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

μPD789881 Subseries

8-Bit Single-Chip Microcontrollers

μPD789881
μPD78F9882
μPD789881(A)

Document No. U15172EJ2V1UD00 (2nd edition)
Date Published August 2005 N CP(K)

© NEC Electronics Corporation 2002
Printed in Japan

[MEMO]

NOTES FOR CMOS DEVICES

① VOLTAGE APPLICATION WAVEFORM AT INPUT PIN

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

② HANDLING OF UNUSED INPUT PINS

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

③ PRECAUTION AGAINST ESD

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

④ STATUS BEFORE INITIALIZATION

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

⑤ POWER ON/OFF SEQUENCE

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

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

⑥ INPUT OF SIGNAL DURING POWER OFF STATE

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

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

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

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

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

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

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

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

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

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

(Note)

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

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

M8E 02. 11-1

Regional Information

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

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

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

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

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

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

- **Sucursal en España**
  Madrid, Spain
  Tel: 091-504 27 87
- **Succursale Française**
  Vélizy-Villacoublay, France
  Tel: 01-30-67 58 00
- **Filiale Italiana**
  Milano, Italy
  Tel: 02-66 75 41
- **Branch The Netherlands**
  Eindhoven, The Netherlands
  Tel: 040-265 40 10
- **Tyskland Filial**
  Taeby, Sweden
  Tel: 08-63 87 200
- **United Kingdom Branch**
  Milton Keynes, UK
  Tel: 01908-691-133

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

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

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

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

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

J05.6

Major Revisions in This Edition

Pages	Description
Major revisions in modification version (U15172EJ2V1UD00)	
pp. 23, 24	**CHAPTER 1 GENERAL** • Addition of lead-free products
p. 235	**CHAPTER 21 RECOMMENDED SOLDERING CONDITIONS** • Addition of soldering conditions of lead-free products in **Table 21-1 Surface Mounting Type Soldering Conditions**

The mark ★ shows major revised points.

INTRODUCTION

Target Readers

This manual is intended to give user engineers an understanding of the functions of the μPD789881 Subseries to design and develop its application systems and programs.

Target products:

- μPD789881 Subseries: μPD789881, 78F9882, 789881(A)

Purpose

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

Organization

Two manuals are available for the μPD789881 Subseries:
This manual and the instruction manual (common to the 78K/0S Series).

μPD789881 Subseries User's Manual	78K/0S Series User's Manual Instructions
• Pin functions • Internal block functions • Interrupts • Other internal peripheral functions • Electrical specifications	• CPU function • Instruction set • Instruction description

How to Use This Manual

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

- For users who use this document as the manual for the μPD789881(A)
  → The only differences between standard products and (A) products are the quality grades (refer to **1.9 Differences Between Standard Quality Grade Products and (A) Products**). For the (A) products, read the part numbers in the following manner.
  μPD789881 → μPD789881(A)
- To understand the overall functions of the μPD789881 Subseries
  → Read this manual in the order of the **CONTENTS**.
- How to read register formats
  → The name of a bit whose number is enclosed in brackets is reserved for the assembler and is defined for the C compiler by the header file sfrbit.h.
- To learn the detailed functions of a register whose register name is known
  → Refer to **APPENDIX B REGISTER INDEX**.
- To learn the details of the instruction functions of the 78K/0S Series
  → Refer to **78K/0S Series Instructions User's Manual (U11047E)** separately available.
- To know the electrical specifications of the μPD789881 Subseries
  → Refer to **CHAPTER 19 ELECTRICAL SPECIFICATIONS**.

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

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.
μPD789881 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-789882-NS-EM1 Emulation Board	U16431E

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

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

Other Related Documents

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

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

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

CONTENTS

LIST OF FIGURES (1/4)

Figure No. Title Page

LIST OF FIGURES (2/4)

Figure No. Title Page

LIST OF FIGURES (3/4)

LIST OF FIGURES (4/4)

LIST OF TABLES (1/2)

Table No. Title Page

LIST OF TABLES (2/2)

CHAPTER 1 GENERAL

1.1 Features

- ROM and RAM capacities

Item Part Number	Program Memory (ROM)		Data Memory	
			Internal High-Speed RAM	LCD Display RAM
μPD789881	Mask ROM	16 KB	512 bytes	26 × 4 bits
μPD78F9882	Flash memory	32 KB		

- Minimum instruction execution time can be changed from high speed (4 μs: @500 kHz (TYP.) operation with main system clock) to ultra low speed (30.5 μs: @32.768 kHz × 4 operation with subsystem clock)
- I/O ports: 28 (N-ch open-drain: 4)
- Timer: 4 channels
- Serial interface (UART): 1 channel
- LCD controller/driver (capacitor step down type)
  Segment signals: 26, common signals: 4
- On-chip multiplier: 8 bits × 8 bits = 16 bits
- On-chip voltage halver circuit and regulator circuit
- Supply voltage:
  μPD789881: V_{DD} = 2.7 to 3.6 V
  μPD78F9882: V_{DD} = 3.0 to 3.6 V

1.2 Applications

Heating meters, gas meters, water meters, healthcare equipment, etc.

★ 1.3 Ordering Information

Part Number	Package	Internal ROM
μPD789881GB-×××-8EU	64-pin plastic LQFP (10 × 10)	Mask ROM
μPD78F9882GB-8EU	64-pin plastic LQFP (10 × 10)	Flash memory
μPD789881GB-×××-8EU-A	64-pin plastic LQFP (10 × 10)	Mask ROM
μPD78F9882GB-8EU-A	64-pin plastic LQFP (10 × 10)	Flash memory
μPD789881GB(A)-×××-8EU	64-pin plastic LQFP (10 × 10)	Mask ROM

Remarks 1. Products that have the part numbers suffixed by "-A" are lead-free products.

2. ××× indicates ROM code suffix.

1.4 Quality Grade

	Part Number	Package	Quality Grade
	μPD789881GB-×××-8EU	64-pin plastic LQFP (10 × 10)	Standard
	μPD78F9882GB-8EU	64-pin plastic LQFP (10 × 10)	Standard
★	μPD789881GB-×××-8EU-A	64-pin plastic LQFP (10 × 10)	Standard
★	μPD78F9882GB-8EU-A	64-pin plastic LQFP (10 × 10)	Standard
	μPD789881GB(A)-×××-8EU	64-pin plastic LQFP (10 × 10)	Special

Remarks 1. Products that have the part numbers suffixed by "-A" are lead-free products.

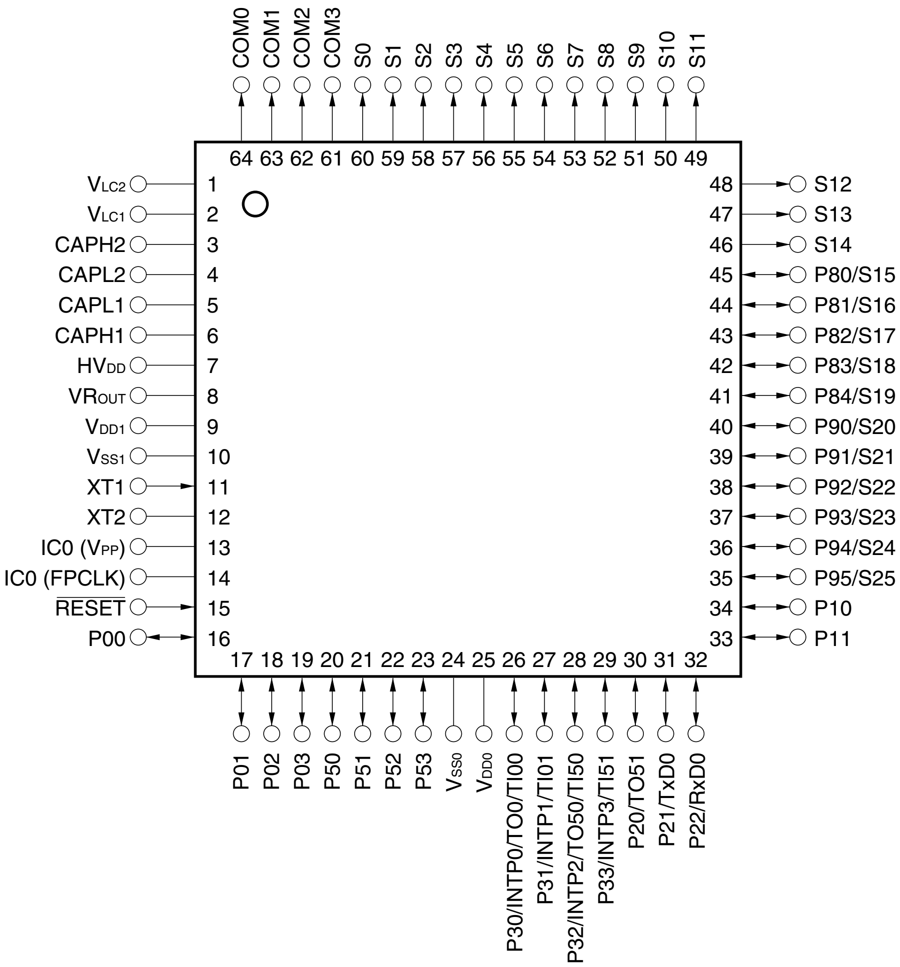
2. ××× indicates ROM code suffix.

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

1.5 Pin Configuration (Top View)

64-pin plastic LQFP (10 × 10)

★	μPD789881GB-×××-8EU	μPD789881GB-×××-8EU-A
★	μPD78F9882GB-8EU	μPD78F9882GB-8EU-A
	μPD789881GB(A)-×××-8EU	



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

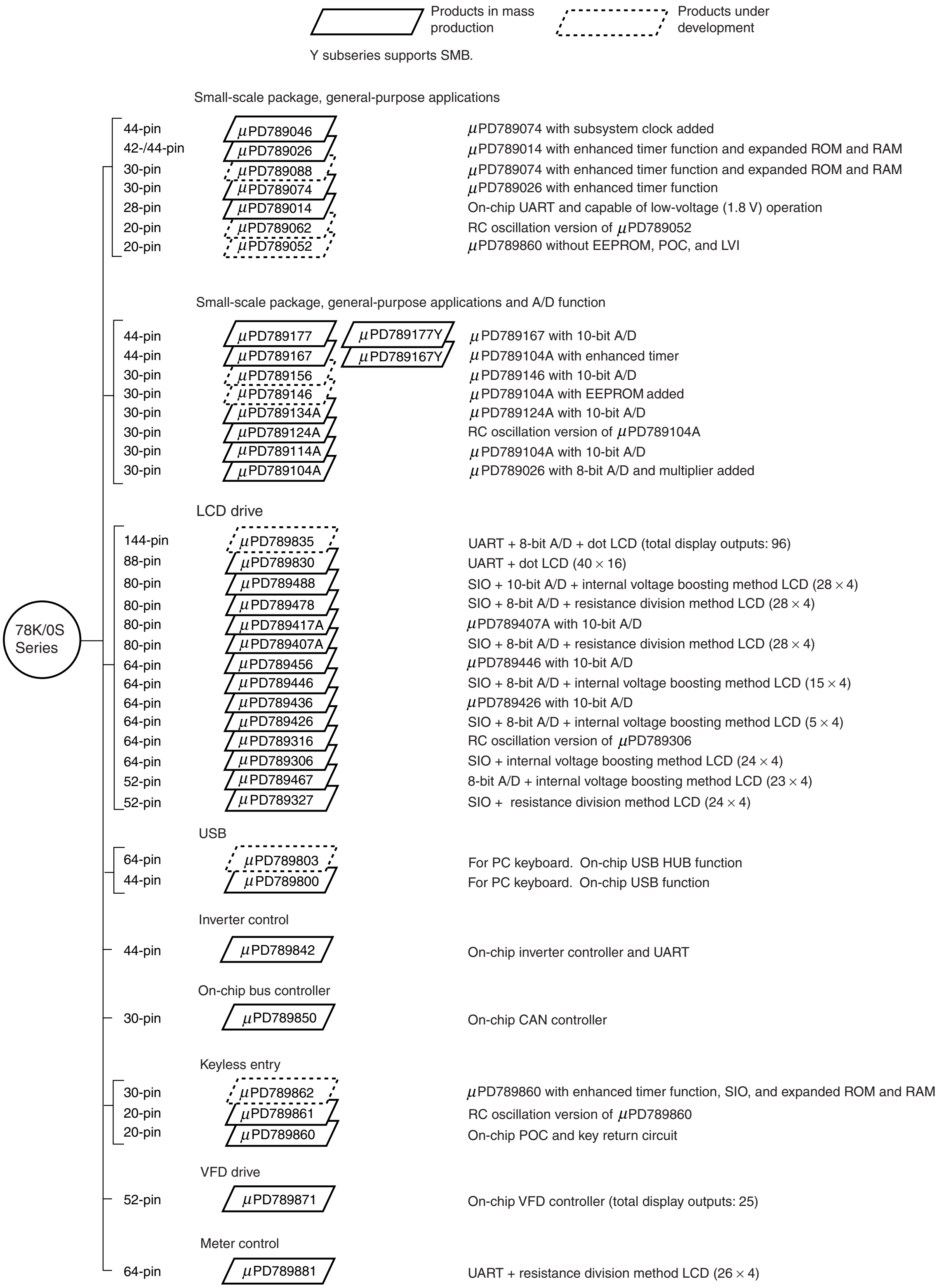
Remark The parenthesized values apply to the μPD78F9882.

Pin Name

CAPH1, CAPL1:	Voltage halver capacitance
CAPH2, CAPL2:	LCD power supply capacitance control
COM0 to COM3:	Common output
FPCLK:	Flash clock input
HV$_{DD}$:	Voltage halver capacitance
IC0:	Internally connected
INTP0 to INTP3:	External interrupt input
P00 to P03:	Port 0
P10, P11:	Port 1
P20 to P22:	Port 2
P30 to P33:	Port 3
P50 to P53:	Port 5
P80 to P84:	Port 8
P90 to P95:	Port 9
$\overline{\text{RESET}}$:	Reset
RxD0:	Receive data
S0 to S25:	Segment output
TI00, 01, 50, 51:	Timer input
TO00, 50, 51:	Timer output
TxD0:	Transmit data
V$_{DD0}$, V$_{DD1}$:	Power supply
V$_{LC1}$, V$_{LC2}$:	Power supply for LCD
V$_{PP}$:	Programming power supply
VR$_{OUT}$:	Regulator capacitance
V$_{SS0}$, V$_{SS1}$:	Ground
X1, X2:	Crystal (Main system clock)
XT1, XT2:	Crystal (Subsystem clock)

1.6 78K/0S Series Lineup

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


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

Small-scale package, general-purpose applications

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

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

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

LCD drive

144-pin	μPD789835	UART + 8-bit A/D + dot LCD (total display outputs: 96)
88-pin	μPD789830	UART + dot LCD (40 × 16)
80-pin	μPD789488	SIO + 10-bit A/D + internal voltage boosting method LCD (28 × 4)
80-pin	μPD789478	SIO + 8-bit A/D + resistance division method LCD (28 × 4)
80-pin	μPD789417A	μPD789407A with 10-bit A/D
80-pin	μPD789407A	SIO + 8-bit A/D + resistance division method LCD (28 × 4)
64-pin	μPD789456	μPD789446 with 10-bit A/D
64-pin	μPD789446	SIO + 8-bit A/D + internal voltage boosting method LCD (15 × 4)
64-pin	μPD789436	μPD789426 with 10-bit A/D
64-pin	μPD789426	SIO + 8-bit A/D + internal voltage boosting method LCD (5 × 4)
64-pin	μPD789316	RC oscillation version of μPD789306
64-pin	μPD789306	SIO + internal voltage boosting method LCD (24 × 4)
52-pin	μPD789467	8-bit A/D + internal voltage boosting method LCD (23 × 4)
52-pin	μPD789327	SIO + resistance division method LCD (24 × 4)

USB

64-pin	μPD789803	For PC keyboard. On-chip USB HUB function
44-pin	μPD789800	For PC keyboard. On-chip USB function

Inverter control

44-pin	μPD789842	On-chip inverter controller and UART

On-chip bus controller

30-pin	μPD789850	On-chip CAN controller

Keyless entry

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

VFD drive

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

Meter control

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

Remark Vacuum Fluorescent Display (VFD) is a general expression, however, some documents refer it to Fluorescent Indicator Panel (FIP™). VFD and FIP provide equivalent functions.

The major differences between subseries are shown below.

Series for General-Purpose and LCD Drive

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

Note Flash memory version: 3.0 V

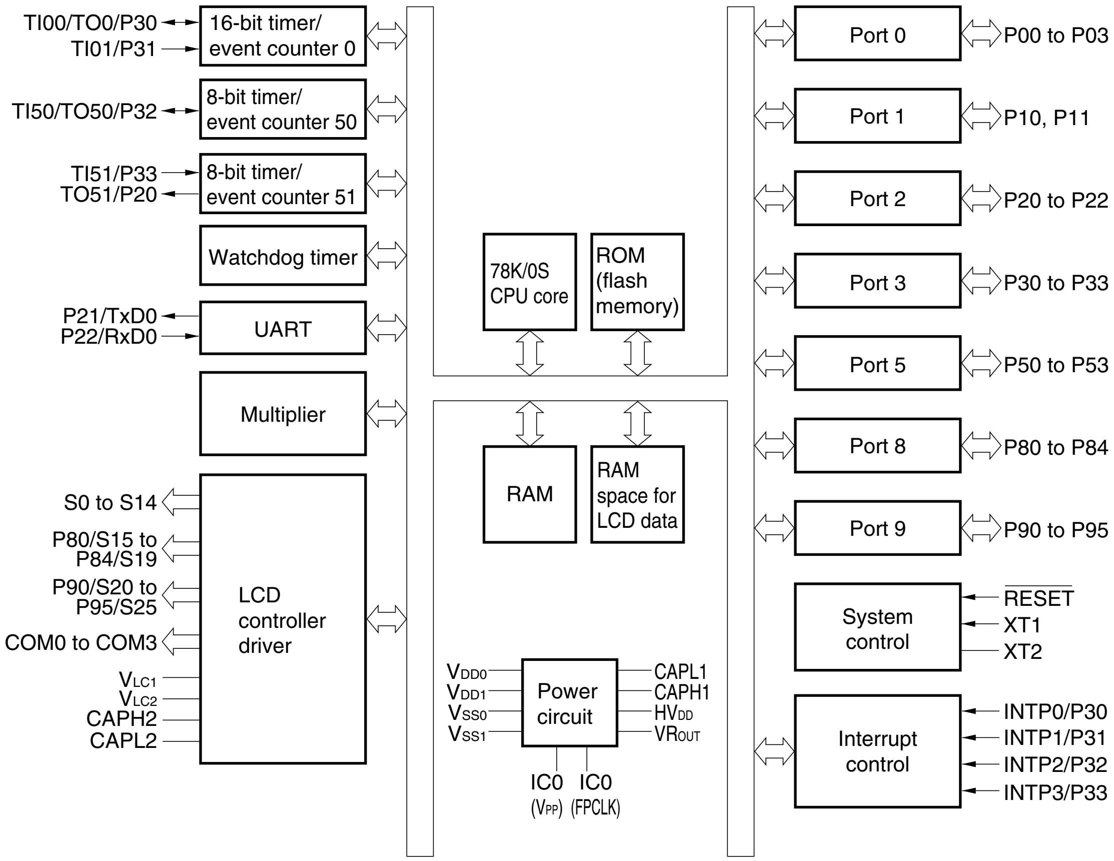
Series for ASSP

Subseries	Function	ROM Capacity (Bytes)	Timer				8-Bit A/D	10-Bit A/D	Serial Interface	I/O	V_{DD}	Remarks
			8-Bit	16-Bit	Watch	WDT					MIN.Value	
USB	μPD789803	8 K to 16 K	2 ch	–	–	1 ch	–	–	2 ch (USB: 1 ch)	41	3.6 V	–
	μPD789800	8 K								31	4.0 V	
Inverter control	μPD789842	8 K to 16 K	3 ch	**Note 1**	1 ch	1 ch	8 ch	–	1 ch (UART: 1 ch)	30	4.0 V	–
On-chip bus controller	μPD789850	16 K	1 ch	1 ch	–	1 ch	4 ch	–	2 ch (UART: 1 ch)	18	4.0 V	–
Keyless entry	μPD789861	4 K	2 ch	–	–	1 ch	–	–	–	14	1.8 V	RC-oscillation version, on-chip EEPROM
	μPD789860											On-chip EEPROM
	μPD789862	16 K	1 ch	2 ch					1 ch (UART: 1 ch)	22		
VFD drive	μPD789871	4 K to 8 K	3 ch	–	1 ch	1 ch	–	–	1 ch	33	2.7 V	–
Meter control	μPD789881	16 K	2 ch	1 ch	–	1 ch	–	–	1 ch (UART: 1 ch)	28	2.7 V[Note 2]	–

Notes 1. 10-bit timer: 1 channel

2. Flash memory version: 3.0 V

1.7 Block Diagram



Remark The parenthesized values apply to the μPD78F9882.

1.8 Overview of Functions

Item		μPD789881	μPD78F9882
Internal memory	ROM	16 KB	32 KB (flash memory)
	High-speed RAM	512 bytes	
	LCD display RAM	26 × 4 bytes	
Minimum instruction execution time		4 μs (@500 kHz (TYP.) operation with main system clock)	
		30.5 μs (@131 kHz operation with ×4 subsystem clock)	
General-purpose registers		8 bits × 8 registers	
Instruction set		• 16-bit operations • Bit manipulation (set, reset, test), etc.	
Multiplier		8 bits × 8 bits = 16 bits	
I/O ports		Total: 28 CMOS I/O: 24 N-ch open-drain I/O (3.6 V withstand voltage): 4	
Timers		• 16-bit timer/event counter: 1 channel • 8-bit timer/event counter: 2 channels • Watchdog timer: 1 channel	
Timer outputs		3	
Serial interface		UART: 1 channel	
LCD controller/driver		• Segment signal outputs: 26 • Common signal outputs: 4	
Vectored interrupt sources	Maskable	Internal: 8, External: 4	
	Non-maskable	Internal: 1	
Reset		• Reset by $\overline{\text{RESET}}$ signal input • Internal reset by watchdog timer	
Supply voltage		V_{DD} = 2.7 to 3.6 V	V_{DD} = 3.0 to 3.6 V
Operating ambient temperature		T_A = –40 to +85°C	
Package		• 64-pin plastic LQFP (10 × 10)	

An outline of the timers is shown below.

		16-Bit Timer/ Event Counter 0	8-Bit Timer/ Event Counters 50, 51	Watchdog Timer
Operating mode	Interval timer	1 channel	2 channels	1 channel$^{\text{Note}}$
	External event counter	1 channel	2 channels	–
Function	Timer output	1 output	2 outputs	–
	PWM output	–	2 outputs	–
	Square wave output	1 output	2 outputs	–
	Capture	1 input	–	–
	Interrupt source	1	2	2

Note Since the watchdog timer provides the watchdog timer function and interval timer function, select the one out of two functions.

1.9 Differences Between Standard Quality Grade Products and (A) Products

A more stringent quality assurance program is applied to the μPD789881(A) than the μPD789881 (standard quality grade product) (NEC classifies the former as a special quality grade product).

The only difference between the standard quality grade product (μPD789881) and (A) product (μPD789881(A)) in the μPD789881 Subseries is the quality grade. The other features (functions and electrical specifications) are the same.

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

Part Number Item	Standard Quality Grade Product μPD789881	(A) Product μPD789881(A)
Quality grade	Standard	Special
Other (functions, electrical specifications, etc.)	Same	

CHAPTER 2 PIN FUNCTIONS

2.1 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 in 1-bit units by means of pull-up resistor option register 0 (PUB0).	Input	–
P10, P11	I/O	Port 1. 2-bit I/O port. Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be specified in 1-bit units by means of pull-up resistor option register 1 (PUB1).	Input	–
P20	I/O	Port 2. 3-bit I/O port. Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be specified in 1-bit units by means of pull-up resistor option register 2 (PUB2).	Input	TO51
P21				TxD0
P22				RxD0
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 in 1-bit units by means of pull-up resistor option register 3 (PUB3).	Input	INTP0/TO0/TI00
P31				INTP1/TI01
P32				INTP2/TO50/TI50
P33				INTP3/TI51
P50 to P53	I/O	Port 5. 4-bit N-ch open-drain I/O port. Input/output can be specified in 1-bit units. For mask ROM version, an on-chip pull-up resistor can be specified by means of mask option.	Input	–
P80 to P84	I/O	Port 8. 5-bit I/O port. Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be specified in 1-bit units by means of pull-up resistor option register 8 (PUB8).	Input	S15 to S19
P90 to P95	I/O	Port 9. 6-bit I/O port. Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be specified in 1-bit units by means of pull-up resistor option register 9 (PUB9).	Input	S20 to S25

(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/TO0/TI00
INTP1				P31/TI01
INTP2				P32/TO50/TI50
INTP3				P33/TI51
TO0	Input	16-bit timer/event counter 0 output	Input	P30/INTP0/TI00
TI00	Input	External count clock input to 16-bit timer/event counter 0	Input	P30/INTP0/TO0
TI01				P31/INTP1
TO50	Output	8-bit timer/event counter 50 output	Input	P32/INTP2/TI50
TO51	Output	8-bit timer/event counter 51 output	Input	P20
TI50	Input	External count clock input to 8-bit timer/event counter 50 (TM50)	Input	P32/INTP2/TO50
TI51		External count clock input to 8-bit timer/event counter 51 (TM51)	Input	P33/INTP3
TxD0	Output	Serial data output of asynchronous serial interface	Input	P21
RxD0	Input	Serial data input of asynchronous serial interface	Input	P22
S0 to S14	Output	LCD controller/driver segment signal outputs	Low-level output	–
S15 to S19				P80 to P84
S20 to S25				P90 to P95
COM0 to COM3	Output	LCD controller/driver common signal outputs	Low-level output	–
V_{LC1}, V_{LC2}	–	LCD drive voltage	–	–
CAPH2, CAPL2	–	LCD drive capacitor connection pin	–	–
CAPH1, CAPL1	–	Power supply circuit capacitor connection (recommended value: 0.47 μF)	–	–
HV_{DD}	–		–	–
VR_{OUT}	–		–	–
XT1	Input	Connecting crystal resonator for subsystem clock oscillation	–	–
XT2	–		–	–
$\overline{\text{RESET}}$	Input	System reset input	Input	–
V_{DD0}	–	Positive power supply for ports	–	–
V_{DD1}	–	Positive power supply (except for ports)	–	–
V_{SS0}	–	Ground potential for ports	–	–
V_{SS1}	–	Ground potential (except for ports)	–	–
IC0	–	Internally connected. Connect to V_{SS0} or V_{SS1} directly.	–	–
FPCLK	–	External clock input in flash memory programming mode	–	–
V_{PP}	–	Sets flash memory programming mode. Applies high voltage when a program is written or verified.	–	–

2.2 Description of Pin Functions

2.2.1 P00 to P03 (Port 0)

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 (PUB0) in 1-bit units.

2.2.2 P10, P11 (Port 1)

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

2.2.3 P20 to P22 (Port 2)

These pins constitute a 3-bit I/O port. In addition, these pins enable UART data I/O and timer output.

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

(1) Port mode

In this mode, P20 to P22 function as a 3-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 2 (PUB2) in 1-bit units.

(2) Control mode

In this mode, P20 to P22 function as the UART data I/O and timer output.

(a) RxD0, TxD0

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

(b) TO51

This is the timer output pin of the 8-bit timer/event counter 51.

Caution When using P20 to P22 as serial interface pins, the I/O mode and output latch must be set according to the functions to be used. For the setting method, refer to 9.3 Registers Controlling Serial Interface UART0.

2.2.4 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, P30 to P33 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 3 (PUB3) in 1-bit units.

(2) Control mode

In this mode, P30 to P33 function as timer I/O and external interrupt input.

(a) TI00

This is an external count clock input pin for 16-bit timer/event counter 0 and a capture trigger signal input pin for the capture registers (CR00 and CR01) of 16-bit timer/event counter 0.

(b) TI01

This is a capture trigger signal input pin for the capture register (CR00) of 16-bit timer/event counter 0.

(c) TI50, TI51

These are the external clock input pins to 8-bit timer/event counters 50 and 51.

(d) TO0, TO50

These are the timer output pins.

(e) 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.

2.2.5 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 in 1-bit units.

2.2.6 P80 to P84 (Port 8)

These pins function as a 5-bit I/O port. In addition, they also function as segment output.

Port 8 can be specified in the following operation modes in 1-bit units.

(1) Port mode

These pins function as a 5-bit I/O port. Port 8 can be set in the input or output port mode in 1-bit units by port mode register 8 (PM8). When used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 8 (PUB8) in 1-bit units.

(2) Control mode

These pins function as segment output pins (S15 to S19).

2.2.7 P90 to P95 (Port 9)

These pins function as a 6-bit I/O port. In addition, they also function as segment output.

Port 9 can be specified in the following operation modes in 1-bit units.

(1) Port mode

These pins function as a 6-bit I/O port. Port 9 can be set in the input or output port mode in 1-bit units by port mode register 9 (PM9). When used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 9 (PUB9) in 1-bit units.

(2) Control mode

These pins function as segment output pins (S20 to S25).

2.2.8 S0 to S14

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

2.2.9 COM0 to COM3

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

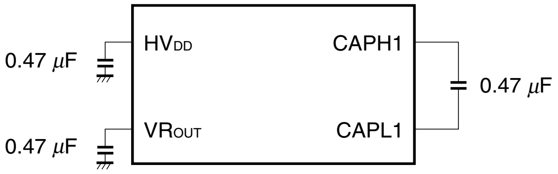
2.2.10 V_{LC1}, V_{LC2}

These pins are power supply voltage pins to drive the LCD. Connect the 0.47 μF capacitor (recommended value).

2.2.11 CAPH1, CAPL1, HV_{DD}, VR_{OUT}

These pins are used to connect a capacitor for the power supply circuit to realize ultra-low power consumption.

An example of connection of the power supply circuit is shown below.



2.2.12 CAPH2, CAPL2

These pins are capacitor connection pins to drive the LCD. Connect the 0.47 μF capacitor (recommended value).

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

This pin inputs an active-low system reset signal.

2.2.14 XT1, XT2

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

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

2.2.15 V_{DD0}, V_{DD1}

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

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

2.2.16 V_{SS0}, V_{SS1}

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

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

2.2.17 V_{PP} (μPD78F9882 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.

2.2.18 FPCLK (μPD78F9882 only)

This is the external clock input pin in the flash memory programming mode.

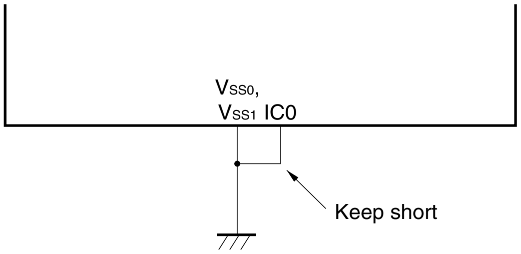
Connect this pin directly to V_{SS0} or V_{SS1} in the normal operation mode.

2.2.19 IC0 (mask ROM version only)

The IC0 (Internally Connected) pin is used to set the μPD789881 in the test mode before shipment. In the normal operation mode, connect this pin directly to the V_{SS0} or V_{SS1} pin with as short a wiring length as possible.

If a potential difference is generated between the IC0 pin and V_{SS0} or V_{SS1} pin due to a long wiring length, or an external noise superimposed on the IC0 pin, the user program may not run correctly.

- Connect the IC0 pin directly to the V_{SS0} or V_{SS1} pin.



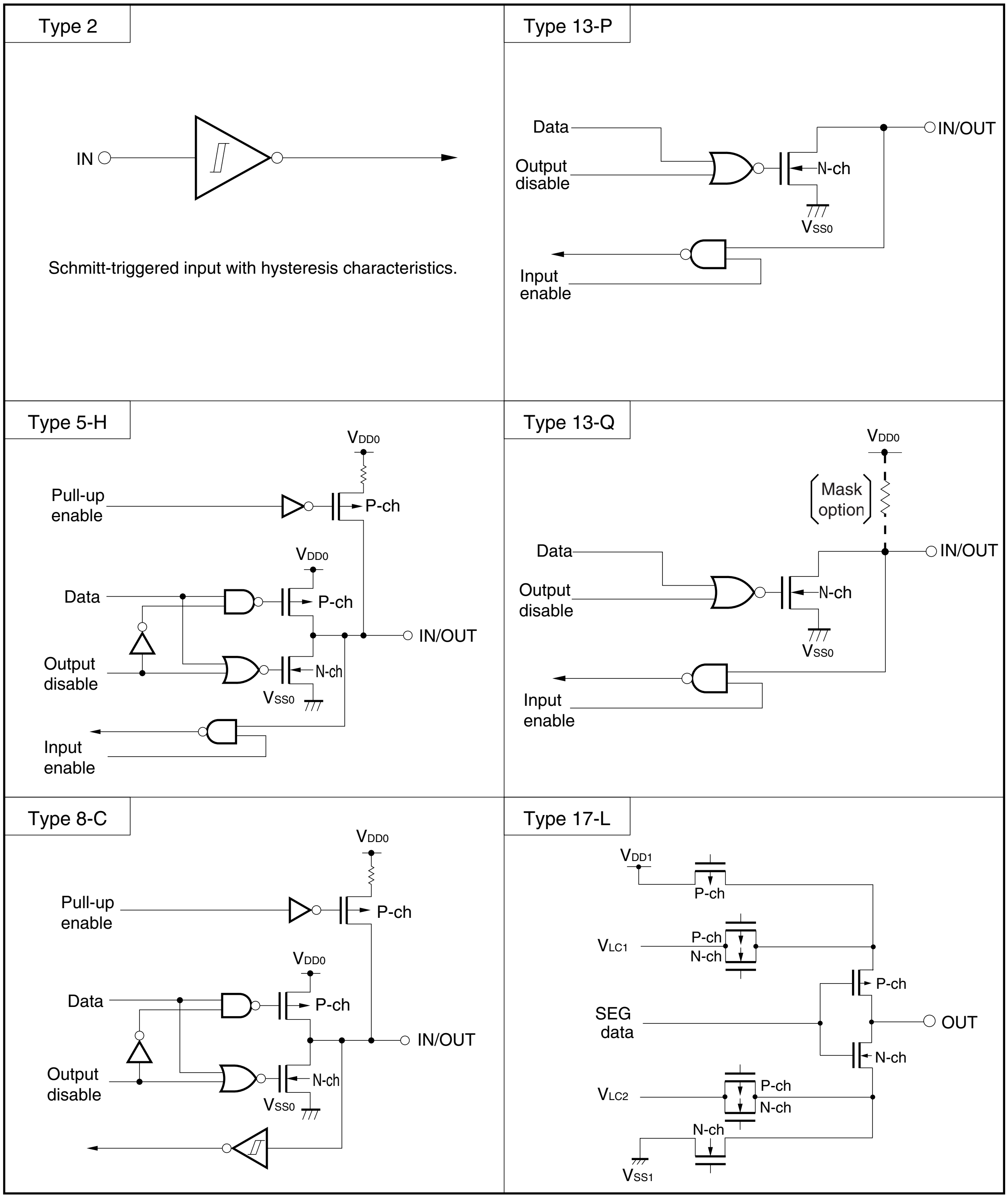
2.3 Pin I/O Circuits and Recommended Connection of Unused Pins

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

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

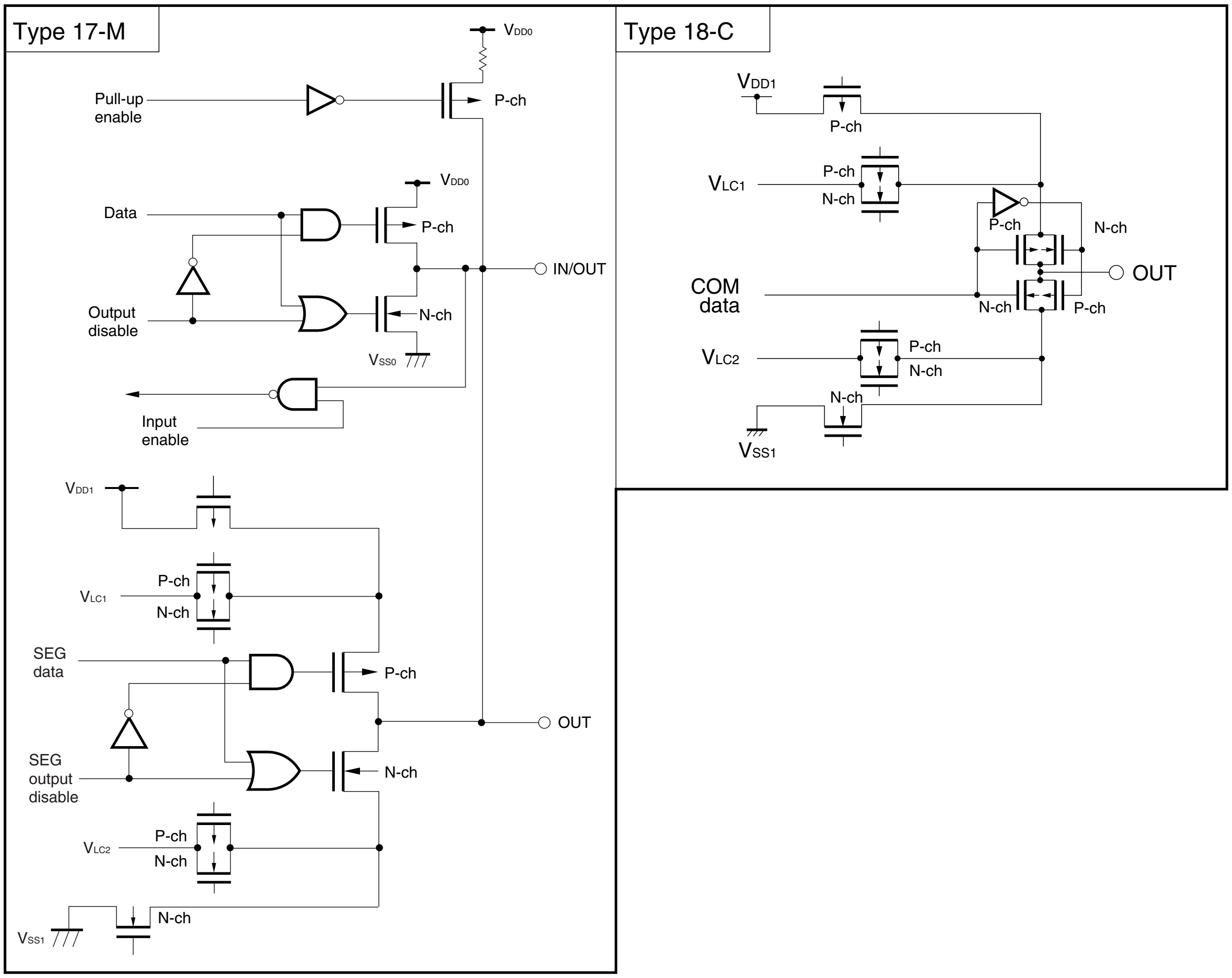
Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00 to P03	5-H	I/O	Input: Independently connect to any one of V_{DD0}, V_{DD1}, V_{SS0}, or V_{SS1} via a resistor. Output: Leave open.
P10, P11			
P20/TO51	8-C		
P21/TxD0			
P22/RxD0			
P30/INTP0/TO0/TI00			Input: Independently connect to V_{SS0} or V_{SS1} via a resistor. Output: Leave open.
P31/INTP1/TI01			
P32/INTP2/TO50/TI50			
P33/INTP3/TI51			
P50 to P53 (Mask ROM version)	13-Q		Input: Connect to V_{SS0} or V_{SS1}. Output: Leave open with low-level output.
P50 to P53 (μPD78F9882)	13-P		
P80/S15 to P84/S19	17-M		Input: Independently connect to any one of V_{DD0}, V_{DD1}, V_{SS0}, or V_{SS1} via a resistor. Output: Leave open.
P90/S20 to P95/S25			
S0 to S14	17-L	Output	Leave open.
COM0 to COM3	18-C		
V_{LC1}, V_{LC2}	–	–	
CAPH2, CAPL2			
$\overline{\text{RESET}}$	2	Input	–
IC0 (Mask ROM version)	–	–	Connect directly to V_{SS0} or V_{SS1}.
FPCLK (μPD78F9882)			
V_{PP} (μPD78F9882)			Independently connect a 10 kΩ pull-down resistor or connect directly to V_{SS0} or V_{SS1}.

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



Schmitt-triggered input with hysteresis characteristics.

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



3.1 Memory Space

The μPD789881 Subseries can access 64 KB of memory space. Figures 3-1 and 3-2 show the memory maps.

Figure 3-1. Memory Map (μPD789881)

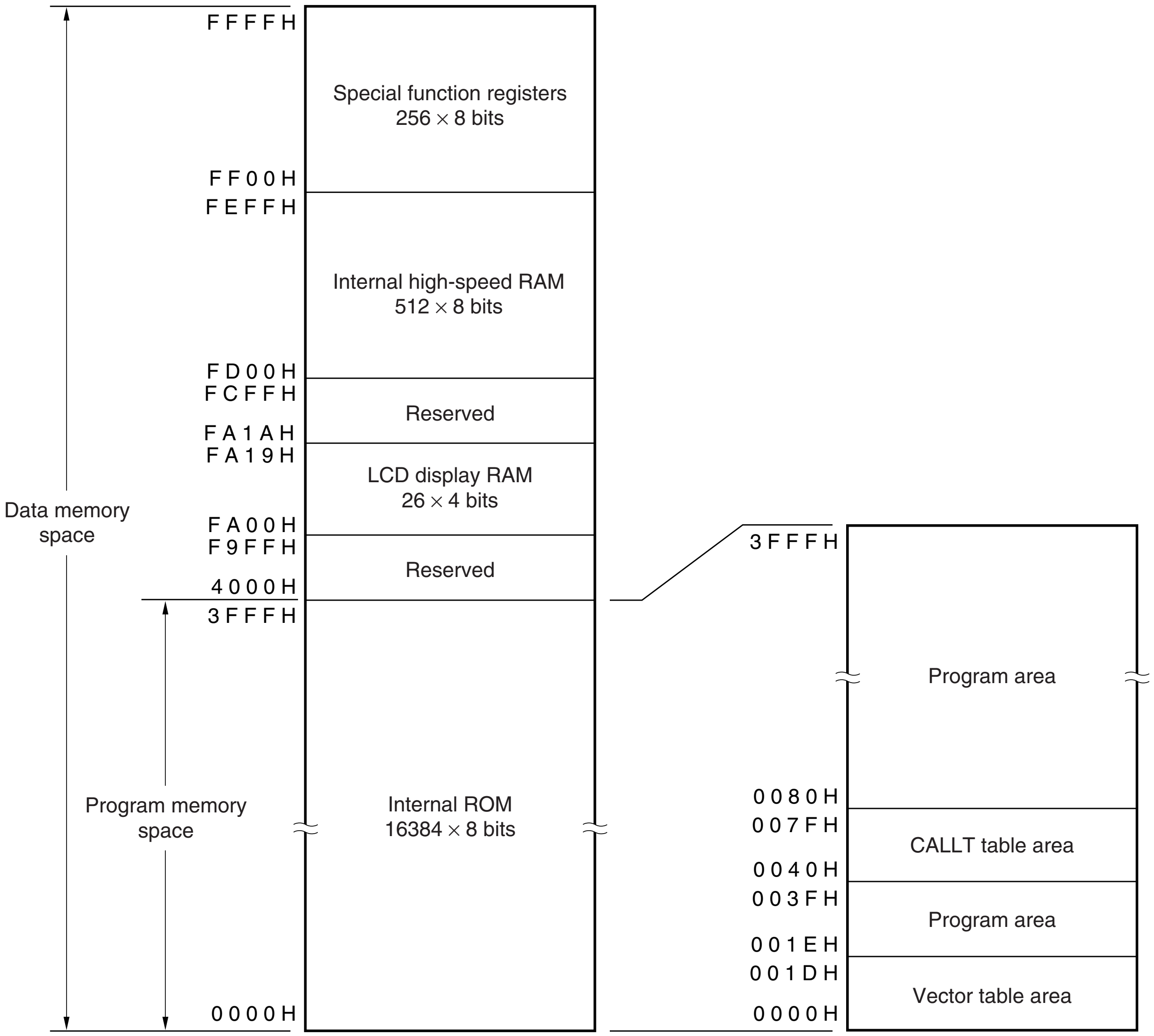
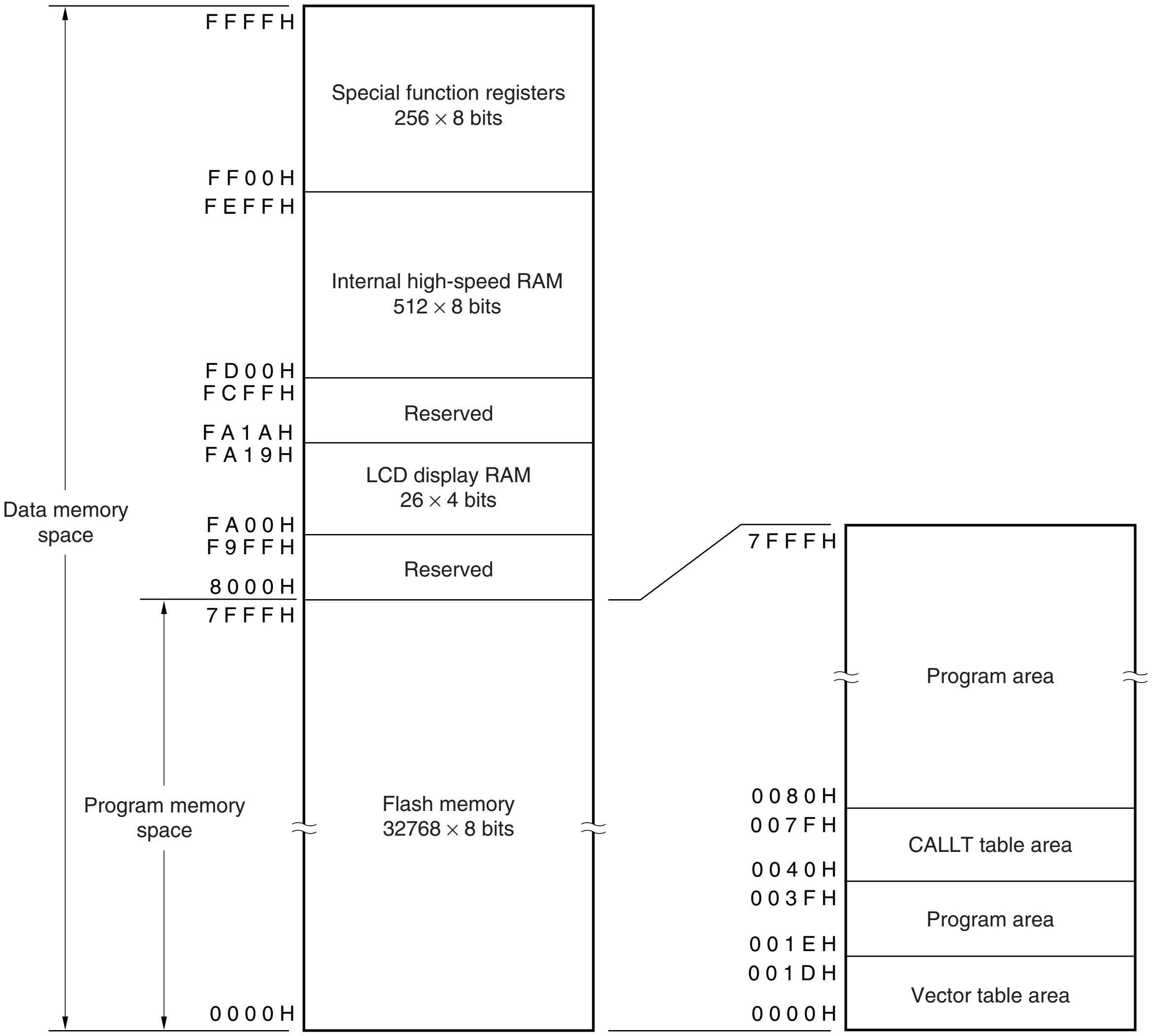


Figure 3-2. Memory Map (μPD78F9882)



3.1.1 Internal program memory space

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

The μPD789881 Subseries provides internal ROM (or flash memory) with the following capacity for each product.

Table 3-1. Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
μPD789881	Mask ROM	16384 × 8 bits
μPD78F9882	Flash memory	32768 × 8 bits

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

(1) Vector table area

The 30-byte area of addresses 0000H to 001DH 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 3-2. Vector Table

Vector Table Address	Interrupt Request	Vector Table Address	Interrupt Request
0000H	$\overline{\text{RESET}}$ input	0010H	INTSR0
0004H	INTWDT	0012H	INTST0
0006H	INTP0	0014H	INTTM50
0008H	INTP1	0016H	INTTM51
000AH	INTP2	0018H	INTTM00
000CH	INTP3	001AH	INTTM01
000EH	INTSRE0	001CH[Note]	–

Note There are no interrupt requests corresponding to vector table address 001CH.

(2) CALLT instruction table area

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

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

The μPD789881 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 incorporated in the area between FA00H and FA19H.

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

3.1.3 Special function register (SFR) area

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

3.1.4 Data memory addressing

The μPD789881 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 3-3 and 3-4 show the data memory addressing modes.

