table**	**CHAPTER 5 CPU ARCHITECTURE**
	Modification of descriptions about **Symbol** in **5.2.3 Special function registers (SFRs)**	
	Correction of **Figure 6-3. Block Diagram of P10 to P13**	**CHAPTER 6 PORT FUNCTIONS**
	Addition of note about Feedback resistor in **Figure 7-3** and **Figure 8-3**	**CHAPTER 7 CLOCK GENERATOR (μPD789306 SUBSERIES)** **CHAPTER 8 CLOCK GENERATOR (μPD789316 SUBSERIES)**
	Modification of descriptions in **9.4.1 Operation as timer interrupt** and **9.4.2 Operation as timer output**	**CHAPTER 9 16-BIT TIMER 20**
	Addition of **9.5 Cautions on Using 16-Bit Timer 20**	
	10.2 8-Bit Timer 30, 40 Configuration • Modification of **Figure 10-3. Block Diagram of Output Controller (Timer 40)** • Modification of cautions in **(1) 8-bit compare register 30 (CR30)** • Addition of descriptions in **(2) 8-bit compare register 40 (CR40)** • Addition of descriptions in **(3) 8-bit compare register H40 (CRH40)**	**CHAPTER 10 8-BIT TIMER 30, 40**
	Addition of cautions in **Figure 10-6. Format of Carrier Generator Output Control Register 40**	
	Addition of descriptions and cautions in **10.4.3 Operation as carrier generator**	
	10.5 Notes on Using 8-Bit Timer 30, 40 • Modification of descriptions in **(1) Error on starting timer** • Addition of **(2) Count value if external clock input from TMI40 pin is selected**	

(2/2)

Edition	Description	Applied to:
2nd	Modification of **Figure 14-1. Block Diagram of Serial Interface 20**	**CHAPTER 14 SERIAL INTERFACE 20**
	Modification of description about PE flag in **Figure 14-5. Format of Asynchronous Serial Interface Status Register 20**	
	Modification of cautions in **Figure 14-6. Format of Baud Rate Generator Control Register 20**	
	Addition of description about reading receive data in **14.4.2 Asynchronous serial interface (UART) mode**	
	Division of **Figure 14-11. 3-Wire Serial I/O Mode Timing** into Master operation and Slave operation.	
	Addition of **Figure 15-1. Correspondence with LCD Display RAM**	**CHAPTER 15 LCD CONTROLLER/DRIVER**
	Modification of **Figure 15-2. Block Diagram of LCD Controller/Driver**	
	15.3 Registers Controlling LCD Controller/Driver • Modification of description about LCDON0, VAON0 in **(1) LCD display mode register 0 (LCDM0)** • Addition of description about frame frequencie in **(2) LCD clock control register 0 (LCDC0)** • Modification of description about GAIN in **(3) LCD voltage amplification control register 0 (LCDVA0)**	
	Addition of **15.8 Supplying LCD Drive Voltages V_{LC0}, V_{LC1}, and V_{LC2}**	
	Addition of cautions in **Figure 16-2. Format of Interrupt Request Flag Registers**	**CHAPTER 16 INTERRUPT FUNCTIONS**
	Addition of cautions in **Figure 16-7. Format of Key Return Mode Register 00**	
	Overall revision of contents related to flash memory programming as **19.1 Flash Memory Characteristics**	**CHAPTER 19 FLASH MEMORY**
	Addition of electrical specifications	**CHAPTER 22 ELECTRICAL SPECIFICATIONS**
	Addition of characteristics curves of LCD comtroller/driver (reference values)	**CHAPTER 23 CHARACTERISTICS CURVES OF LCD CONTROLLER/DRIVER (REFERENCE VALUES)**
	Addition of package drawings	**CHAPTER 24 PACKAGE DRAWINGS**
	Addition of recommended soldering conditions	**CHAPTER 25 RECOMMENDED SOLDERING CONDITIONS**
	Overall revision of contents of **APPENDIX A DEVELOPMENT TOOLS** Deletion of embedded software	**APPENDIX A DEVELOPMENT TOOLS**
	Addition of cautions on designing target system	**APPENDIX B CAUTIONS ON DESIGNING TARGET SYSTEM**
	Addition of revision history	**APPENDIX D REVISION HISTORY**