Figure 3-3. Data Memory Addressing (μPD789881)

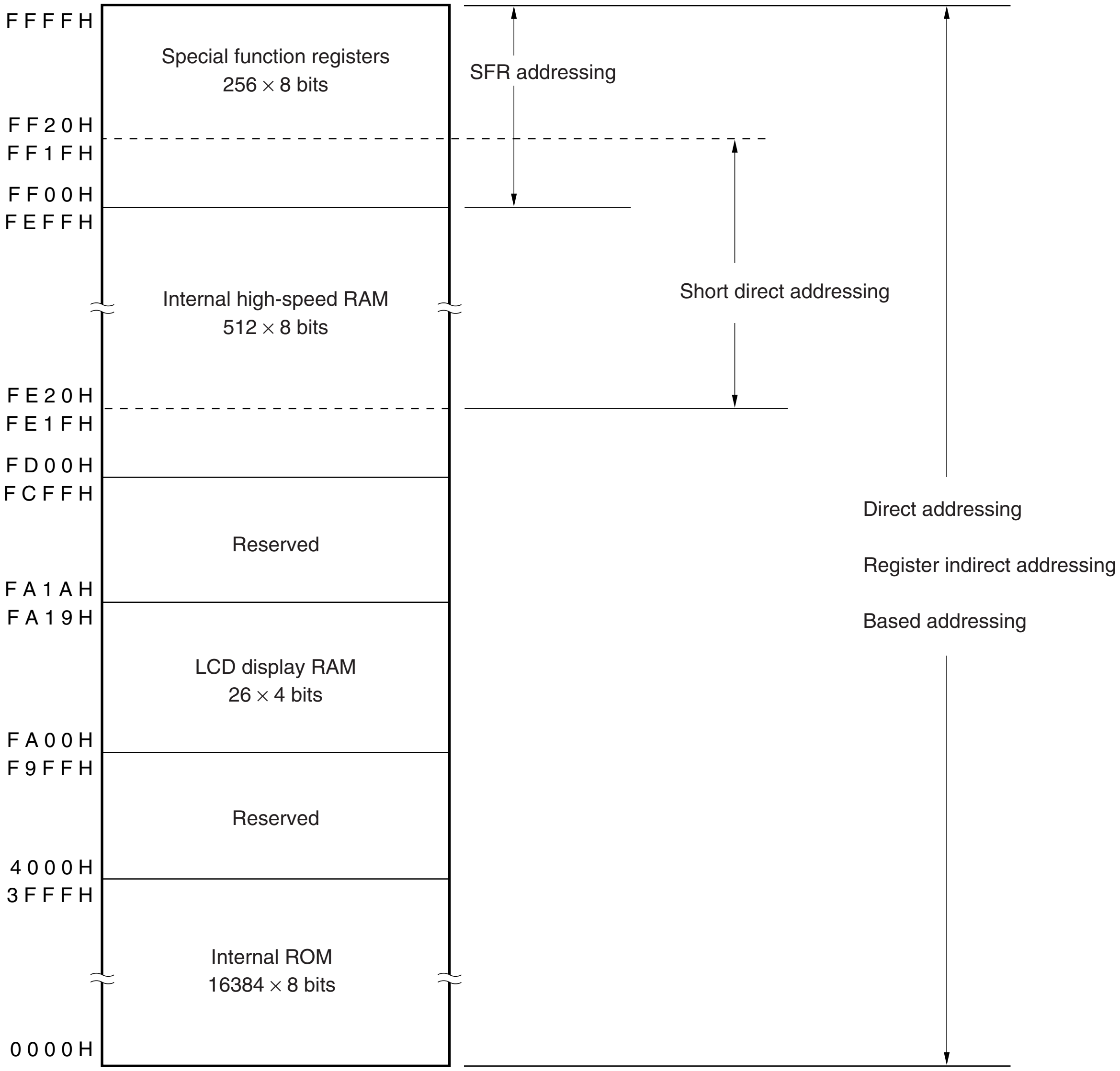
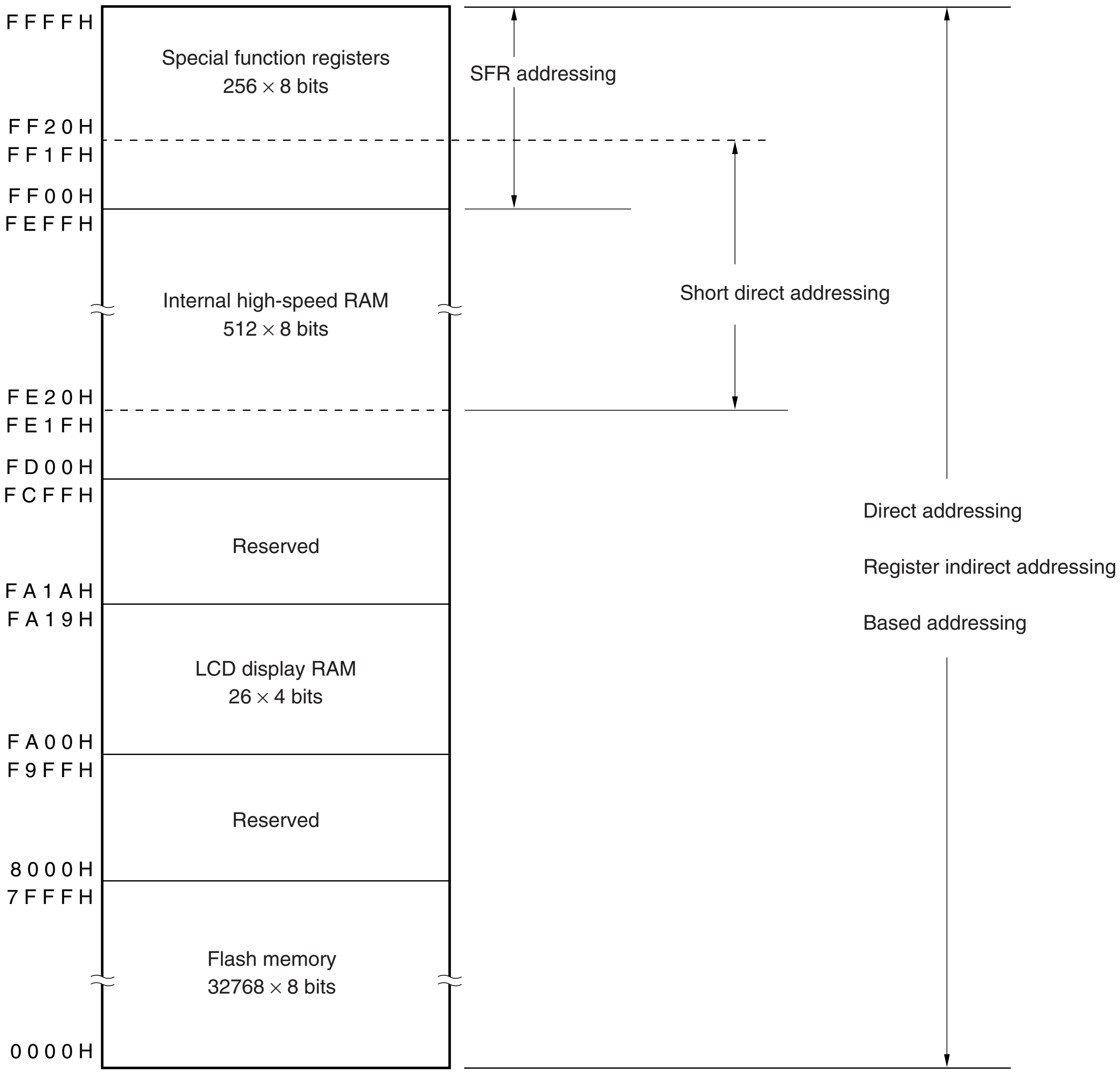


Figure 3-4. Data Memory Addressing (μPD78F9882)



3.2 Processor Registers

The μPD789881 Subseries provides the following on-chip processor registers.

3.2.1 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 3-5. Program Counter Configuration

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

(2) Program status word (PSW)

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

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

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

Figure 3-6. 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 acknowledgment 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 acknowledgment enable is controlled with an interrupt mask flag for various interrupt sources.

IE is reset to 0 upon DI instruction execution or interrupt acknowledgment and is set to 1 upon EI instruction execution.

(b) Zero flag (Z)

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

(c) Auxiliary carry flag (AC)

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

(d) Carry flag (CY)

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

(3) Stack pointer (SP)

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

Figure 3-7. Stack Pointer Configuration

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

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

Each stack operation saves/restores data as shown in Figures 3-8 and 3-9.

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

Figure 3-8. Data to Be Saved to Stack Memory

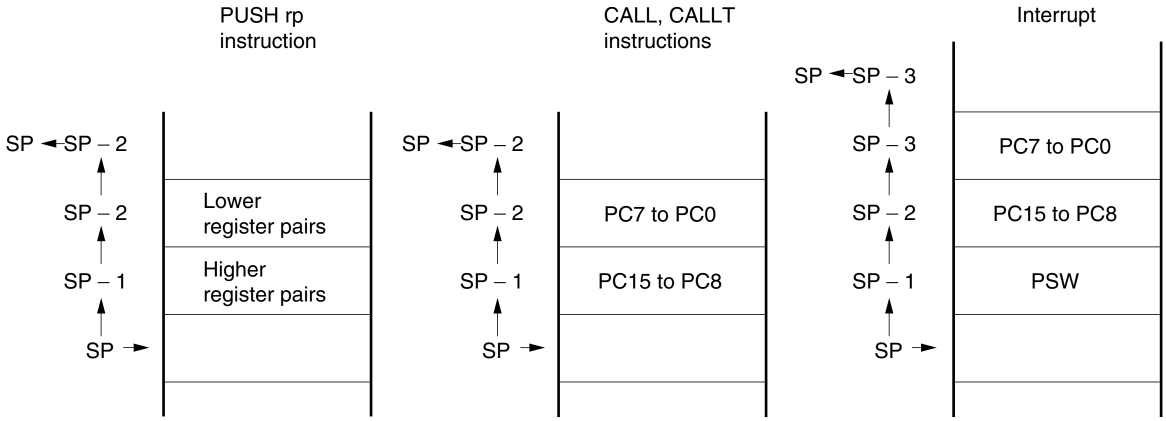
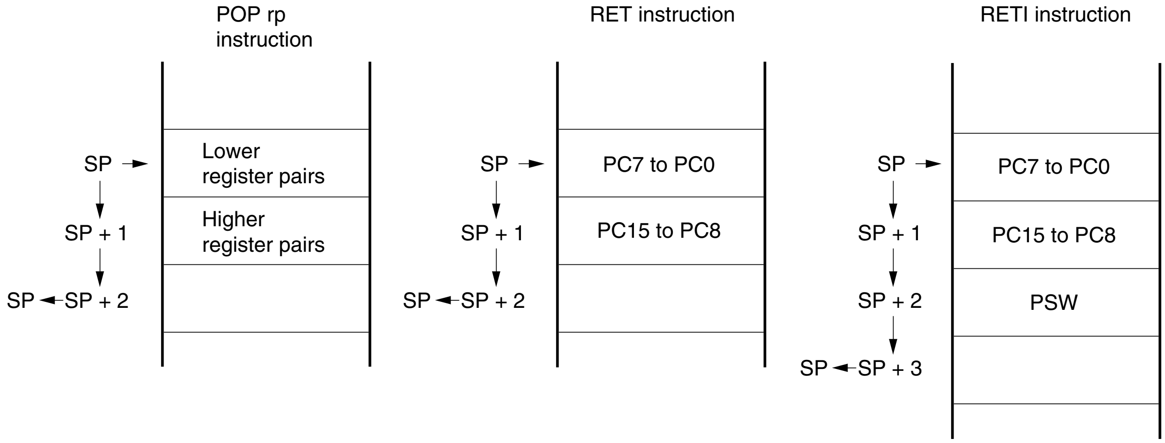


Figure 3-9. Data to Be Restored from Stack Memory



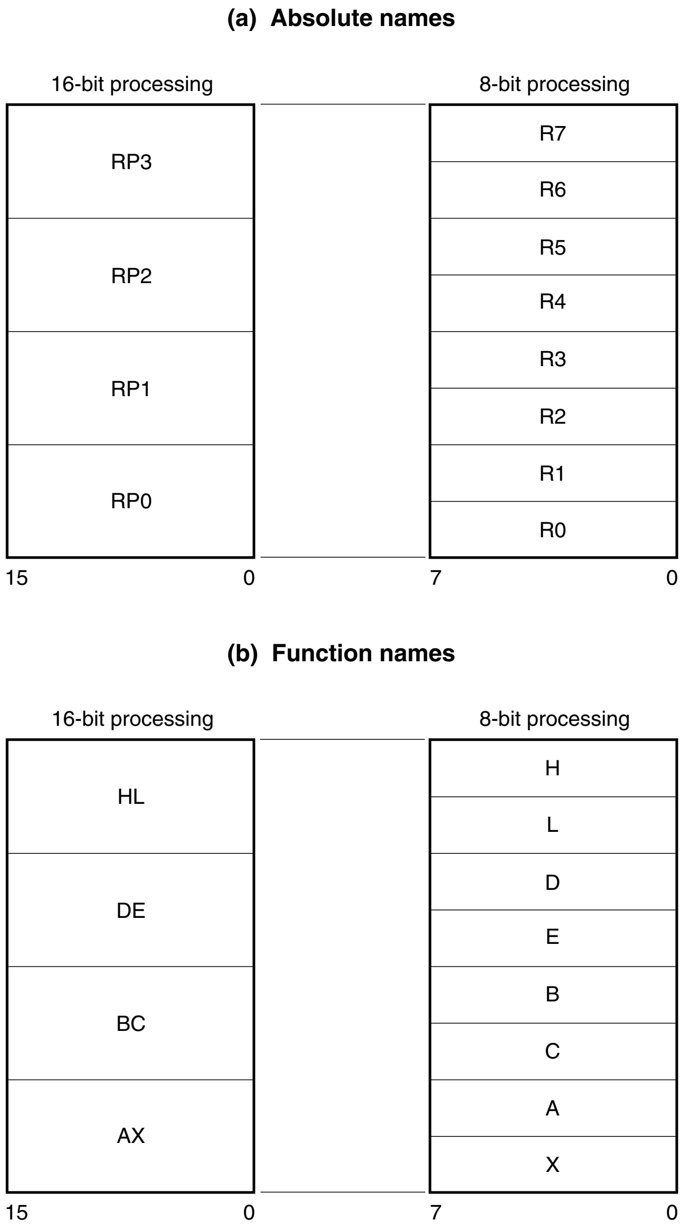
3.2.2 General-purpose registers

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

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

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

Figure 3-10. General-Purpose Register Configuration



(a) Absolute names

(b) Function names

3.2.3 Special function registers (SFRs)

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

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

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

The manipulatable bits can be specified as follows.

- 1-bit manipulation
  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 3-3 lists the special function registers. The meanings of the symbols in this table are as follows.

- Symbol
  Indicates the addresses of the implemented special function registers. The symbols shown in this column are the reserved words of the assembler, and have already been defined in the header file called “sfrbit.h” of the C compiler. Therefore, these symbols can be used as instruction operands if an assembler or integrated debugger is used.

- R/W
  Indicates whether the special function register in question can be read or written.

R/W:	Read/write
R:	Read only
W:	Write only

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

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

Table 3-3. Special Function Registers (1/2)

Address	Special Function Register (SFR) Name	Symbol		R/W	Bit Unit for Manipulation			After Reset
					1 Bit	8 Bits	16 Bits	
FF00H	Port 0	P0		R/W	√	√	–	00H
FF01H	Port 1	P1			√	√	–	
FF02H	Port 2	P2			√	√	–	
FF03H	Port 3	P3			√	√	–	
FF05H	Port 5	P5			√	√	–	
FF08H	Port 8	P8			√	√	–	
FF09H	Port 9	P9			√	√	–	
FF12H	16-bit multiplication result store register L	MUL0L	MUL0	R	–	√	√	Undefined
FF13H	16-bit multiplication result store register H	MUL0H			–	√		
FF14H	16-bit timer counter 0	TM0			–	–	√	0000H
FF15H								
FF16H	16-bit capture/compare register 00	CR00		R/W	–	–	√	
FF17H								
FF18H	16-bit capture/compare register 01	CR01			–	–	√	
FF19H								
FF20H	Port mode register 0	PM0			√	√	–	FFH
FF21H	Port mode register 1	PM1			√	√	–	
FF22H	Port mode register 2	PM2			√	√	–	
FF23H	Port mode register 3	PM3			√	√	–	
FF25H	Port mode register 5	PM5			√	√	–	
FF28H	Port mode register 8	PM8			√	√	–	
FF29H	Port mode register 9	PM9			√	√	–	
FF30H	Pull-up resistor option register 0	PUB0			√	√	–	00H
FF31H	Pull-up resistor option register 1	PUB1			√	√	–	
FF32H	Pull-up resistor option register 2	PUB2			√	√	–	
FF33H	Pull-up resistor option register 3	PUB3			√	√	–	
FF38H	Pull-up resistor option register 8	PUB8			√	√	–	
FF39H	Pull-up resistor option register 9	PUB9			√	√	–	
FF49H	Watchdog timer mode register	WDTM			√	√	–	
FF4AH	Watchdog timer clock select register	WDCS			–	√	–	
FF58H	Port function register 8	PF8		W	–	√	–	
FF59H	Port function register 9	PF9			–	√	–	
FF66H	16-bit timer mode control register 0	TMC0		R/W	√	√	–	
FF67H	16-bit timer prescaler mode register 0	PRM0			√	√	–	
FF68H	16-bit capture/compare control register 0	CRC0			√	√	–	
FF69H	16-bit timer output control register 0	TOC0			√	√	–	

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

Address	Special Function Register (SFR) Name	Symbol	R/W	Bit Unit for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
FF70H	8-bit timer counter 50	TM50	R	–	√	–	00H
FF71H	8-bit compare register 50	CR50	R/W	–	√	–	
FF72H	8-bit timer clock select register 50	TCL50		–	√	–	
FF73H	8-bit timer mode control register 50	TMC50		√	√	–	
FF78H	8-bit timer counter 51	TM51	R	–	√	–	
FF79H	8-bit compare register 51	CR51	R/W	–	√	–	
FF7AH	8-bit timer clock select register 51	TCL51		–	√	–	
FF7BH	8-bit timer mode control register 51	TMC51		√	√	–	
FF90H	Asynchronous serial interface mode register 0	ASIM0		√	√	–	01H
FF91H	Baud rate generator control register 0	BRGC0		√	√	–	1FH
FF92H	Receive buffer register 0	RXB0	R	–	√	–	FFH
FF93H	Asynchronous serial interface status register 0	ASIS0		–	√	–	00H
FF94H	Transmit shift register 0	TXS0	W	–	√	–	FFH
FFB0H	LCD display mode register 0	LCDM0	R/W	√	√	–	00H
FFB2H	LCD clock control register 0	LCDC0		–	√	–	
FFD0H	Multiplication data register A0	MRA0	W	–	√	–	Undefined
FFD1H	Multiplication data register B0	MRB0		–	√	–	
FFD2H	Multiplier control register 0	MULC0	R/W	√	√	–	00H
FFE0H	Interrupt request flag register 0	IF0		√	√	–	
FFE1H	Interrupt request flag register 1	IF1		√	√	–	
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		–	√	–	
FFF2H	Subclock control register	CSS		√	√	–	
FFFBH	Processor clock control register	PCC		√	√	–	02H

3.3 Instruction Address Addressing

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

3.3.1 Relative addressing

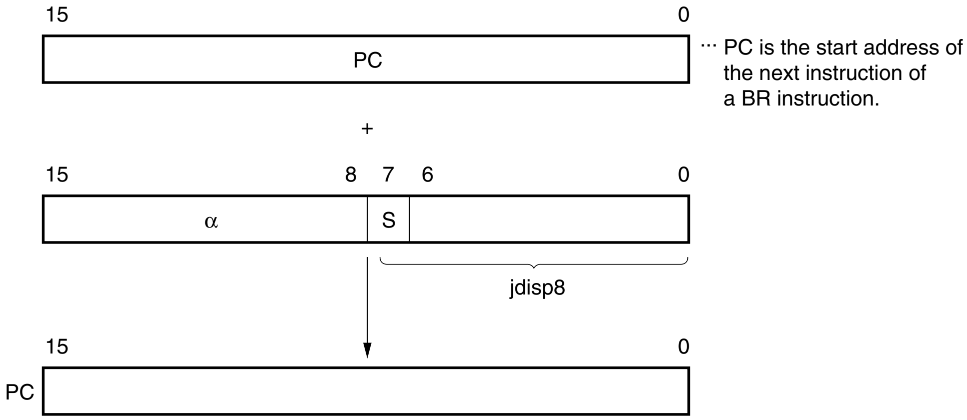
[Function]

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

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

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

[Illustration]


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.

3.3.2 Immediate addressing

[Function]

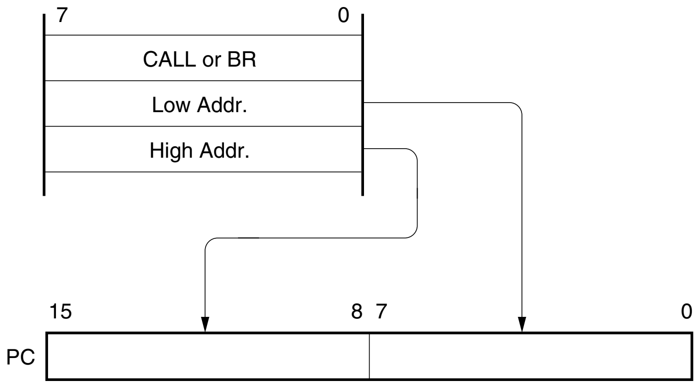
Immediate data in the instruction word is transferred to the program counter (PC) and branched.

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

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

[Illustration]

In case of CALL !addr16 and BR !addr16 instructions



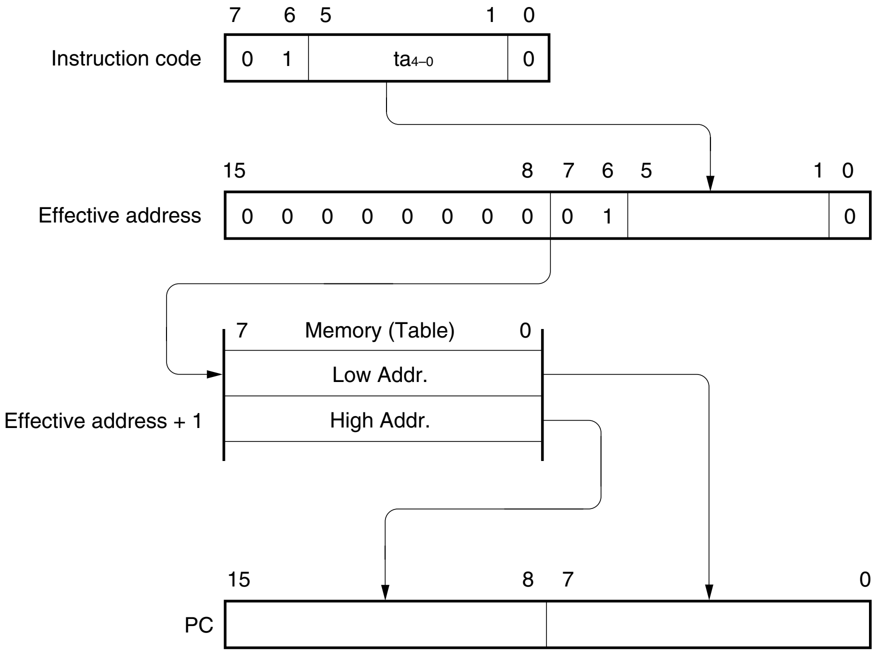
3.3.3 Table indirect addressing

[Function]

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

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

[Illustration]



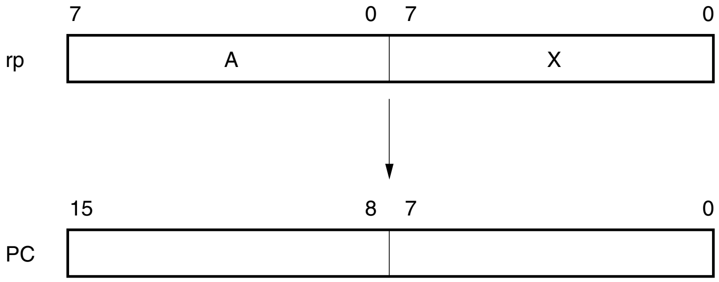
3.3.4 Register addressing

[Function]

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

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

[Illustration]



3.4 Operand Address Addressing

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

3.4.1 Direct addressing

[Function]

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

[Operand format]

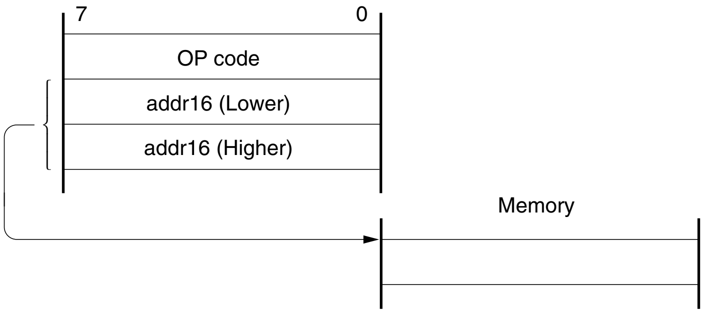
Identifier	Description
addr16	Label or 16-bit immediate data

[Description example]

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

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

[Illustration]



3.4.2 Short direct addressing

[Function]

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

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

The SFR area (FF00H to FF1FH) where short direct addressing is applied is a part of the whole SFR area. Ports that are frequently accessed in a program and the compare register of the timer 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 cleared to 0. When it is at 00H to 1FH, bit 8 is set to 1. See **[Illustration]** below.

[Operand format]

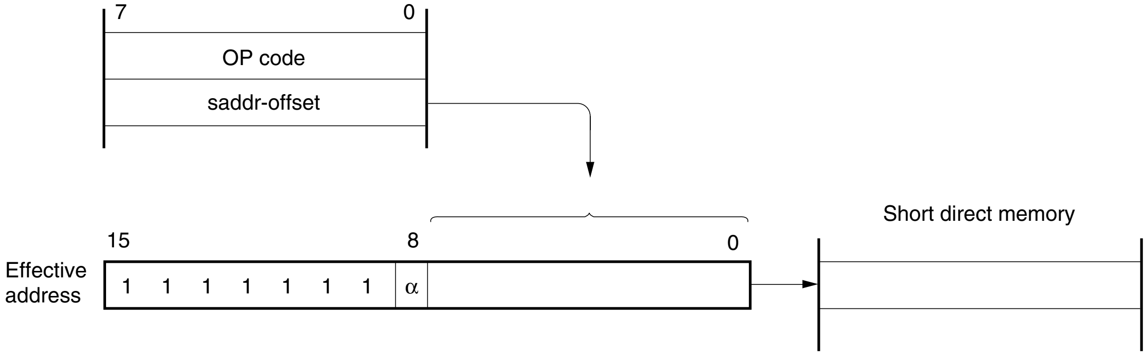
Identifier	Description
saddr	Label or FE20H to FF1FH immediate data
saddrp	Label or FE20H to FF1FH immediate data (even address only)

[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.

3.4.3 Special function register (SFR) addressing

[Function]

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

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

[Operand format]

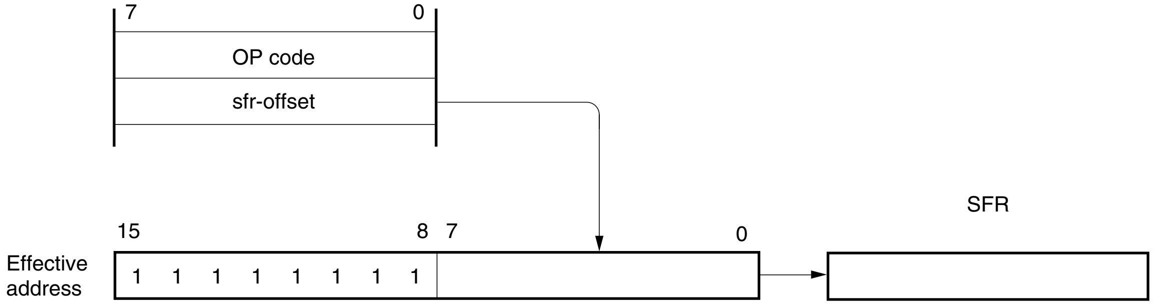
Identifier	Description
sfr	Special function register name

[Description example]

MOV PM0, A; When selecting PM0 for sfr

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

[Illustration]



3.4.4 Register addressing

[Function]

In the register addressing mode, general-purpose registers are accessed as operands. The general-purpose register to be accessed is specified by a register specification code or functional name in the instruction code.

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

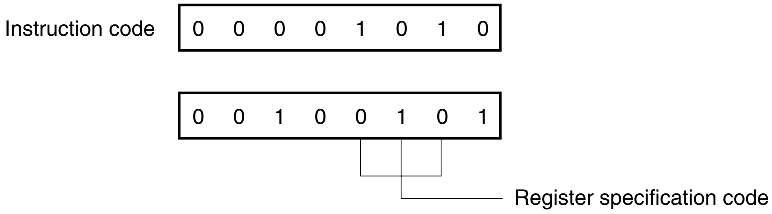
[Operand format]

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

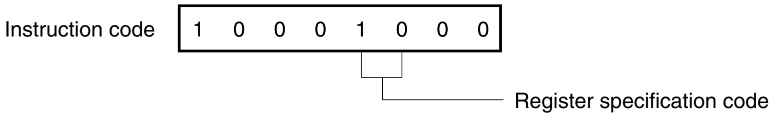
r and rp can be described with absolute names (R0 to R7 and RP0 to RP3) as well as function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL).

[Description example]

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



INCW DE; When selecting the DE register pair for rp



3.4.5 Register indirect addressing

[Function]

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

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

[Operand format]

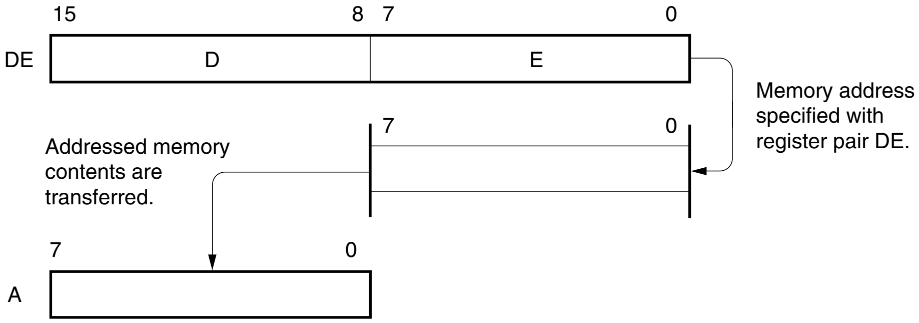
Identifier	Description
–	[DE], [HL]

[Description example]

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

Instruction code	0	0	1	0	1	0	1	1

[Illustration]


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

3.4.6 Based addressing

[Function]

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

[Operand format]

Identifier	Description
–	[HL+byte]

[Description example]

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

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

3.4.7 Stack addressing

[Function]

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

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

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

[Description example]

In the case of PUSH DE

Instruction code	1	0	1	0	1	0	1	0

CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

The μPD789881 Subseries provides the ports shown in Figure 4-1, enabling various methods of control. The functions of each port are shown in Table 4-1.

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

Figure 4-1. Port Types

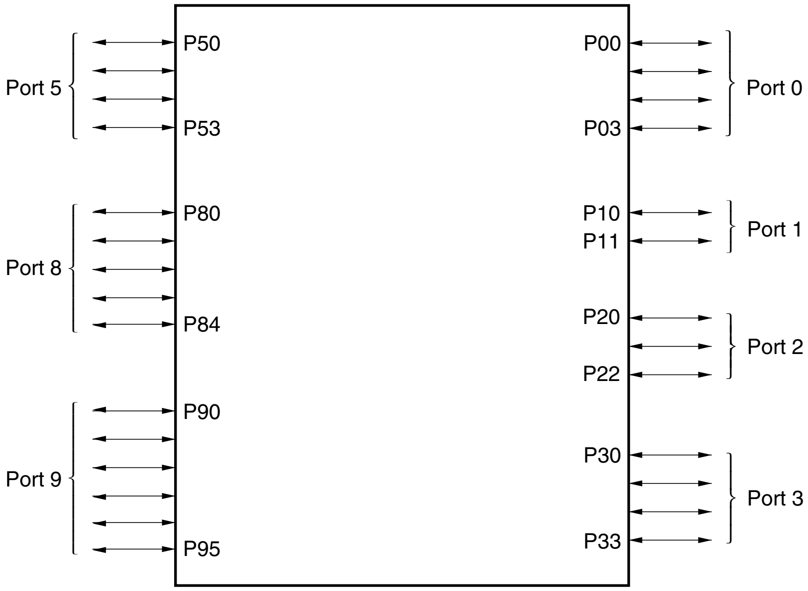


Table 4-1. Port Functions

Port Name	Pin Name	Function
Port 0	P00 to P03	I/O port. Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be specified in 1-bit units by means of pull-up resistor option register 0 (PUB0).
Port 1	P10, P11	I/O port. Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be specified in 1-bit units by means of pull-up resistor option register 1 (PUB1).
Port 2	P20 to P22	I/O port. Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be specified in 1-bit units by means of pull-up resistor option register 2 (PUB2).
Port 3	P30 to P33	I/O port. Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be specified in 1-bit units by means of pull-up resistor option register 3 (PUB3).
Port 5	P50 to P53	N-ch open-drain I/O port. Input/output can be specified in 1-bit units. An on-chip pull-up resistor can be specified by mask option.
Port 8	P80 to P84	I/O port. Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be specified in 1-bit units by means of pull-up resistor option register 8 (PUB8).
Port 9	P90 to P95	I/O port. Input/output can be specified in 1-bit units. When used as an input port, an on-chip pull-up resistor can be specified in 1-bit units by means of pull-up resistor option register 9 (PUB9).

4.2 Port Configuration

Ports have the following hardware configuration.

Table 4-2. Configuration of Port

Item	Configuration
Control registers	Port mode registers (PMm: m = 0 to 3, 5, 8, 9) Pull-up resistor option registers (PUBm: m = 0 to 3, 8, 9) Port function registers (PF8 and PF9)
Ports	Total: 28 (CMOS I/O: 24, N-ch open-drain I/O: 4)
Pull-up resistors	• Mask ROM version Total: 28 (software control: 24, mask option specification: 4) • Flash memory version Total: 24 (software control only)

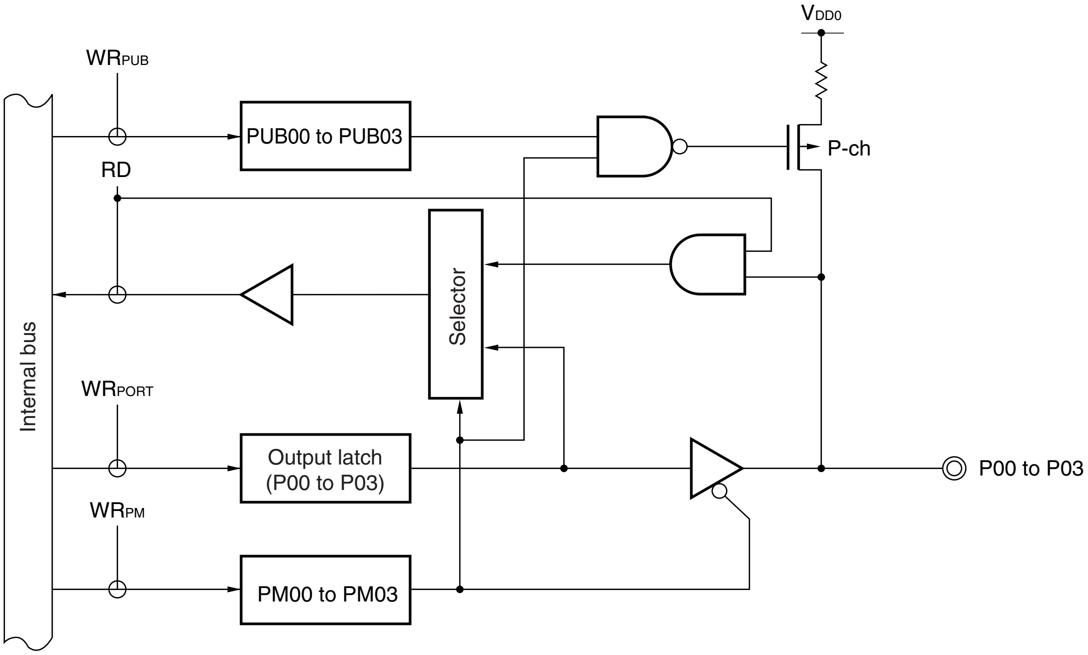
4.2.1 Port 0

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 port mode register 0 (PM0). When the P00 to P03 pins are used as input port pins, on-chip pull-up resistors can be specified in 1-bit units by pull-up resistor option register 0 (PUB0).

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

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

Figure 4-2. Block Diagram of P00 to P03



PUB0:	Pull-up resistor option register 0
PM:	Port mode register
RD:	Port 0 read signal
WR:	Port 0 write signal

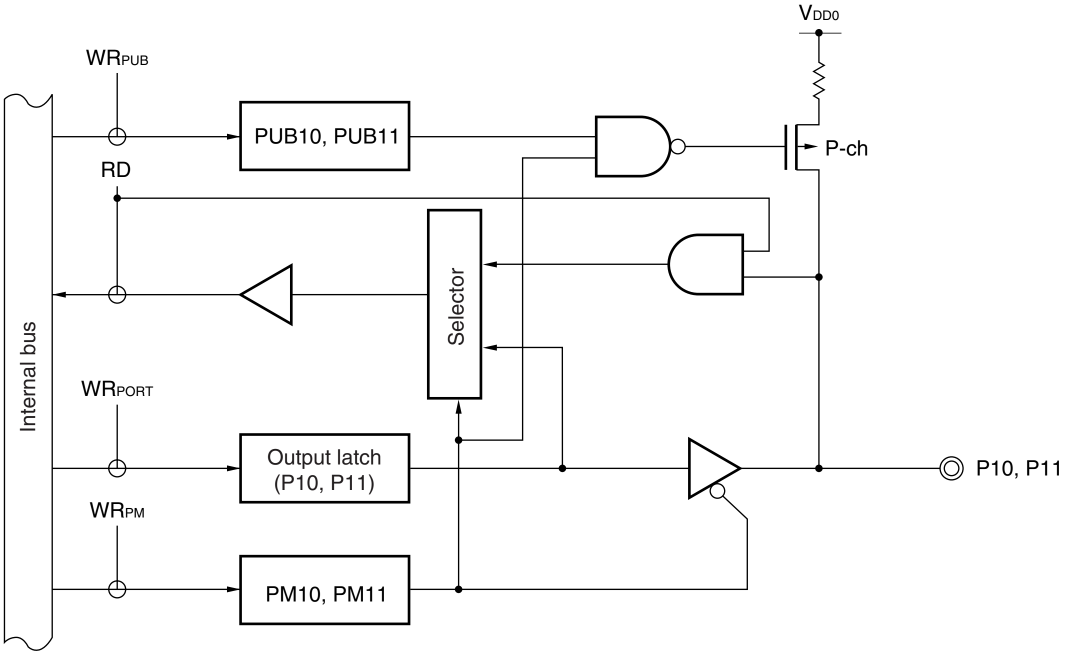
4.2.2 Port 1

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

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

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

Figure 4-3. Block Diagram of P10 and P11



PUB1: Pull-up resistor option register 1

PM: Port mode register

RD: Port 1 read signal

WR: Port 1 write signal

4.2.3 Port 2

This is a 3-bit I/O port with an output latch. Port 2 can be specified in the input or output mode in 1-bit units by port mode register 2 (PM2). When using the P20 to P22 pins as input port pins, on-chip pull-up resistors can be specified in 1-bit units by pull-up resistor option register 2 (PUB2).

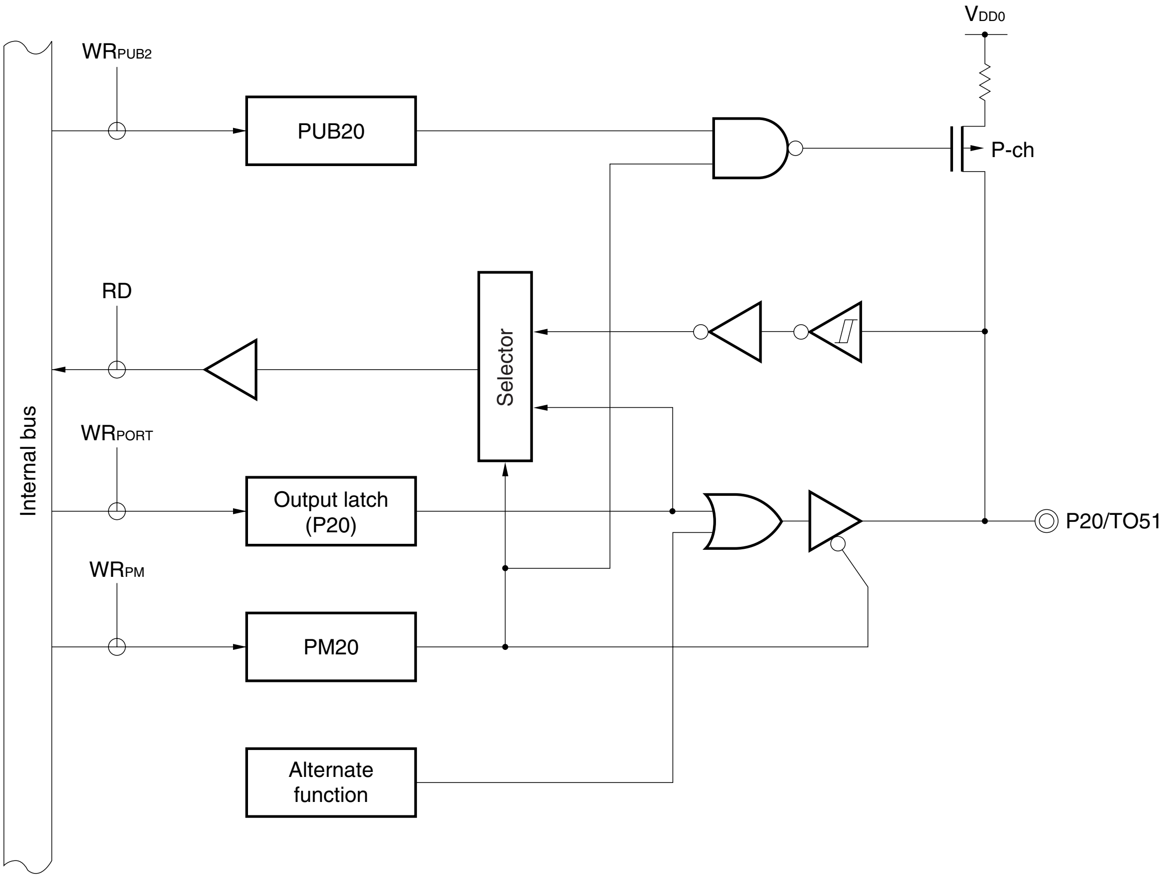
The port is also used as the serial interface I/O and timer output.

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

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

Caution When using the pins of port 2 as the serial interface, the I/O or output latch must be set according to the function to be used. For the setting method, refer to 9.3 Registers Controlling Serial Interface UART0.

Figure 4-4. Block Diagram of P20



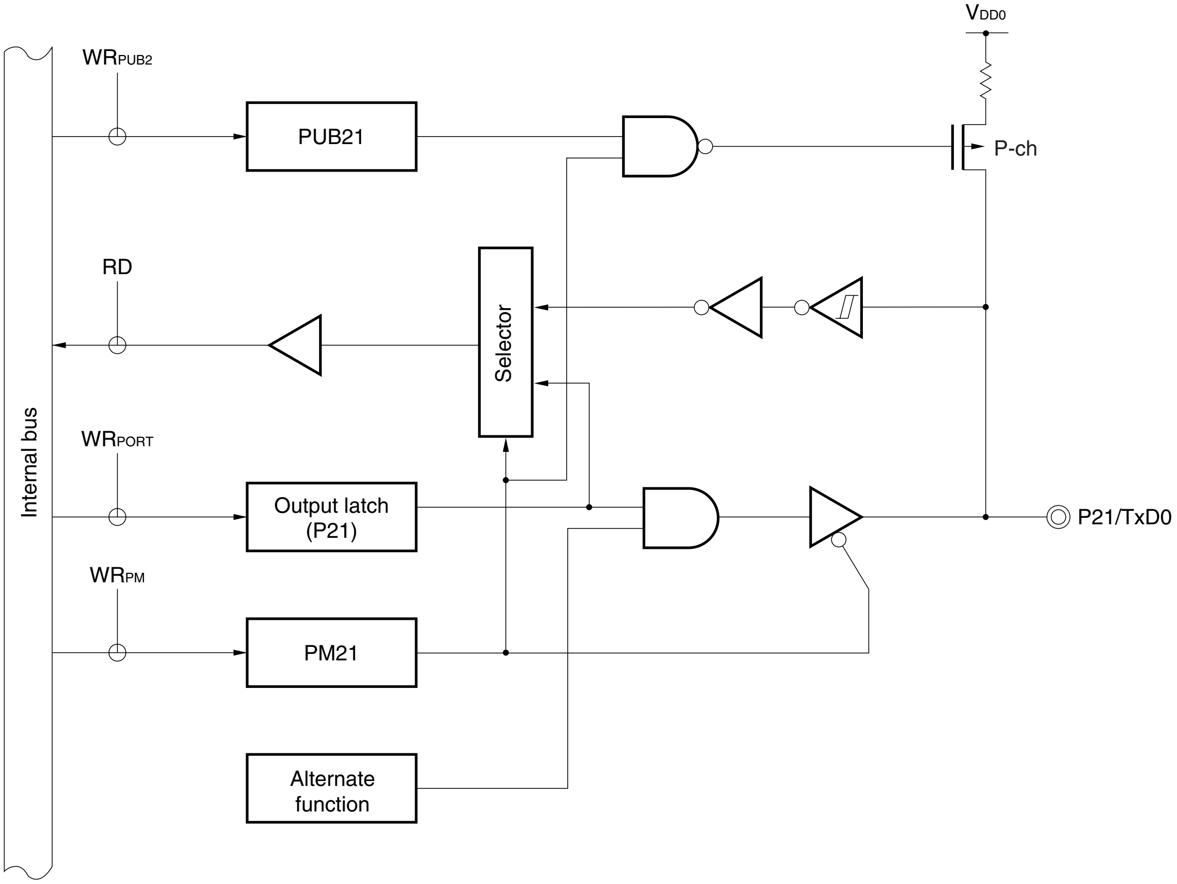
PUB2: Pull-up resistor option register 2

PM: Port mode register

RD: Port 2 read signal

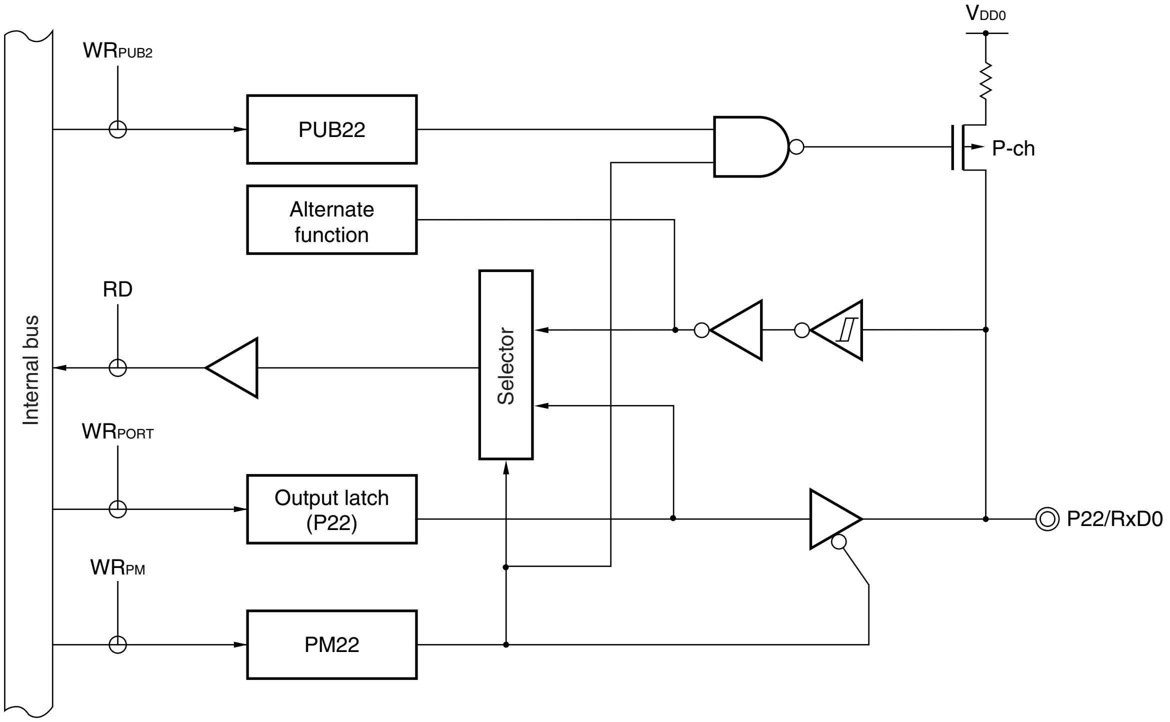
WR: Port 2 write signal

Figure 4-5. Block Diagram of P21



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

Figure 4-6. Block Diagram of P22



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

4.2.4 Port 3

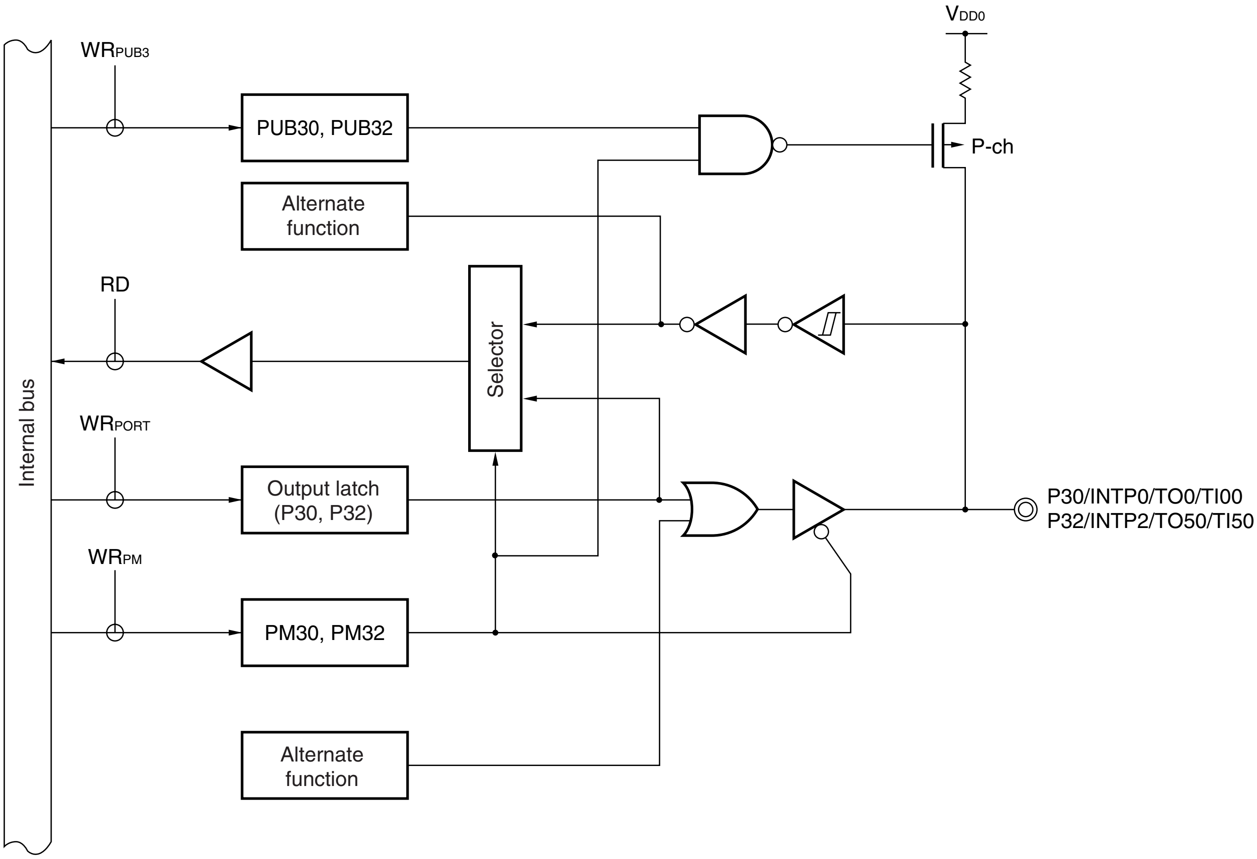
This is a 4-bit I/O port with an output latch. Port 3 can be specified in the input or output mode in 1-bit units by port mode register 3 (PM3). When using the P30 to P33 pins as input port pins, on-chip pull-up resistors can be specified in 1-bit units by pull-up resistor option register 3 (PUB3).

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

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

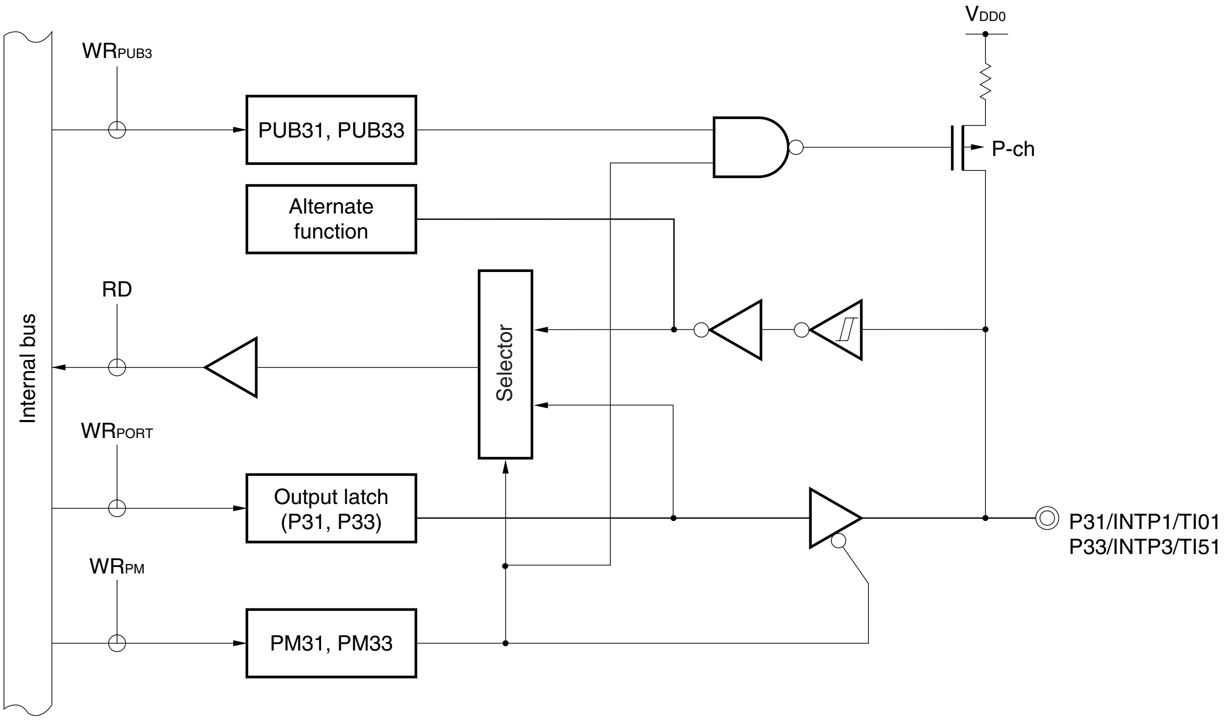
Figures 4-7 and 4-8 show block diagrams of port 3.

Figure 4-7. Block Diagram of P30 and P32



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

Figure 4-8. Block Diagram of P31 and P33



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

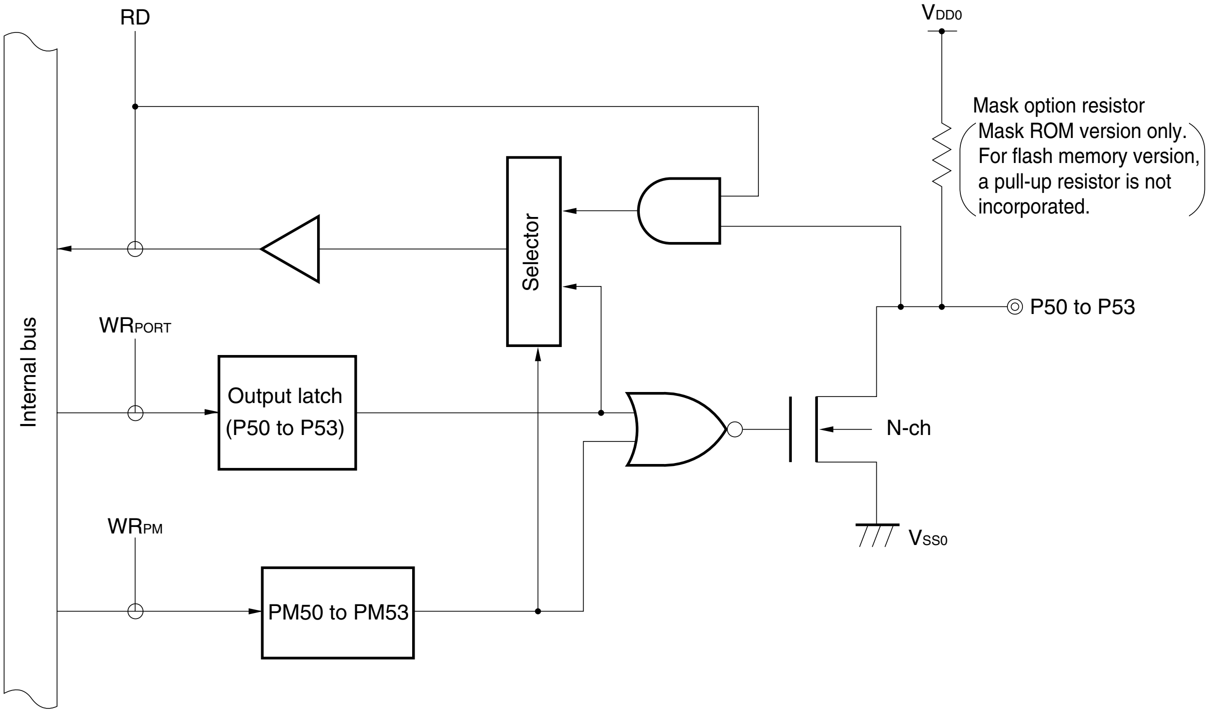
4.2.5 Port 5

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

$\overline{\text{RESET}}$ input sets port 5 to input mode.

Figure 4-9 shows a block diagram of port 5.

Figure 4-9. Block Diagram of P50 to P53



PM: Port mode register

RD: Port 5 read signal

WR: Port 5 write signal

4.2.6 Port 8

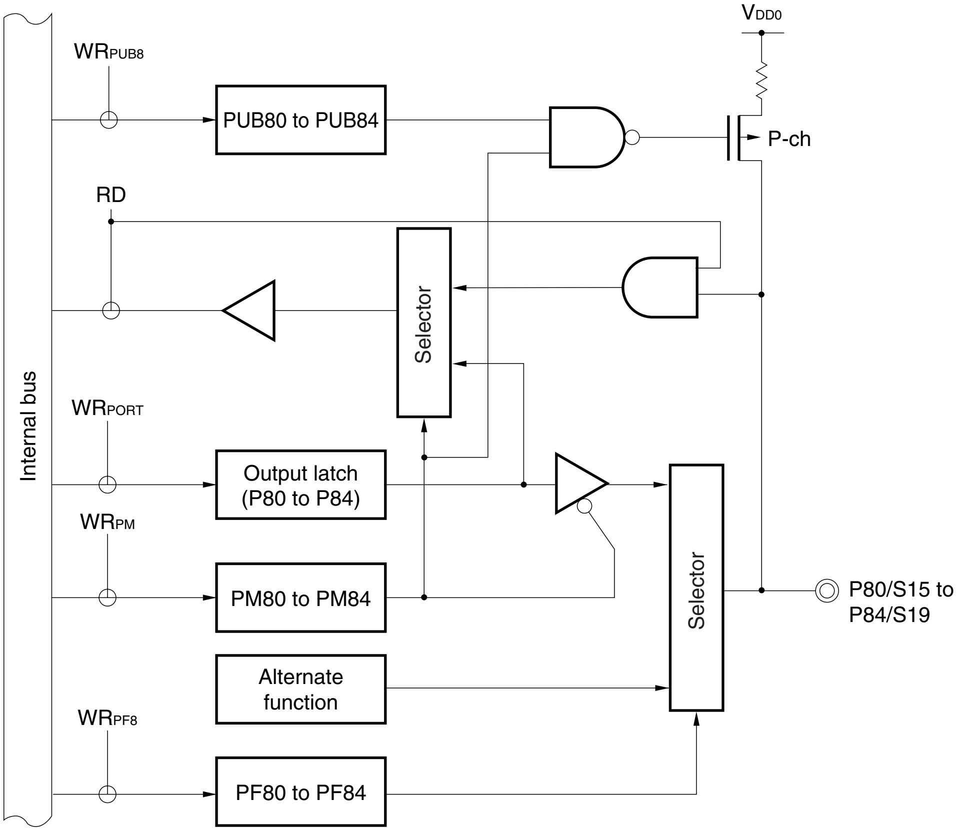
This is a 5-bit I/O port with an output latch. Only the bits for which a port function is selected by port function register 8 (PF8) can be used.

Port 8 can be specified in the input or output mode in 1-bit units by port mode register 8 (PM8). When the P80 to P84 pins are used as input port pins, on-chip pull-up resistors can be specified in 1-bit units by pull-up resistor option register 8 (PUB8).

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

Figure 4-10 shows a block diagram of port 8.

Figure 4-10. Block Diagram of P80 to P84



PUB8:	Pull-up resistor option register 8
PM:	Port mode register
PF8:	Port function register 8
RD:	Port 8 read signal
WR:	Port 8 write signal

4.2.7 Port 9

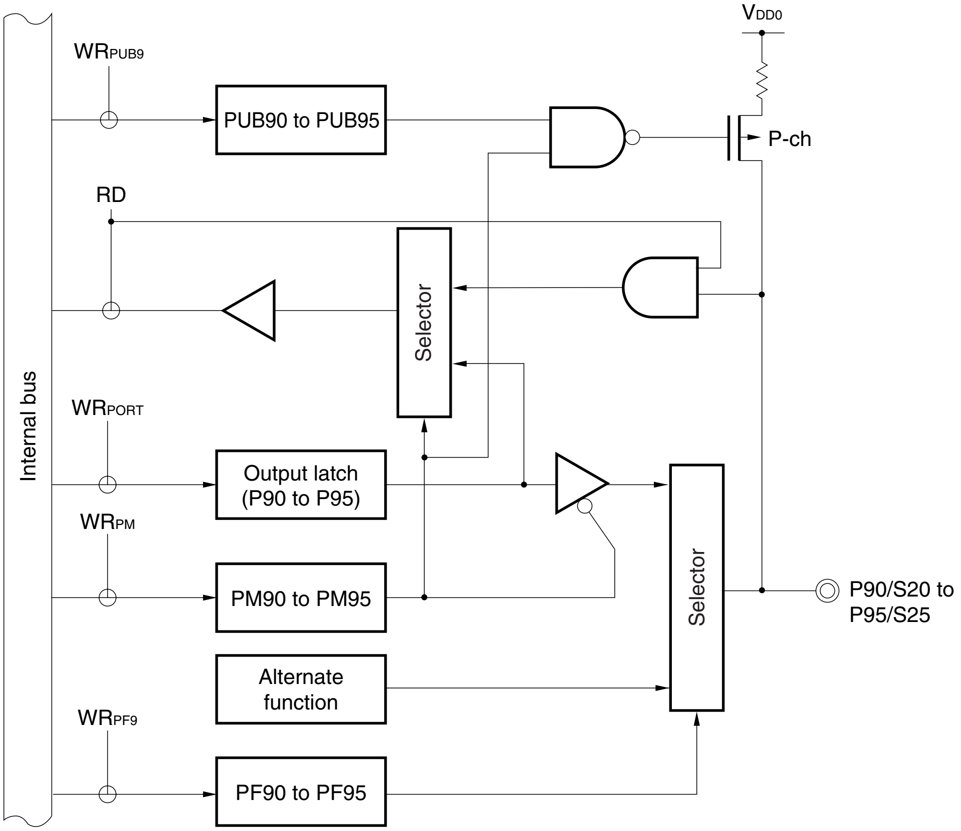
This is a 6-bit I/O port with an output latch. Only the bits for which a port function is selected by port function register 9 (PF9) can be used.

Port 9 can be specified in the input or output mode in 1-bit units by port mode register 9 (PM9). When the P90 to P95 pins are used as input port pins, on-chip pull-up resistors can be specified in 1-bit units by pull-up resistor option register 9 (PUB9).

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

Figure 4-11 shows a block diagram of port 9.

Figure 4-11. Block Diagram of P90 to P95



PUB9: Pull-up resistor option register 9

PM: Port mode register

PF9: Port function register 9

RD: Port 9 read signal

WR: Port 9 write signal

4.3 Registers Controlling Port Function

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

- Port mode registers (PM0 to PM3, PM5, PM8, PM9)
- Pull-up resistor option registers (PUB0 to PUB3, PUB8, PUB9)
- Port function registers (PF8, PF9)

(1) Port mode registers (PM0 to PM3, PM5, PM8, PM9)

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

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

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

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

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

Figure 4-12. Format of Port Mode Registers

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	1	1	PM11	PM10	FF21H	FFH	R/W
PM2	1	1	1	1	1	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
PM8	1	1	1	PM84	PM83	PM82	PM81	PM80	FF28H	FFH	R/W
PM9	1	1	PM95	PM94	PM93	PM92	PM91	PM90	FF29H	FFH	R/W

PMmn	Pmn pin input/output mode selection (m = 0 to 3, 5, 8, 9, n = 0 to 5)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

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

Pin Name	Alternate Function		PM××	P××
	Name	I/O		
P20	TO51	Output	0	0
P30	INTP0	Input	1	×
	TO0	Output	0	0
	TI00	Input	1	×
P31	INTP1	Input	1	×
	TI01	Input	1	×
P32	INTP2	Input	1	×
	TO50	Output	0	0
	TI50	Input	1	×
P33	INTP3	Input	1	×
	TI51	Input	1	×

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

Cautions 1. When using port 2 as a serial interface pin, the I/O and output latch must be set according to the function to be used. For the setting method, refer to 9.3 Registers Controlling Serial Interface UART0.

2. When using ports 8 and 9 as segment pins of the LCD controller/driver, set these ports using port function registers (PF8 and PF9).

(2) Pull-up resistor option registers (PUB0 to PUB3, PUB8, PUB9)

These registers set whether to use on-chip pull-up resistors for pins P00 to P03, P10, P11, P20 to P22, P30 to P33, P80 to P84, and P90 to P95. An on-chip pull-up resistor can be used only for those bits set to the input mode of a port for which the use of the on-chip pull-up resistor has been specified using PUB0 to PUB3, PUB8, and PUB9.

For those bits set to the output mode, on-chip pull-up resistors cannot be used, regardless of the setting of PUB0 to PUB3, PUB8, and PUB9. This also applies to alternate-function pins used as output pins[Note].

PUB0 to PUB3, PUB8, and PUB9 can be set by a 1-bit or 8-bit memory manipulation instruction.

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

Note When using ports 8 and 9 as segment outputs, be sure to clear the corresponding bits of PUB8 and PUB9 to 0.

Figure 4-13. Format of Pull-Up Resistor Option Registers

Symbol	7	6	5	4	<3>	<2>	<1>	<0>	Address	After reset	R/W
PUB0	0	0	0	0	PUB03	PUB02	PUB01	PUB00	FF30H	00H	R/W
	7	6	5	4	3	2	<1>	<0>			
PUB1	0	0	0	0	0	0	PUB11	PUB10	FF31H	00H	R/W
	7	6	5	4	3	<2>	<1>	<0>			
PUB2	0	0	0	0	0	PUB22	PUB21	PUB20	FF32H	00H	R/W
	7	6	5	4	<3>	<2>	<1>	<0>			
PUB3	0	0	0	0	PUB33	PUB32	PUB31	PUB30	FF33H	00H	R/W
	7	6	5	<4>	<3>	<2>	<1>	<0>			
PUB8	0	0	0	PUB84	PUB83	PUB82	PUB81	PUB80	FF38H	00H	R/W
	7	6	<5>	<4>	<3>	<2>	<1>	<0>			
PUB9	0	0	PUB95	PUB94	PUB93	PUB92	PUB91	PUB90	FF39H	00H	R/W

PUBmn	Pmn on-chip pull-up resistor selection (m = 0 to 3, 8, 9, n = 0 to 5)
0	An on-chip pull-up resistor is not connected.
1	An on-chip pull-up resistor is connected.

(3) Port function registers (PF8, PF9)

These registers set in 1-bit units whether to use port 8 and port 9 as port or segment pins.

PF8 and PF9 can be set by an 8-bit memory manipulation instruction.

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

Figure 4-14. Format of Port Function Registers

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PF8	0	0	0	PF84	PF83	PF82	PF81	PF80	FF58H	00H	W
	7	6	5	4	3	2	1	0			
PF9	0	0	PF95	PF94	PF93	PF92	PF91	PF90	FF59H	00H	W

PFmn	Pmn function selection (m = 8, 9, n = 0 to 5)
0	Pmn is used as I/O port
1	Pmn is used as segment output

Cautions 1. Each bit of ports 8 and 9 function as segment output pins when PFmn = 1, regardless of the values of the corresponding port mode register and output latch. At this time, the values of the port mode register and output latch are invalid.

2. If an on-chip pull-up resistor is connected (PUBmn = 1), the pull-up resistor is not disconnected even when PFmn is set to 1. Be sure to clear PUBmn to 0 when setting PFmn to 1.

4.4 Port Function Operation

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

4.4.1 Writing to I/O port

(1) In output mode

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

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.

4.4.2 Reading from I/O port

(1) In output mode

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

(2) In input mode

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

4.4.3 Arithmetic operation of I/O port

(1) In output mode

An arithmetic operation can be performed 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.

CHAPTER 5 CLOCK GENERATOR

5.1 Clock Generator Functions

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

- **Main system clock oscillator**
  This circuit oscillates a clock of 500 kHz (TYP.). Oscillation can be stopped by setting the processor clock control register (PCC).
- **Subsystem clock oscillator**
  This circuit oscillates a clock of 32.768 kHz. Oscillation cannot be stopped.
  A clock whose frequency is four times that of the subsystem clock can be selected and supplied to the CPU and peripheral hardware by using a ×4 multiplication circuit.

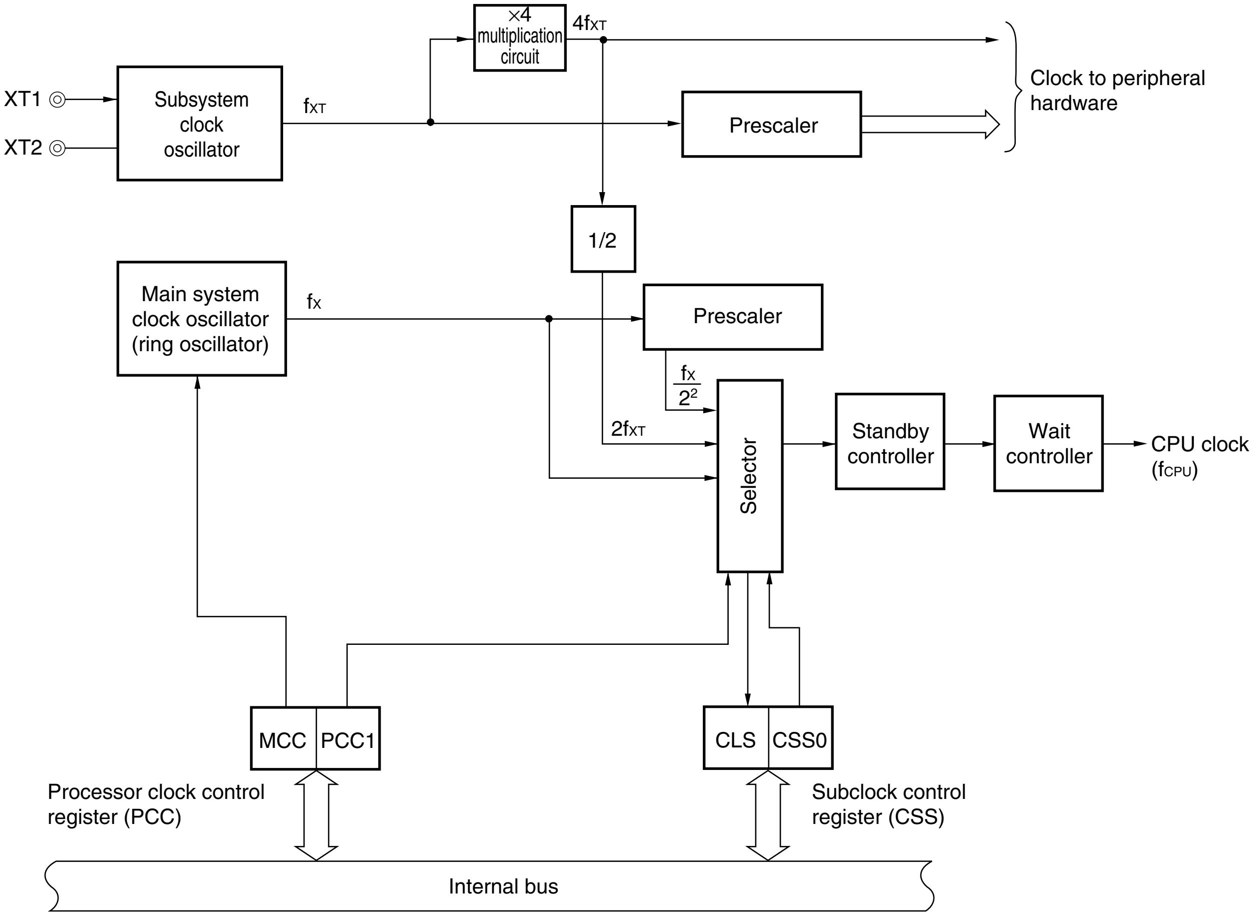
5.2 Clock Generator Configuration

The clock generator includes the following hardware.

Table 5-1. Configuration of Clock Generator

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

Figure 5-1. Block Diagram of Clock Generator


×4 multiplication circuit
4fXT
XT1
XT2
Subsystem clock oscillator
fXT
Prescaler
Clock to peripheral hardware
1/2
Main system clock oscillator (ring oscillator)
fX
Prescaler
fX/2²
2fXT
Selector
Standby controller
Wait controller
CPU clock (fCPU)
MCC
PCC1
CLS
CSS0
Processor clock control register (PCC)
Subclock control register (CSS)
Internal bus

5.3 Registers Controlling Clock Generator

The clock generator is controlled by the following two registers.

- Processor clock control register (PCC)
- Subclock control register (CSS)

(1) Processor clock control register (PCC)

This register is used to select the CPU clock and set the main system clock oscillator operation/stop.
PCC can be set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets PCC to 02H.

Figure 5-2. Format of Processor Clock Control Register

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

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

CSS0	PCC1	CPU clock (f_{CPU}) selection[Note]	Minimum instruction execution time: $2/f_{CPU}$
			At f_X = 500 kHz (TYP.), f_{XT} = 32.768 kHz operation
0	0	f_X	4 μs
0	1	$f_X/2^2$	16 μs
1	0	$4f_{XT}/2$	30.5 μs
1	1	Setting prohibited	

Note The CPU clock is selected by a combination of flag settings in the PCC and CSS registers (refer to **5.3 (2) Subclock control register (CSS)**).

Cautions 1. Be sure to clear bits 0 and 2 to 6 to 0.

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

3. After reset has been released, the main system clock is selected as the CPU clock. To select the subsystem clock, clear PCC to 0 and then set CSS0 to 1.

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

3. The minimum instruction execution time shown above when the main system clock is selected is for reference only.

(2) Subclock control register (CSS)

This register is used to select the CPU clock and specifies the CPU clock operation status.

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

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

Figure 5-3. 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 main system clock (ring oscillator)
1	Operation based on the output of the ×4 subsystem clock multiplication circuit

CSS0	Selection of the CPU clock
0	Output from the main system clock (ring oscillator)
1	Output from the ×4 subsystem clock multiplication circuit

Note Bit 5 is read only.

Caution Be sure to clear bits 0 to 3, 6, and 7 to 0.

5.4 System Clock Oscillators

5.4.1 Main system clock oscillator

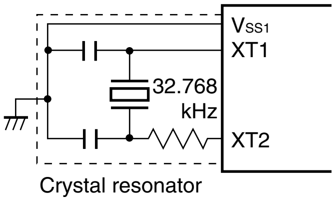
The main system clock oscillator is oscillated by the internal ring oscillator (500 kHz (TYP.)).

5.4.2 Subsystem clock oscillator

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

Figure 5-4 shows the external circuit of the subsystem clock oscillator.

Figure 5-4. External Circuit of Subsystem Clock Oscillator



Caution Wire as follows in the area enclosed by the broken lines in Figure 5-4 to avoid an adverse effect from wiring capacitance.

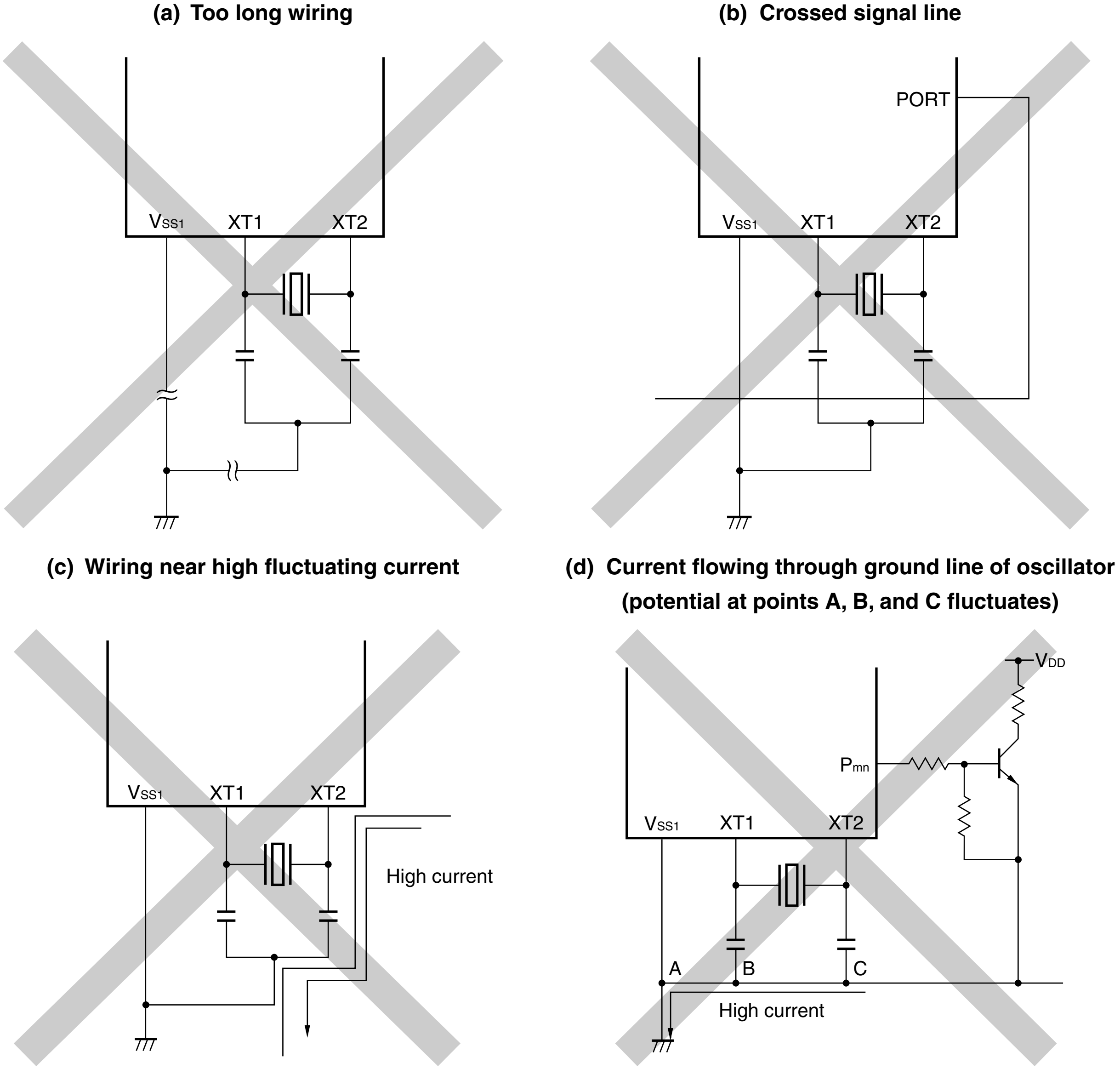
- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as V_{SS}. Do not ground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

5.4.3 Example of incorrect resonator connection

Figure 5-5 shows examples of incorrect resonator connection.

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



(a) Too long wiring

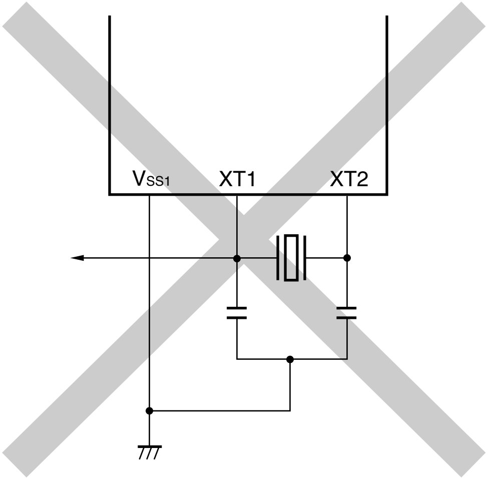
(b) Crossed signal line

(c) Wiring near high fluctuating current

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

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

(e) Signal is fetched



5.4.4 Divider circuit

The divider circuit divides the output of the subsystem clock oscillator (f_{XT}) to generate various clocks.

5.4.5 ×4 subsystem clock multiplication circuit

This circuit multiplies the subsystem clock by four and supplies the resultant clock to the CPU and peripheral hardware.

It stops operating during the HALT period (to reduce the power consumption).

5.5 Clock Generator Operation

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

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

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

(a) The low-speed mode (16 μs: at 500 kHz (TYP.) 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) To change the CPU clock from the main system clock to the subsystem clock, clear bit 1 (PCC1) of PCC to 0 in advance.

(c) Three minimum instruction execution time levels (4 μs, 16 μs: with main system clock (at 500 kHz (TYP.) operation), 30.5 μs: subsystem clock (at 32.768 kHz operation) multiplied by four) can be selected by setting PCC and CSS.

(d) 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 as a standby function.

(e) The clock pulse for the peripheral hardware is generated by dividing the frequency of the subsystem clock. Peripheral hardware can continue running even in standby (HALT) mode. However, a clock multiplied by four cannot be supplied to the peripheral hardware in the HALT mode because the ×4 multiplication circuit stops.

5.6 Changing Setting of System Clock and CPU Clock

5.6.1 Time required for switching between system clock and CPU clock

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

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

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

CPU Clock	Set Value Before Switching	Set Value After Switching	Maximum Time for Switching
Low-speed main → High-speed main	CSS0 = 0, PCC1 = 1	CSS0 = 0, PCC1 = 0	2 clocks
High-speed main → Sub multiplied by four	CSS0 = 0, PCC1 = 0	CSS0 = 1, PCC1 = 0	$2f_X/4f_{XT}$ clocks (8 clocks)
Sub multiplied by four → High-speed main	CSS0 = 1, PCC1 = 0	CSS0 = 0, PCC1 = 0	2 clocks
Other switching			Setting prohibited

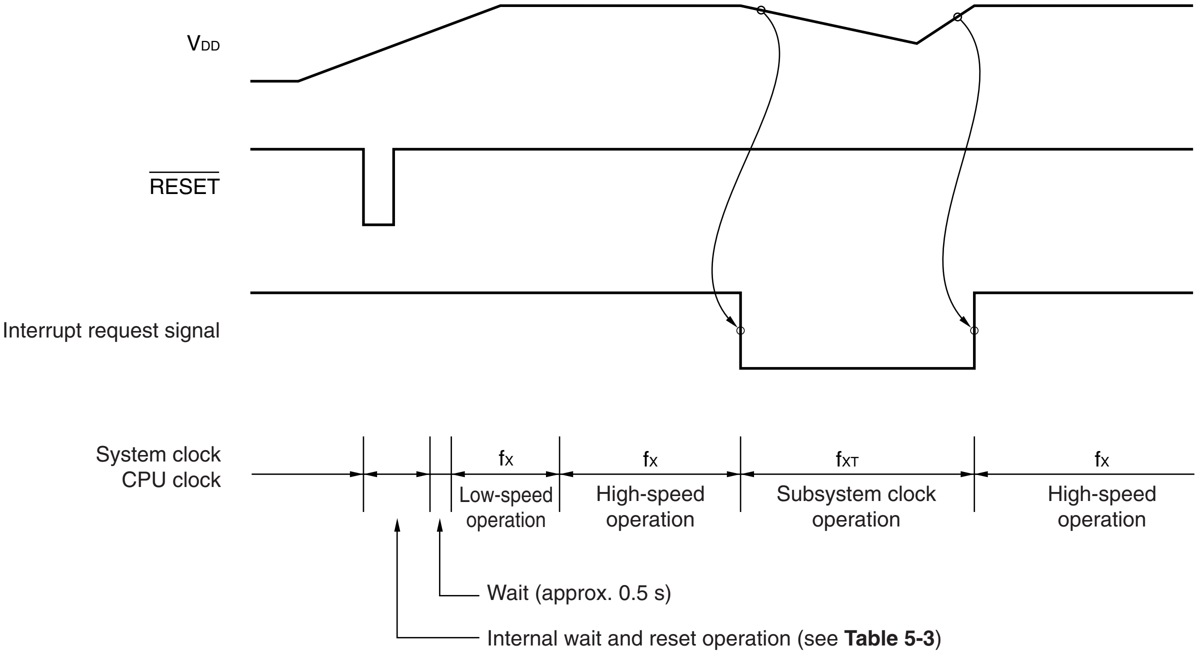
Remarks 1. Two clocks are the minimum instruction execution time of the CPU clock before switching.

2. The parenthesized values apply to operation at f_X = 500 kHz (TYP.) or f_{XT} = 32.768 kHz.

5.6.2 Switching between system clock and CPU clock

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

Figure 5-6. Switching Between System Clock and CPU Clock



<1> The CPU is reset when the $\overline{\text{RESET}}$ pin is made low on power application. When the $\overline{\text{RESET}}$ pin is later made high, an internal wait time of several seconds (refer to **Table 5-3**) elapses to ensure that subclock oscillation is stabilized. After that, a wait time of about 0.5 seconds elapses until the internal regulator is stabilized.

<2> The CPU starts executing instructions at the low speed of the main system clock (16 μs: at 500 kHz (TYP.) operation). Change the speed to high (4 μs: at 500 kHz (TYP.) operation) by rewriting bit 1 (PCC1) of the processor clock control register (PCC) at the initial stage of the user program.

<3> Detect a drop in the V_{DD} voltage by using an interrupt request signal, and set CSS0 to 1 to select the subsystem clock multiplied by four (it is also possible to stop the main system clock by later setting bit 7 (MCC) of PCC to 1).

<4> When a recovery of the V_{DD} voltage is detected by an interrupt request signal, clear CSS0 to 0 and change the speed of the main system clock to high (if the main system clock has been stopped, clear bit 7 (MCC) of PCC to 0 to start main system clock oscillation before rewriting CSS0). At this time, no wait time is necessary to stabilize oscillation[Note].

Note Oscillation of the main system clock is stabilized during the period of one subsystem clock. It is therefore not necessary to ensure the lapse of the oscillation stabilization time when the main system clock operation is started while the main system clock is stopped (during subsystem clock operation).

Table 5-3. Internal Reset Wait Time

Supply Voltage	Wait Time (Reference)
When V_{DD} = 3.5 V	Approx. 3 to 7 s
When V_{DD} = 3.0 V	Approx. 5 to 9.5 s
When V_{DD} = 2.7 V	Approx. 6 to 12 s

Remark The values in this table are for reference only as the wait time varies depending on the temperature, voltage, and production variation.

CHAPTER 6 16-BIT TIMER/EVENT COUNTER 0

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

6.1 Functions of 16-Bit Timer/Event Counter 0

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

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

(1) Interval timer

16-bit timer/event counter 0 generates interrupt requests at a preset time interval.

(2) External event counter

16-bit timer/event counter 0 can measure the number of pulses of an externally input signal.

(3) PPG output

16-bit timer/event counter 0 can output a rectangular wave whose frequency and output pulse width can be set freely.

(4) Pulse width measurement

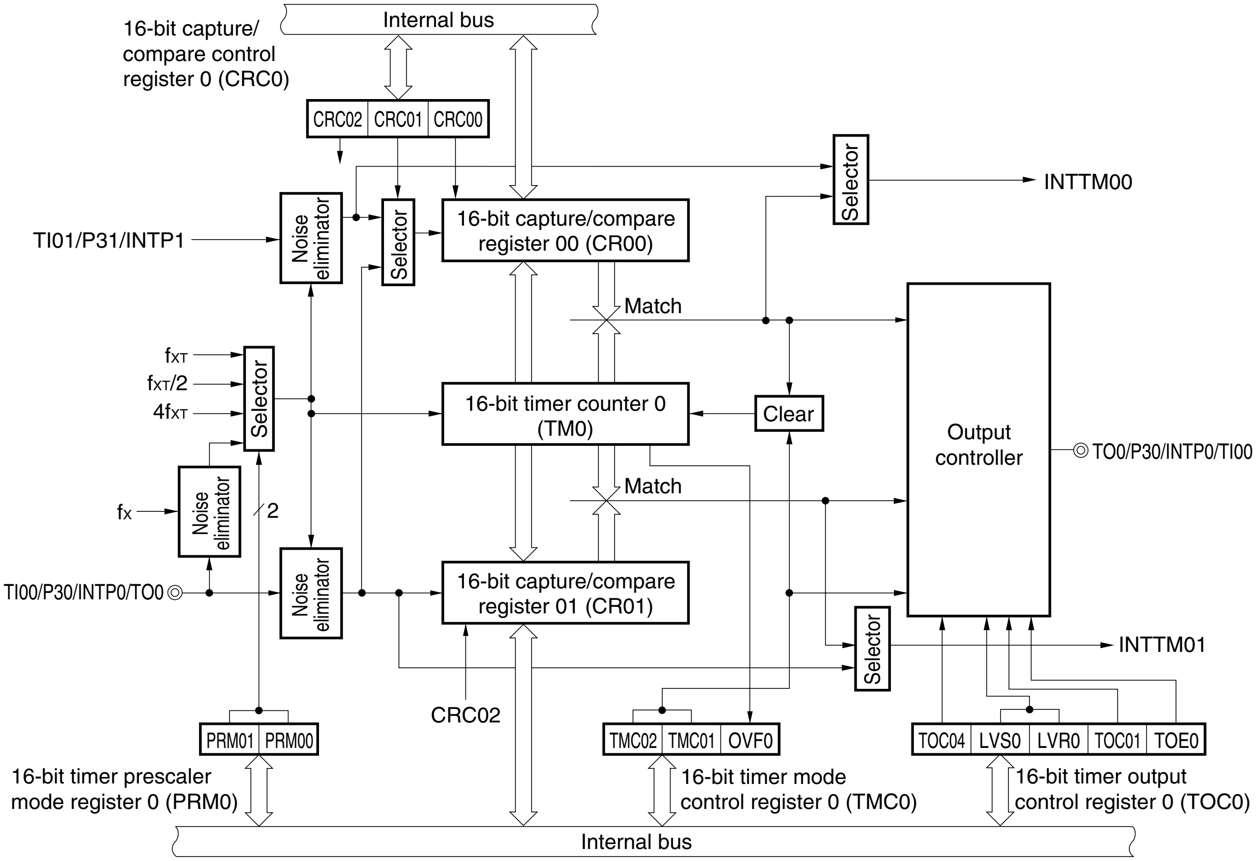
16-bit timer/event counter 0 can measure the pulse width of an externally input signal.

(5) Square wave output

16-bit timer/event counter 0 can output a square wave of any selected frequency.

Figure 6-1 shows the block diagram of 16-bit timer/event counter 0.

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



6.2 Configuration of 16-Bit Timer/Event Counter 0

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

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

Item	Configuration
Timer register	16 bits × 1 (TM0)
Register	Capture/compare register: 16 bits × 2 (CR00 and CR01)
Timer output	1 (TO0)
Control registers	16-bit timer mode control register 0 (TMC0) 16-bit capture/compare control register 0 (CRC0) 16-bit timer output control register 0 (TOC0) 16-bit timer prescaler mode register 0 (PRM0) Port mode register 3 (PM3)

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

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

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

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

<2> If TMC01 and TMC02 are cleared to 0

<3> If the valid edge of TI00 is input in the clear & start mode entered by inputting the valid edge of TI00

<4> If TM0 and CR00 match in the clear & start mode entered on a match between TM0 and CR00

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

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

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

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

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

  It is possible to use the valid edge of the TI00/TO0/P30/INTP0 pin or TI01/P31/INTP1 pin as the capture trigger. When CR00 is captured at the valid edge of the TI01 pin, an interrupt request (INTTM00) is generated. The valid edges of TI00 and TI01 pins are specified by setting 16-bit timer prescaler mode register 0 (PRM0).

  When CR00 is specified as a capture register and the valid edge of the TI00/TO0/P30/INTP0 pin is specified as the capture trigger, the situation is as shown in Table 6-2. On the other hand, when the valid edge of the TI01/P31/INTP1 pin is specified as the capture trigger, the situation is as shown in Table 6-3.

Table 6-2. TI00 Pin Valid Edge and Capture Trigger of CR00 and CR01

ES01	ES00	TI00/TO0/P30/INTP0 Pin Valid Edge	CR00 Capture Trigger	CR01 Capture Trigger
0	0	Falling edge	Rising edge	Falling edge
0	1	Rising edge	Falling edge	Rising edge
1	0	Setting prohibited	Setting prohibited	Setting prohibited
1	1	Both rising and falling edges	No capture operation	Both rising and falling edges

Table 6-3. TI01 Pin Valid Edge and Capture Trigger of CR00

ES11	ES10	TI01/P31/INTP1 Pin Valid Edge	CR00 Capture Trigger
0	0	Falling edge	Falling edge
0	1	Rising edge	Rising edge
1	0	Setting prohibited	Setting prohibited
1	1	Both rising and falling edges	Both rising and falling edges

CR00 can be set by a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CR00 to 0000H.

Cautions 1. Set a value other than 0000H in CR00 in the mode in which clear & start occurs on a match of TM00 and CR00.

If CR00 is set to 0000H in the free-running mode and in the clear mode using the valid edge of the TI00 pin, an interrupt request (INTTM00) is generated when the value of CR00 changes from 0000H to 0001H following TM00 overflow (FFFFH). Moreover, INTTM00 is generated after a match of TM00 and CR00 is detected or a valid edge of the TI01 pin is detected.

2. When pin P30 is used as the valid edge of TI00, it cannot be used as the timer output (TO0). Also, if it is used as TO0, it cannot be used as the valid edge of TI00.

3. CR00 does not generate an interrupt request if it is captured at the valid edge of the TI00 pin.

4. Do not select the TI00 valid edge as the count clock when using TI00 as a capture trigger.

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

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

- **When CR01 is used as a compare register**

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

- **When CR01 is used as a capture register**

  It is possible to use the valid edge of the TI00/TO0/P30/INTP0 pin as the capture trigger. When CR01 is captured, an interrupt request (INTTM01) is generated. The valid edge of TI00/TO0/P30/INTP0 is specified by setting 16-bit timer prescaler mode register 0 (PRM0).

CR01 can be set by a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CR01 to 0000H.

Cautions 1. If CR01 is cleared to 0000H, an interrupt request (INTTM01) is generated when the value of CR01 changes from 0000H to 0001H following TM00 overflow (FFFFH). Moreover, INTTM01 is generated after a match of TM00 and CR01 is detected or a valid edge of the TI00 pin is detected.

2. When using CR01 as a capture register, do not select the TI00 valid edge as the count clock.

6.3 16-Bit Timer/Event Counter 0 Control Registers

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

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

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

This register is used to detect an overflow and to set the 16-bit timer operation mode, the 16-bit timer counter 0 clear mode, and output timing.

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

$\overline{\text{RESET}}$ input clears TMC0 to 00H.

Caution 16-bit timer counter 0 starts operation at the moment TMC01 and TMC02 are set to values other than 0, 0 (operation stop mode) respectively. To stop operation, be sure to clear TMC01 and TMC02 to 0, 0.

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

Address: FF66H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
TMC0	0	0	0	0	TMC02	TMC01	0	OVF0

TMC02	TMC01	Operation mode and clear mode selection	TO0 output timing selection
0	0	Operation stop (TM0 cleared to 0)	Not changed
0	1	Free-running mode	Match between TM0 and CR00 or TM0 and CR01
1	0	Clear & start on valid edge of TI00	–
1	1	Clear & start on match between TM0 and CR00	Match between TM0 and CR00 or TM0 and CR01

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

Cautions 1. For bits other than the OVF0 flag, writing should be performed after timer operation has stopped.

2. The valid edge of the TI00/TO0/INTP0/P30 pin is specified by setting 16-bit timer prescaler mode register 0 (PRM0).

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

4. When clear & start mode on the valid edge of TI00 is selected, do not select the TI00 valid edge as the count clock.

5. Be sure to clear bits 1 and 4 to 7 to 0.

Remark

TO0:	16-bit timer/event counter output pin
TI00:	16-bit timer/event counter input pin
TM0:	16-bit timer counter 0
CR00:	16-bit capture/compare register 00
CR01:	16-bit capture/compare register 01

(2) 16-bit capture/compare control register 0 (CRC0)

This register is used to control the operation of the 16-bit capture/compare registers (CR00 and CR01).

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

$\overline{\text{RESET}}$ input clears CRC0 to 00H.

Figure 6-3. Format of 16-Bit Capture/Compare Control Register 0

Address: FF68H After reset: 00H R/W

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

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

CRC01	CR00 capture trigger selection
0	Captures on valid edge of TI01
1	Captures on inverted valid edge of TI00[Note]

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

Note If both the rising and falling edges are selected as the valid edge of TI00, CR00 cannot perform a capture operation.

Cautions 1. Timer operation must be stopped before setting CRC0.

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

3. To surely perform the capture operation, the capture trigger requires a pulse two times longer than the count clock selected by 16-bit timer prescaler mode register 0 (PRM0).

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

This register controls the operation of the 16-bit timer/event counter 0 output controller. It controls R-S type flip-flop (LV0) set/reset, output inversion enable/disable, and timer output enable/disable of 16-bit timer/event counter 0.

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

$\overline{\text{RESET}}$ input clears TOC0 to 00H.

Figure 6-4 shows the TOC0 format.

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

Address: FF69H After reset: 00H R/W

Symbol	7	6	5	4	<3>	<2>	1	<0>
TOC0	0	0	0	TOC04	LVS0	LVR0	TOC01	TOE0

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

LVS0	LVR0	16-bit timer/event counter 0 timer output F/F status setting
0	0	Not changed
0	1	Timer output F/F reset to 0
1	0	Timer output F/F set to 1
1	1	Setting prohibited

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

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

Cautions 1. Timer operation must be stopped before setting TOC0.

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

3. Be sure to clear bits 5 to 7 to 0.

(4) 16-bit timer prescaler mode register 0 (PRM0)

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

PRM0 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears PRM0 to 00H.

Figure 6-5. Format of 16-Bit Timer Prescaler Mode Register 0

Address: FF67H After reset: 00H R/W

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

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

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

PRM01	PRM00	Count clock selection
0	0	f_{XT} (32.768 kHz)
0	1	$f_{XT}/2$ (16.384 kHz)
1	0	$4f_{XT}$ (131 kHz)[Note 1]
1	1	TI00 valid edge[Note 2]

Notes 1. Because the ×4 multiplication clock circuit stops during HALT operation, $4f_{XT}$ cannot be selected as the count clock.

2. The external clock (TI00) must have a pulse width wider than two main system clocks (f_X). It can be selected only when the main system clock is operating (MCC = 0) because the main system clock is used as a sampling clock for eliminating noise.

Cautions 1. To select the valid edge of TI00 as a count clock, do not set the mode in which the timer is cleared and started by the valid edge of TI00, and do not specify TI00 as a capture trigger. In this case, P30/TI00/TO0/INTP0 pin cannot also be used as a timer output (TO0).

2. Timer operation must be stopped before setting PRM0.

3. When the TI00 or TI01 pin is high immediately after system reset, if the valid edge of TI00 or TI01 pin is specified as the rising edge or both rising and falling edges and the operation of 16-bit timer counter 0 (TM0) is enabled, the rising edge will be detected immediately after these settings. Therefore, be careful in cases when the TI00 or TI01 pin is pulled up. No rising edge will be detected, however, when the operation of TM0 is enabled again after being stopped.

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

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

4. Values in parentheses apply to the operation at f_{XT} = 32.768 kHz.

(5) Port mode register 3 (PM3)

This register is used to set the input/output mode of port 3 in 1-bit units.
When the P30/TI00/TO0/INTP0 pin is used as a timer input (TI00), set PM30 to 1.
When it is used as a timer output, clear PM30 and the output latch of P30 to 0.
When the P31/TI01/INTP1 pin is used as a timer input (TI01), set PM31 to 1.
PM3 can be set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets PM3 to FFH.

Figure 6-6. Format of Port Mode Register 3

Address: FF23H After reset: FFH R/W

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

PM3n	P3n pin input/output mode selection (n = 0 to 3)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

6.4 16-Bit Timer/Event Counter 0 Operations

6.4.1 Interval timer operation

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

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

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

The value of the compare register cannot be changed during timer count operation.

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

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

					TMC02	TMC01		OVF0
TMC0	0	0	0	0	1	1	0	0

TMC02, TMC01: Clear & start mode on match between TM0 and CR00

(b) 16-bit capture/compare control register 0 (CRC0)

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

CRC00: Uses CR00 as compare register

Figure 6-8. Configuration Diagram for Interval Timer

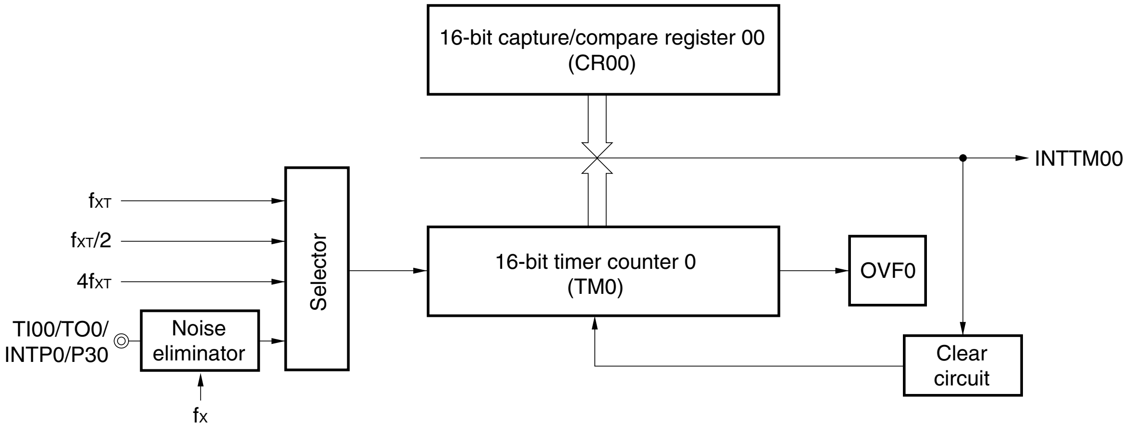
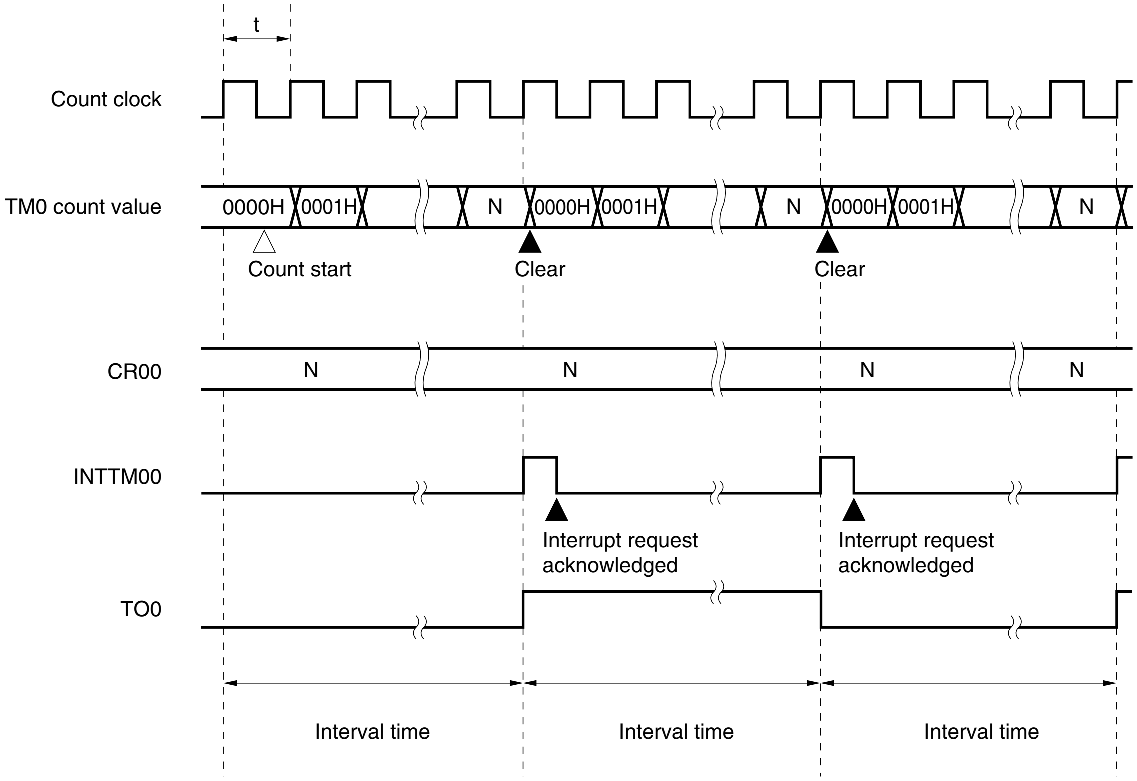


Figure 6-9. Interval Timer Operation Timing



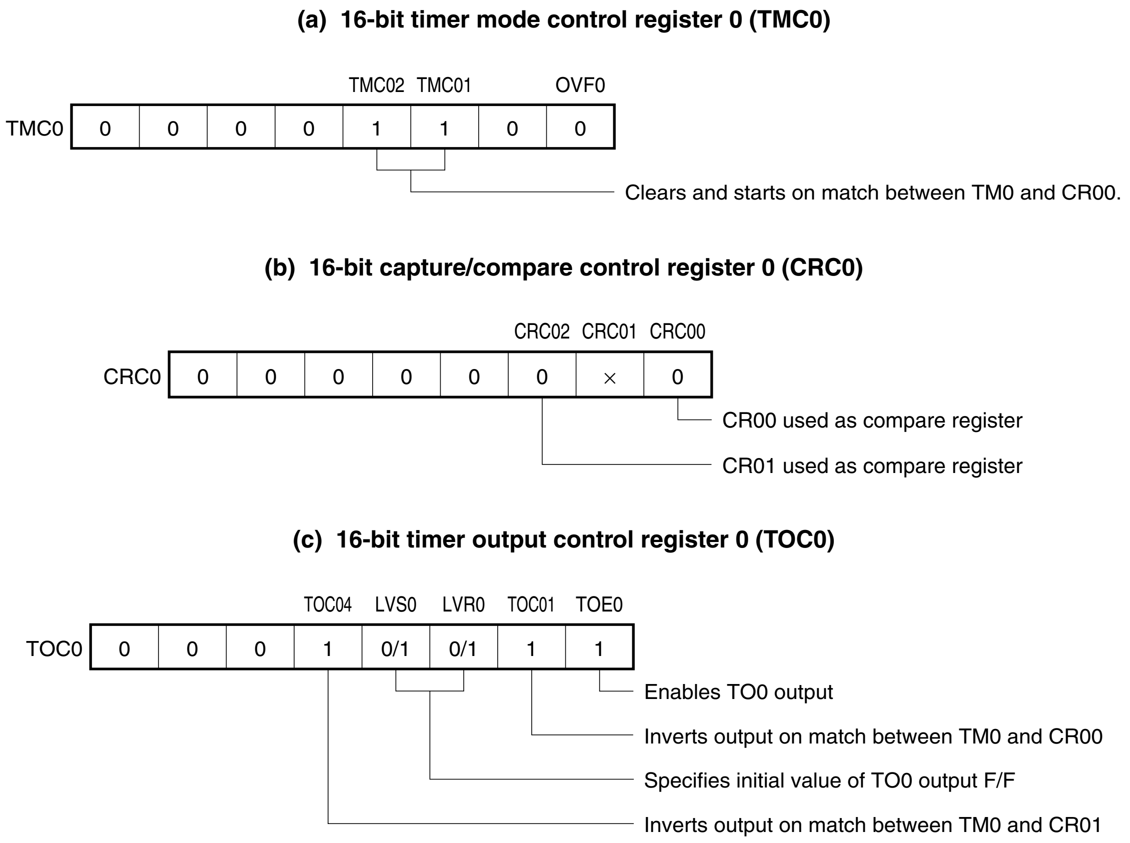
Remark Interval time = (N + 1) × t: N = 00H to FFH

6.4.2 PPG output operations

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

In the PPG output operation, rectangular waves are output from the TO0/TI00/INTP0/P30 pin with the pulse width and the cycle that correspond to the count values preset in 16-bit capture/compare register 01 (CR01) and in 16-bit capture/compare register 00 (CR00), respectively.

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



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

(b) 16-bit capture/compare control register 0 (CRC0)

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

TMC02
TMC01
OVF0
TMC0
0 0 0 0 1 1 0 0
Clears and starts on match between TM0 and CR00.
CRC02
CRC01
CRC00
CRC0
0 0 0 0 0 0 × 0
CR00 used as compare register
CR01 used as compare register
TOC04
LVS0
LVR0
TOC01
TOE0
TOC0
0 0 0 1 0/1 0/1 1 1
Enables TO0 output
Inverts output on match between TM0 and CR00
Specifies initial value of TO0 output F/F
Inverts output on match between TM0 and CR01

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

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

Remark ×: don’t care

Figure 6-11. Configuration of PPG Output

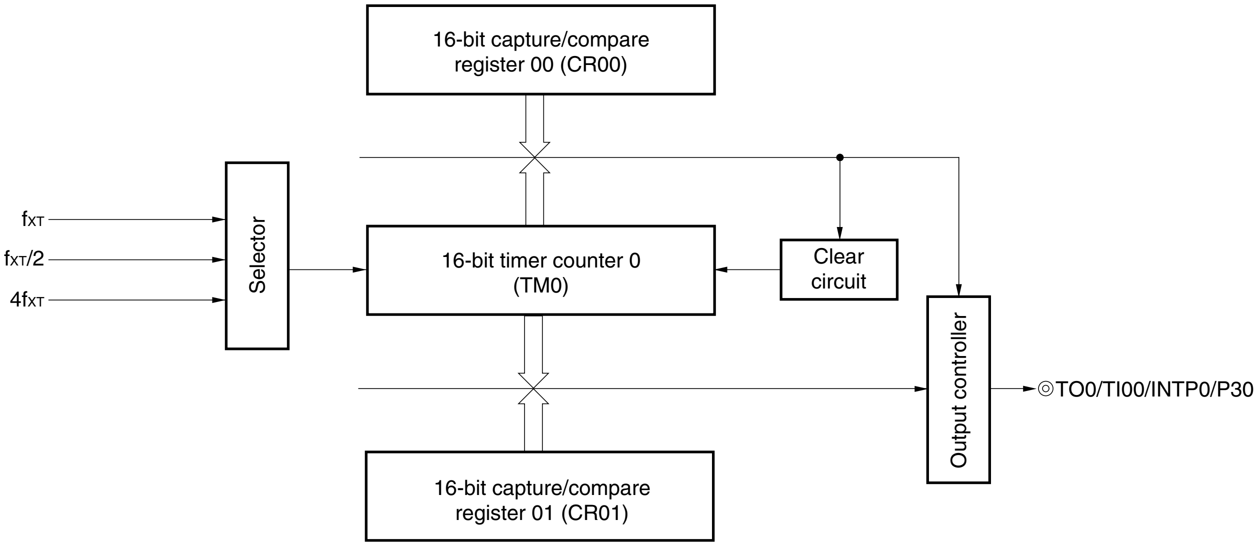
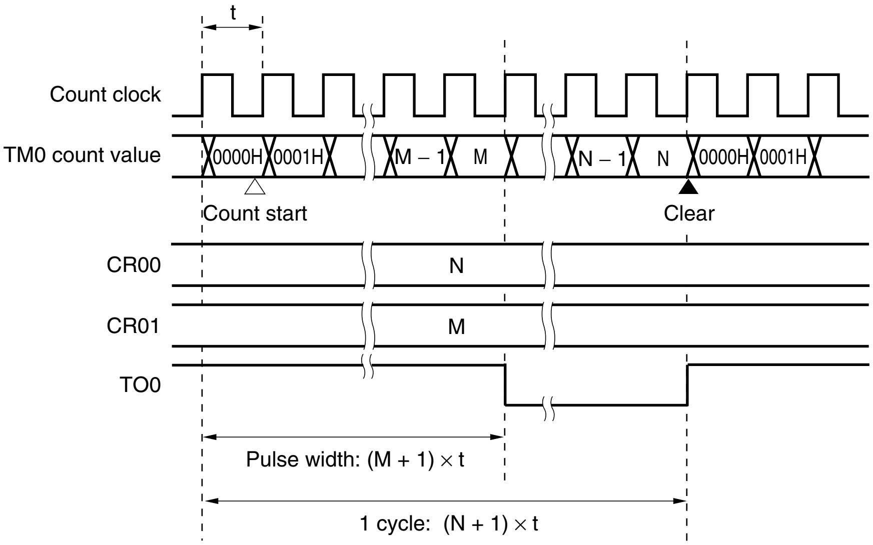

16-bit capture/compare register 00 (CR00)
fXT
fXT/2
4fXT
Selector
16-bit timer counter 0 (TM0)
Clear circuit
Output controller
◎TO0/TI00/INTP0/P30
16-bit capture/compare register 01 (CR01)

Figure 6-12. PPG Output Operation Timing



Remark 0000H < M < N ≤ FFFFH

6.4.3 Pulse-width measurement operations

It is possible to measure the pulse width of the signal input to the TI00/TO0/P30/INTP0 pin and TI01/P31/INTP1 pin using 16-bit timer counter 0 (TM0).

There are two measurement methods: measuring with TM0 used in free-running mode, and measuring by restarting the timer in synchronization with the edge of the signal input to the TI00/TO0/P30/INTP0 pin.

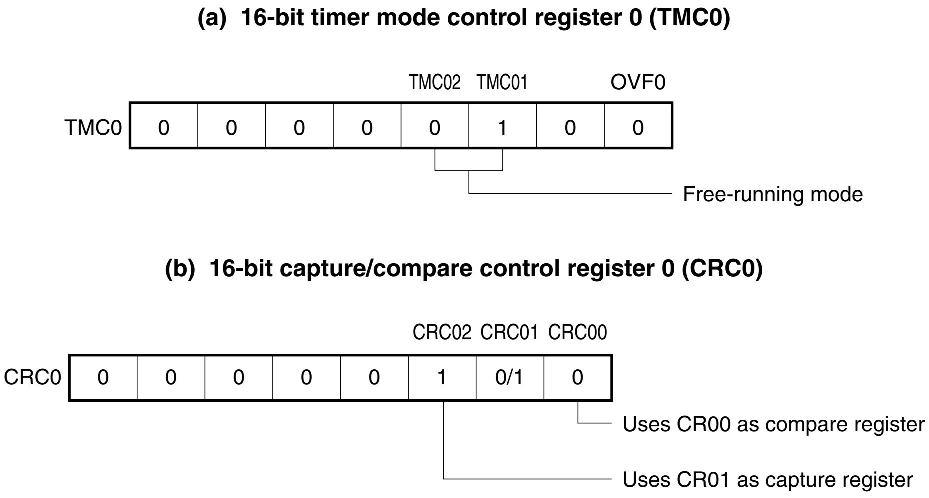
(1) Pulse width measurement with free-running counter and one capture register

When 16-bit timer counter 0 (TM0) is operated in free-running mode (see the register setting in **Figure 6-10**), if the edge specified by 16-bit timer prescaler mode register 0 (PRM0) is input to the TI00/TO0/P30/INTP0 pin, the value of TM0 is taken into 16-bit capture/compare register 01 (CR01) and an external interrupt request signal (INTTM01) is set.

The valid edge of the TI00 pin is specified by bits 4 and 5 (ES00 and ES01) of PRM0, and the rising edge, falling edge, or both edges can be selected.

With valid edge detection, sampling is performed at the count clock interval selected using PRM0, and a capture operation is only performed when a valid level is detected twice, thus eliminating noise with a short-pulse width.

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



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

(b) 16-bit capture/compare control register 0 (CRC0)

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

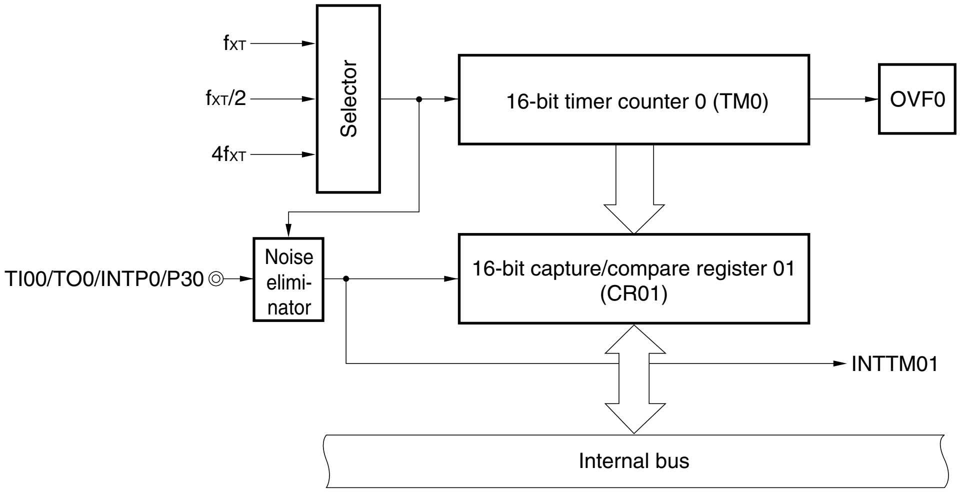
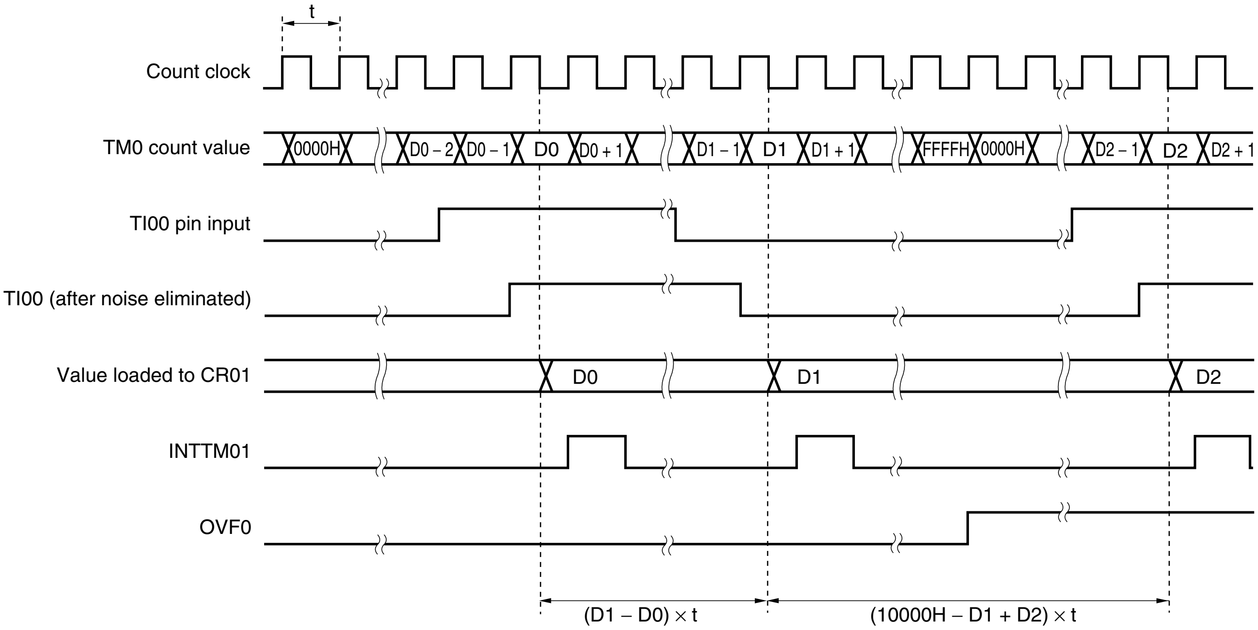


Figure 6-15. Timing of Pulse-Width Measurement Operation with Free-Running Counter and One Capture Register (When Both Edges Are Specified as Valid Edge)



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

When 16-bit timer counter 0 (TM0) is operated in free-running mode (see the register setting in **Figure 6-16**), it is possible to simultaneously measure the pulse widths of the signal input to the TI00/TO0/INTP0/P30 pin and TI01/INTP1/P31 pin.

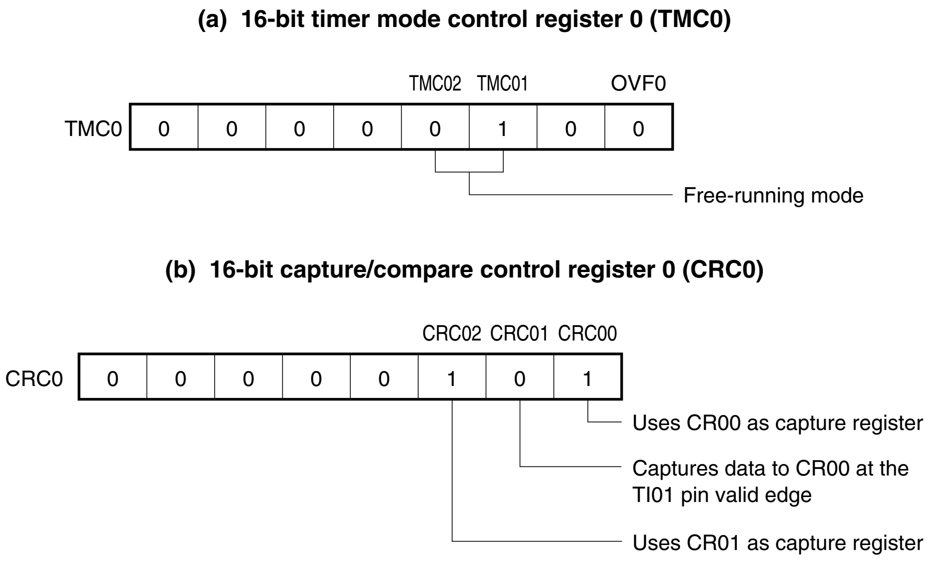
When the edge specified by bits 4 and 5 (ES00 and ES01) of 16-bit timer prescaler mode register 0 (PRM0) is input to the TI00/TO0/INTP0/P30 pin, the value of TM0 is taken into 16-bit capture/compare register 01 (CR01) and an external interrupt request signal (INTTM01) is set.

Also, when the edge specified by bits 6 and 7 (ES10 and ES11) of PRM0 is input to the TI01/P31/INTP1 pin, the value of TM0 is taken into 16-bit capture/compare register 00 (CR00) and an external interrupt request signal (INTTM00) is set.

The valid edges of the TI00 and TI01 pins are specified by bits 4 and 5 (ES00 and ES01) and bits 6 and 7 (ES10 and ES11) of PRM0. It is possible to select the rising edge, falling edge, or both edges as the valid edge.

With valid edge detection, sampling is performed at the count clock interval selected using PRM0, and a capture operation is only performed when a valid level is detected twice, thus eliminating noise with a short-pulse width.

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



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

TMC02
TMC01
OVF0
TMC0
0 0 0 0 0 1 0 0
Free-running mode

(b) 16-bit capture/compare control register 0 (CRC0)

CRC02
CRC01
CRC00
CRC0
0 0 0 0 0 1 0 1
Uses CR00 as capture register
Captures data to CR00 at the TI01 pin valid edge
Uses CR01 as capture register

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

  The capture register operation when a capture trigger is input is shown below.

Figure 6-17. CR01 Capture Operation When Rising Edge Is Specified as Valid Edge

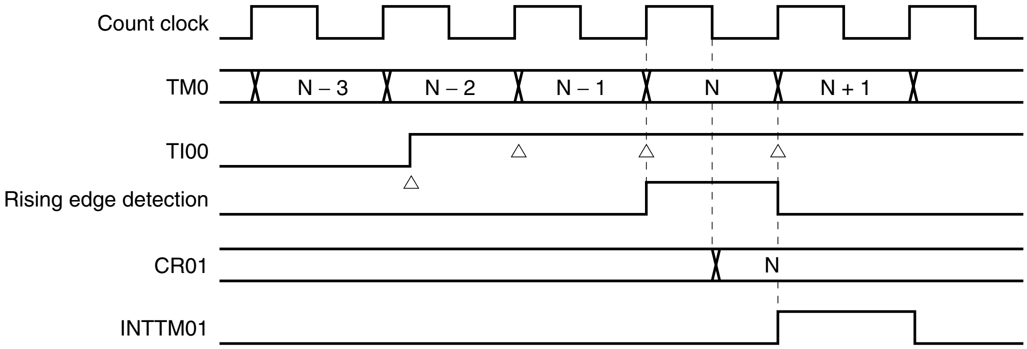
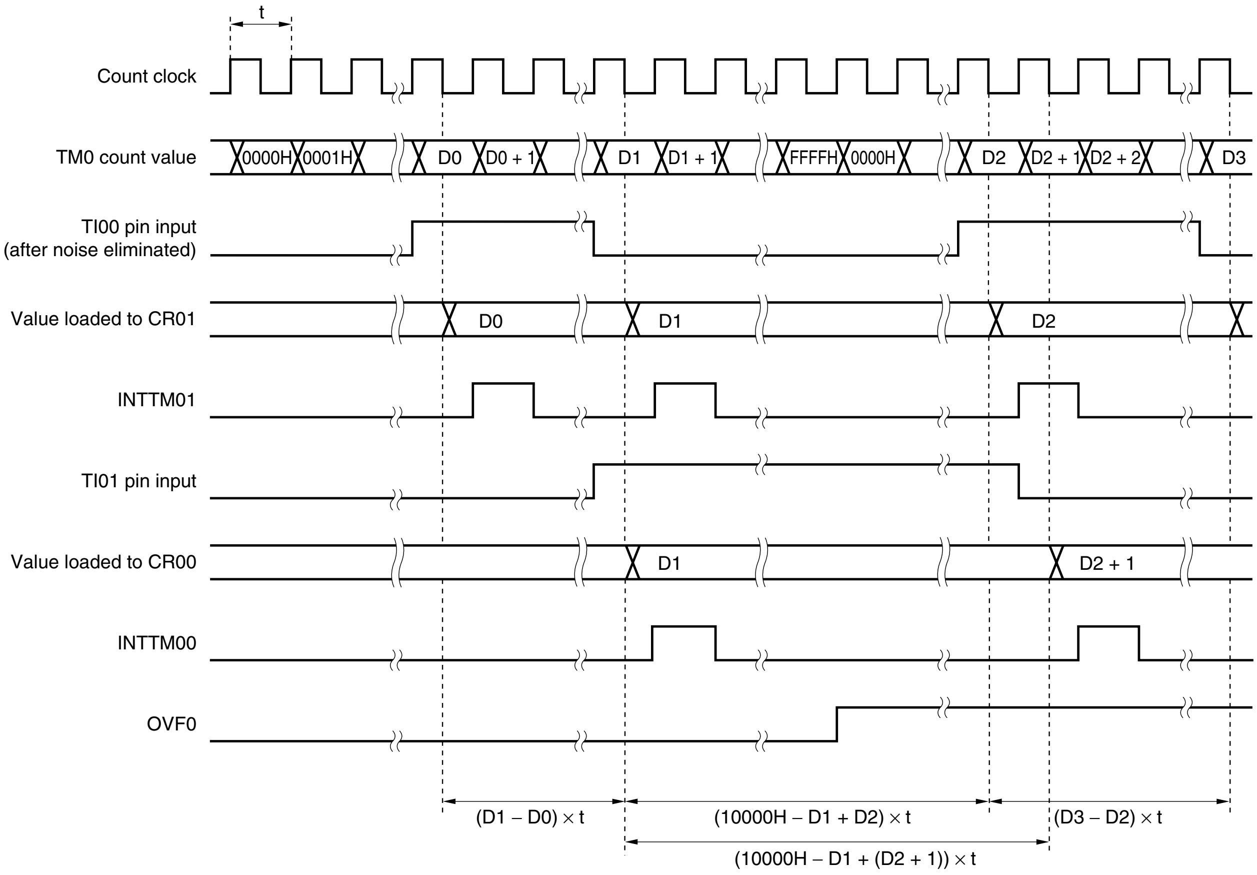


Figure 6-18. Timing of Pulse-Width Measurement Operation with Free-Running Counter (When Both Edges Are Specified as Valid Edge)



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

When 16-bit timer counter 0 (TM0) is operated in free-running mode (see the register setting in **Figure 6-19**), it is possible to measure the pulse width of the signal input to the TI00/TO0/INTP0/P30 pin.

When the edge specified by bits 4 and 5 (ES00 and ES01) of 16-bit timer prescaler mode register 0 (PRM0) is input to the TI00/TO0/INTP0/P30 pin, the value of TM0 is taken into 16-bit capture/compare register 01 (CR01) and an external interrupt request signal (INTTM01) is set.

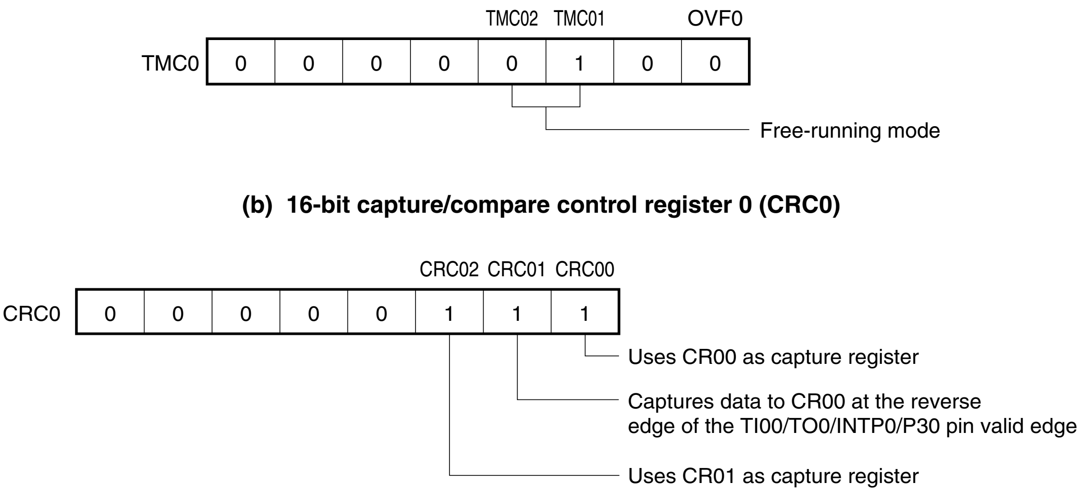
Also, on input of the inverse edge to that of the capture operation into CR01, the value of TM0 is taken into 16-bit capture/compare register 00 (CR00).

The valid edge of the TI00 pin is specified by bits 4 and 5 (ES00 and ES01) of PRM0, and it is possible to select the rising edge or falling edge.

With valid edge detection, sampling is performed at the count clock interval selected using PRM0, and a capture operation is only performed when a valid level is detected twice, thus eliminating noise with a short-pulse width.

Caution When both the rising and falling edges are selected as the valid edge of the TI00/TO0/INTP0/P30 pin, capture/compare register 00 (CR00) cannot perform the capture operation.

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



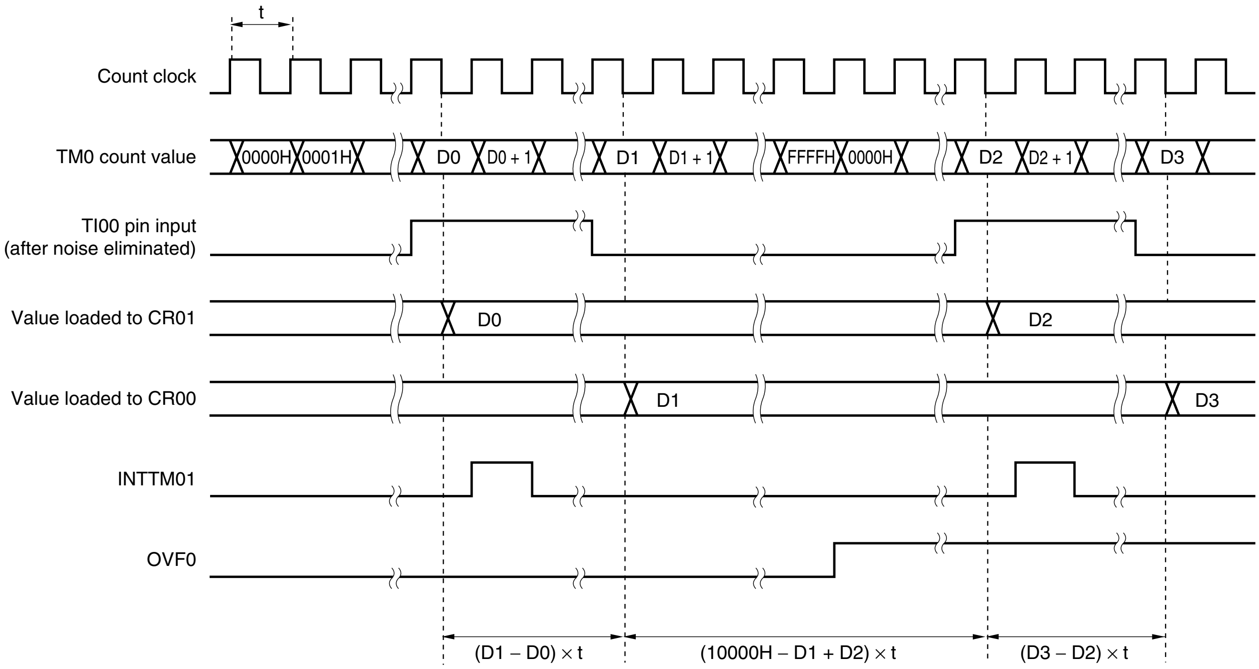
(a) 16-bit timer mode control register 0 (TMC0)

TMC02
TMC01
OVF0
TMC0
0
0
0
0
0
1
0
0
Free-running mode

(b) 16-bit capture/compare control register 0 (CRC0)

CRC02
CRC01
CRC00
CRC0
0
0
0
0
0
1
1
1
Uses CR00 as capture register
Captures data to CR00 at the reverse edge of the TI00/TO0/INTP0/P30 pin valid edge
Uses CR01 as capture register

Figure 6-20. Timing of Pulse-Width Measurement Operation with Free-Running Counter and Two Capture Registers (When Rising Edge Is Specified as Valid Edge)



(4) Pulse width measurement by means of restart

When input of a valid edge to the TI00/TO0/INTP0/P30 pin is detected, the count value of 16-bit timer counter 0 (TM0) is taken into 16-bit capture/compare register 01 (CR01), and then the pulse width of the signal input to the TI00/TO0/INTP0/P30 pin is measured by clearing TM0 and restarting the count (refer to **Figure 6-21**).

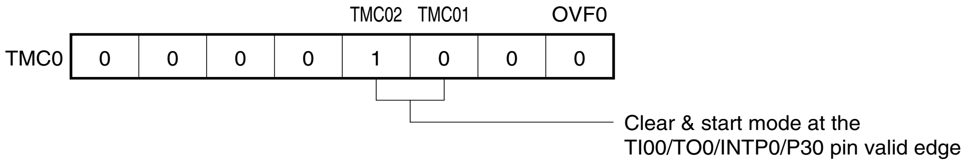
The valid edge of the TI00 pin is specified by bits 4 and 5 (ES00 and ES01) of 16-bit timer prescaler mode register 0 (PRM0), and it is possible to select the rising edge or falling edge.

With valid edge detection, sampling is performed at the count clock interval selected using PRM0, and a capture operation is only performed when a valid level is detected twice, thus eliminating noise with a short-pulse width.

Caution When both the rising and falling edges are selected as the valid edge of the TI00/TO0/INTP0/P30 pin, capture/compare register 00 (CR00) cannot perform the capture operation.

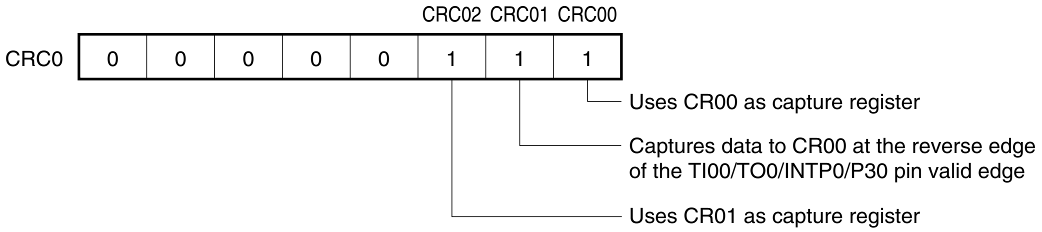
Figure 6-21. Control Register Settings for Pulse-Width Measurement by Means of Restart

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



Clear & start mode at the TI00/TO0/INTP0/P30 pin valid edge

(b) 16-bit capture/compare control register 0 (CRC0)



Uses CR00 as capture register

Captures data to CR00 at the reverse edge of the TI00/TO0/INTP0/P30 pin valid edge

Uses CR01 as capture register

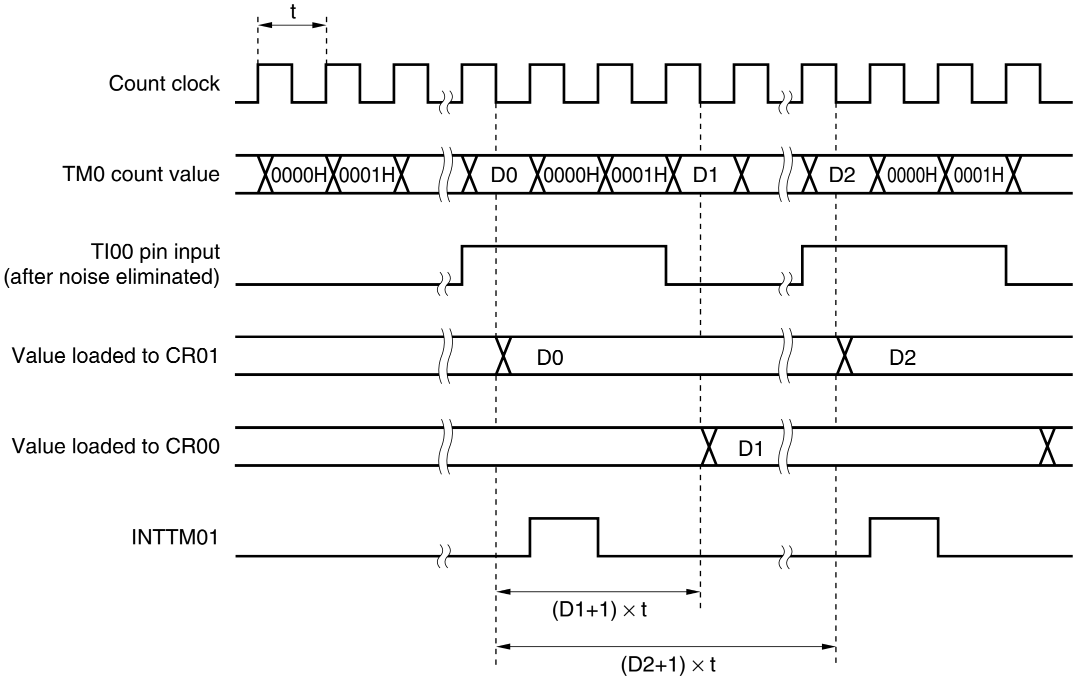


Figure 6-22. Timing of Pulse-Width Measurement Operation by Means of Restart (When Rising Edge Is Specified as Valid Edge)

6.4.4 External event counter operation

The external event counter counts the number of external clock pulses to be input to the TI00/TO0/INTP0/P30 pin by 16-bit timer counter 0 (TM0).

TM0 is incremented each time the specified valid edge is input to the TI00/TO0/INTP0/P30 pin when the valid edge of TI00 is selected as the count clock by 16-bit timer prescaler mode register 0 (PRM0).

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

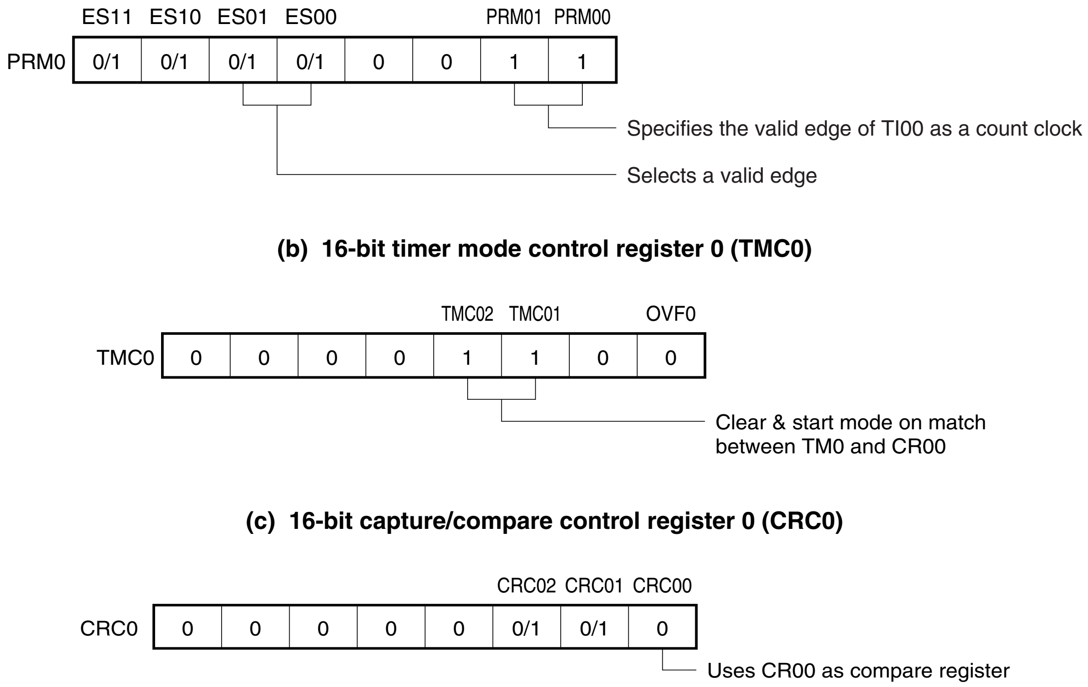
A value other than 0000H should be set for CR00 (1 pulse count operation is not possible).

The valid edge of the TI00 pin is specified by bits 4 and 5 (ES00 and ES01) of PRM0, and the rising edge, falling edge, or both edges can be selected.

With valid edge detection, sampling is performed at the main system clock interval (fx), and a capture operation is only performed when a valid level is detected twice, thus eliminating noise with a short-pulse width.

Caution The TI00/TO0/INTP0/P30 pin cannot be used as a timer output (TO0) when 16-bit timer/event counter 0 is used as an external event counter.

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



(a) 16-bit timer prescaler mode register 0 (PRM0)

	ES11	ES10	ES01	ES00			PRM01	PRM00
PRM0	0/1	0/1	0/1	0/1	0	0	1	1

Specifies the valid edge of TI00 as a count clock
Selects a valid edge

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

					TMC02	TMC01		OVF0
TMC0	0	0	0	0	1	1	0	0

Clear & start mode on match between TM0 and CR00

(c) 16-bit capture/compare control register 0 (CRC0)

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

Uses CR00 as compare register

Figure 6-24. Configuration Diagram for External Event Counter

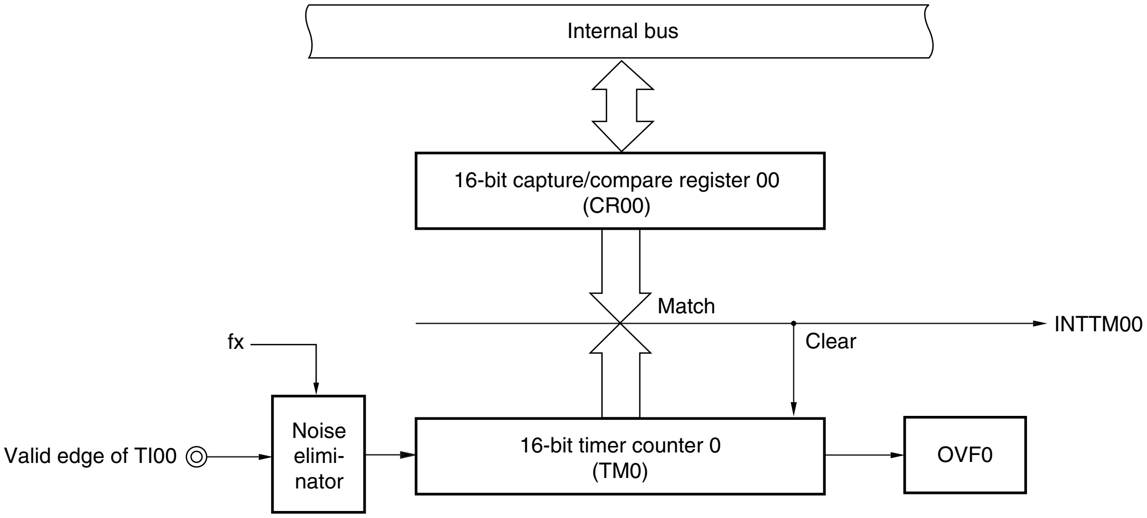
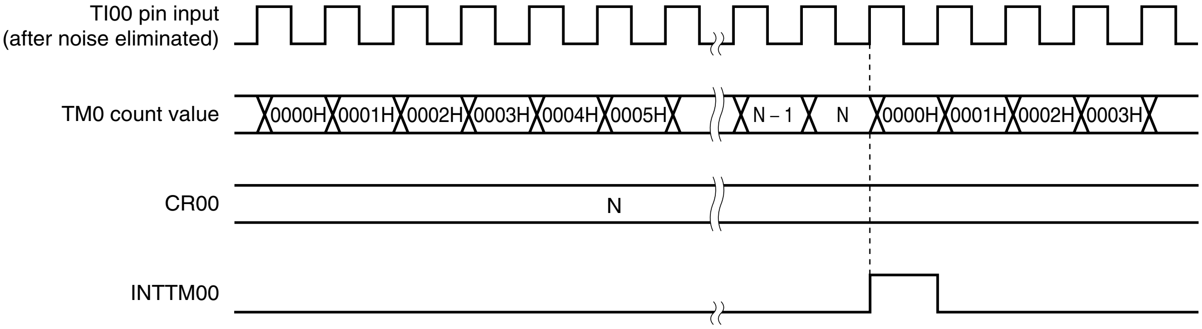


Figure 6-25. External Event Counter Operation Timing (When Rising Edge Is Specified as Valid Edge)



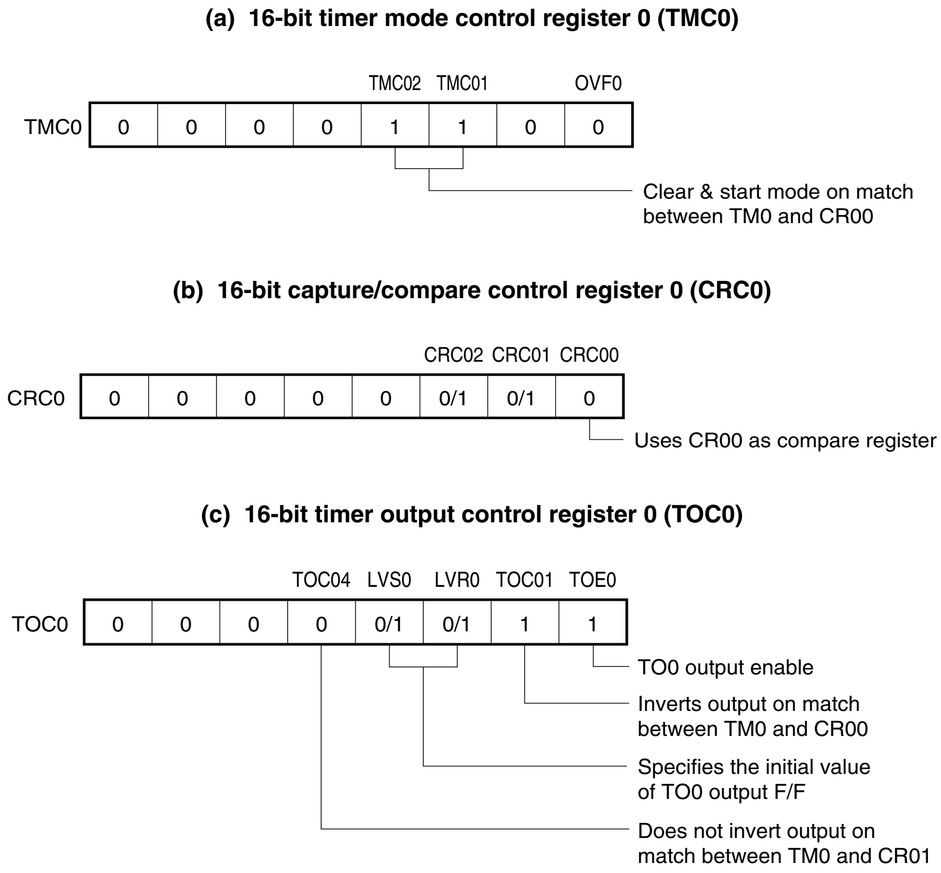
Caution When reading the external event counter count value, TM0 should be read.

6.4.5 Square wave output operation

This is an operation whereby a square wave of any selected frequency is output using the count value preset in 16-bit capture/compare register 00 (CR00).

The TO0/TI00/INTP0/P30 pin output status is reversed at the intervals of the count value preset in CR00 by setting bit 0 (TOE0) and bit 1 (TOC01) of 16-bit timer output control register 0 (TOC0) to 1. This enables a square wave of any selected frequency to be output.

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



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

TMC02
TMC01
OVF0
TMC0
0
0
0
0
1
1
0
0
Clear & start mode on match between TM0 and CR00

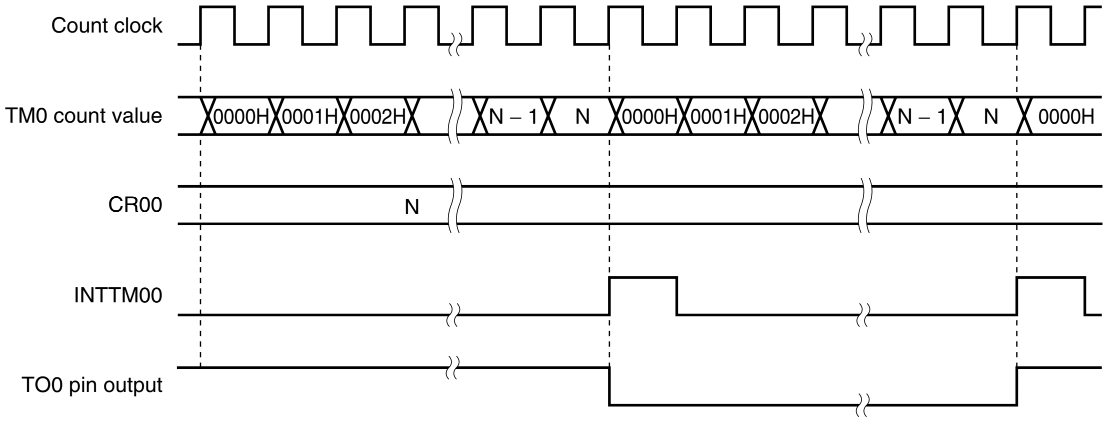
(b) 16-bit capture/compare control register 0 (CRC0)

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

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

TOC04
LVS0
LVR0
TOC01
TOE0
TOC0
0
0
0
0
0/1
0/1
1
1
TO0 output enable
Inverts output on match between TM0 and CR00
Specifies the initial value of TO0 output F/F
Does not invert output on match between TM0 and CR01

Figure 6-27. Square Wave Output Operation Timing

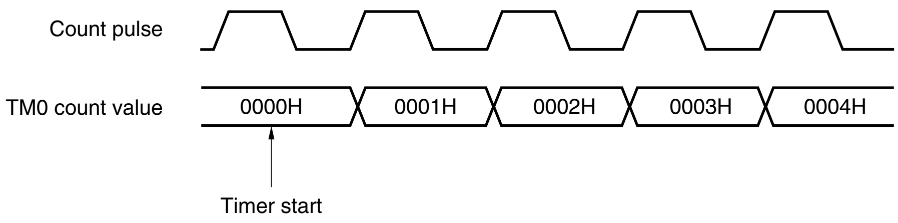


6.5 Cautions for 16-Bit Timer/Event Counter 0 Operation

(1) Timer start errors

An error of a maximum of one clock may occur in the time required for a match signal to be generated after timer start. This is because 16-bit timer counter 0 (TM0) is started asynchronously with the count pulse.

Figure 6-28. 16-Bit Timer Counter 0 Start Timing



(2) 16-bit compare register setting

Set a value other than 0000H to 16-bit capture/compare register 00 (CR00).

Accordingly, when using 16-bit timer/event counter 0 as an event counter, a one-pulse count operation cannot be performed.

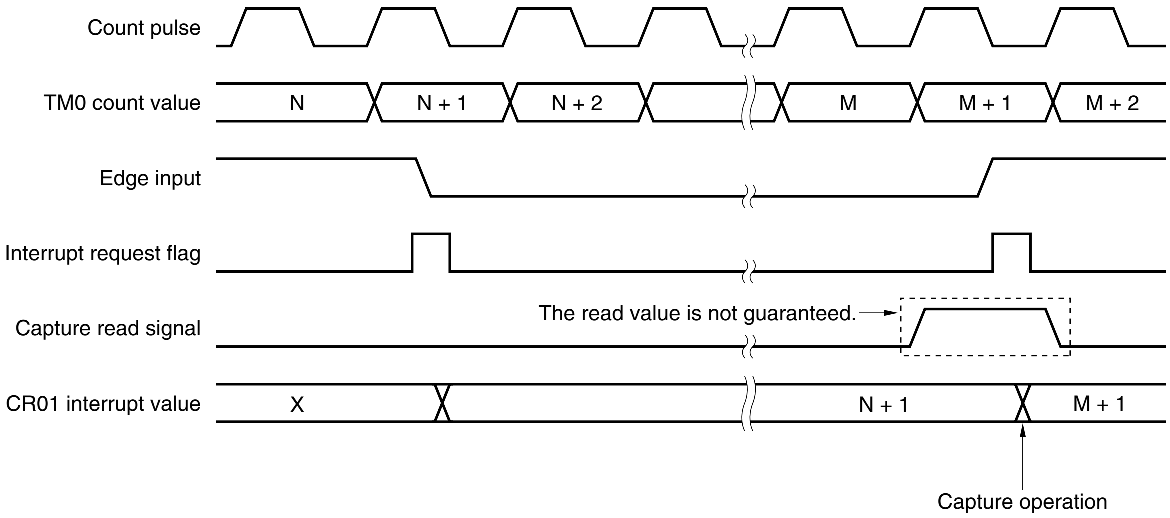
(3) Prohibition of compare register change during timer count operation

The value of the 16-bit capture/compare registers (CR00 and CR01) cannot be changed while the timer is operating. To change the value of these registers, be sure to stop the timer.

(4) Capture register data retention timing

If the capture trigger is input during 16-bit capture/compare registers (CR00 and CR01) read, CR00 and CR01 perform a capture operation normally, but the read value at this time is not guaranteed. The interrupt request flags (TMIF00 and TMIF01) are set upon detection of the valid edge.

Figure 6-29. Capture Register Data Retention Timing



(5) Valid edge setting

Set the valid edge of the TI00 and TI01 pins after the timer operation is stopped by clearing bits 2 and 3 (TMC01 and TMC02) of 16-bit timer mode control register 0 (TMC0) to 0, 0, respectively. The valid edges of the TI00 and TI01 pins are specified by setting bits 4 and 5 (ES00 and ES01) and bits 6 and 7 (ES10 and ES11) of 16-bit timer prescaler mode register 0 (PRM0).

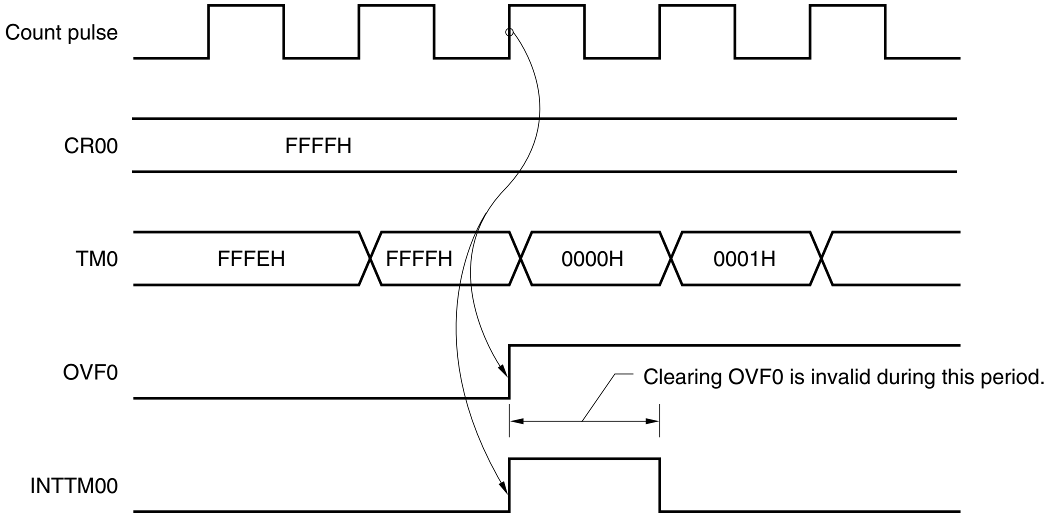
(6) Operation of OVF0 flag

(a) When the mode in which TM0 is cleared and started when its value matches the value of CR00 is selected, the OVF0 flag is set to 1 in the following case.

CR00 is set to FFFFH.
↓
When TM0 has counted up from FFFFH to 0000H.

(b) After TM0 overflows, it is reset and the clear instruction becomes invalid even if the OVF0 flag is cleared before the next count clock (before TM0 becomes 0001H).

Figure 6-30. Operation Timing of OVF0 Flag



(7) Conflicting operations

(a) Conflicting between the read period of a 16-bit capture/compare register (CR00 or CR01) and capture trigger input (CR00 or CR01 used as a capture register)
→ The capture trigger input has priority (CR00 or CR01 captured data is normal).
The data read from CR00 or CR01 is undefined.

(b) Conflicting between the capture trigger input of a 16-bit capture/compare register (CR00 or CR01) and the stop input (TMC01 = TMC02 = 0) of CR00 or CR01 and 16-bit timer counter 0 (TM0) (CR00 or CR01 used as a capture register)
→ The CR00 or CR01 capture data is undefined.

(8) Timer operation

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

(b) Regardless of the operation mode of the CPU, if the timer is stopped, the noise of the external input (TI00 or TI01) is not eliminated.

(9) Capture operation

(a) If TI00 is specified as the valid edge of the count clock, capture operation by the capture register specified as the trigger for TI00 is not possible.

(b) CR00 does not perform a capture operation when both the rising and falling edges are selected as the valid edge of TI00.

(c) To ensure the reliability of the capture operation, the capture trigger requires a pulse two times longer than the count clock selected by 16-bit timer prescaler mode register 0 (PRM0).

(d) The capture operation is performed at the fall of the count clock. An interrupt request input (INTTM00, INTTM01), however, is generated at the rise of the next count clock.

(10) Compare operation

(a) Do not change the value of the 16-bit capture/compare registers (CR00 and CR01) while the timer is operating. Timer operation must be stopped before changing the value of CR00 and CR01.

(b) CR00 and CR01 set in the compare mode cannot perform the capture operation even if the capture trigger is input.

(c) CR00 can be set to a value of 0001H to FFFFH and CR01 can be set to a value of 0000H to FFFFH. If CR01 is set to 0000H, an interrupt request (INTTM01) is generated after TM0 overflows or matches the value of CR00.

(11) Edge detection

(a) When the TI00 pin is high immediately after system reset, if the valid edge of TI00 pin is set to rising edge or both rising and falling edges and the operation of 16-bit timer counter 0 (TM0) is enabled, the rising edge will be detected immediately after these settings. Therefore, be careful in cases when the TI00 or TI01 pin is pulled up. No rising edge will be detected, however, when the operation of TM0 is enabled again after being stopped.

(b) The sampling clock used to eliminate noise differs when a TI00 pin valid edge is used as count clock and when it is used as a capture trigger. In the former case, the count clock is f_X, and in the latter case the count clock is selected by 16-bit timer prescaler mode register 0 (PRM0). When a valid level of the TI00 pin is detected twice by sampling with the above-mentioned sampling clock, the capture operation is started, therefore noise with short pulse widths can be eliminated.

(c) If the TI00 pin is selected as the count clock, it cannot be used while the main system clock is stopped because the main system clock is used as the sampling clock to eliminate noise.

(12) HALT operation

Because the ×4 multiplication clock circuit stops during HALT operation, $4f_{XT}$ or the valid edge of TI00 cannot be selected as the count clock. Use a count clock other than above when using 16-bit timer/event counter 0 in the HALT mode.

7.1 Functions of 8-Bit Timer/Event Counters 50 and 51

8-bit timer/event counters 50 and 51 have the following functions.

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

(1) 8-bit interval timer

Interrupts are generated at preset time intervals.

(2) External event counter

The number of pulses of signals input externally can be counted.

(3) Square-wave output

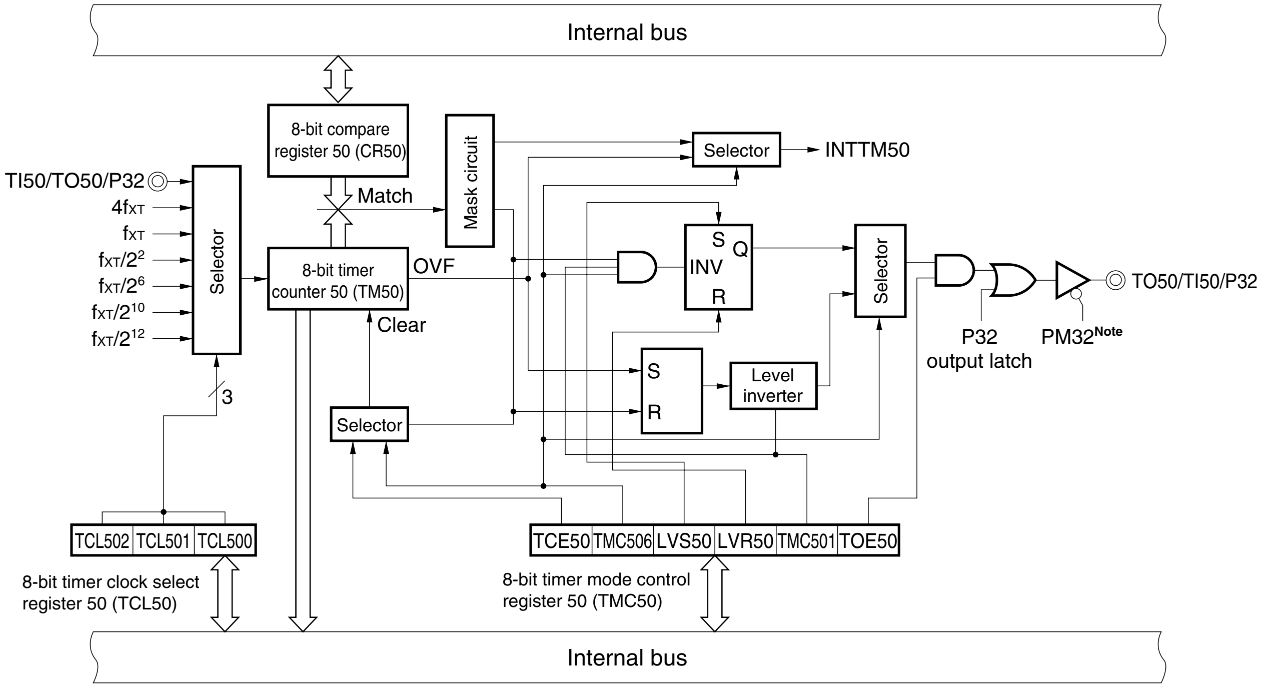
Square waves of any frequency can be output.

(4) PWM output

PWM output with 8-bit resolution is possible.

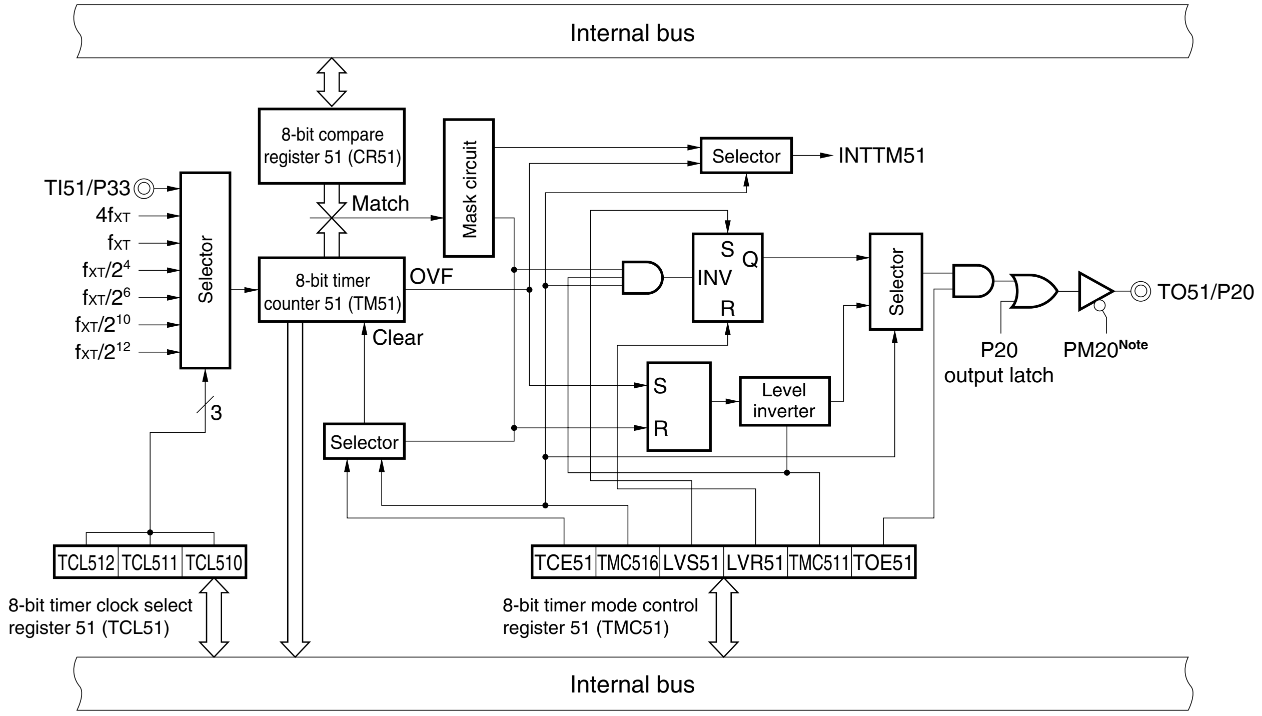
Figures 7-1 and 7-2 show the block diagrams of 8-bit timer/event counters 50 and 51.

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



Note Bit 2 of port mode register 3 (PM3)

Figure 7-2. Block Diagram of 8-Bit Timer/Event Counter 51



Note Bit 0 of port mode register 2 (PM2)

7.2 Configuration of 8-Bit Timer/Event Counters 50 and 51

8-bit timer/event counters 50 and 51 consist of the following hardware.

Table 7-1. Configuration of 8-Bit Timer/Event Counters 50 and 51

Item	Configuration
Timer register	8 bits × 2 (TM50 and TM51)
Register	Compare register: 8 bits × 2 (CR50 and CR51)
Timer output	2 (TO50 and TO51)
Control registers	8-bit timer clock select registers (TCL50 and TCL51) 8-bit timer mode control registers (TMC50 and TMC51) Port mode registers (PM2 and PM3)

(1) 8-bit timer counter 5n (TM5n)

TM5n is an 8-bit read-only register that counts the count pulses.

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

The count value is cleared to 00H in the following cases.

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

<2> If TCE5n is cleared

<3> If TM5n and CR5n match in the clear & start mode on a match between TM5n and CR5n

Remark n = 0, 1

(2) 8-bit compare register 5n (CR5n)

The value set in CR5n is constantly compared with 8-bit timer counter 5n (TM5n) count value, and an interrupt request (INTTM5n) is generated if they match (in a mode other than PWM mode).

CR5n can be set by an 8-bit memory manipulation instruction. It cannot be set by a 16-bit memory manipulation instruction. The value of CR5n can be set within the range of 00H to FFH, and can be overwritten during a count operation.

$\overline{\text{RESET}}$ input clears CR5n to 00H.

Caution When using the 8-bit timer in PWM mode, set TMC5n to PWM mode after setting the CR5n value.

Remark n = 0, 1

7.3 8-Bit Timer/Event Counters 50 and 51 Control Registers

The following three registers are used to control 8-bit timer/event counters 50 and 51.

- 8-bit timer clock select register 5n (TCL5n)
- 8-bit timer mode control register 5n (TMC5n)
- Port mode registers 2 and 3 (PM2 and PM3)

(1) 8-bit timer clock select register 5n (TCL5n)

This register is used to set the count clock of 8-bit timer counter 5n (TM5n).
TCL5n can be set by an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears TCL5n to 00H.

Remark n = 0, 1

Figure 7-3. Format of 8-Bit Timer Clock Select Register 50

Address: FF72H After reset: 00H R/W

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

TCL502	TCL501	TCL500	TM50 count clock selection
0	0	0	TI50 falling edge[Note]
0	0	1	TI50 rising edge[Note]
0	1	0	$4f_{XT}$ (131 kHz)
0	1	1	f_{XT} (32.768 kHz)
1	0	0	$f_{XT}/2^2$ (8.192 kHz)
1	0	1	$f_{XT}/2^6$ (512 Hz)
1	1	0	$f_{XT}/2^{10}$ (32 Hz)
1	1	1	$f_{XT}/2^{12}$ (8 Hz)

Note Timer output (PWM output) cannot be used when a clock is input externally.

Cautions 1. Timer operation must be stopped before rewriting TCL50 with values other than the identical data.

2. Be sure to clear bits 3 to 7 to 0.

3. Because the ×4 multiplication clock circuit stops in the HALT mode, $4f_{XT}$ cannot be selected as the count clock.

Remarks 1. f_{XT}: Subsystem clock oscillation frequency

2. Values in parentheses apply to the operation with f_{XT} = 32.768 kHz.

Figure 7-4. Format of 8-Bit Timer Clock Select Register 51

Address: FF7AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TCL51	0	0	0	0	0	TCL512	TCL511	TCL510

TCL512	TCL511	TCL510	TM51 count clock selection
0	0	0	TI51 falling edge
0	0	1	TI51 rising edge
0	1	0	4fXT (131 kHz)
0	1	1	fXT (32.768 kHz)
1	0	0	fXT/2^4 (2.048 kHz)
1	0	1	fXT/2^6 (512 Hz)
1	1	0	fXT/2^{10} (32 Hz)
1	1	1	fXT/2^{12} (8 Hz)

Cautions 1. Timer operation must be stopped before rewriting TCL51 with values other than the identical data.

2. Be sure to clear bits 3 to 7 to 0.

3. Because the ×4 multiplication clock circuit stops in the HALT mode, 4fXT cannot be selected as the count clock.

Remarks 1. fXT: Subsystem clock oscillation frequency

2. Values in parentheses apply to the operation with fXT = 32.768 kHz.

(2) 8-bit timer mode control register 5n (TMC5n)

This register is used to control the following five settings.

<1> Control of 8-bit timer counter 5n (TM5n) count operation
<2> Selection of 8-bit timer counter 5n (TM5n) operation mode
<3> Setting of timer output F/F (flip-flop) status
<4> Timer F/F control or selection of active level in PWM (free-running) mode
<5> Control of timer output

TMC5n can be set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears TMC5n to 00H.

Remark n = 0, 1

Figure 7-5. Format of 8-Bit Timer Mode Control Registers 50 and 51

Address: FF73H (TMC50), FF7BH (TMC51) After reset: 00H R/W

Symbol	<7>	6	5	4	<3>	<2>	1	<0>
TMC5n	TCE5n	TMC5n6	0	0	LVS5n	LVR5n	TMC5n1	TOE5n

TCE5n	TM5n count operation control
0	Count operation disabled after clearing counter to 0 (prescaler disabled)
1	Count operation started

TMC5n6	TM5n operation mode selection
0	Clear & start mode on match between TM5n and CR5n
1	PWM (free-running) mode

LVS5n	LVR5n	Timer output F/F status setting
0	0	No change
0	1	Timer output F/F is reset to 0.
1	0	Timer output F/F is set to 1.
1	1	Setting prohibited

TMC5n1	Other than PWM mode (TMC5n6 = 0)	PWM mode (TMC5n6 = 1)
	Timer F/F control	Active level selection
0	Inversion operation disabled	High active
1	Inversion operation enabled	Low active

TOE5n	Timer output control
0	Output disabled (port mode)
1	Output enabled

Cautions 1. Be sure to clear bits 4 and 5 to 0.

2. Bits 2 and 3 are write only.

3. Timer operation must be stopped before changing the operation mode by using bit 6.

4. Do not rewrite bits 0 and 1 simultaneously.

5. Before clearing TCE5n to 0, set the interrupt mask flag (TMMK5n) to 1. This is because the interrupt may occur after TCE5n has been cleared. Clear TCE5n with the following procedure.

```
TMMK5n = 1;   Sets mask.
TCE5n = 0;    Clears timer.
TMIF5n = 0;   Clears interrupt request flag.
TMMK5n = 0;   Clears mask.
       :
TCE5n = 1;    Starts timer.
       :
```

Remarks 1. By clearing TCE5n to 0 in PWM mode, the PWM output level becomes inactive.

2. 0 will be read out if LVS5n and LVR5n are read after setting data.

3. n = 0, 1

(3) Port mode registers 2 and 3 (PM2 and PM3)

These registers set the input/output mode of ports 2 and 3 in 1-bit units.

When the P20/TO51 pin is used as a timer output (TO51), clear PM20 and the output latch of P20 to 0.

When the P32/TO50/TI50/INTP2 pin is used as a timer output (TO50), clear PM32 and the output latch of P32 to 0. When it is used as a timer input (TI50), set PM32 to 1.

When the P33/TI51/INTP3 pin is used as a timer input (TI51), set PM33 to 1.

PM2 and PM3 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PM2 and PM3 to FFH.

Figure 7-6. Format of Port Mode Registers 2 and 3

Address: FF22H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM2	1	1	1	1	1	PM22	PM21	PM20

Address: FF23H After reset: FFH R/W

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

PMmn	Pmn pin input/output mode selection
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Remark mn = 20 to 22, 30 to 33

7.4 8-Bit Timer/Event Counters 50 and 51 Operations

7.4.1 Interval timer (8-bit) operation

The 8-bit timer can operate as an interval timer by setting 8-bit timer mode control register 5n (TMC5n) as shown in Figure 7-7. Interrupt requests are repeatedly generated using the count value preset in 8-bit compare register 5n (CR5n) as the interval.

When the count value of 8-bit timer counter 5n (TM5n) matches the value set in CR5n, counting continues, with the TM5n value cleared to 0, and the interrupt request signal (INTTM5n) is generated. The count clock of 8-bit timer counter 5n (TM5n) can be selected using 8-bit timer clock select register 5n (TCL5n).

The value of the compare register cannot be changed during count operation for the interval timer. Timer operation must be stopped before changing compare register value.

[Setting]

<1> Set the registers.

- TCL5n: Select count clock.
- CR5n: Compare value
- TMC5n: Clear and start mode by match of TM5n and CR5n
  (TMC5n = 0000×××0B × = don’t care)

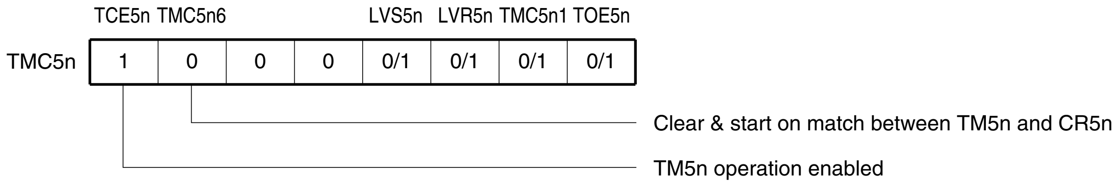
<2> After TCE5n = 1 is set, count operation starts.

<3> If the values of TM5n and CR5n match, INTTM5n is generated (TM5n is cleared to 00H).

<4> INTTM5n generates repeatedly at the same interval. Clear TCE5n to 0 to stop count operation.

Remark n = 0, 1

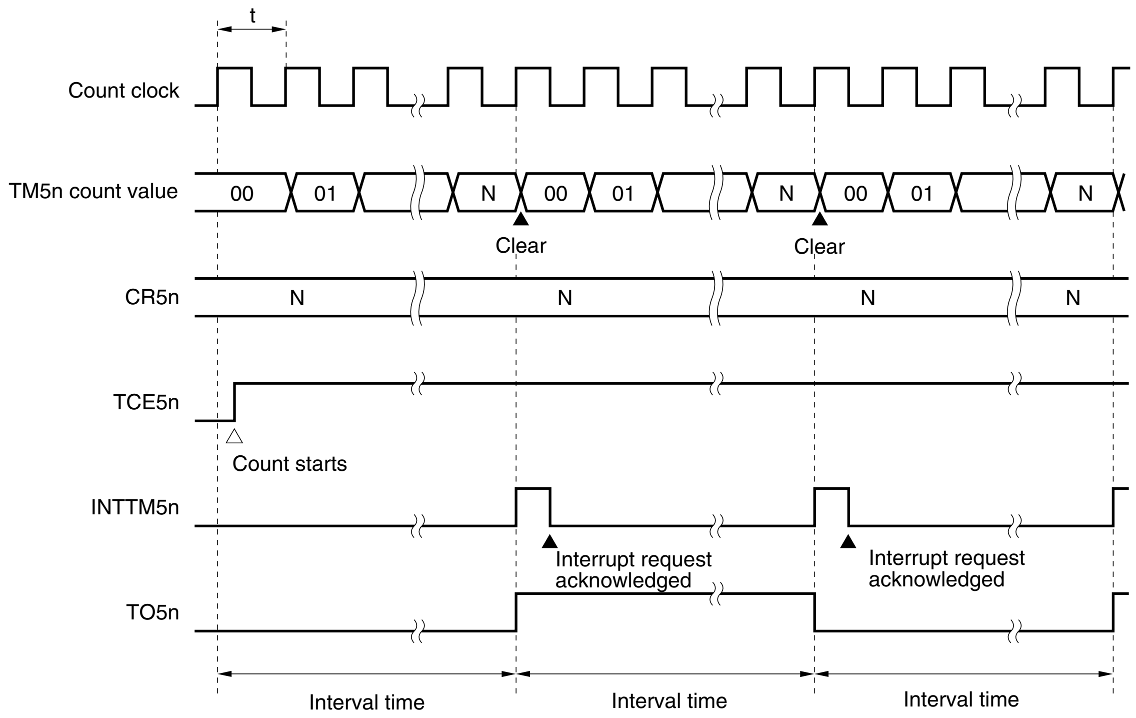
Figure 7-7. Description of 8-Bit Timer Mode Control Register Settings During Interval Timer Operation



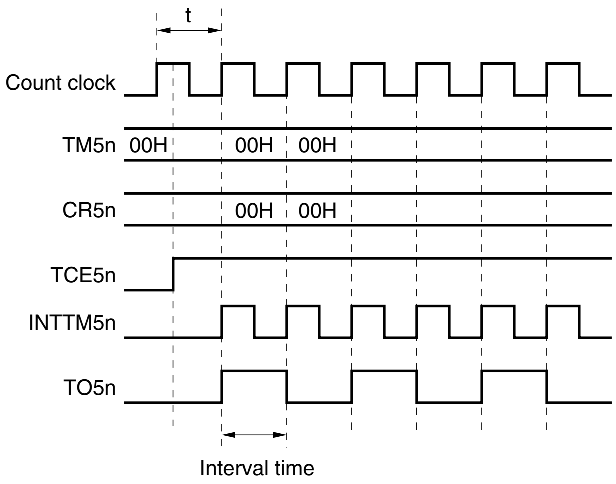
	TCE5n	TMC5n6			LVS5n	LVR5n	TMC5n1	TOE5n
TMC5n	1	0	0	0	0/1	0/1	0/1	0/1

Figure 7-8. Timing of Interval Timer Operation (1/2)

(a) Basic operation



(b) When CR5n = 00H



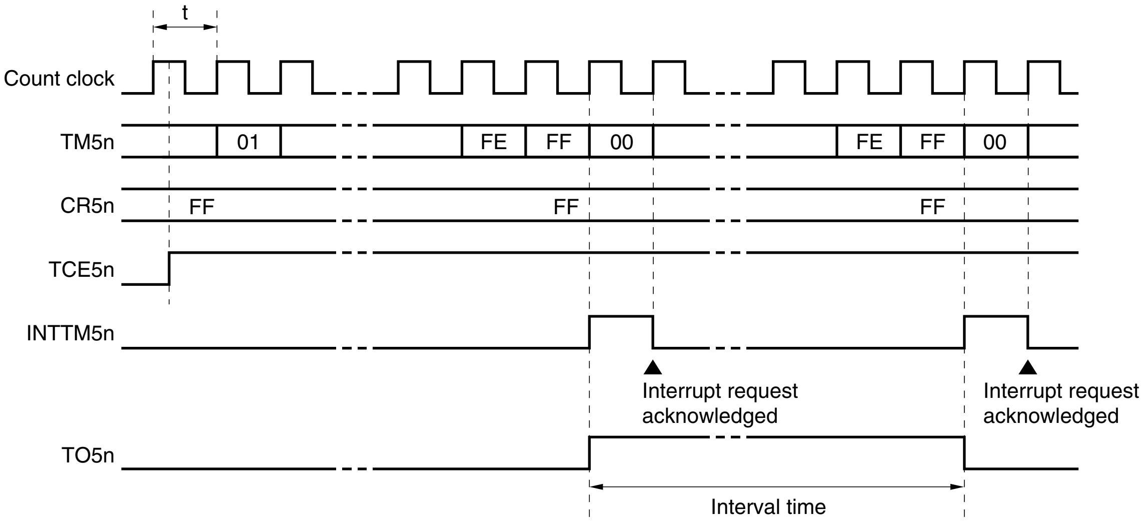
Remarks 1. Interval time = (N + 1) × t

N = 00H to FFH

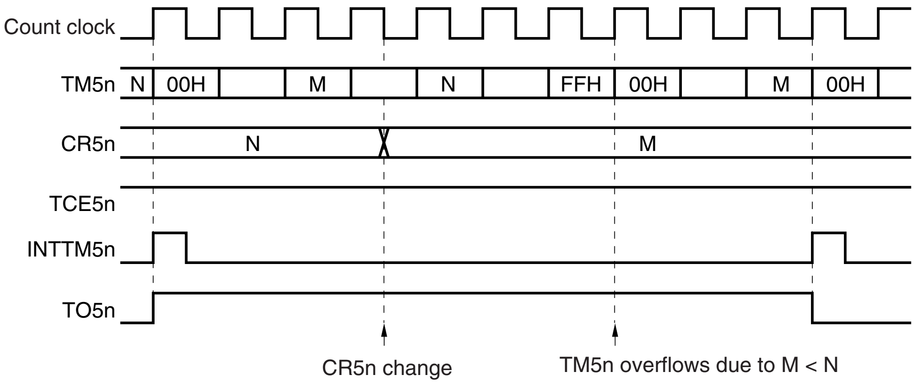
2. n = 0, 1

Figure 7-8. Timing of Interval Timer Operation (2/2)

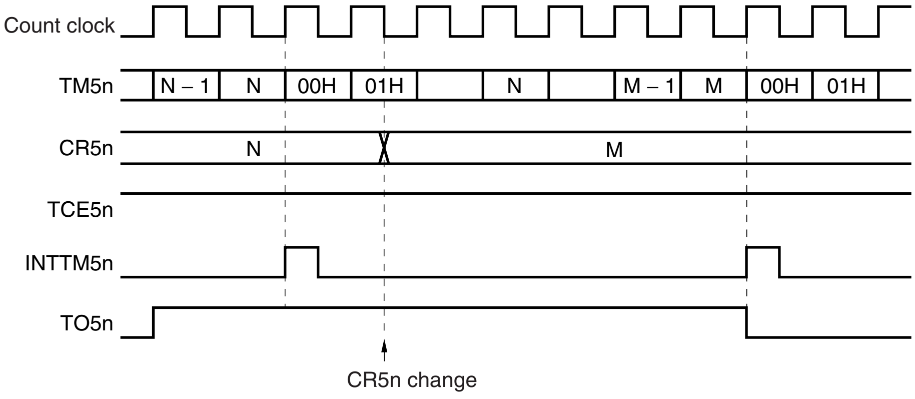
(c) When CR5n = FFH



(d) Operation according to change of CR5n (M < N)



(e) Operation according to change of CR5n (M > N)



Remark n = 0, 1

Table 7-2. Interval Time of 8-Bit Timer/Event Counter 50

Minimum Interval Time	Maximum Interval Time	Resolution
$1/4f_{XT}$ (7.62 μs)	$2^6 \times 1/f_{XT}$ (1.95 ms)	$1/4f_{XT}$ (7.62 μs)
$1/f_{XT}$ (30.5 μs)	$2^8 \times 1/f_{XT}$ (7.81 ms)	$1/f_{XT}$ (30.5 μs)
$2^2 \times 1/f_{XT}$ (122 μs)	$2^{10} \times 1/f_{XT}$ (31.2 ms)	$2^2 \times 1/f_{XT}$ (122 μs)
$2^6 \times 1/f_{XT}$ (1.95 ms)	$2^{14} \times 1/f_{XT}$ (500 ms)	$2^6 \times 1/f_{XT}$ (1.95 ms)
$2^{10} \times 1/f_{XT}$ (31.2 ms)	$2^{18} \times 1/f_{XT}$ (8 s)	$2^{10} \times 1/f_{XT}$ (31.2 ms)
$2^{12} \times 1/f_{XT}$ (125 ms)	$2^{20} \times 1/f_{XT}$ (32 s)	$2^{12} \times 1/f_{XT}$ (125 ms)

Remarks 1. f_{XT}: Subsystem clock oscillation frequency

2. Values in parentheses apply to the operation at f_{XT} = 32.768 kHz.

Table 7-3. Interval Time of 8-Bit Timer/Event Counter 51

Minimum Interval Time	Maximum Interval Time	Resolution
$1/4f_{XT}$ (7.62 μs)	$2^6 \times 1/f_{XT}$ (1.95 ms)	$1/4f_{XT}$ (7.62 μs)
$1/f_{XT}$ (30.5 μs)	$2^8 \times 1/f_{XT}$ (7.81 ms)	$1/f_{XT}$ (30.5 μs)
$2^4 \times 1/f_{XT}$ (488 μs)	$2^{12} \times 1/f_{XT}$ (125 ms)	$2^4 \times 1/f_{XT}$ (488 μs)
$2^6 \times 1/f_{XT}$ (1.95 ms)	$2^{14} \times 1/f_{XT}$ (500 ms)	$2^6 \times 1/f_{XT}$ (1.95 ms)
$2^{10} \times 1/f_{XT}$ (31.2 ms)	$2^{18} \times 1/f_{XT}$ (8 s)	$2^{10} \times 1/f_{XT}$ (31.2 ms)
$2^{12} \times 1/f_{XT}$ (125 ms)	$2^{20} \times 1/f_{XT}$ (32 s)	$2^{12} \times 1/f_{XT}$ (125 ms)

Remarks 1. f_{XT}: Subsystem clock oscillation frequency

2. Values in parentheses apply to the operation at f_{XT} = 32.768 kHz.

7.4.2 External event counter operation

The external event counter counts the number of external clock pulses to be input to the TI5n by 8-bit timer counter 5n (TM5n).

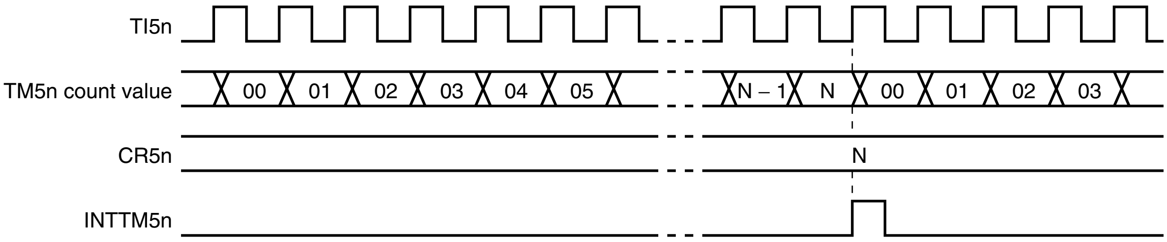
TM5n is incremented each time the valid edge specified with 8-bit timer clock select register 5n (TCL5n) is input. Either the rising or falling edge can be selected.

When the TM5n counted values match the values of 8-bit compare register 5n (CR5n), TM5n is cleared to 0 and the interrupt request signal (INTTM5n) is generated.

Whenever the TM5n counted value matches the value of CR5n, INTTM5n is generated.

Remark n = 0, 1

Figure 7-9. External Event Counter Operation Timing (with Rising Edge Specified)



n = 0, 1

7.4.3 Square-wave output (8-bit resolution) operation

The 8-bit timer operates as a square-wave output of any frequency using a count value preset in 8-bit compare register 5n (CR5n) as the interval.

When bit 1 (TMC5n1) and bit 0 (TOE5n) of 8-bit timer mode control register 5n (TMC5n) are set to 1, the output status of the TO5n pin is inverted at the interval time specified by the count value preset in CR5n. In this way, a square-wave of any frequency (duty factor = 50%) can be output.

[Setting]

<1> Set each register.

- Clear port latch and port mode register to 0.
- TCL5n: Select count clock.
- CR5n: Compare value
- TMC5n: Clear and start mode by match of TM5n and CR5n

LVS5n	LVR5n	Timer Output F/F Status Setting
1	0	High-level output
0	1	Low-level output

Timer output F/F reverse enable
Timer output enable → TOE5n = 1

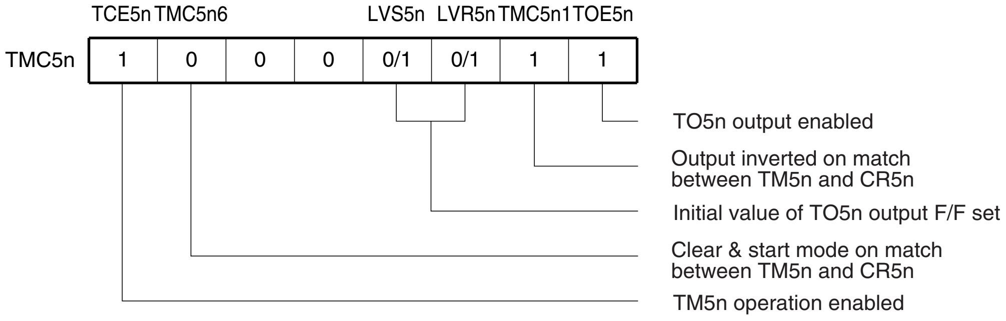
<2> After TCE5n = 1 is set, count operation starts.

<3> Timer output F/F is reversed by match of TM5n and CR5n.
After INTTM5n is generated, TM5n is cleared to 00H.

<4> Timer output F/F is reversed at the same interval and square wave is output from TO5n.

Remark n = 0, 1

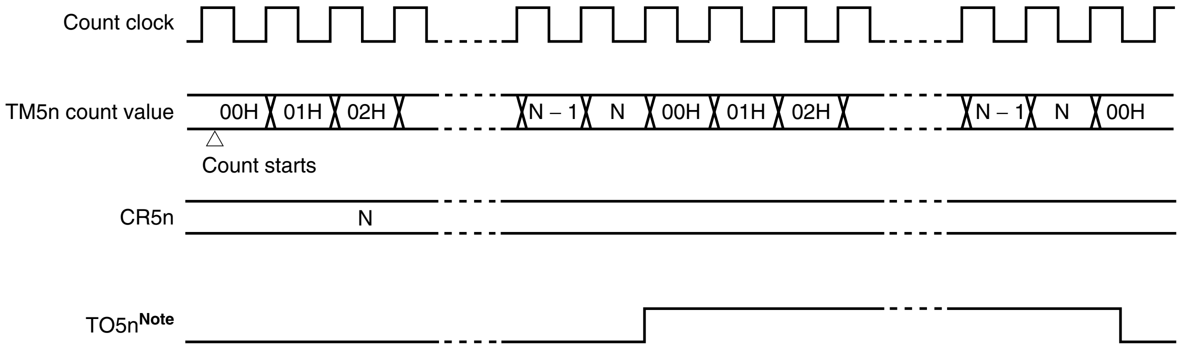
Figure 7-10. Description of 8-Bit Timer Mode Control Register Settings During Square-Wave Output Operation


TCE5n
TMC5n6
LVS5n
LVR5n
TMC5n1
TOE5n
TMC5n
1
0
0
0
0/1
0/1
1
1
TO5n output enabled
Output inverted on match between TM5n and CR5n
Initial value of TO5n output F/F set
Clear & start mode on match between TM5n and CR5n
TM5n operation enabled

Caution When the TO50/P32/TI50/INTP2 pin is used as a timer output, clear port mode register 3 (PM3) and the output latch to 0.
When the TO51/P20 pin is used as a timer output, clear port mode register 2 (PM2) and the output latch to 0.

Remark n = 0, 1

Figure 7-11. Timing of Square-Wave Output Operation



Note The initial value of the TO5n output can be specified by setting bits 2 and 3 (LVR5n, LVS5n) of 8-bit timer mode control register 5n (TMC5n).

Remark n = 0, 1

Table 7-4. Square-Wave Output Range for 8-Bit Timer/Event Counter 50

Minimum Pulse Width	Maximum Pulse Width	Resolution
1/4fXT (7.62 μs)	2^6 × 1/fXT (1.95 ms)	1/4fXT (7.62 μs)
1/fXT (30.5 μs)	2^8 × 1/fXT (7.81 ms)	1/fXT (30.5 μs)
2^2 × 1/fXT (122 μs)	2^{10} × 1/fXT (31.2 ms)	2^2 × 1/fXT (122 μs)
2^6 × 1/fXT (1.95 ms)	2^{14} × 1/fXT (500 ms)	2^6 × 1/fXT (1.95 ms)
2^{10} × 1/fXT (31.2 ms)	2^{18} × 1/fXT (8 s)	2^{10} × 1/fXT (31.2 ms)
2^{12} × 1/fXT (125 ms)	2^{20} × 1/fXT (32 s)	2^{12} × 1/fXT (125 ms)

Remarks 1. fXT: Subsystem clock oscillation frequency

2. Values in parentheses apply to the operation at fXT = 32.768 kHz.

Table 7-5. Square-Wave Output Range for 8-Bit Timer/Event Counter 51

Minimum Pulse Width	Maximum Pulse Width	Resolution
1/4fXT (7.62 μs)	2^6 × 1/fXT (1.95 ms)	1/4fXT (7.62 μs)
1/fXT (30.5 μs)	2^8 × 1/fXT (7.81 ms)	1/fXT (30.5 μs)
2^4 × 1/fXT (488 μs)	2^{12} × 1/fXT (125 ms)	2^4 × 1/fXT (488 μs)
2^6 × 1/fXT (1.95 ms)	2^{14} × 1/fXT (500 ms)	2^6 × 1/fXT (1.95 ms)
2^{10} × 1/fXT (31.2 ms)	2^{18} × 1/fXT (8 s)	2^{10} × 1/fXT (31.2 ms)
2^{12} × 1/fXT (125 ms)	2^{20} × 1/fXT (32 s)	2^{12} × 1/fXT (125 ms)

Remarks 1. fXT: Subsystem clock oscillation frequency

2. Values in parentheses apply to the operation at fXT = 32.768 kHz.

7.4.4 8-bit PWM output operations

The PWM output operation can be performed by setting 8-bit timer mode control register 5n (TMC5n) as shown in Figure 7-12.

A pulse with a duty factor determined by the value preset in 8-bit compare register 5n (CR5n) is output from the TO5n pin.

The active level of the PWM pulse is set with bit 1 (TMC5n1) of TMC5n.

This PWM pulse has 8-bit resolution. This pulse can be converted to analog voltage by integrating an external low pass filter (LPF).

PWM output can be enabled or disabled using bit 0 (TOE5n) of TMC5n.

[Setting]

<1> With timer 50, set P32 in the output mode (PM32 = 0), and clear the output latch of P32 to 0.
With timer 51, set P20 in the output mode (PM20 = 0), and clear the output latch of P20 to 0.

<2> Set active level width with 8-bit compare register 5n (CR5n).

<3> Select count clock with 8-bit timer clock select register 5n (TCL5n).

<4> Set active level with bit 1 (TMC5n1) of TMC5n.

<5> Count operation starts when bit 7 (TCE5n) of TMC5n is set to 1.
Clear TCE5n to 0 to stop count operation.

[PWM output operation]

<1> PWM output (output from TO5n) outputs inactive level after count operation starts until overflow is generated.

<2> When overflow is generated, the active level set in <4> of setting is output.
The active level is output until CR5n matches the count value of 8-bit timer counter 5n (TM5n).

<3> After the CR5n matches the count value, PWM output outputs the inactive level again until overflow is generated.

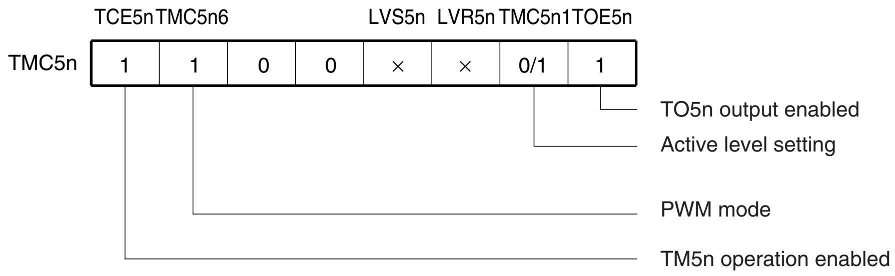
<4> Operations <2> and <3> are repeated until the count operation stops.

<5> When the count operation is stopped with TCE5n = 0, PWM output comes to inactive level.

Caution CR5n can only be overwritten once per cycle in PWM mode.

Remark n = 0, 1

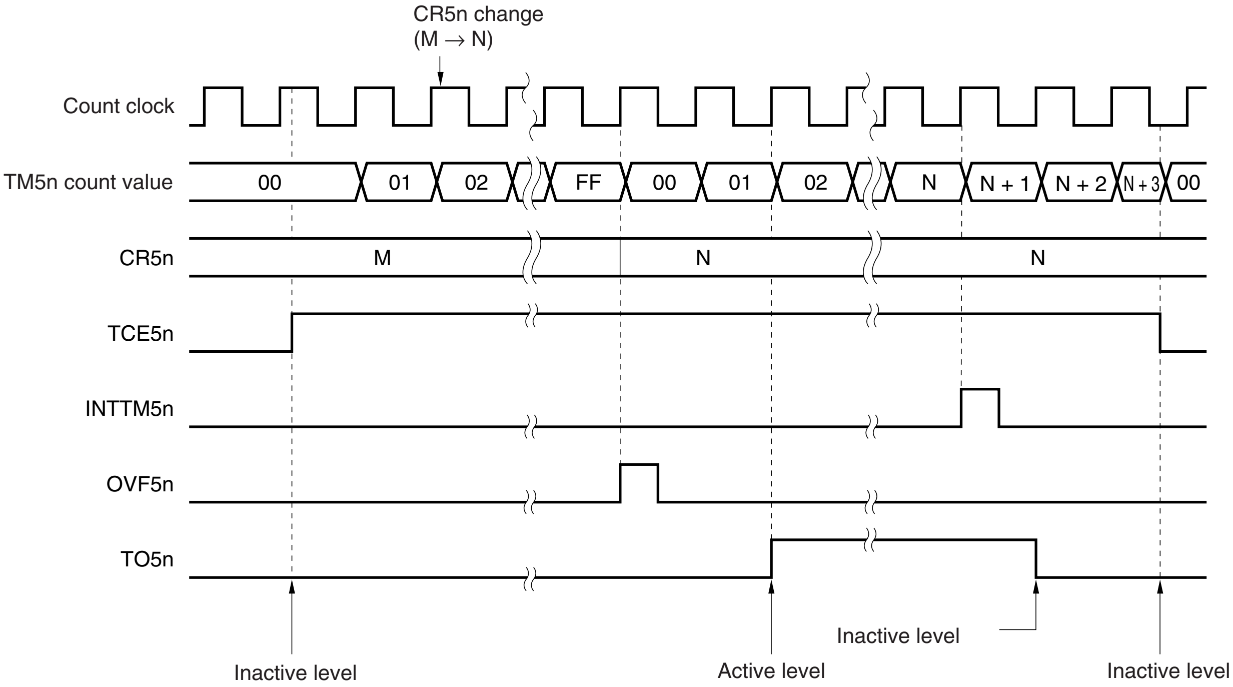
Figure 7-12. Description of 8-Bit Timer Control Register 5n Settings During PWM Output Operation



Remarks 1. ×: don't care

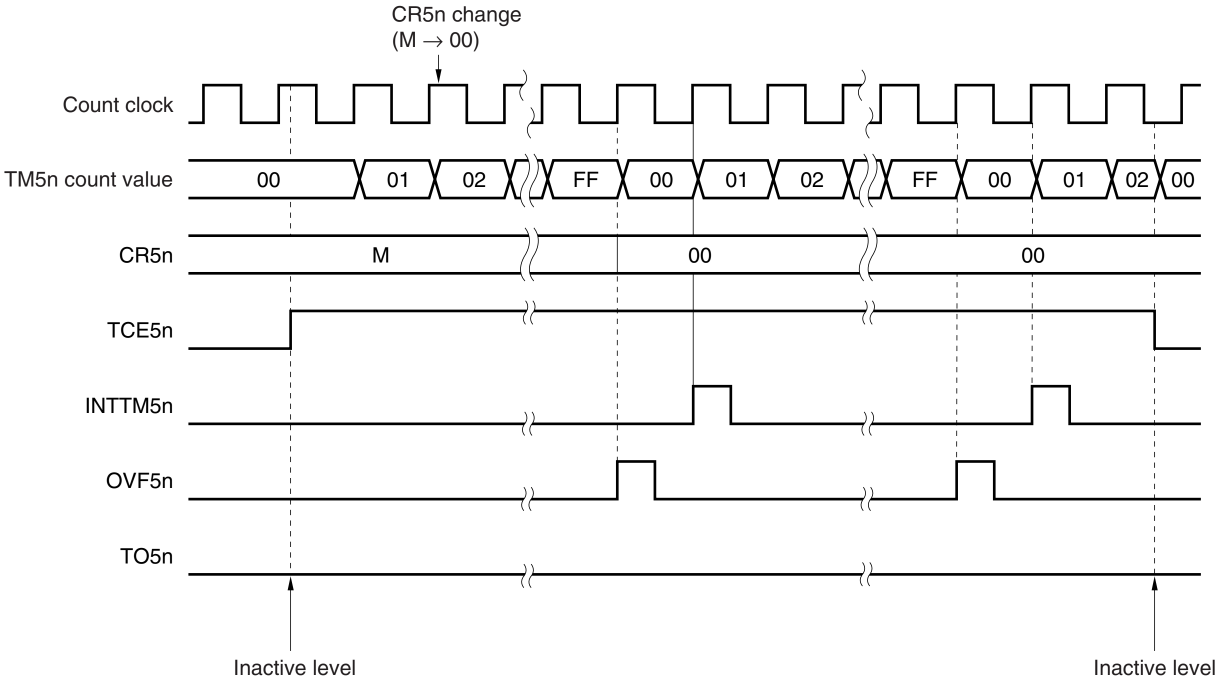
2. n = 0, 1

Figure 7-13. Timing of PWM Output Operation (When Active Level = H)



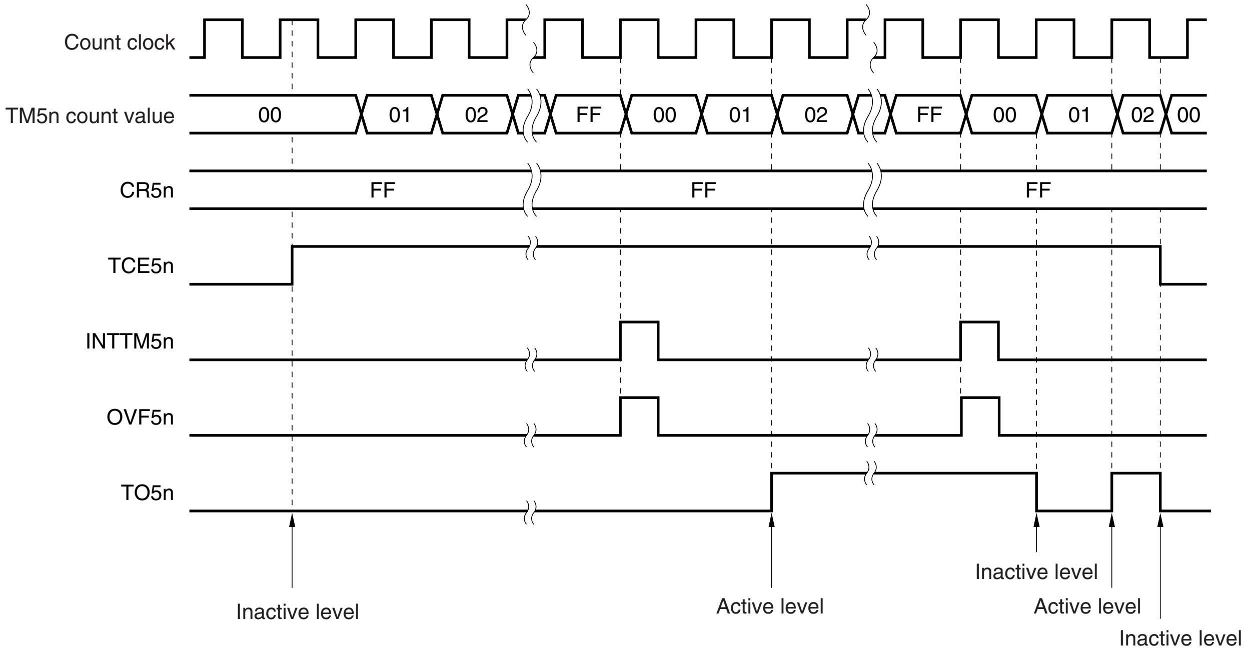
Remark n = 0, 1

Figure 7-14. Timing of PWM Output Operation (When CR5n = 00H, Active Level = H)



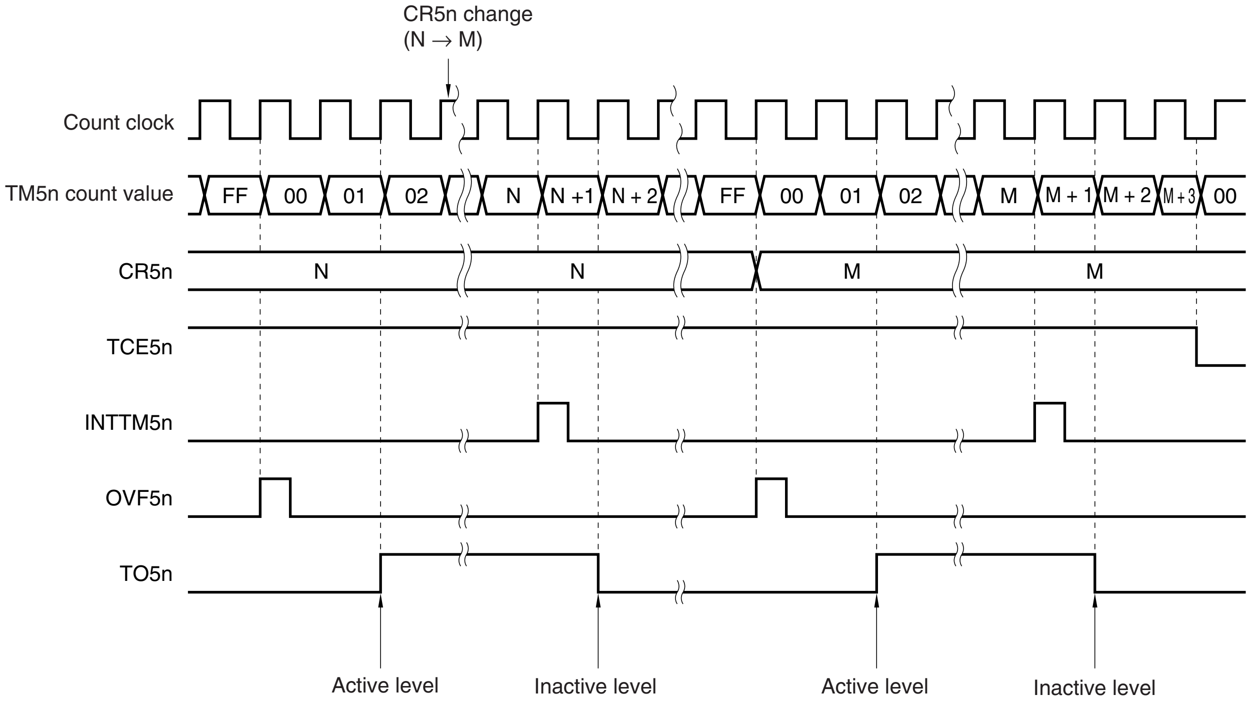
Remark n = 0, 1

Figure 7-15. Timing of PWM Output Operation (When CR5n = FFH, Active Level = H)



Remark n = 0, 1

Figure 7-16. Timing of PWM Output Operation (When CR5n Is Changed, Active Level = H)



Caution When CR5n is changed during TM5n operation, the changed value is not reflected until TM5n overflows.

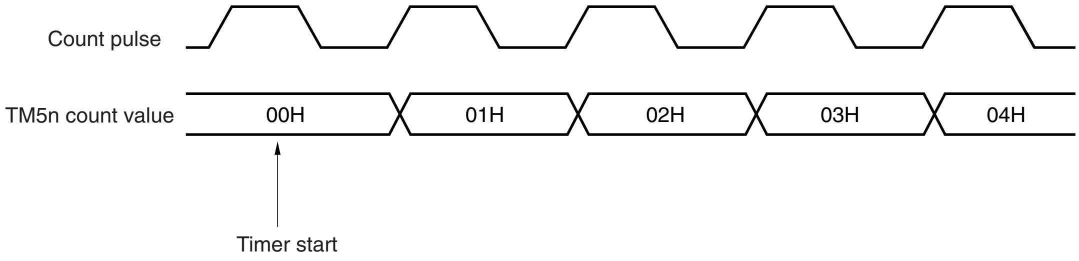
Remark n = 0, 1

7.5 Cautions for 8-Bit Timer/Event Counters 50 and 51 Operation

(1) Timer start errors

An error of a maximum of one clock may occur in the time required for a match signal to be generated after timer start. This is because 8-bit timer counter 5n (TM5n) is started asynchronously with the count pulse.

Figure 7-17. 8-Bit Timer Counter 5n (TM5n) Start Timing



Remark n = 0, 1

(2) Prohibition of compare register change during timer count operation in a mode other than PWM mode

The value of 8-bit compare register 5n (CR5n) cannot be changed in a mode other than PWM mode while the timer is operating. Timer operation must be stopped before changing the value of this register.

(3) HALT operation

Because the ×4 multiplication clock circuit stops during HALT operation, $4f_{XT}$ cannot be selected as the count clock. Use a count clock other than $4f_{XT}$ when using 8-bit timer/event counters 50 and 51 in the HALT mode.

(4) TI50 and TO50 pins

Because TI50 and TO50 are assigned to the same pin, both the functions cannot be used at the same time.

CHAPTER 8 WATCHDOG TIMER

8.1 Watchdog Timer Functions

The watchdog timer has the following functions.

- Watchdog timer
- Interval timer

Caution Select the watchdog timer mode or interval timer mode by using the watchdog timer mode register (WDTM) (the watchdog timer and interval timer cannot be used at the same time).

(1) Watchdog timer

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

Table 8-1. Watchdog Timer Runaway Detection Time

Runaway Detection Time	At f_{XT} = 32.768 kHz Operation	At f_{XT} = 38.4 kHz Operation
$2^{12} \times 1/f_{XT}$	125 ms	106.7 ms
$2^{13} \times 1/f_{XT}$	250 ms	213.3 ms
$2^{14} \times 1/f_{XT}$	500 ms	426.7 ms
$2^{15} \times 1/f_{XT}$	1 s	853.3 ms
$2^{16} \times 1/f_{XT}$	2 s	1.71 s
$2^{17} \times 1/f_{XT}$	4 s	3.41 s
$2^{18} \times 1/f_{XT}$	8 s	6.83 s
$2^{20} \times 1/f_{XT}$	32 s	27.3 s

f_{XT}: Subsystem clock oscillation frequency

(2) Interval timer

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

Table 8-2. Interval Time

Interval	At f_{XT} = 32.768 kHz Operation	At f_{XT} = 38.4 kHz Operation
$2^{12} \times 1/f_{XT}$	125 ms	106.7 ms
$2^{13} \times 1/f_{XT}$	250 ms	213.3 ms
$2^{14} \times 1/f_{XT}$	500 ms	426.7 ms
$2^{15} \times 1/f_{XT}$	1 s	853.3 ms
$2^{16} \times 1/f_{XT}$	2 s	1.71 s
$2^{17} \times 1/f_{XT}$	4 s	3.41 s
$2^{18} \times 1/f_{XT}$	8 s	6.83 s
$2^{20} \times 1/f_{XT}$	32 s	27.3 s

f_{XT}: Subsystem clock oscillation frequency

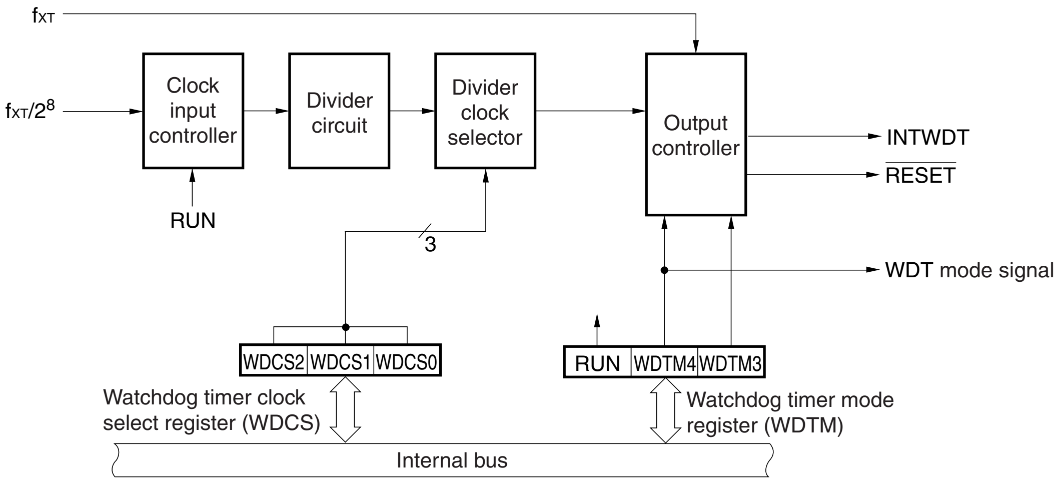
8.2 Watchdog Timer Configuration

The watchdog timer includes the following hardware.

Table 8-3. Configuration of Watchdog Timer

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

Figure 8-1. Block Diagram of Watchdog Timer



8.3 Watchdog Timer Control Registers

The watchdog timer is controlled by the following two registers.

- 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 can be set by an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears WDCS to 00H.

Figure 8-2. 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	WDCS0	FF4AH	00H	R/W

WDCS2	WDCS1	WDCS0	Watchdog timer/interval timer overflow time		
				At f_{XT} = 32.768 kHz operation	At f_{XT} = 38.4 kHz operation
0	0	0	$2^{12} \times 1/f_{XT}$	125 ms	106.7 ms
0	0	1	$2^{13} \times 1/f_{XT}$	250 ms	213.3 ms
0	1	0	$2^{14} \times 1/f_{XT}$	500 ms	426.7 ms
0	1	1	$2^{15} \times 1/f_{XT}$	1 s	853.3 ms
1	0	0	$2^{16} \times 1/f_{XT}$	2 s	1.71 s
1	0	1	$2^{17} \times 1/f_{XT}$	4 s	3.41 s
1	1	0	$2^{18} \times 1/f_{XT}$	8 s	6.83 s
1	1	1	$2^{20} \times 1/f_{XT}$	32 s	27.3 s

Remark f_{XT}: Subsystem clock oscillation frequency

(2) Watchdog timer mode register (WDTM)

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

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

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

Figure 8-3. Format of Watchdog Timer Mode Register

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
WDTM	RUN	0	0	WDTM4	WDTM3	0	0	0	FF49H	00H	R/W

RUN	Watchdog timer operation selection[Note 1]
0	Stop counting.
1	Clears counter and starts counting.

WDTM4	WDTM3	Watchdog timer operation mode selection[Note 2]
0	1	Interval timer mode (generates a maskable interrupt (INTWDT) upon overflow occurrence.)[Note 3]
1	0	Watchdog timer mode 1 (generates a non-maskable interrupt (INTWDT) upon overflow occurrence.)
1	1	Watchdog timer mode 2 (starts reset operation upon overflow occurrence.)

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

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

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

Cautions 1. When the watchdog timer is cleared by setting RUN to 1, the actual overflow time is up to $2^8/f_{XT}$ seconds shorter than the time set by the watchdog timer clock select register (WDCS).

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

3. The vector address of INTWDT is 0002H in the development tool (ICE) and 0004H in the device. Set both 0002H and 0004H as the interrupt servicing branch address.

8.4 Watchdog Timer Operation

8.4.1 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 runaway detection time interval of the watchdog timer can be selected by bits 0 to 2 (WDCS0 to WDCS2) of 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, a system reset signal or a non-maskable interrupt is generated, depending on the value of bit 3 (WDTM3) of WDTM.

Cautions 1. The actual runaway detection time may be up to $2^8/f_{XT}$ seconds shorter than the set time.

2. The vector address of INTWDT is 0002H in the development tool (ICE) and 0004H in the device. Set both 0002H and 0004H as the interrupt servicing branch address.

Table 8-4. Watchdog Timer Runaway Detection Time

WDCS2	WDCS1	WDCS0	Runaway Detection Time	At f_{XT} = 32.768 kHz Operation	At f_{XT} = 38.4 kHz Operation
0	0	0	$2^{12} \times 1/f_{XT}$	125 ms	106.7 ms
0	0	1	$2^{13} \times 1/f_{XT}$	250 ms	213.3 ms
0	1	0	$2^{14} \times 1/f_{XT}$	500 ms	426.7 ms
0	1	1	$2^{15} \times 1/f_{XT}$	1 s	853.3 ms
1	0	0	$2^{16} \times 1/f_{XT}$	2 s	1.71 s
1	0	1	$2^{17} \times 1/f_{XT}$	4 s	3.41 s
1	1	0	$2^{18} \times 1/f_{XT}$	8 s	6.83 s
1	1	1	$2^{20} \times 1/f_{XT}$	32 s	27.3 s

f_{XT}: Subsystem clock oscillation frequency

8.4.2 Operation as interval timer

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

Select a count clock (or interval) by setting bits 0 to 2 (WDCS0 to WDCS2) of the watchdog timer clock 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 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.

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

2. The interval time may be up to $2^8/f_{XT}$ seconds shorter than the set time when WDTM has just been set.

Table 8-5. Interval Time of Interval Timer

WDCS2	WDCS1	WDCS0	Interval	At f_{XT} = 32.768 kHz Operation	At f_{XT} = 38.4 kHz Operation
0	0	0	$2^{12} \times 1/f_{XT}$	125 ms	106.7 ms
0	0	1	$2^{13} \times 1/f_{XT}$	250 ms	213.3 ms
0	1	0	$2^{14} \times 1/f_{XT}$	500 ms	426.7 ms
0	1	1	$2^{15} \times 1/f_{XT}$	1 s	853.3 ms
1	0	0	$2^{16} \times 1/f_{XT}$	2 s	1.71 s
1	0	1	$2^{17} \times 1/f_{XT}$	4 s	3.41 s
1	1	0	$2^{18} \times 1/f_{XT}$	8 s	6.83 s
1	1	1	$2^{20} \times 1/f_{XT}$	32 s	27.3 s

f_{XT}: Subsystem clock oscillation frequency

9.1 Functions of Serial Interface UART0

The serial interface UART0 has the following two modes.

(1) Operation stop mode

This mode is used when serial transfer is not performed to reduce power consumption.

For details, see **9.4.1 Operation stop mode**.

(2) Asynchronous serial interface (UART) mode

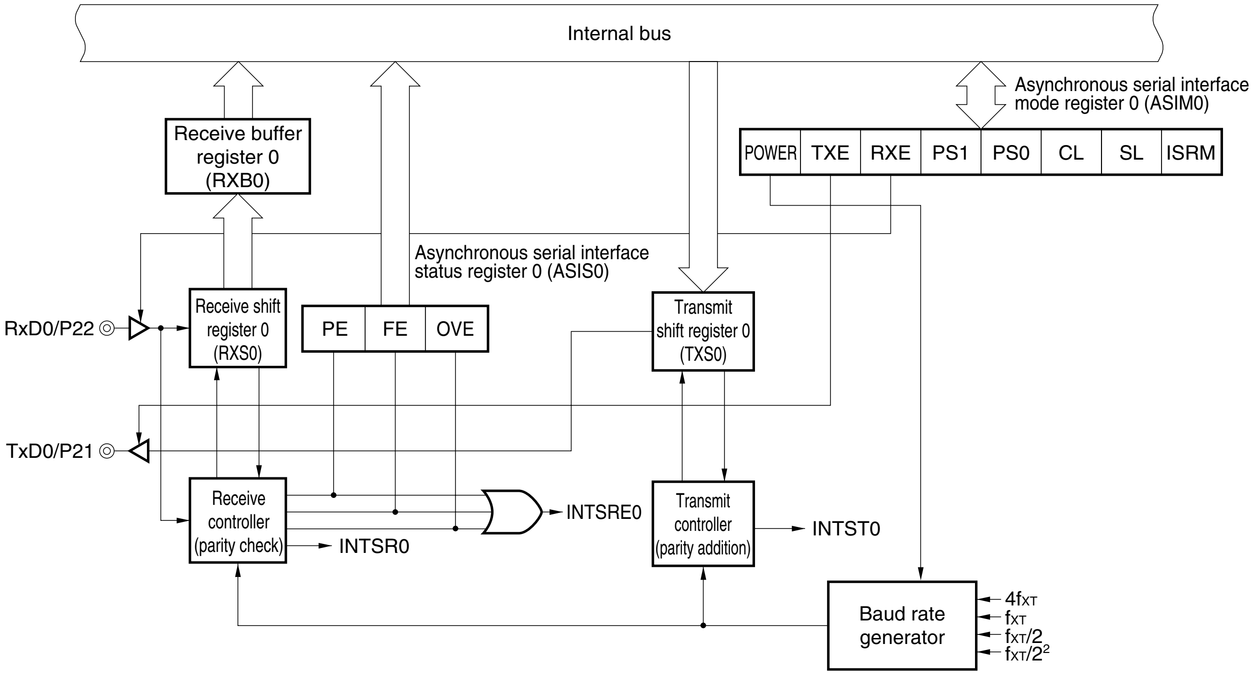
In this mode, one byte of data starting with the start bit is transmitted/received, and full-duplex operation is possible.

A UART-dedicated baud rate generator is incorporated, allowing communication over a wide range of baud rates.

For details, see **9.4.2 Asynchronous serial interface (UART) mode**.

Figure 9-1 shows the serial interface UART0 block diagram.

Figure 9-1. Block Diagram of Serial Interface UART0


Internal bus
Asynchronous serial interface mode register 0 (ASIM0)
POWER
TXE
RXE
PS1
PS0
CL
SL
ISRM
Receive buffer register 0 (RXB0)
Asynchronous serial interface status register 0 (ASIS0)
RxD0/P22
Receive shift register 0 (RXS0)
PE
FE
OVE
Transmit shift register 0 (TXS0)
TxD0/P21
Receive controller (parity check)
INTSRE0
INTSR0
Transmit controller (parity addition)
INTST0
Baud rate generator
4fXT
fXT
fXT/2
fXT/2²

Remark Baud rate generator operation (operate/stop) can be controlled by setting bit 7 (POWER) of asynchronous serial interface mode register 0 (ASIM0).

9.2 Configuration of Serial Interface UART0

The serial interface UART0 consists of the following hardware.

Table 9-1. Configuration of Serial Interface UART0

Item	Configuration
Registers	Transmit shift register 0 (TXS0) Receive shift register 0 (RXS0) Receive buffer register 0 (RXB0)
Control registers	Asynchronous serial interface mode register 0 (ASIM0) Asynchronous serial interface status register 0 (ASIS0) Baud rate generator control register 0 (BRGC0)

(1) Transmit shift register 0 (TXS0)

This register is used to set the transmit data. The data written in TXS0 is transmitted as serial data.
If the data length is specified as 7 bits, bits 0 to 6 of the data written in TXS0 are transmitted as transmit data.
Writing data to TXS0 starts the transmit operation.
TXS0 can be written by an 8-bit memory manipulation instruction. It cannot be read.
$\overline{\text{RESET}}$ input sets TXS0 to FFH.

Caution TXS0 must not be written to during a transmit operation. Write the next transmit data to TXS0 after the transmission completion interrupt (INTST) is generated.

(2) Receive shift register 0 (RXS0)

This register is used to convert serial data input to the RxD0 pin to parallel data. When one byte of data is received, the receive data is transferred to receive buffer register 0 (RXB0).
RXS0 cannot be directly manipulated by a program.

(3) Receive buffer register 0 (RXB0)

This register holds receive data. Each time one byte of data is received, new receive data is transferred from receive shift register 0 (RXS0).
If the data length is specified as 7 bits, the receive data is transferred to bits 0 to 6 of RXB0, and the MSB of RXB0 is always cleared to 0.
RXB0 can be read by an 8-bit memory manipulation instruction. It cannot be written to.
$\overline{\text{RESET}}$ input sets RXB0 to FFH.

(4) Transmit controller

This circuit controls transmit operations such as the addition of a start bit, parity bit, and stop bit to data written in transmit shift register 0 (TXS0) in accordance with the contents set in asynchronous serial interface mode register 0 (ASIM0).

(5) Receive controller

This circuit controls receive operations in accordance with the contents set in asynchronous serial interface mode register 0 (ASIM0). It also performs error checks for parity errors, etc., during receive operations, and if an error is detected, sets a value that corresponds to the error in asynchronous serial interface status register 0 (ASIS0).

9.3 Registers Controlling Serial Interface UART0

The serial interface UART0 is controlled by the following three registers.

- Asynchronous serial interface mode register 0 (ASIM0)
- Asynchronous serial interface status register 0 (ASIS0)
- Baud rate generator control register 0 (BRGC0)

(1) Asynchronous serial interface mode register 0 (ASIM0)

This is an 8-bit register that sets UART operation enable/disable and controls the serial transfer operation of the UART interface.

ASIM0 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ASIM0 to 01H.

Figure 9-2 shows the format of this register.

Caution Set the port mode register 2 (PM2) and its output latch in the UART mode as shown below.

- **For receive**
  Set P22 (RxD0) to input mode (PM22 = 1).
- **For transmit**
  Set P21 (TxD0) to output mode (PM21 = 0), and set output latch to 1.
- **For transmit and receive**
  Set P22 to input mode and P21 to output mode, and set P21 output latch to 1.

Figure 9-2. Format of Asynchronous Serial Interface Mode Register 0

Address: FF90H After reset: 01H R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
ASIM0	POWER	TXE	RXE	PS1	PS0	CL	SL	ISRM

POWER	Specifies operation mode
0	Operation disabled (internal circuit reset asynchronously)
1	Operation enabled

TXE	RXE	Operation mode	Function of RxD0/P22 pin	Function of TxD0/P21 pin
0	0	Operation stops	Port function (P22)	Port function (P21)
0	1	UART mode (receive only)	Serial function (RxD0)	
1	0	UART mode (transmit only)	Port function (P22)	Serial function (TxD0)
1	1	UART mode (transmit/receive)	Serial function (RxD0)	

PS1	PS0	Specifies parity bit
0	0	No parity
0	1	Transmission: Always 0 parity added Reception: Parity not checked (parity error not generated)
1	0	Odd parity
1	1	Even parity

CL	Specifies character length of transmit/receive data
0	7 bits
1	8 bits

SL	Specifies number of stop bits of transmit data
0	1 bit
1	2 bits

ISRM	Reception end interrupt control upon occurrence of error
0	Does not generate reception completion interrupt request (INTSR0) and generates reception error generation interrupt (INTSRE0) on occurrence of error.
1	Generates reception completion interrupt request (INTSR0) on occurrence of error (does not generate reception error generation interrupt (INTSRE0)).

Cautions 1. Before changing the operation mode, be sure to stop serial transmission/reception.

2. The TxD0 pin outputs a low level when POWER = 0, and outputs a high level when POWER = 1. When a serial operation is enabled, therefore, be sure to set TxD0/P21 to output mode after POWER is set to 1 (operation enabled) to prevent the communication partner from malfunctioning.

(2) Asynchronous serial interface status register 0 (ASIS0)

This register indicates the error contents when a receive error occurs in the UART mode.

ASIS0 can be read by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ASIS0 to 00H.

Figure 9-3. Format of Asynchronous Serial Interface Status Register 0

Address: FF93H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS0	0	0	0	0	0	PE	FE	OVE

PE	Parity error flag
0	Parity error does not occur.
1	Parity error occurs (specified parity of transmit data does not match receive data parity).

FE	Framing error flag
0	Framing error does not occur.
1	Framing error occurs[Note 1] (when stop bit is not detected).

OVE	Overrun error flag
0	Overrun error does not occur.
1	Overrun error occurs[Notes 2, 3] (when next receive is completed before data is read from receive buffer register).

Notes 1. Even if the stop bit length is set to 2 bits by setting bit 1 (SL) of asynchronous serial interface mode register 0 (ASIM0) to 1, only 1 stop bit is detected during reception.

2. Be sure to read receive buffer register 0 (RXB0) when an overrun error occurs.
Until RXB0 is read, an overrun error persistently occurs each time data has been received.

3. If an overrun error occurs, the next receive data is not written to RXB0, but discarded.

(3) Baud rate generator control register 0 (BRGC0)

This register sets the serial clock of the UART interface.

BRGC0 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets BRGC0 to 1FH.

Figure 9-4 shows the format of BRGC0.

Figure 9-4. Format of Baud Rate Generator Control Register 0

Address: FF91H After reset: 1FH R/W

Symbol	7	6	5	4	3	2	1	0
BRGC0	TPS1	TPS0	0	MDL4	MDL3	MDL2	MDL1	MDL0

TPS1	TPS0	Selects input clock (f_{XCLK})
0	0	4f_{XT} (131 kHz)
0	1	f_{XT} (32.768 kHz)
1	0	$f_{XT}/2$ (16.4 kHz)
1	1	$f_{XT}/2^2$ (8.2 kHz)

MDL4	MDL3	MDL2	MDL1	MDL0	Selects 5-bit counter divide value	k
0	1	0	0	0	$f_{XCLK}/8$	8
0	1	0	0	1	$f_{XCLK}/9$	9
0	1	0	1	0	$f_{XCLK}/10$	10
0	1	0	1	1	$f_{XCLK}/11$	11
0	1	1	0	0	$f_{XCLK}/12$	12
0	1	1	0	1	$f_{XCLK}/13$	13
0	1	1	1	0	$f_{XCLK}/14$	14
0	1	1	1	1	$f_{XCLK}/15$	15
1	0	0	0	0	$f_{XCLK}/16$	16
1	0	0	0	1	$f_{XCLK}/17$	17
1	0	0	1	0	$f_{XCLK}/18$	18
1	0	0	1	1	$f_{XCLK}/19$	19
1	0	1	0	0	$f_{XCLK}/20$	20
1	0	1	0	1	$f_{XCLK}/21$	21
1	0	1	1	0	$f_{XCLK}/22$	22
1	0	1	1	1	$f_{XCLK}/23$	23
1	1	0	0	0	$f_{XCLK}/24$	24
1	1	0	0	1	$f_{XCLK}/25$	25
1	1	0	1	0	$f_{XCLK}/26$	26
1	1	0	1	1	$f_{XCLK}/27$	27
1	1	1	0	0	$f_{XCLK}/28$	28
1	1	1	0	1	$f_{XCLK}/29$	29
1	1	1	1	0	$f_{XCLK}/30$	30
1	1	1	1	1	$f_{XCLK}/31$	31
Other than above					Setting prohibited	−

Cautions 1. If a write to BRGC0 is performed during communication, the output of the baud rate generator may be disrupted, preventing normal communication. The BRGC0 register should therefore not be written to during communication.

2. Be sure to clear POWER (bit 7 of ASIM0) to 0 before rewriting BRGC0.

Remark k: Value set by bits MDL0 to MDL4 ($8 \le k \le 31$)

9.4 Operation of Serial Interface UART0

The two serial interface UART0 modes are described below.

9.4.1 Operation stop mode

Since serial transfer is not performed in this mode, the power consumption can be reduced.
In addition, the pins can be used as ordinary port pins in this mode.

(1) Register setting

The operation stop mode can be set by asynchronous serial interface mode register 0 (ASIM0).
ASIM0 can be set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets ASIM0 to 01H.

Address: FF90H After reset: 01H R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
ASIM0	POWER	TXE	RXE	PS1	PS0	CL	SL	ISRM

POWER	Specifies operation mode
0	Operation disabled (internal circuit reset asynchronously)
1	Operation enabled

TXE	RXE	Operation mode	Function of RxD0/P22 pin	Function of TxD0/P21 pin
0	0	Operation stops	Port function (P22)	Port function (P21)
0	1	UART mode (receive only)	Serial function (RxD0)	
1	0	UART mode (transmit only)	Port function (P22)	Serial function (TxD0)
1	1	UART mode (transmit/receive)	Serial function (RxD0)	

Cautions 1. Before changing the operation mode, be sure to stop serial transmission/reception.

2. TxD0 pin outputs low level when POWER = 0, and outputs high level when POWER = 1. When a serial operation is enabled, therefore, be sure to set TxD0/P21 to output mode after POWER is set to 1 (operation enabled) to prevent the communication partner from malfunctioning.

9.4.2 Asynchronous serial interface (UART) mode

In this mode, one byte of data starting with the start bit is transmitted/received, and full-duplex operation is possible. A UART-dedicated baud rate generator is incorporated, allowing communication over a wide range of baud rates.

(1) Register setting

The UART mode is set by using asynchronous serial interface mode register 0 (ASIM0), asynchronous serial interface status register 0 (ASIS0), and baud rate generator control register 0 (BRGC0).

(a) Asynchronous serial interface mode register 0 (ASIM0)

ASIM0 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ASIM0 to 01H.

Caution Set the port mode register 2 (PM2) in the UART mode as shown below. Clear each output latch to 0.

- **For receive**
  Set P22 (RxD0) to input mode (PM22 = 1).
- **For transmit**
  Set P21 (TxD0) to output mode (PM21 = 0), and set output latch to 1.
- **For transmit and receive**
  Set P22 to input mode and P21 to output mode, and set P21 output latch to 1.

Address: FF90H After reset: 01H R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
ASIM0	POWER	TXE	RXE	PS1	PS0	CL	SL	ISRM

POWER	Specifies operation mode
0	Operation disabled (internal circuit reset asynchronously)
1	Operation enabled

TXE	RXE	Operation mode	Function of RxD0/P22 pin	Function of TxD0/P21 pin
0	0	Operation stops	Port function (P22)	Port function (P21)
0	1	UART mode (receive only)	Serial function (RxD0)	
1	0	UART mode (transmit only)	Port function (P22)	Serial function (TxD0)
1	1	UART mode (transmit/receive)	Serial function (RxD0)	

PS1	PS0	Specifies parity bit
0	0	No parity
0	1	Transmission: Always 0 parity added Reception: Parity not checked (parity error not generated)
1	0	Odd parity
1	1	Even parity

CL	Specifies character length
0	7 bits
1	8 bits

SL	Specifies transmit data stop bit length
0	1 bit
1	2 bits

ISRM	Reception end interrupt control upon occurrence of error
0	Does not generate reception completion interrupt request and generates reception error generation interrupt (INTSRE0) on occurrence of error.
1	Generates reception completion interrupt request on occurrence of error (does not generate reception error generation interrupt (INTSRE0)).

Cautions 1. Before changing the operation mode, be sure to stop serial transmission/reception.

2. The TxD0 pin outputs a low level when POWER = 0, and outputs a high level when POWER = 1. When a serial operation is enabled, therefore, be sure to set TxD0/P21 to output mode after POWER is set to 1 (operation enabled) to prevent the communication partner from malfunctioning.

(b) Asynchronous serial interface status register 0 (ASIS0)

ASIS0 can be read by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ASIS0 to 00H.

Address: FF93H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS0	0	0	0	0	0	PE	FE	OVE

PE	Parity error flag
0	Parity error does not occur.
1	Parity error occurs (specified parity of transmit data does not match receive data parity).

FE	Framing error flag
0	Framing error does not occur.
1	Framing error occurs[Note 1] (when stop bit is not detected).

OVE	Overrun error flag
0	Overrun error does not occur.
1	Overrun error occurs[Notes 2, 3] (when next receive is completed before data is read from receive buffer register).

Notes 1. Even if the stop bit length is set to 2 bits by setting bit 1 (SL) of asynchronous serial interface mode register 0 (ASIM0) to 1, only 1 stop bit is detected during reception.

2. Be sure to read receive buffer register 0 (RXB0) when an overrun error occurs.
Until RXB0 is read, an overrun error persistently occurs each time data has been received.

3. If an overrun error occurs, the next receive data is not written to RXB0, but discarded.

(c) Baud rate generator control register 0 (BRGC0)

BRGC0 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets BRGC0 to 1FH.

Address: FF91H After reset: 1FH R/W

Symbol	7	6	5	4	3	2	1	0
BRGC0	TPS1	TPS0	0	MDL4	MDL3	MDL2	MDL1	MDL0

TPS1	TPS0	Selects input clock (f_{XCLK})
0	0	4f_{XT}
0	1	f_{XT} (32.768 kHz)
1	0	f_{XT}/2 (16.4 kHz)
1	1	$f_{XT}/2^2$ (8.2 kHz)

MDL4	MDL3	MDL2	MDL1	MDL0	Selects 5-bit counter divide value	k
0	1	0	0	0	f_{XCLK}/8	8
0	1	0	0	1	f_{XCLK}/9	9
0	1	0	1	0	f_{XCLK}/10	10
0	1	0	1	1	f_{XCLK}/11	11
0	1	1	0	0	f_{XCLK}/12	12
0	1	1	0	1	f_{XCLK}/13	13
0	1	1	1	0	f_{XCLK}/14	14
0	1	1	1	1	f_{XCLK}/15	15
1	0	0	0	0	f_{XCLK}/16	16
1	0	0	0	1	f_{XCLK}/17	17
1	0	0	1	0	f_{XCLK}/18	18
1	0	0	1	1	f_{XCLK}/19	19
1	0	1	0	0	f_{XCLK}/20	20
1	0	1	0	1	f_{XCLK}/21	21
1	0	1	1	0	f_{XCLK}/22	22
1	0	1	1	1	f_{XCLK}/23	23
1	1	0	0	0	f_{XCLK}/24	24
1	1	0	0	1	f_{XCLK}/25	25
1	1	0	1	0	f_{XCLK}/26	26
1	1	0	1	1	f_{XCLK}/27	27
1	1	1	0	0	f_{XCLK}/28	28
1	1	1	0	1	f_{XCLK}/29	29
1	1	1	1	0	f_{XCLK}/30	30
1	1	1	1	1	f_{XCLK}/31	31
Other than above					Setting prohibited	–

Cautions 1. If a write to BRGC0 is performed during communication, the output of the baud rate generator may be disrupted, preventing normal communication. The BRGC0 register should therefore not be written to during communication.

2. Be sure to clear POWER (bit 7 of ASIM0) to 0 before rewriting BRGC0.

Remark k: Value set by bits MDL0 to MDL4 ($8 \leq k \leq 31$)

(2) Baud rate setting

The transmit/receive clock for the baud rate to be generated is obtained by dividing the subsystem clock.

(a) Generating transmit/receive clock for baud rate from subsystem clock

The transmit/receive clock is generated by dividing the subsystem clock. The baud rate generated from the subsystem clock can be calculated from the following expression.

$$[\text{Baud rate}] = \frac{f_{XCLK}}{2 \times k} \ [\text{bps}]$$

f_{XCLK}: 5-bit counter input clock
k: Value set by MDL0 to MDL4 ($8 \leq k \leq 31$)

(b) Error of baud rate

The error of generated baud rate can be calculated from the following expression.

$$[\text{Baud rate error}] = \frac{(\text{Baud rate})}{(\text{Target baud rate value})} \times 100 - 100 \ [\%]$$

Table 9-2 lists the relationship between input clock, target baud rate, and baud rate error.

Table 9-2. Relationship Between Input Clock, Target Baud Rate, and Baud Rate Error

Target Baud Rate (bps)	At f_{XT} = 32.768 kHz Operation								At f_{XT} = 38.4 kHz Operation							
	$f_{XCLK} = f_{XT}/2^2$		$f_{XCLK} = f_{XT}/2$		$f_{XCLK} = f_{XT}$		$f_{XCLK} = 4f_{XT}$		$f_{XCLK} = f_{XT}/2^2$		$f_{XCLK} = f_{XT}/2$		$f_{XCLK} = f_{XT}$		$f_{XCLK} = 4f_{XT}$	
	k	Error (%)	k	Error (%)	k	Error (%)	k	Error (%)	k	Error (%)	k	Error (%)	k	Error (%)	k	Error (%)
200	20	2.4	–	–	–	–	–	–	24	0	–	–	–	–	–	–
300	14	–2.48	28	–2.48	–	–	–	–	16	0	–	–	–	–	–	–
600	–	–	14	–2.48	28	–2.48	–	–	8	0	16	0	–	–	–	–
1200	–	–	–	–	14	–2.48	–	–	–	–	8	0	16	0	–	–
2400	–	–	–	–	–	–	28	–2.48	–	–	–	–	8	0	–	–
4800	–	–	–	–	–	–	14	–2.48	–	–	–	–	–	–	16	0
9600	–	–	–	–	–	–	–	–	–	–	–	–	–	–	8	0

Remarks 1. f_{XCLK}: 5-bit counter input clock
2. k: Value set by MDL0 to MDL4 ($8 \leq k \leq 31$)

Cautions 1. Because the ×4 multiplication clock circuit is stopped in the HALT mode, $4f_{XT}$ cannot be selected as the input clock.
2. When a baud rate of 9600 bps is necessary, use a 38.4 kHz resonator for the subsystem clock.

(c) Error torelance range for baud rate

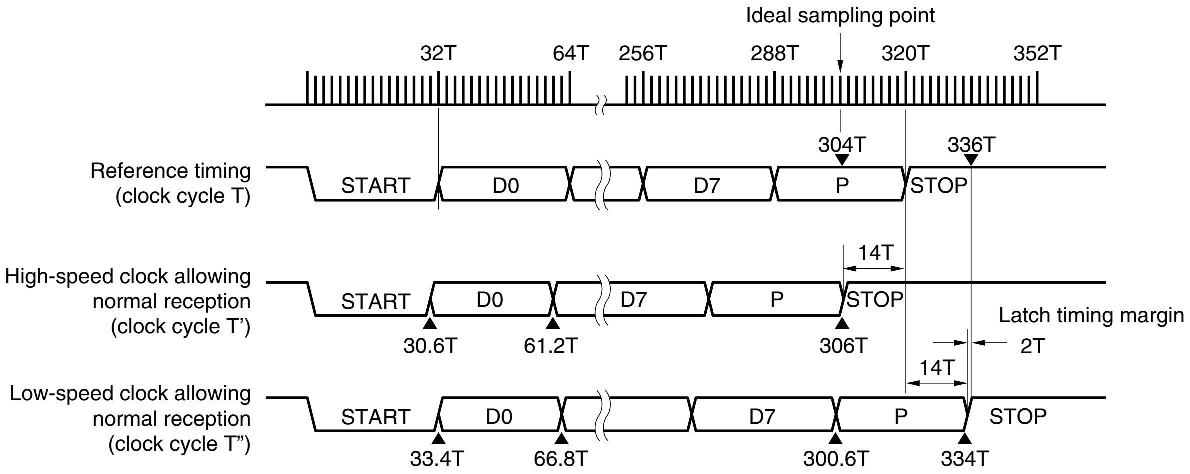
The tolerance range for the baud rate depends on the number of bits per frame and the counter's division ratio (k).

Transmission/reception can be performed normally if the difference between the baud rate error at the reception side and the baud rate error at the transmission side is within this tolerance range.

Figure 9-5 shows an example of a baud rate error tolerance.

Figure 9-5. Example of Baud Rate Error Tolerance Considering Margin

<Condition> Data character length: 8 bits
Parity: Provided (1 bit)
Stop bit: 1 bit
When k = 16



Remark T: 5-bit counter source clock cycle

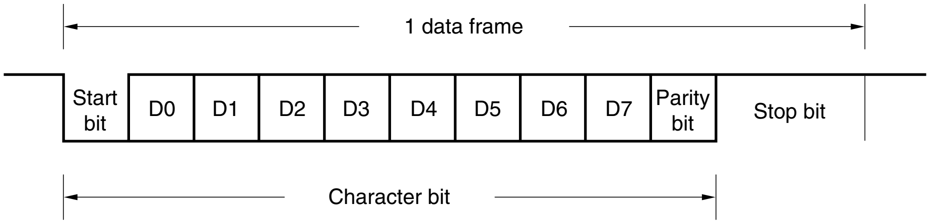
$$[\text{Baud rate error tolerance}]\ (\text{when } k = 16) = \frac{\pm 14}{320} \times 100 = 4.375\ (\%)$$

(3) Communication operation

(a) Data format

Figure 9-6 shows the transmit/receive data format.

Figure 9-6. Asynchronous Serial Interface Transmit/Receive Data Format



One data frame consists of the following bits.

- 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

The character bit length, parity bit, and stop bit length for each data frame are specified with asynchronous serial interface mode register 0 (ASIM0).

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 asynchronous serial interface mode register 0 (ASIM0) and baud rate generator control register 0 (BRGC0).

If a serial data receive error occurs, the receive error contents can be determined by reading the status of asynchronous serial interface status register 0 (ASIS0).

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

(i) Even parity

- Transmission

  The number of bits with a value of 1, including the parity bit, in the transmit data is controlled to be even. The value of the parity bit is as follows.

  Number of bits with a value of 1 in transmit data is odd: 1
  Number of bits with a value of 1 in transmit data is even: 0

- Reception

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

(ii) Odd parity

- Transmission

  Conversely to the situation with even parity, the number of bits with a value of 1, including the parity bit, in the transmit data is controlled to be odd. The value of the parity bit is as follows.

  Number of bits with a value of 1 in transmit data is odd: 0
  Number of bits with a value of 1 in transmit data is even: 1

- Reception

  The number of bits with a value of 1, including the parity bit, in the receive data is counted. If it is even, a parity error occurs.

(iii) 0 parity

When transmitting, the parity bit is cleared to 0 irrespective of the transmit data.

At reception, no parity bit check is performed. Therefore, no parity error is generated, irrespective of whether the parity bit is set to 0 or 1.

(iv) No parity

No parity bit is added to the transmit data.

At reception, data is received assuming that there is no parity bit. Since there is no parity bit, no parity error is generated.

(c) Transmission

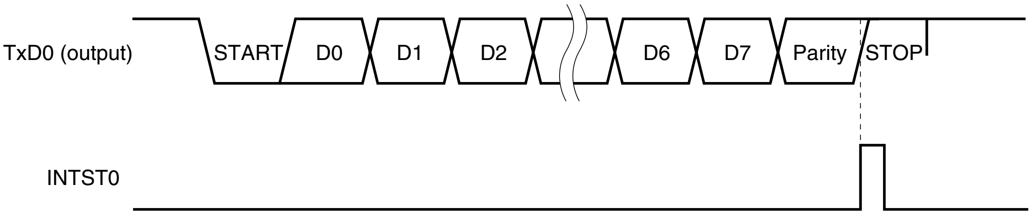
Transmit operation is enabled when bit 7 (TXE0) of asynchronous serial interface mode register 0 (ASIM0) is set to 1, and is started when transmit data is written to transmit shift register 0 (TXS0). The start bit, parity bit, and stop bit(s) are added automatically.

When the transmit operation starts, the data in TXS0 is shifted out, and when that register becomes empty, a transmission completion interrupt request (INTST0) is generated.

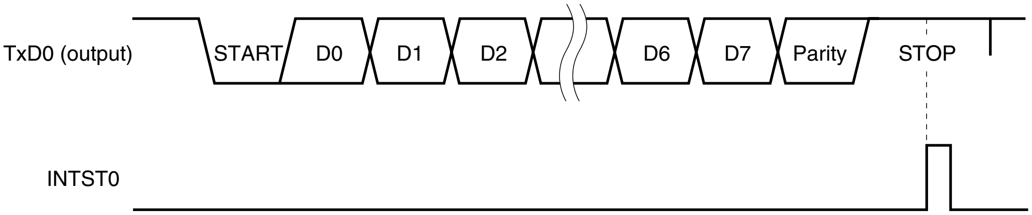
Figure 9-7 shows the generation timing of the transmission completion interrupt request.

Figure 9-7. Asynchronous Serial Interface Transmission Completion Interrupt Request Generation Timing

(i) Stop bit length: 1



(ii) Stop bit length: 2



Caution Writing to asynchronous serial interface mode register 0 (ASIM0) should not be performed during a transmit operation. If writing of ASIM0 register is done during transmission, subsequent transmit operations may not be possible (the normal state is restored by $\overline{\text{RESET}}$ input).

It is possible to determine whether transmission is in progress by software using a transmission completion interrupt request (INTST0) or the interrupt request flag (STIF0) set by INTST0.

(d) Reception

When bit 5 (RXE) of asynchronous serial interface mode register 0 (ASIM0) is set to 1, a receive operation is enabled and sampling of the RxD0 pin input is performed.

RxD0 pin input sampling is performed using the serial clock specified by ASIM0.

When the RxD0 pin input becomes low, the 5-bit counter of the baud rate generator starts counting, and at the time when the half time determined by the specified baud rate has passed, the data sampling start timing signal is output. If the RxD0 pin input sampled again as the result of this start timing signal is low, it is identified as a start bit, the 5-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.

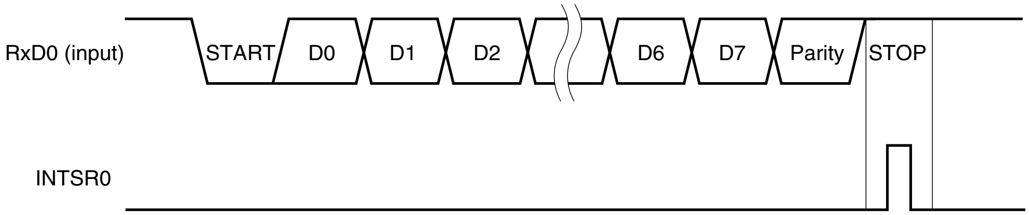
When one frame of data has been received, the receive data in the shift register is transferred to receive buffer register 0 (RXB0), and a reception completion interrupt request (INTSR0) is generated.

Even if an error is generated, the receive data in which the error was generated is still transferred to RXB0. INTSR0 is generated if bit 0 (ISRM) of ASIM0 is set to 1 at occurrence of the error. If the ISRM bit is cleared to 0 at this time, INTSR0 is not generated (see **Figure 9-9**).

If the RXE bit is reset to 0 during the receive operation, the receive operation is stopped immediately. In this case, the contents of RXB0 and ASIS0 are not changed, and INTSR0 and INTSRE0 are not generated.

Figure 9-8 shows the asynchronous serial interface reception completion interrupt request generation timing.

Figure 9-8. Asynchronous Serial Interface Reception Completion Interrupt Request Generation Timing



Caution Receive buffer register 0 (RXB0) must be read even if a receive error is generated. If RXB0 is not read, an overrun error will be generated when the next data is received, and the receive error state will continue indefinitely.

(e) Receive errors

Three types of errors can be generated during a receive operation: a parity error, a framing error, and an overrun error. If, as the result of data receive, an error flag is set in asynchronous serial interface status register 0 (ASIS0), a receive error interrupt request (INTSRE0) is generated. The receive error interrupt is generated before the reception completion interrupt request (INTSR0). Table 9-3 shows the receive error causes.

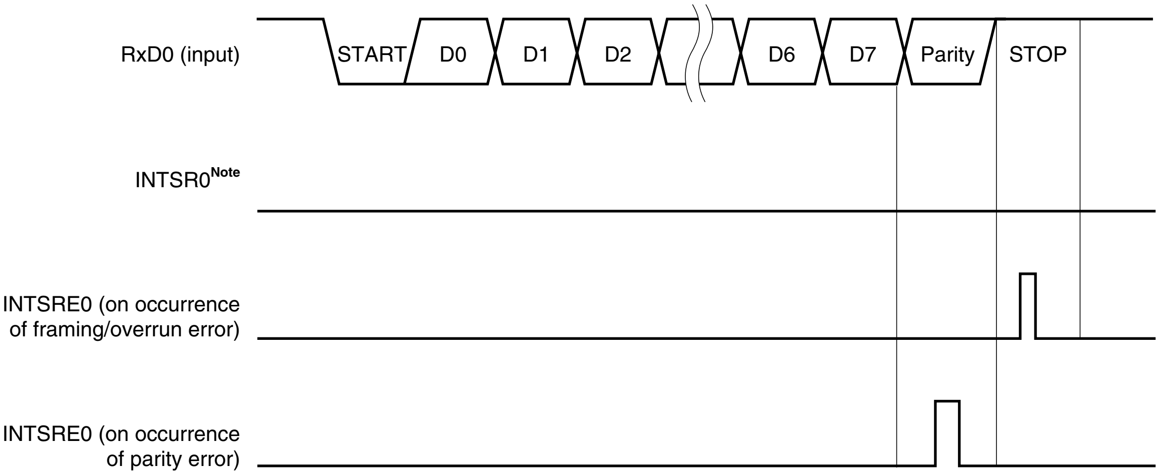
Reading the contents of ASIS0 during receive error interrupt servicing (INTSRE0) allows detection of what error has been generated during reception (see **Table 9-3** and **Figure 9-9**).

The contents of ASIS0 are reset to 0 by reading receive buffer register 0 (RXB0) or receiving the next data (if there is an error in the next data, the corresponding error flag is set).

Table 9-3. Receive Error Causes

Receive Error	Cause	ASIS0 Value
Parity error	The parity specified at transmission and the parity of the receive data 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 9-9. Receive Error Timing


RxD0 (input)
START
D0
D1
D2
D6
D7
Parity
STOP
INTSR0Note
INTSRE0 (on occurrence of framing/overrun error)
INTSRE0 (on occurrence of parity error)

Note INTSR0 is not generated if a receive error is generated when ISRM bit is cleared to 0.

Cautions 1. The contents of asynchronous serial interface status register 0 (ASIS0) are reset to 0 by reading receive buffer register 0 (RXB0) or receiving the next data. To ascertain the error contents, ASIS0 must be read before reading RXB0.

2. Receive buffer register 0 (RXB0) must be read even if a receive error is generated. If RXB0 is not read, an overrun error will be generated when the next data is received, and the receive error state will continue indefinitely.

CHAPTER 10 LCD CONTROLLER/DRIVER

10.1 LCD Controller/Driver Functions

The functions of the LCD controller/driver of the μPD789881 Subseries are as follows.

(1) Automatic output of segment and common signals based on automatic display data memory read

(2) Two different display modes:

- Static
- 1/4 duty (1/3 bias)

(3) Four different frame frequencies, selectable in each display mode

(4) 15 to 26 segment signal outputs
(S0 to S14, S15 to S25 (usable by port function register)), 4 common signal outputs (COM0 to COM3)

The maximum number of displayable pixels is shown in Table 10-1 below.

Table 10-1. Maximum Number of Display Pixels

Bias Mode	Time Division	Common Signals Used	Maximum Number of Segments	Maximum Number of Display Pixels
Static		COM0	26	26 (26 segments × 1 common)[Note 1]
1/3	4	COM0 to COM3		104 (26 segments × 4 commons)[Note 2]

Notes 1. The LCD panel of the figure 8. consists of 3 rows with 8 segments per row.

2. The LCD panel of the figure 8. consists of 13 rows with 2 segments per row.

10.2 LCD Controller/Driver Configuration

The LCD controller/driver includes the following hardware.

Table 10-2. Configuration of LCD Controller/Driver

Item	Configuration
Display outputs	Segment signals: 15 to 26 Common signals: 4 (COM0 to COM3)
Control registers	LCD display mode register 0 (LCDM0) LCD clock control register 0 (LCDC0) Port function registers 8 and 9 (PF8 and PF9)

The correspondence with the LCD display RAM is shown in Figure 10-1 below.

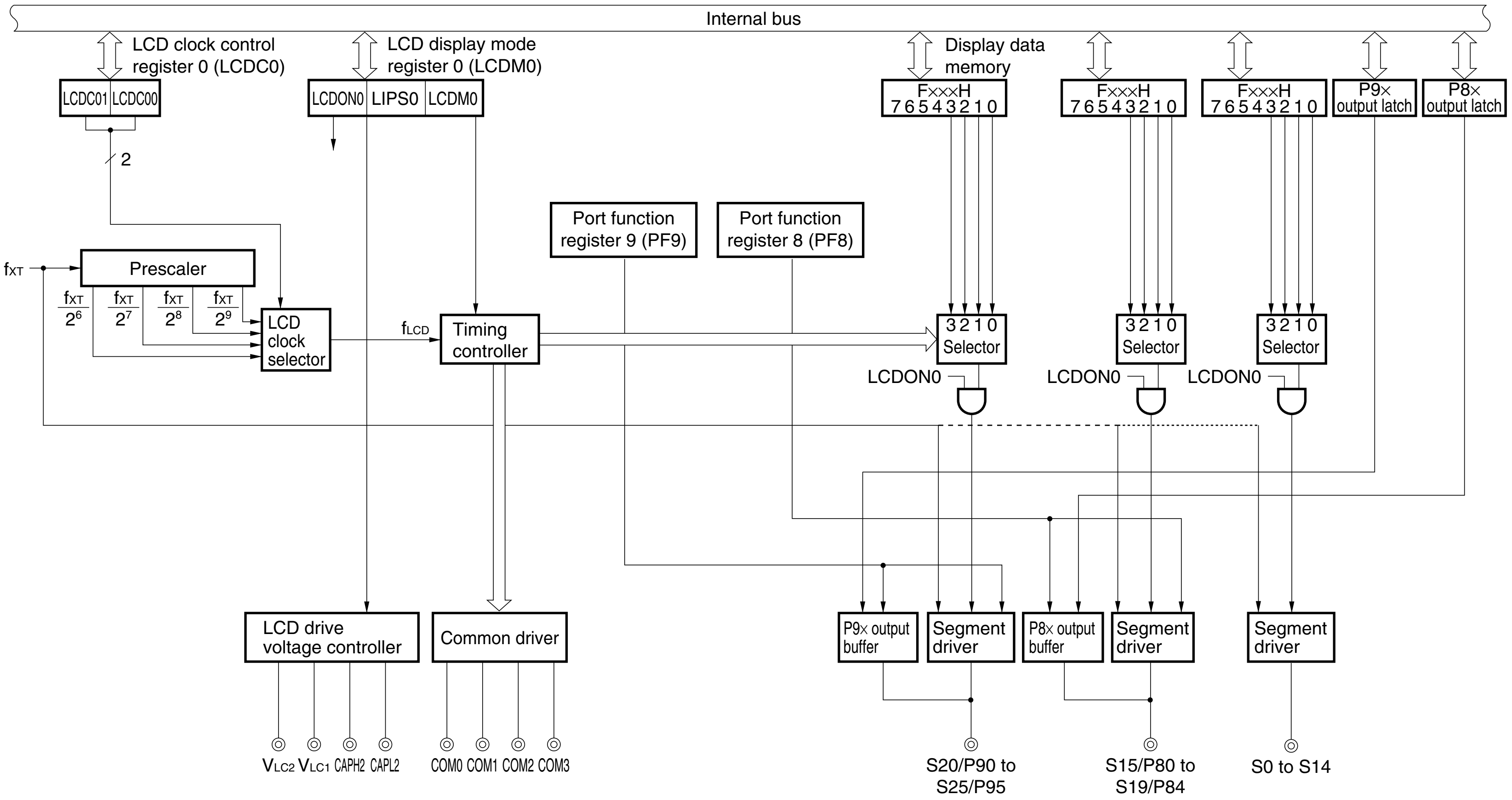
Figure 10-1. Correspondence with LCD Display RAM

Address	Bit								Segment
	7	6	5	4	3	2	1	0	
FA19H	0	0	0	0					→ S25[Note]
FA18H	0	0	0	0					→ S24[Note]
FA17H	0	0	0	0					→ S23[Note]
FA16H	0	0	0	0					→ S22[Note]
FA15H	0	0	0	0					→ S21[Note]
FA14H	0	0	0	0					→ S20[Note]
FA13H	0	0	0	0					→ S19[Note]
FA12H	0	0	0	0					→ S18[Note]
FA11H	0	0	0	0					→ S17[Note]
FA10H	0	0	0	0					→ S16[Note]
FA0FH	0	0	0	0					→ S15[Note]
FA0EH	0	0	0	0					→ S14
FA0DH	0	0	0	0					→ S13
FA0CH	0	0	0	0					→ S12
FA0BH	0	0	0	0					→ S11
FA0AH	0	0	0	0					→ S10
FA09H	0	0	0	0					→ S9
FA08H	0	0	0	0					→ S8
FA07H	0	0	0	0					→ S7
FA06H	0	0	0	0					→ S6
FA05H	0	0	0	0					→ S5
FA04H	0	0	0	0					→ S4
FA03H	0	0	0	0					→ S3
FA02H	0	0	0	0					→ S2
FA01H	0	0	0	0					→ S1
FA00H	0	0	0	0					→ S0
				Common	↑ COM3	↑ COM2	↑ COM1	↑ COM0	

Note Segments S15 to S25 are selected in 1-bit units via a port function register (segment output pin/port pin).

Remark Bits 4 to 7 are fixed to 0.

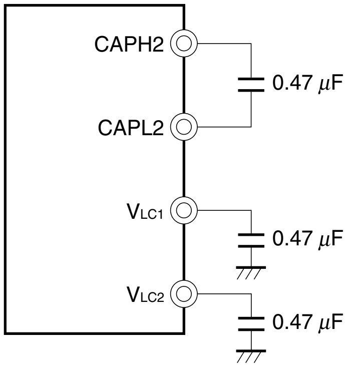
Figure 10-2. Block Diagram of LCD Controller/Driver



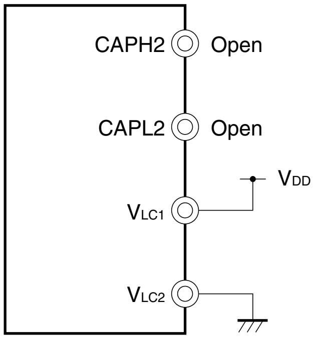
Remark Connect the LCD drive pins (V_{LC1}, V_{LC2}, CAPH2, and CAPL2) as shown in Figure 10-3, according to the display mode used.

Figure 10-3. Connection Example of LCD Drive Pin

(a) 1/4 duty, 1/3 bias, etc.



(b) Static mode



10.3 Registers Controlling LCD Controller/Driver

- LCD display mode register 0 (LCDM0)
- LCD clock control register 0 (LCDC0)
- Port function registers 8 and 9 (PF8 and PF9)

(1) LCD display mode register 0 (LCDM0)

This register specifies whether to enable display operation. It also specifies the LCD drive power supply and display mode.

LCDM0 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears LCDM0 to 00H.

Figure 10-4. Format of LCD Display Mode Register 0

Symbol	<7>	6	5	<4>	3	2	1	0	Address	After reset	R/W
LCDM0	LCDON0	0	0	LIPS0	0	LCDM0	0	0	FFB0H	00H	R/W

LCDON0	LCD display enable/disable
0	Display off (all segment outputs are deselect signal outputs)
1	Display on

LIPS0	LCD drive power supply[Note]
0	LCD drive power not supplied
1	LCD drive power supplied

LCDM0	LCD controller/driver display mode selection	
	Number of time slices	Bias mode
0	4	1/3
1	Static	

Note When the LCD display panel is not used, LIPS0 must be cleared to 0 to reduce power consumption.

Cautions 1. Be sure to set LCDON0 (display on) after setting LIPS0 (LCD drive power supply).

2. To use the 1/3 bias mode, be sure to set LCDON0 at least 500 ms after setting LIPS0.

3. In the static mode, clear LIPS0 to 0.

4. Be sure to change the display mode after turning off the display (LCDON0 = 0).

(2) LCD clock control register 0 (LCDC0)

This register specifies the LCD clock. The frame frequency is determined depending on the LCD clock and number of time divisions.

LCDC0 can be set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears LCDC0 to 00H.

Figure 10-5. 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	0	0	LCDC01	LCDC00	FFB2H	00H	R/W

LCDC01	LCDC00	LCD clock (LCDCL) selection
0	0	$f_{XT}/2^6$ (512 Hz)
0	1	$f_{XT}/2^7$ (256 Hz)
1	0	$f_{XT}/2^8$ (128 Hz)
1	1	$f_{XT}/2^9$ (64 Hz)

Cautions 1. Be sure to clear bits 2 to 7 to 0.

2. Do not rewrite LCDC0 while display is on (LCDON0 = 1).

Remarks 1. f_{XT}: Subsystem clock oscillation frequency

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

Table 10-3 lists the frame frequencies.

Table 10-3. Frame Frequencies (Hz)

LCD Clock (LCDCL) Display Duty Ratio	$f_{XT}/2^9$ (64 Hz)	$f_{XT}/2^8$ (128 Hz)	$f_{XT}/2^7$ (256 Hz)	$f_{XT}/2^6$ (512 Hz)
Static	64	128	256	512
1/4	16	32	64	128

(3) Port function registers (PF8, PF9)

These registers set in 1-bit units whether to use port 8 and port 9 as port or segment pins.

PF8 and PF9 can be set by an 8-bit memory manipulation instruction.

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

Figure 10-6. Format of Port Function Registers

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PF8	0	0	0	PF84	PF83	PF82	PF81	PF80	FF58H	00H	W
	7	6	5	4	3	2	1	0			
PF9	0	0	PF95	PF94	PF93	PF92	PF91	PF90	FF59H	00H	W

PFmn	Pmn function selection (m = 8, 9, n = 0 to 5)
0	Pmn is used as I/O port
1	Pmn is used as segment output

Cautions 1. Each bit of ports 8 and 9 function as segment output pins when PFmn = 1, regardless of the values of the corresponding port mode register and output latch. At this time, the values of the port mode register and output latch are invalid.

2. If an on-chip pull-up resistor is connected (PUBmn = 1), the pull-up resistor is not disconnected even when PFmn is set to 1. Be sure to clear PUBmn to 0 when setting PFmn to 1.

10.4 Setting LCD Controller/Driver

Set the LCD controller/driver using the following procedure.

<1> Set the LCD display data memory (FA00H to FA19H) to the default value.
<2> Set the pins used for segment output by using the port function registers (PF8 and PF9).
<3> Select the display mode by using LCD display mode register 0 (LCDM0).
<4> Set the LCD frame frequency by using LCD clock control register 0 (LCDC0).
<5> In the 1/4 duty, 1/3 bias mode, set bit 4 (LIPS0) of LCDM0 to 1 and then wait for 500 ms or more.
<6> Start display output corresponding to the LCD display data memory by setting bit 7 (LCDON0) of LCDM0 to 1.

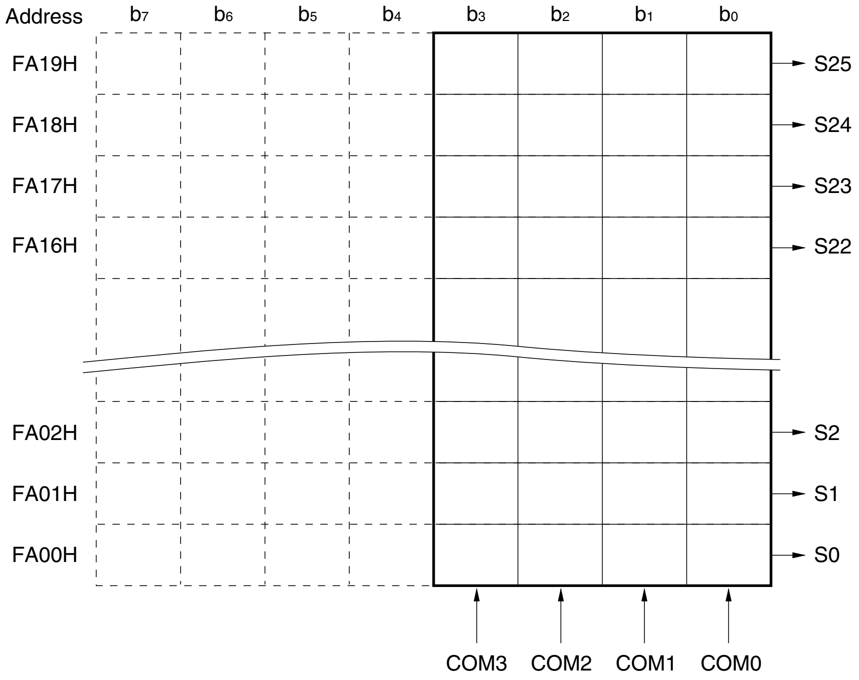
10.5 LCD Display Data Memory

The LCD display data memory is mapped at addresses FA00H to FA19H. Data in the LCD display data memory can be displayed on the LCD panel using the LCD controller/driver.

Figure 10-7 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 10-7. Relationship Between LCD Display Data Memory Contents and Segment/Common Outputs (When Using S15 to S25)



Caution No memory has been installed as the higher 4 bits of the LCD display data memory. Be sure to clear them to 0.

10.6 Common and Segment Signals

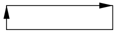
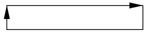
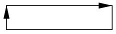
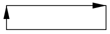
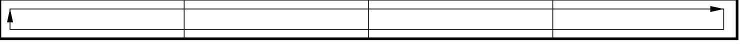
Each pixel of the LCD panel turns on when the potential difference between the corresponding common and segment signals becomes higher than a specific voltage (LCD drive voltage, V_{LCD}). It turns off when the potential difference becomes lower than V_{LCD}.

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 10-4. In the static mode, the same signal is output to COM0 to COM3.

Table 10-4. COM Signals

Number of Time Slots \ COM Signal	COM0	COM1	COM2	COM3
Static				
Four-time slot mode				

(2) Segment signals

The segment signals correspond to LCD display data memory. Bits 0, 1, 2, and 3 of each byte are read in synchronization with COM0, COM1, COM2, and COM3, respectively. If the contents of each bit are 1, 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.

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.

Bits 1 to 3 of the LCD display data memory are not used for LCD display in the static 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

Voltages listed in Table 10-5 are output as 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.

Table 10-5. LCD Drive Voltage

(a) Static display mode

Segment Signal Common Signal	Select Signal Level	Deselect Signal Level
	V_{SS0}/V_{LC0}	V_{LC0}/V_{SS0}
V_{LC0}/V_{SS0}	$-V_{LCD}/+V_{LCD}$	0 V/0 V

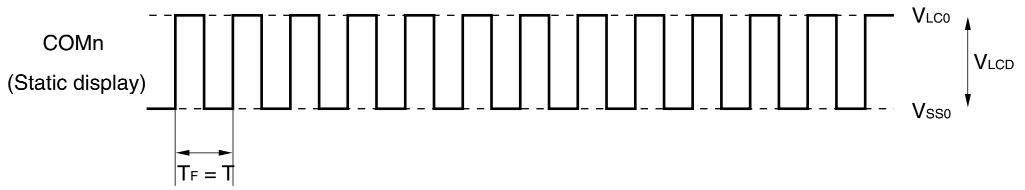
(b) 1/3 bias mode

Segment Signal Common Signal		Select Signal Level	Deselect Signal Level
		V_{SS0}/V_{LC0}	V_{LC1}/V_{LC2}
Select signal level	V_{LC0}/V_{SS0}	$-V_{LCD}/+V_{LCD}$	$-\frac{1}{3}V_{LCD}/+\frac{1}{3}V_{LCD}$
Deselect signal level	V_{LC2}/V_{LC1}	$-\frac{1}{3}V_{LCD}/+\frac{1}{3}V_{LCD}$	$-\frac{1}{3}V_{LCD}/+\frac{1}{3}V_{LCD}$

Figure 10-8 shows the common signal waveforms, and Figure 10-9 shows the voltages and phases of the common and segment signals.

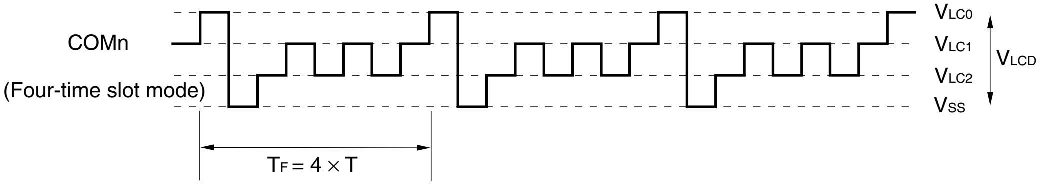
Figure 10-8. Common Signal Waveforms

(a) Static display mode



T: One LCD clock period TF: Frame frequency

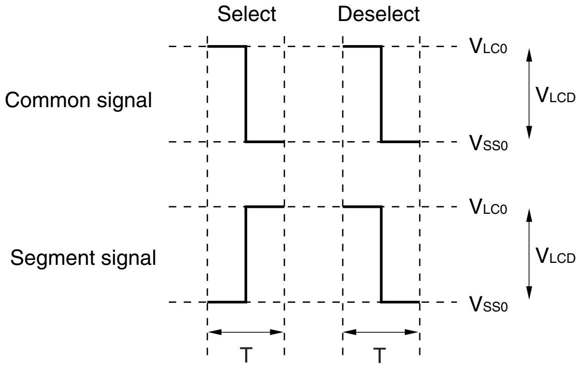
(b) 1/3 bias mode



T: One LCD clock period TF: Frame frequency

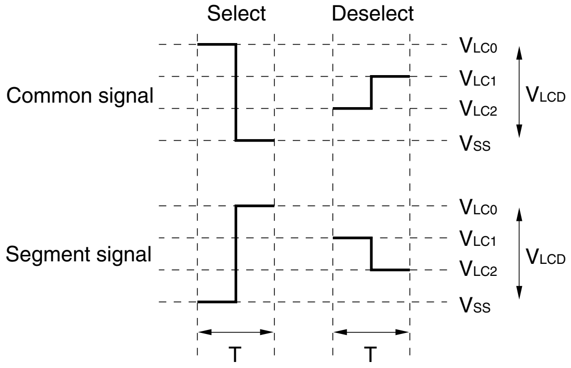
Figure 10-9. Voltages and Phases of Common and Segment Signals

(a) Static display mode



T: One LCD clock period

(b) 1/3 bias mode



T: One LCD clock period

10.7 Display Modes

10.7.1 Static display example

Figure 10-11 shows how the three-digit LCD panel having the display pattern shown in Figure 10-10 is connected to the segment signals (S0 to S23) and the common signal (COM0) of the μPD789881 Subseries chip. This example displays data "12.3" 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 "2." (2.) displayed in the second digit. To display "2." in the LCD panel, it is necessary to apply the select or deselect voltage to the S8 to S15 pins according to Table 10-6 at the timing of the common signal COM0; see Figure 10-10 for the relationships between the segment signals and LCD segments.

Table 10-6. Select and Deselect Voltages (COM0)

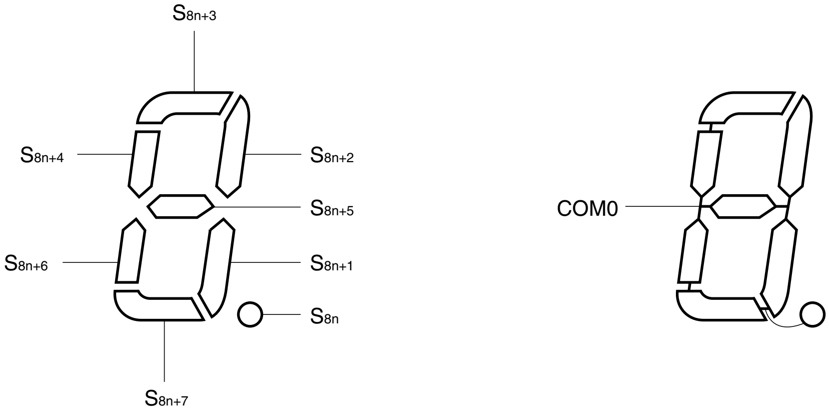
Segment / Common	S8	S9	S10	S11	S12	S13	S14	S15
COM0	Select	Deselect	Select	Select	Deselect	Select	Select	Select

According to Table 10-6, it is determined that the bit-0 pattern of the display data memory locations (FA08H to FA0FH) must be 10110111.

Figure 10-12 shows the LCD drive waveforms of S11 and S12, and COM0. When the select voltage is applied to S11 at the timing of COM0, an alternate rectangle waveform, $+V_{LCD}/-V_{LCD}$, is generated to turn on the corresponding LCD segment.

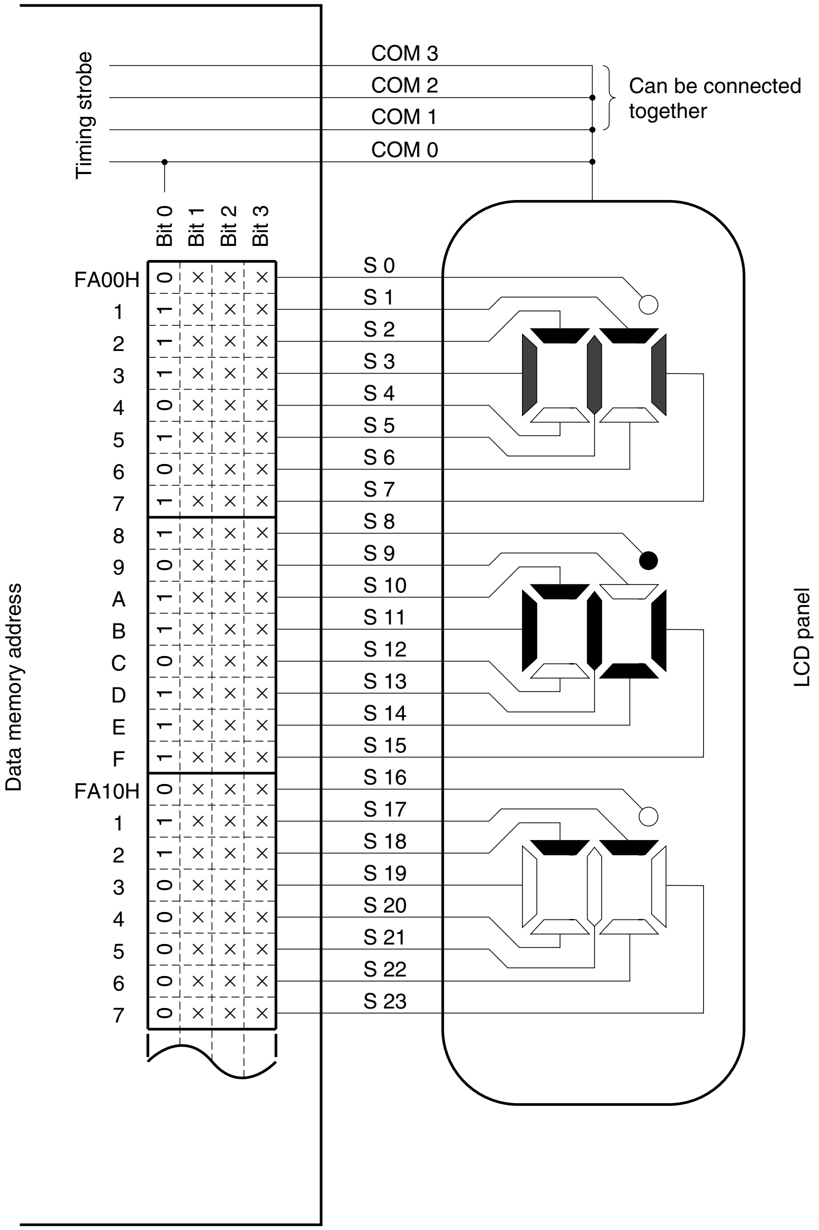
COM1 to COM3 are supplied with the same waveform as for COM0. So, COM0 to COM3 may be connected together to increase the driving capacity.

Figure 10-10. Static LCD Display Pattern and Electrode Connections



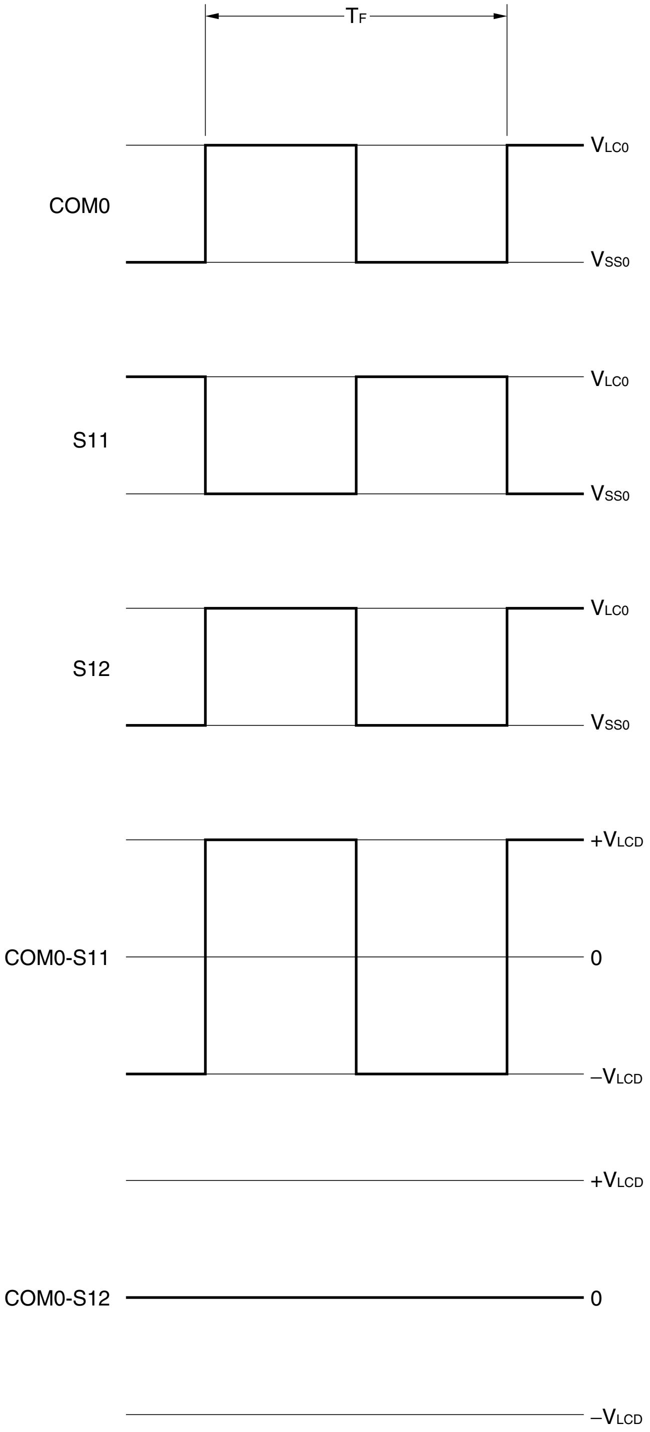
Remark n = 0 to 2

Figure 10-11. Example of Connecting Static LCD Panel



	Bit 0	Bit 1	Bit 2	Bit 3
FA00H	0	×	×	×
1	1	×	×	×
2	1	×	×	×
3	1	×	×	×
4	0	×	×	×
5	1	×	×	×
6	0	×	×	×
7	1	×	×	×
8	1	×	×	×
9	0	×	×	×
A	1	×	×	×
B	1	×	×	×
C	0	×	×	×
D	1	×	×	×
E	1	×	×	×
F	1	×	×	×
FA10H	0	×	×	×
1	1	×	×	×
2	1	×	×	×
3	0	×	×	×
4	0	×	×	×
5	0	×	×	×
6	0	×	×	×
7	0	×	×	×

Figure 10-12. Static LCD Drive Waveform Examples



10.7.2 Four-time slot display example

Figure 10-14 shows how the 13-digit LCD panel having the display pattern shown in Figure 10-13 is connected to the segment signals (S0 to S25) and the common signals (COM0 to COM3) of the μPD789881 Subseries chip. This example displays data "123456.7890123" in the LCD panel. The contents of the display data memory (addresses FA00H to FA19H) correspond to this display.

The following description focuses on numeral "6." (6.) displayed in the eighth digit. To display "6." in the LCD panel, it is necessary to apply the select or deselect voltage to the S14 and S15 pins according to Table 10-7 at the timing of the common signals COM0 to COM3; see Figure 10-13 for the relationships between the segment signals and LCD segments.

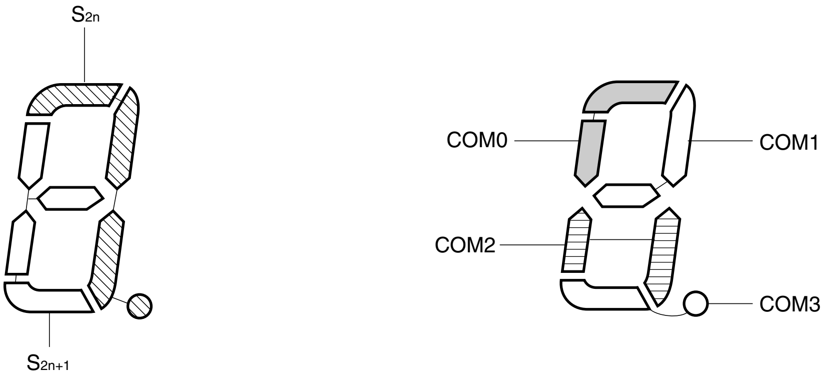
Table 10-7. Select and Deselect Voltages (COM0 to COM3)

Segment Common	S14	S15
COM0	Select	Select
COM1	Deselect	Select
COM2	Select	Select
COM3	Select	Select

According to Table 10-7, it is determined that the display data memory location (FA0EH) that corresponds to S14 must contain 1101.

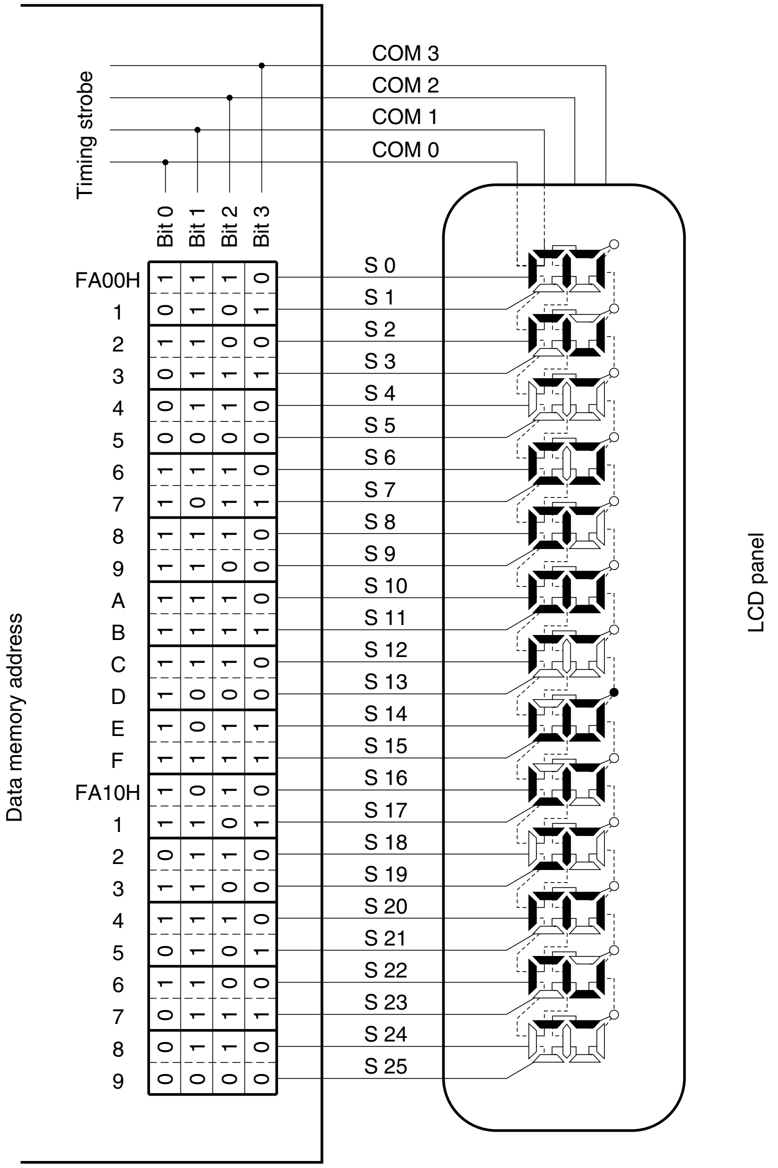
Figure 10-15 shows examples of LCD drive waveforms between the S14 signal and each common signal. When the select voltage is applied to S14 at the timing of COM0, an alternate rectangle waveform, $+V_{LCD}$/$-V_{LCD}$, is generated to turn on the corresponding LCD segment.

Figure 10-13. Four-Time Slot LCD Display Pattern and Electrode Connections



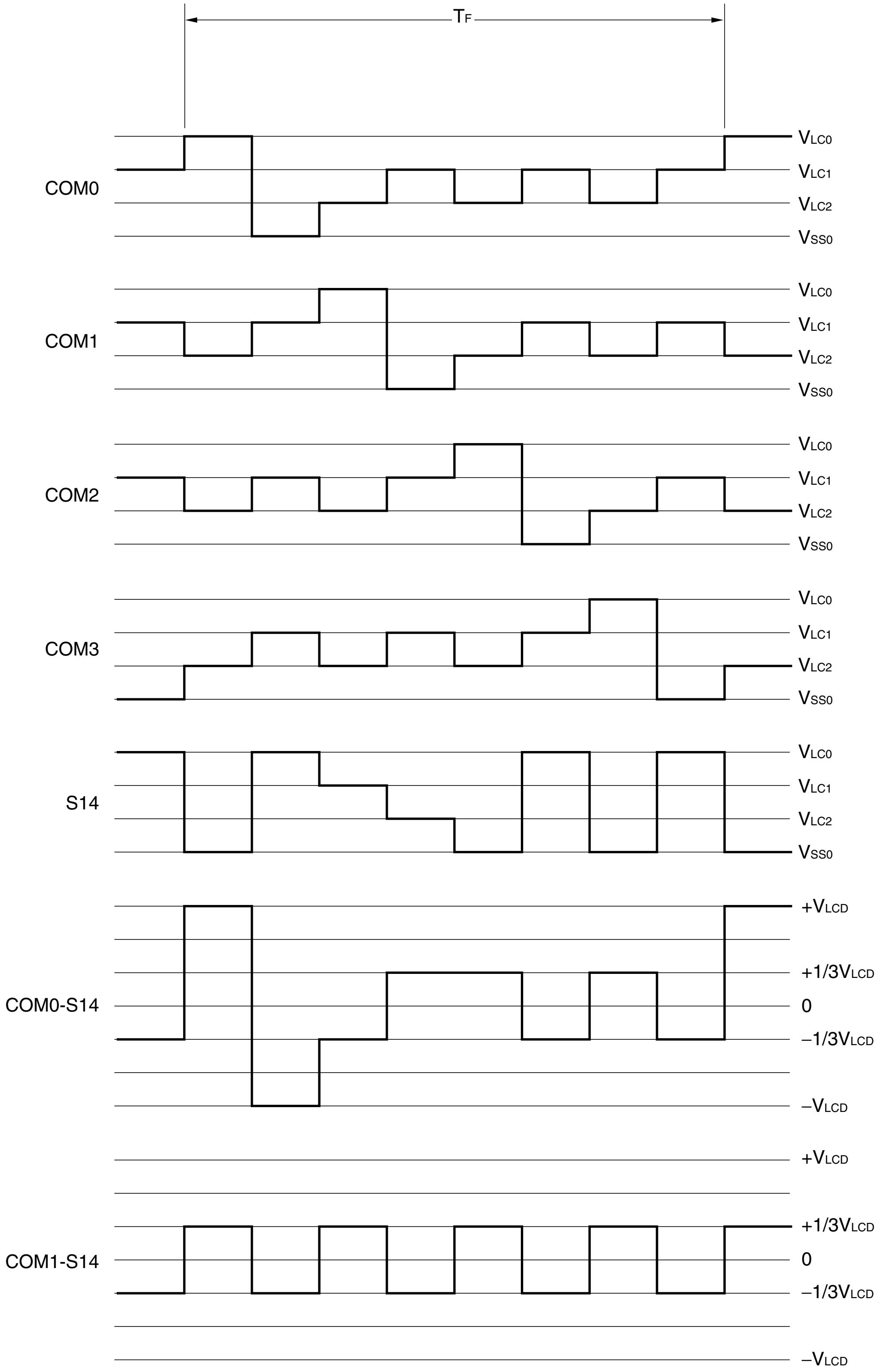
Remark n = 0 to 12

Figure 10-14. Example of Connecting Four-Time Slot LCD Panel



Data memory address	Bit 0	Bit 1	Bit 2	Bit 3	
FA00H	1	1	1	0	S 0
1	0	1	0	1	S 1
2	1	1	0	0	S 2
3	0	1	1	1	S 3
4	0	1	1	0	S 4
5	0	0	0	0	S 5
6	1	1	1	0	S 6
7	1	0	1	1	S 7
8	1	1	1	0	S 8
9	1	1	0	0	S 9
A	1	1	1	0	S 10
B	1	1	1	1	S 11
C	1	1	1	0	S 12
D	1	0	0	0	S 13
E	1	0	1	1	S 14
F	1	1	1	1	S 15
FA10H	1	0	1	0	S 16
1	1	1	0	1	S 17
2	0	1	1	0	S 18
3	1	1	0	0	S 19
4	1	1	1	0	S 20
5	0	1	0	1	S 21
6	1	1	0	0	S 22
7	0	1	1	1	S 23
8	0	1	1	0	S 24
9	0	0	0	0	S 25

Figure 10-15. Four-Time Slot LCD Drive Waveform Examples (1/3 Bias Mode)



Remark The waveforms of COM2-S14 and COM3-S14 are omitted.

CHAPTER 11 MULTIPLIER

11.1 Multiplier Function

The multiplier has the following function.

- Calculation of 8 bits × 8 bits = 16 bits

11.2 Multiplier Configuration

(1) 16-bit multiplication result storage register 0 (MUL0)

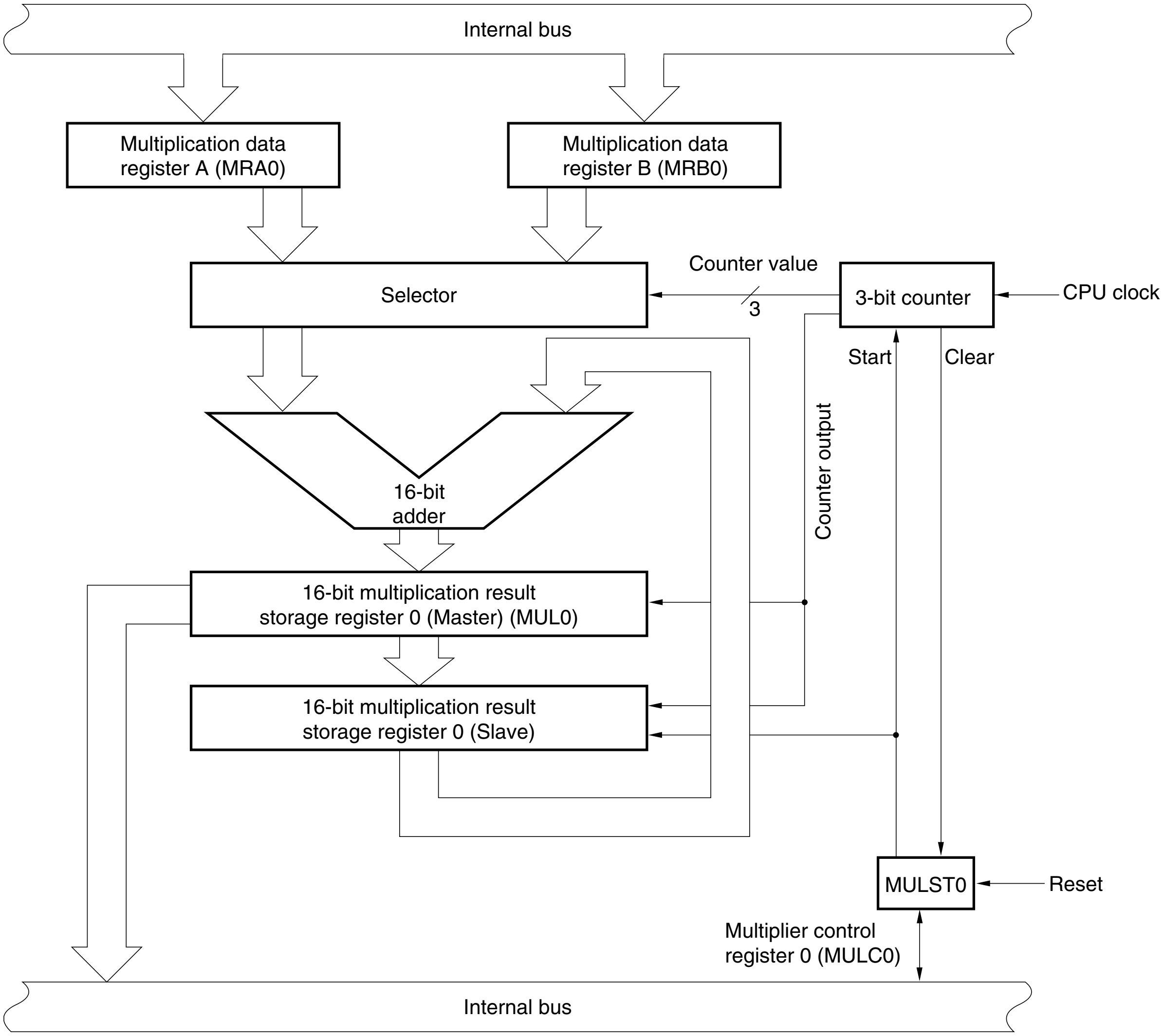
This register stores the 16-bit result of multiplication.
This register holds the result of multiplication after 16 CPU clocks have elapsed.
MUL0 can be read by a 16-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input makes MUL0 undefined.

Caution Although this register is manipulated with a 16-bit memory manipulation instruction, it can be also manipulated with an 8-bit memory manipulation instruction. When using an 8-bit memory manipulation instruction, however, access the register by means of direct addressing.

(2) Multiplication data registers A and B (MRA0 and MRB0)

These are 8-bit multiplication data storage registers. The multiplier multiplies the values of MRA0 and MRB0.
MRA0 and MRB0 can be written by an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input makes these registers undefined.
Figure 11-1 shows the block diagram of the multiplier.

Figure 11-1. Block Diagram of Multiplier


Internal bus
Multiplication data register A (MRA0)
Multiplication data register B (MRB0)
Counter value
Selector
3
3-bit counter
CPU clock
Start
Clear
Counter output
16-bit adder
16-bit multiplication result storage register 0 (Master) (MUL0)
16-bit multiplication result storage register 0 (Slave)
MULST0
Reset
Multiplier control register 0 (MULC0)
Internal bus

11.3 Multiplier Control Register

The multiplier is controlled by the following register.

- Multiplier control register 0 (MULC0)

(1) Multiplier control register 0 (MULC0)

This register indicates the operating status of the multiplier after operation, as well as controls the multiplier.
MULC0 can be set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears MULC0 to 00H.

Figure 11-2. Format of Multiplier Control Register 0

Symbol	7	6	5	4	3	2	1	<0>	Address	After reset	R/W
MULC0	0	0	0	0	0	0	0	MULST0	FFD2H	00H	R/W

MULST0	Multiplier operation start control bit	Operating status of multiplier
0	Stops operation after resetting counter to 0.	Operation stops
1	Enables operation	Operation in progress

Caution Be sure to clear bits 1 to 7 to 0.

11.4 Multiplier Operation

The multiplier of the μPD789881 Subseries can execute the calculation of 8 bits × 8 bits = 16 bits. Figure 11-3 shows the operation timing of the multiplier where MRA0 is set to AAH and MRB0 is set to D3H.

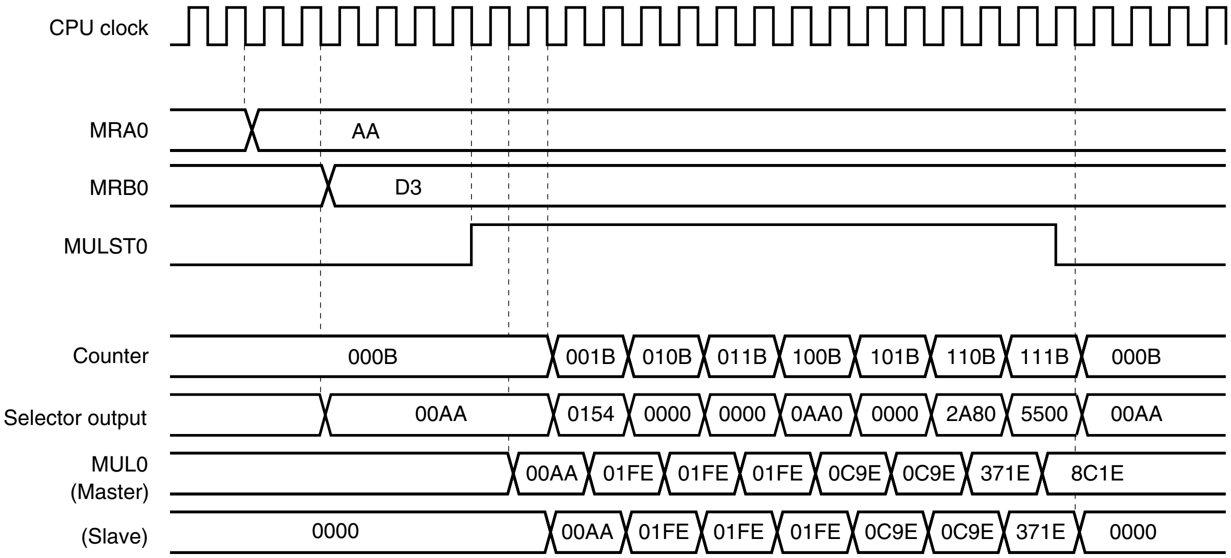
<1> Counting is started by setting MULST0.

<2> The data generated by the selector is added to the data of MUL0 at each CPU clock, and the counter value is incremented by one.

<3> If MULST0 is cleared when the counter value is 111B, the operation is stopped. At this time, MUL0 holds the data.

<4> While MULST0 is low, the counter and slave are cleared.

Figure 11-3. Multiplier Operation Timing (Example of AAH × D3H)



CHAPTER 12 VOLTAGE HALVER CIRCUIT AND REGULATOR CIRCUIT

The μPD789881 Subseries has a voltage halver circuit[Note] and a regulator circuit in its power supply block. These circuits allow the device to operate on an internal voltage lower than V_{DD} and to realize ultra-low power consumption.

Note The voltage halver circuit is not provided in the μPD78F9882.

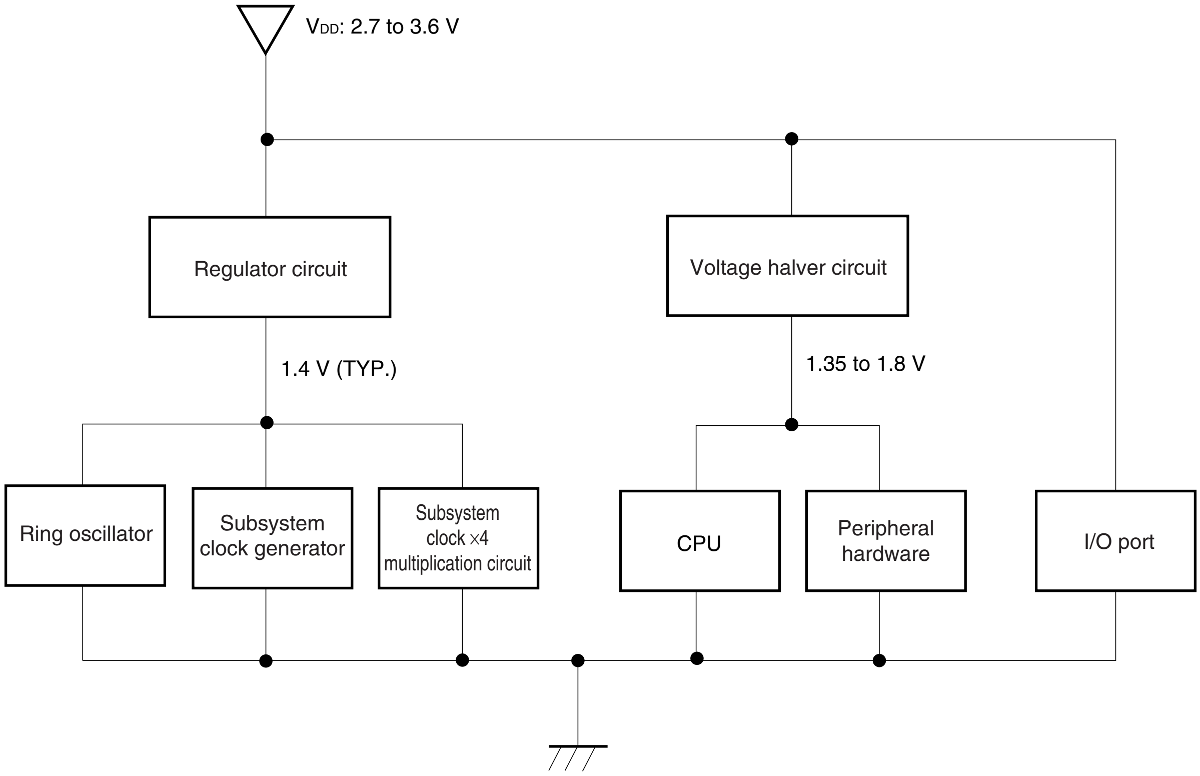
(1) Voltage halver circuit

This circuit internally lowers the V_{DD} voltage to 1/2 (V_{DD}/2) and supplies it to the CPU and peripheral hardware.

(2) Regulator circuit

This circuit internally drops the V_{DD} voltage to 1.4 V (TYP.) and supplies it to the main system clock generator (ring oscillator), subsystem clock generator, and subsystem clock ×4 multiplication circuit.

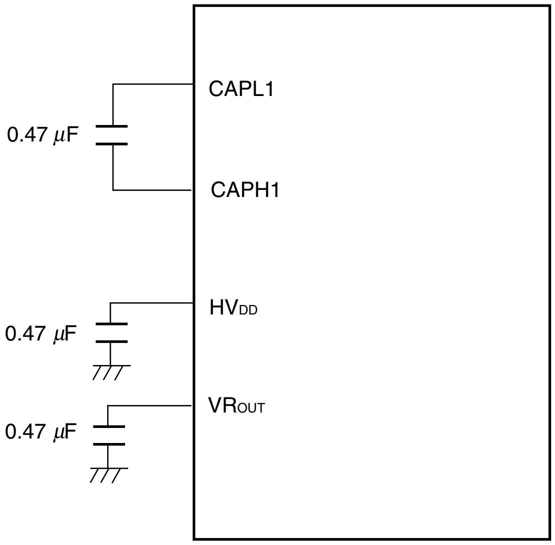
Figure 12-1. Block Diagram of Power Supply Block



Connect 0.47 μF capacitors to the pins of the power supply block (CAPL1, CAPH1, HV_{DD}, and VR_{OUT} pins).

Figure 12-2 illustrates how to connect the capacitors.

Figure 12-2. Connecting Capacitors



CHAPTER 13 INTERRUPT FUNCTIONS

13.1 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 13-1.

A standby release signal is generated.

4 external and 8 internal interrupt sources are incorporated as maskable interrupts.

13.2 Interrupt Sources and Configuration

A total of 13 non-maskable and maskable interrupts are incorporated as interrupt sources (see **Table 13-1**).

Table 13-1. Interrupt Source

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[Note 3]	(A)
Maskable	0	INTWDT	Watchdog timer overflow (with interval timer mode selected)			(B)
	1	INTP0	Pin (INTP0) input edge detection	External	0006H	(C)
	2	INTP1	Pin (INTP1) input edge detection		0008H	
	3	INTP2	Pin (INTP2) input edge detection		000AH	
	4	INTP3	Pin (INTP3) input edge detection		000CH	
	5	INTSRE0	End of UART reception error	Internal	000EH	(B)
	6	INTSR0	UART reception completion		0010H	
	7	INTST0	UART transmission completion		0012H	
	8	INTTM50	Match between TM50 and CR50		0014H	
	9	INTTM51	Match between TM51 and CR51		0016H	
	10	INTTM00	• Match between TM0 and CR00 (when CR00 is specified as compare register) • TI01 pin valid edge detection (when CR00 is specified as capture register)		0018H	
	11	INTTM01	• Match between TM0 and CR01 (when CR01 is specified as compare register) • TI00 pin valid edge detection (when CR01 is specified as capture register)		001AH	

Notes 1. Default priority is the priority order when more than one maskable interrupt request is generated at the same time. 0 is the highest priority and 11 is the lowest.

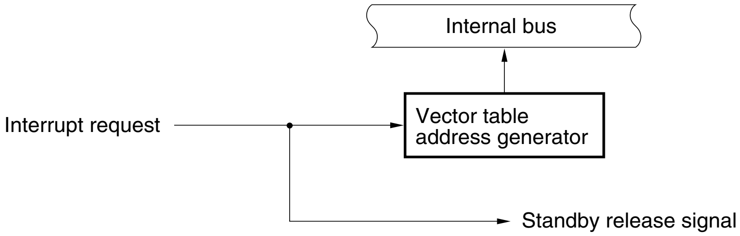
2. Basic configuration types (A), (B), and (C) correspond to (A), (B), and (C) in Figure 13-1.

3. The vector address of INTWDT is 0002H in the development tool (ICE) and 0004H in the device. Set both 0002H and 0004H as the interrupt servicing branch address.

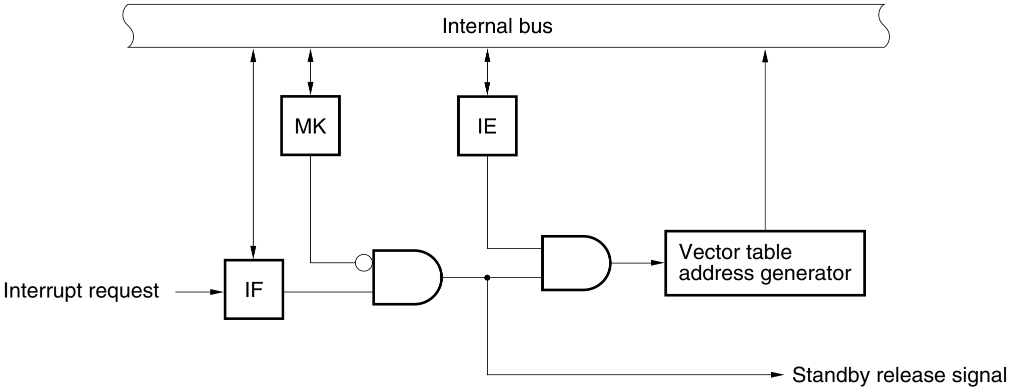
Remark Only one of the two watchdog timer interrupt (INTWDT) sources, non-maskable or maskable (internal), can be selected.

Figure 13-1. Basic Configuration of Interrupt Function

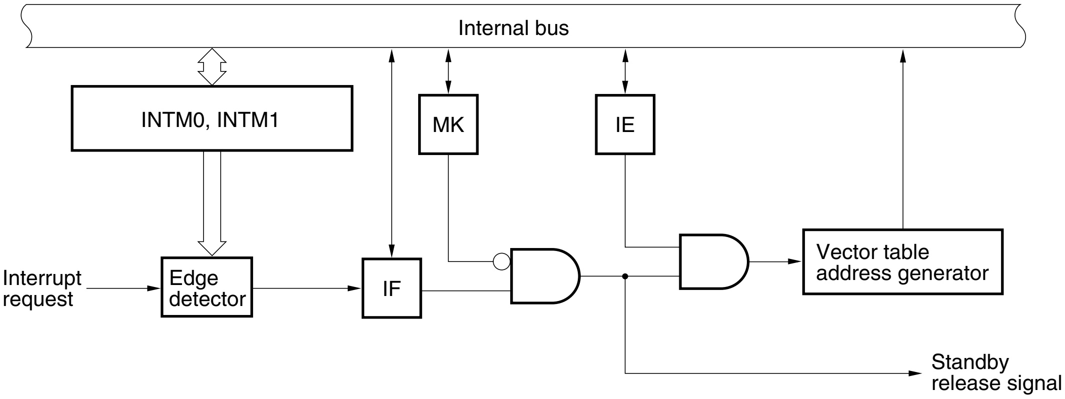
(A) Internal non-maskable interrupt



(B) Internal maskable interrupt



(C) External maskable interrupt



INTM0:	External interrupt mode register 0
INTM1:	External interrupt mode register 1
IF:	Interrupt request flag
IE:	Interrupt enable flag
MK:	Interrupt mask flag

13.3 Registers Controlling Interrupt Function

The following four types of registers are used to control the interrupt functions.

- Interrupt request flag registers (IF0 and IF1)
- Interrupt mask flag registers (MK0 and MK1)
- External interrupt mode registers (INTM0 and INTM1)
- Program status word (PSW)

Table 13-2 gives a listing of interrupt request flag and interrupt mask flag names corresponding to interrupt requests.

Table 13-2. 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
INTSRE0	SREIF0	SREMK0
INTSR0	SRIF0	SRMK0
INTST0	STIF0	STMK0
INTTM50	TMIF50	TMMK50
INTTM51	TMIF51	TMMK51
INTTM00	TMIF00	TMMK00
INTTM01	TMIF01	TMMK01

(1) Interrupt request flag registers (IF0, IF1)

An interrupt request flag is set to 1 when the corresponding interrupt request is generated, or when an instruction is executed. It is cleared to 0 when the interrupt request is acknowledged, when the $\overline{\text{RESET}}$ signal is input, or when an instruction is executed.

IF0 and IF1 can be set by a 1-bit or 8-bit memory manipulation instruction.

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

Figure 13-2. Format of Interrupt Request Flag Registers

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
IF0	STIF0	SRIF0	SREIF0	PIF3	PIF2	PIF1	PIF0	WDTIF	FFE0H	00H	R/W

Symbol	7	6	5	4	<3>	<2>	<1>	<0>	Address	After reset	R/W
IF1	0	0	0	0	TMIF01	TMIF00	TMIF51	TMIF50	FFE1H	00H	R/W

××IF×	Interrupt request flag
0	No interrupt request signal generated
1	An interrupt request signal is generated and an interrupt request made

Cautions 1. The WDTIF flag can be read/written only when the watchdog timer is being used as an interval timer. It must be cleared to 0 if the watchdog timer is used in watchdog timer mode 1 or 2.

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

3. When an interrupt is acknowledged, the interrupt request flag is automatically cleared and then the interrupt routine is entered.

(2) Interrupt mask flag registers (MK0, MK1)

Interrupt mask flags are used to enable and disable the corresponding maskable interrupts.

MK0 and MK1 can be set by a 1-bit or 8-bit memory manipulation instruction.

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

Figure 13-3. Format of Interrupt Mask Flag Registers

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
MK0	STMK0	SRMK0	SREMK0	PMK3	PMK2	PMK1	PMK0	WDTMK	FFE4H	FFH	R/W

Symbol	7	6	5	4	<3>	<2>	<1>	<0>	Address	After reset	R/W
MK1	1	1	1	1	TMMK01	TMMK00	TMMK51	TMMK50	FFE5H	FFH	R/W

××MK	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Cautions 1. When the watchdog timer is being used in watchdog timer mode 1 or 2, any attempt to read the WDTMK flag results in an undefined value being detected.

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

(3) External interrupt mode registers (INTM0, INTM1)

These registers are used to specify the valid edge for INTP0 to INTP3.

INTM0 and INTM1 can be set by an 8-bit memory manipulation instruction.

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

Figure 13-4. Format of External Interrupt Mode Registers

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
INTM0	ES21	ES20	ES11	ES10	ES01	ES00	0	0	FFECH	00H	R/W

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
INTM1	0	0	0	0	0	0	ES31	ES30	FFEDH	00H	R/W

ESn1	ESn0	INTPn valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

Remark n = 0 to 3

Cautions 1. Be sure to clear bits 0 and 1 of INTM0, and 2 to 7 of INTM1 to 0.

2. Before setting INTM0 and INTM1, set the interrupt mask flags (PMK0 to PMK3) to 1 to disable interrupts.

To enable interrupts, clear the interrupt request flags (PIF0 to PIF3) to 0, then clear the interrupt mask flags (PMK0 to PMK3) to 0.

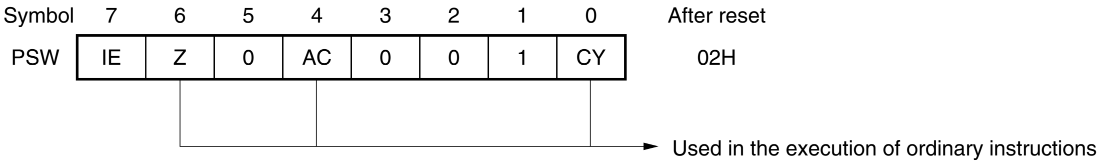
(4) Program status word (PSW)

The program status word is used to hold the instruction execution results and the current status of the interrupt requests. The IE flag, used to enable and disable maskable interrupts, is mapped to the PSW.

PSW can be read and written in 8-bit units, and can be manipulated by using bit manipulation instructions and dedicated instructions (EI and DI). When a vectored interrupt is acknowledged, the PSW is automatically saved to the stack, and the IE flag is reset to 0.

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

Figure 13-5. Program Status Word Configuration



Symbol	7	6	5	4	3	2	1	0	After reset
PSW	IE	Z	0	AC	0	0	1	CY	02H

IE	Interrupt acknowledgment enable/disable
0	Disabled
1	Enabled

13.4 Interrupt Servicing Operation

13.4.1 Non-maskable interrupt request acknowledgment operation

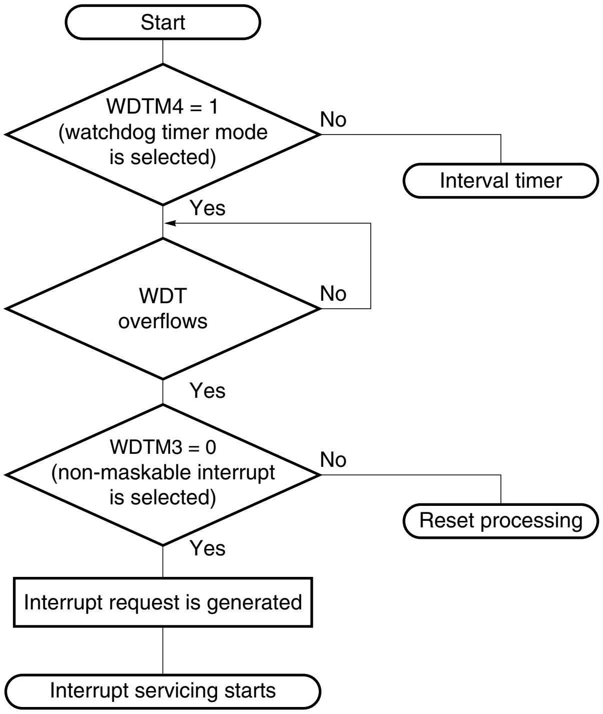
The non-maskable interrupt request is unconditionally acknowledged even when interrupts are disabled. It is not subject to interrupt priority control and takes precedence over all other interrupts.

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 PC, and then program execution branches.

Figure 13-6 shows the flow from non-maskable interrupt request generation to acknowledgment, Figure 13-7 shows the timing of non-maskable interrupt acknowledgment, and Figure 13-8 shows the acknowledgment operation when a number of non-maskable interrupts are generated.

Caution During non-maskable interrupt service program execution, do not input another non-maskable interrupt request; if it is input, the service program will be interrupted and the new non-maskable interrupt request will be acknowledged.

Figure 13-6. Flow from Generation of Non-Maskable Interrupt Request to Acknowledgment



WDTM: Watchdog timer mode register
WDT: Watchdog timer

Figure 13-7. Timing of Non-Maskable Interrupt Request Acknowledgment

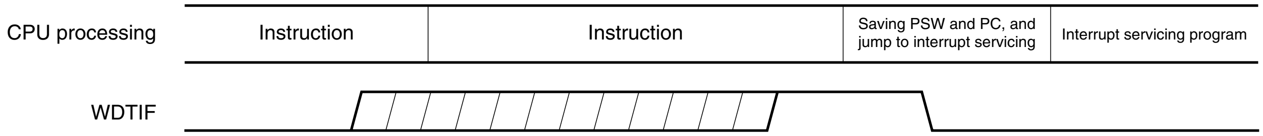
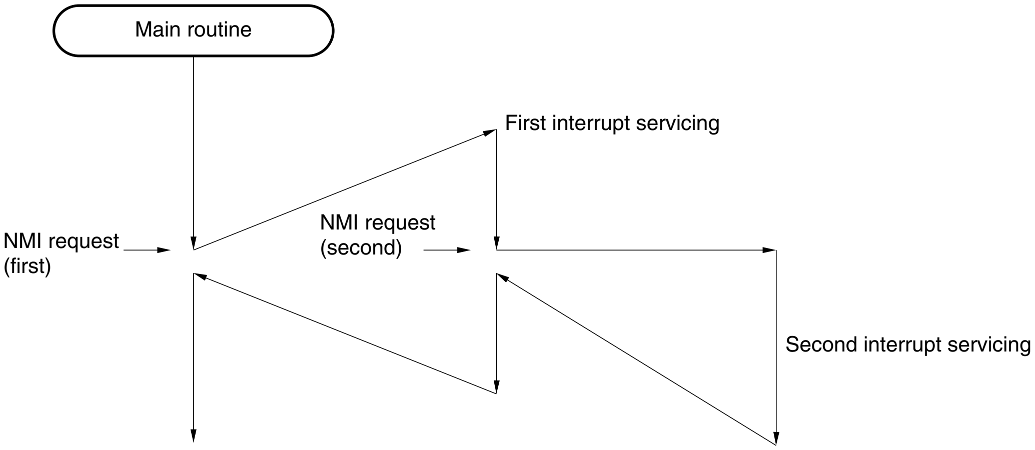


Figure 13-8. Non-Maskable Interrupt Request Acknowledgment



13.4.2 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 13-3.

Refer to Figures 13-10 and 13-11 for the timing of interrupt request acknowledgment.

Table 13-3. 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.

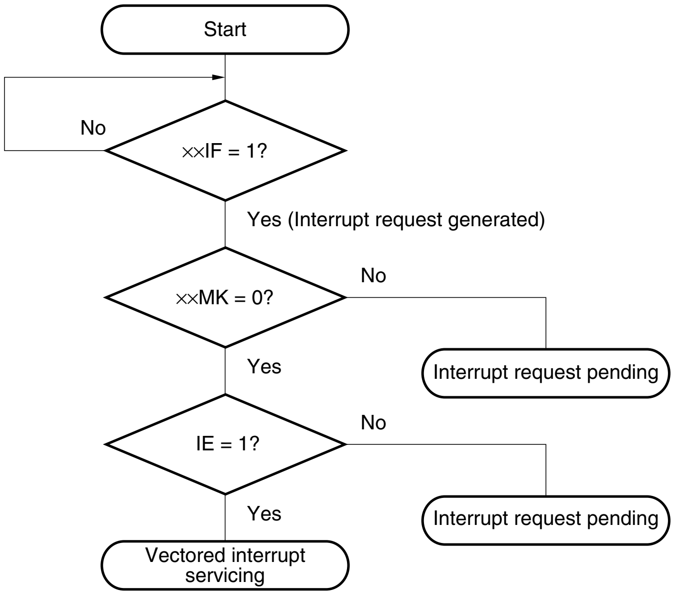
A pending interrupt is acknowledged when the status where it can be acknowledged is set.

Figure 13-9 shows the algorithm of interrupt request acknowledgment.

When a maskable interrupt request is acknowledged, 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.

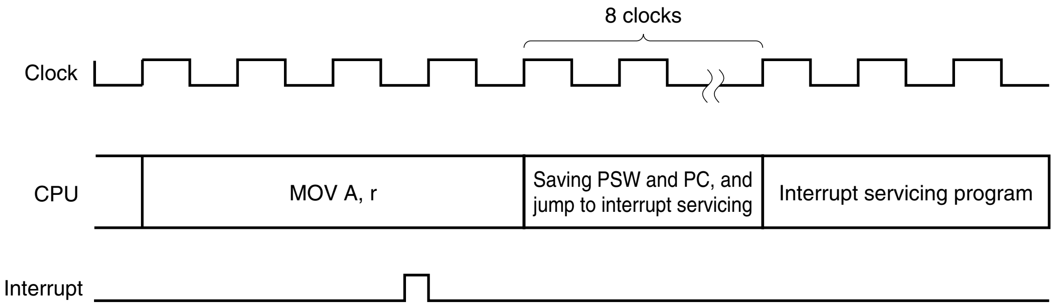
To return from interrupt servicing, use the RETI instruction.

Figure 13-9. Interrupt Request Acknowledgment Program Algorithm


Start
No
××IF = 1?
Yes (Interrupt request generated)
No
××MK = 0?
Yes
Interrupt request pending
No
IE = 1?
Yes
Interrupt request pending
Vectored interrupt servicing

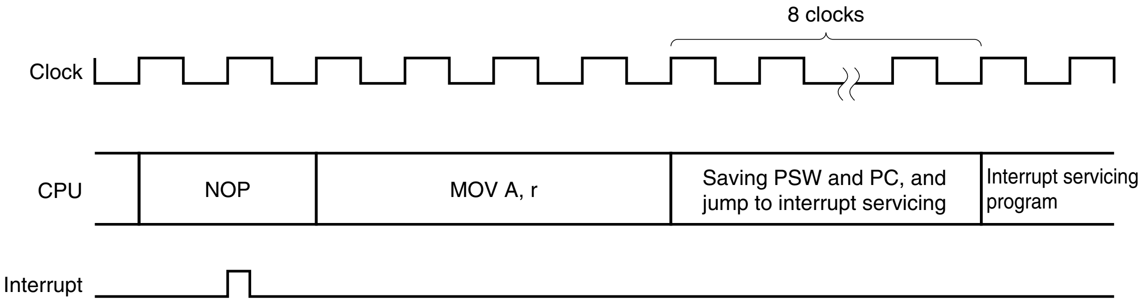
××IF:	Interrupt request flag
××MK:	Interrupt mask flag
IE:	Flag to control maskable interrupt request acknowledgment (1 = enable, 0 = disable)

Figure 13-10. 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 (××IF) by the time the instruction clocks under execution, n clocks (n = 4 to 10), are n – 1, interrupt request acknowledgment processing will start following the completion of the instruction under execution. Figure 13-10 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 13-11. 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 (××IF) is generated in the final clock of the instruction, interrupt request acknowledgment processing will begin after execution of the next instruction is complete.

Figure 13-11 shows an example whereby an interrupt request was generated in the 2nd clock of NOP (a 2-clock instruction). In this case, the interrupt request will be processed after execution of MOV A, r, which follows NOP, is complete.

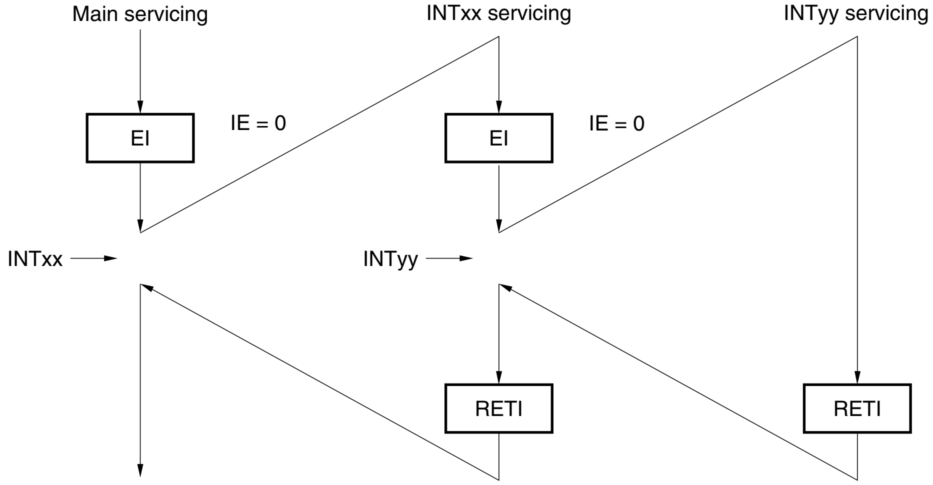
Caution When interrupt request flag registers (IF0 and IF1), or interrupt mask flag registers (MK0 and MK1) are being accessed, interrupt requests will be held pending.

13.4.3 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 13-1**).

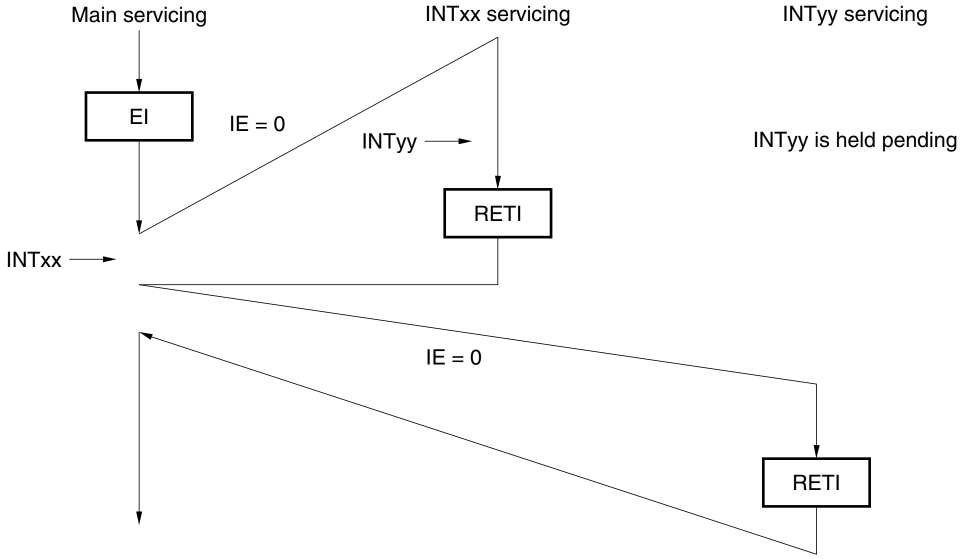
Figure 13-12. Example of Multiple Interrupts

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

13.4.4 Putting interrupt requests on hold

If an interrupt request (such as a maskable, non-maskable, or external interrupt) is generated when a certain type of instruction is being executed, the interrupt request will not be acknowledged until the instruction is completed. Such instructions (interrupt request pending instructions) are as follows.

- Instructions that manipulate interrupt request flag registers (IF0 and IF1)
- Instructions that manipulate interrupt mask flag registers (MK0 and MK1)

CHAPTER 14 STANDBY FUNCTION

14.1 Standby Function and Configuration

The standby function is to reduce the power consumption of the system and the μPD789881 Subseries supports only the HALT mode.

(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. It is also possible to set the HALT mode while the main system clock is stopped (refer to **CHAPTER 5 CLOCK GENERATOR**). This is useful for resuming processing immediately when an interrupt request is generated, or for intermittent operations.

The previous contents of the registers, flags, and data memory before setting the HALT mode are all retained. In addition, the statuses of the output latch of the I/O ports and output buffer are also retained.

Caution The STOP mode is not supported. Therefore, do not execute the STOP instruction.

14.2 Standby Function Operation

14.2.1 HALT mode setting and operation status

The HALT mode is set by executing the HALT instruction.

The operation status in the HALT mode is shown in the following table.

Table 14-1. Operation Statuses in HALT Mode

Item	HALT Mode Operation Status
Clock generator	Oscillation enabled. Clock supply to CPU is stopped
Subsystem clock ×4 multiplication circuit	Operation stopped
CPU	Operation stopped
Ports (output latches)	Status before HALT mode setting retained
16-bit timer/event counter 0	Operable[Notes 1, 2]
8-bit timer/event counters 50 and 51	Operable[Note 1]
Watchdog timer	Operable
Serial interface UART0	Operable[Note 1]
Multiplier	Operable
LCD controller/driver	Operable
External interrupts	Operable[Note 3]

Notes 1. Because the ×4 multiplication clock circuit stops operation, 4f_{XT} cannot be selected as the count clock.

2. When the valid edge of TI00 is selected as the count clock, this timer/event counter can only be used when the main system clock is operating (MCC = 0).

3. Maskable interrupt that is not masked

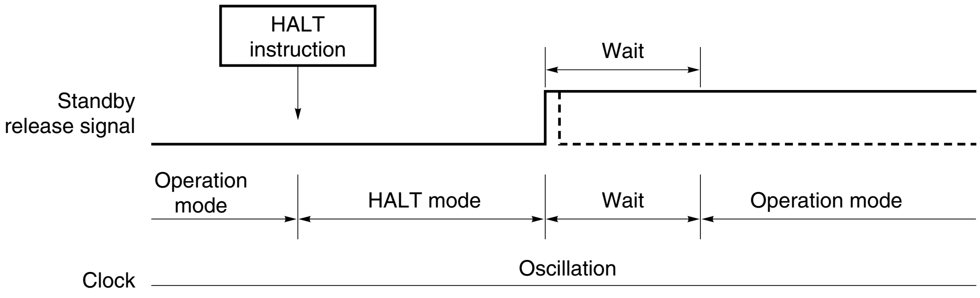
14.2.2 Releasing HALT mode

The HALT mode can be released by the following three types of sources.

(1) Releasing by unmasked interrupt request

When an unmasked interrupt request is generated, the HALT mode is released. If the interrupt is enabled to be acknowledged, vectored interrupt servicing is performed. If the interrupt is disabled, the instruction at the next address is executed.

Figure 14-1. 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 servicing is performed: 9 to 10 clocks
- When vectored interrupt servicing is not performed: 1 to 2 clocks

(2) Releasing by non-maskable interrupt request

When a non-maskable interrupt request is generated, the HALT mode is released regardless of whether the interrupt is enabled or disabled, and vectored interrupt servicing is performed.

(3) Releasing by $\overline{\text{RESET}}$ input

When the $\overline{\text{RESET}}$ signal is input, the HALT mode is released, and then, execution branches to the reset vector address in the same manner as the ordinary reset operation, and program execution is started.

Figure 14-2. Releasing HALT Mode by $\overline{\text{RESET}}$ Input

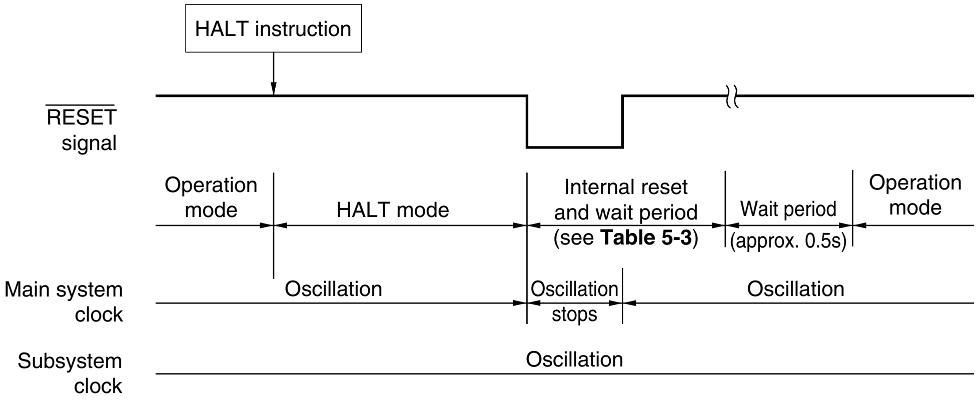


Table 14-2. Operation After Releasing HALT Mode

Releasing Source	MK××	IE	Operation
Maskable interrupt request	0	0	Executes next address instruction
	0	1	Executes interrupt servicing
	1	×	Retains HALT mode
Non-maskable interrupt request	–	×	Executes interrupt servicing
$\overline{\text{RESET}}$ input	–	–	Reset processing

×: don't care

CHAPTER 15 RESET FUNCTION

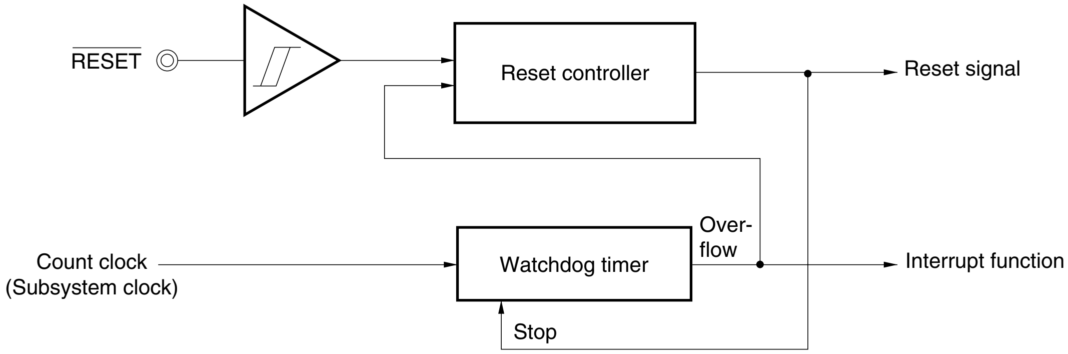
15.1 Reset Types

The following two operations are available to generate reset signals.

(1) External reset input by $\overline{\text{RESET}}$ pin

(2) Internal reset by watchdog timer program loop time detection

Figure 15-1. Block Diagram of Reset Function



15.2 Reset Processing Operation

There is no functional difference between an external reset and an internal reset. Both the reset operations start execution of the program from the address written to addresses 0000H and 0001H when the reset signal is input. Upon reset, each hardware is in the status shown in Table 15-1. While the reset signal is being input and during the regulator stabilization time immediately after the reset signal has been released, the port pins go into a high-impedance state and the subsystem clock continues oscillating.

Caution After reset, the CPU operates in the low-speed mode (16 μs: at 500 kHz (TYP.) operation) of the main system clock. Change the mode to high-speed (4 μs: at 500 kHz (TYP.) operation) by changing bit 1 (PCC1) of the processor clock control register (PCC) at the initial stage of the user program.

(1) External reset input by $\overline{\text{RESET}}$ pin

The system is reset when a low level is input to the $\overline{\text{RESET}}$ pin. If a high level is later input to the $\overline{\text{RESET}}$ pin, internal reset wait time (refer to **Table 5-3**) automatically elapses and the system is released from the reset status. The program execution is started after a wait time (approx. 0.5 s), during which the internal regulator is automatically stabilized (refer to **Figure 15-2**).

In this case, the main system clock stops oscillating during the reset period. It continues oscillating during the internal reset wait period and during the regulator stabilization time immediately after the reset signal has been released.

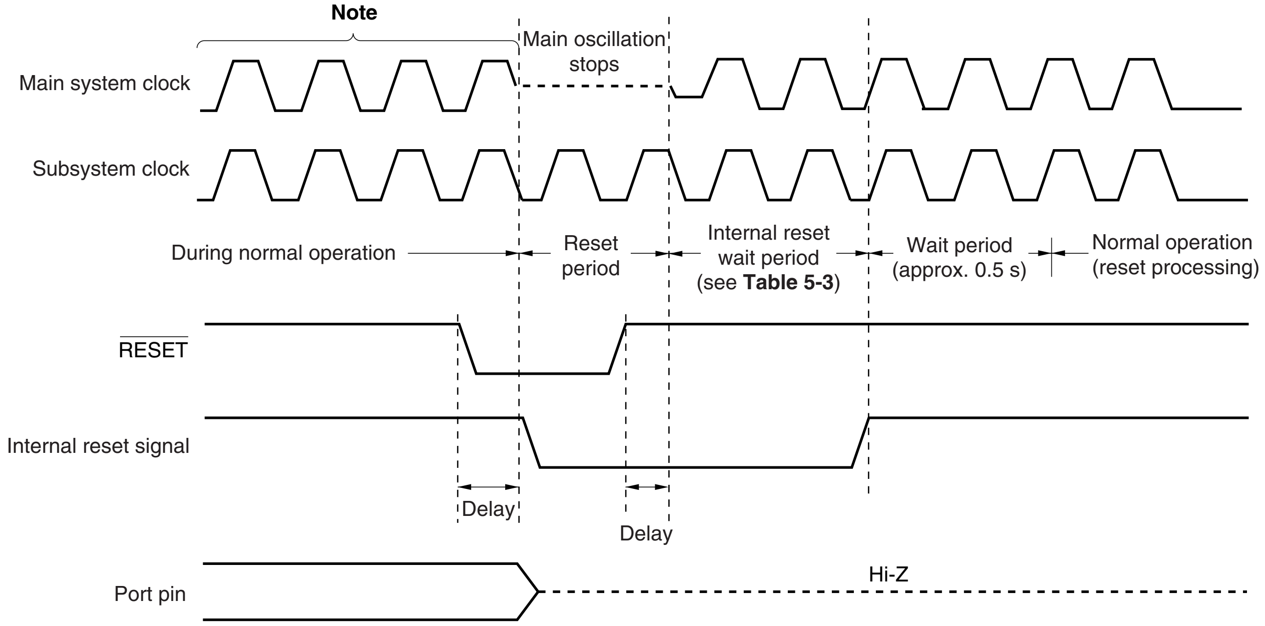
Caution To execute an external reset, input a low level of 10 μs or more to the $\overline{\text{RESET}}$ pin.

(2) Internal reset by detecting program loop time with watchdog timer

When the watchdog timer overflows, an internal reset signal is generated that resets the system. The internal reset signal automatically lasts for the internal reset wait time (refer to **Table 5-3**), and then the reset signal is released. Program execution is started after the lapse of the wait time (approx. 0.5 s), during which the internal regulator is stabilized (refer to **Figure 15-3**).

In this case, the main system clock continues to be in the status before reset during the reset period (oscillates when MCC = 0 and stops when MCC = 1), and oscillates during the internal reset wait period and regulator stabilization time immediately after the reset signal has been released.

Figure 15-2. Timing of Reset by RESET Input

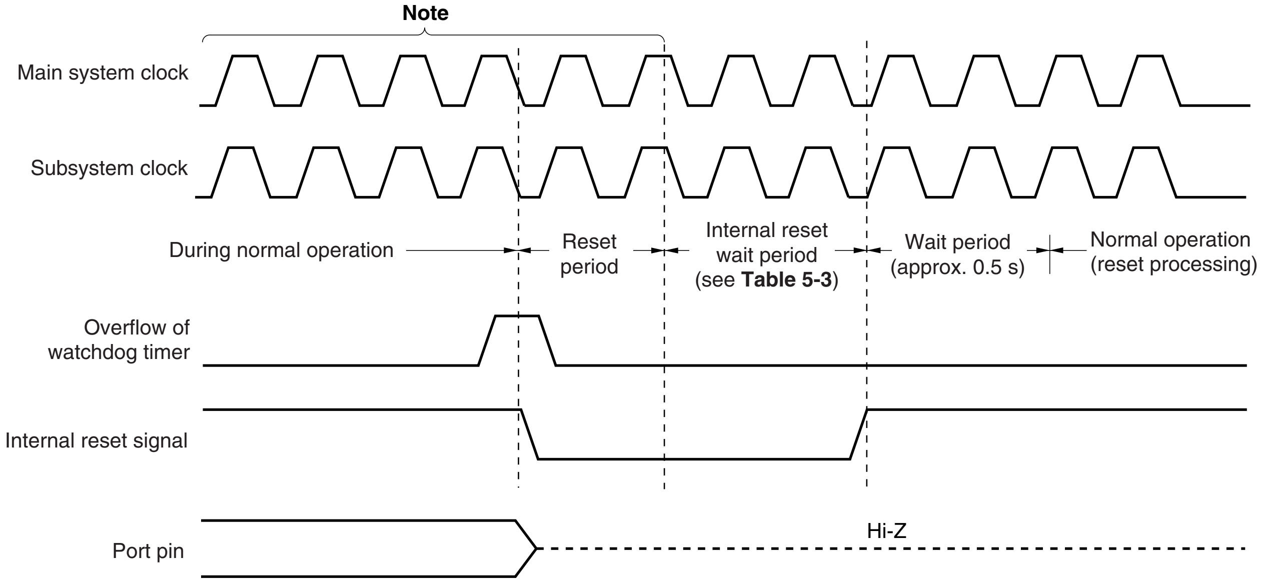


Note The main system clock before reset oscillates when MCC = 0, and stops when MCC = 1.

Remarks 1. The main system clock always stops during the reset period.
It always oscillates during the internal reset wait period and after reset (because the default value is MCC = 0).

2. The subsystem clock always continues oscillating as long as power is supplied (not affected by reset).

Figure 15-3. Timing of Reset by Overflow of Watchdog Timer


Note
Main system clock
Subsystem clock
During normal operation
Reset period
Internal reset wait period (see **Table 5-3**)
Wait period (approx. 0.5 s)
Normal operation (reset processing)
Overflow of watchdog timer
Internal reset signal
Port pin
Hi-Z

Note The main system clock before reset and during the reset period oscillates when MCC = 0 and stops when MCC = 1.

Remarks 1. The main system clock always oscillates during the internal reset wait period and after reset (because the default value is MCC = 0).

2. The subsystem clock always continues oscillating as long as power is supplied (not affected by reset).

Figure 15-4. Timing of Reset by $\overline{\text{RESET}}$ Input on Power Application

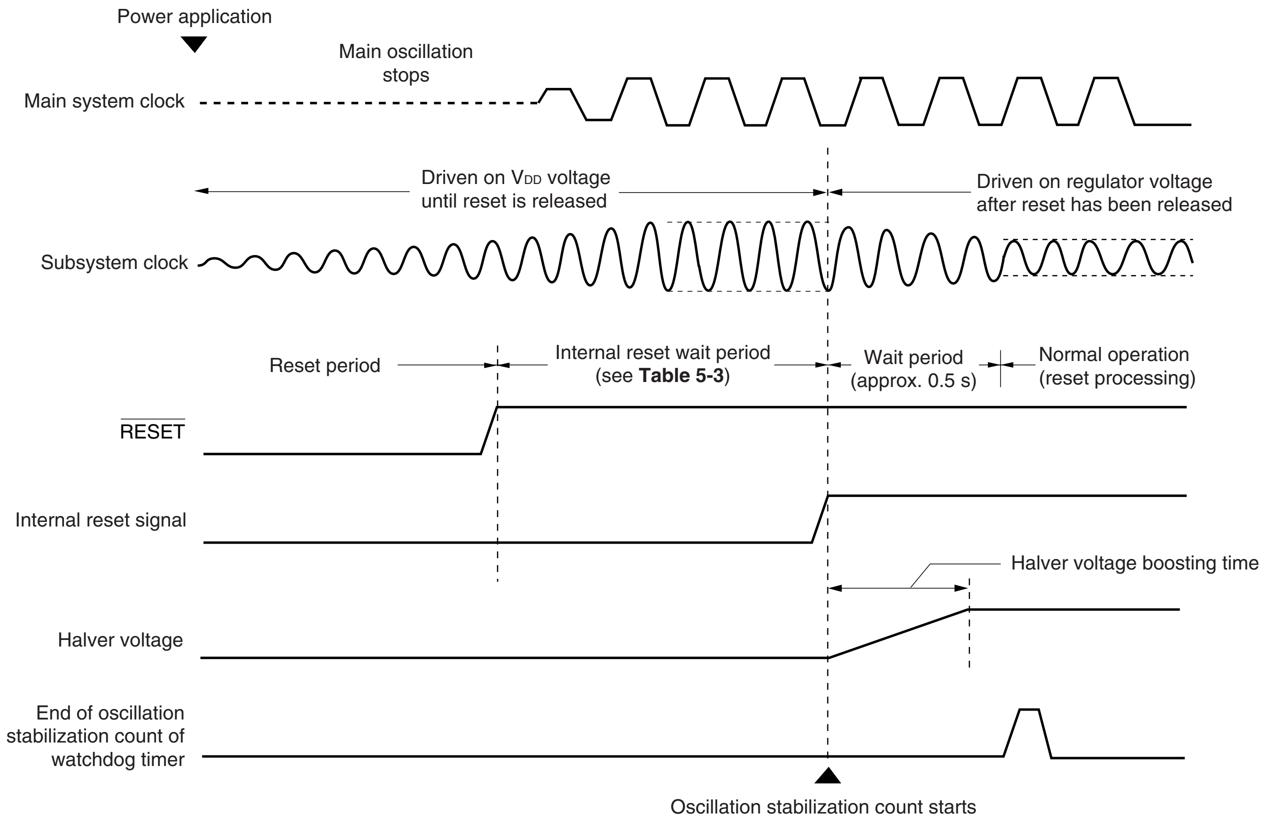

Power application
Main oscillation stops
Main system clock
Driven on VDD voltage until reset is released
Driven on regulator voltage after reset has been released
Subsystem clock
Reset period
Internal reset wait period (see Table 5-3)
Wait period (approx. 0.5 s)
Normal operation (reset processing)
RESET
Internal reset signal
Halver voltage boosting time
Halver voltage
End of oscillation stabilization count of watchdog timer
Oscillation stabilization count starts

Table 15-1. Status of Hardware After Reset

Hardware		Status After Reset
Program counter (PC)[Note 1]		Contents of reset vector table (0000H, 0001H) set
Stack pointer (SP)		Undefined
Program status word (PSW)		02H
RAM	Data memory	Undefined[Note 2]
	General-purpose registers	Undefined[Note 2]
Ports (P0 to P3, P5, P8, P9) (output latches)		00H
Port mode registers (PM0 to PM3, PM5, PM8, PM9)		FFH
Port function registers (PF8, PF9)		00H
Pull-up resistor option registers (PUB0 to PUB3, PUB8, PUB9)		00H
Processor clock control register (PCC)		02H
Subclock control register (CSS)		00H
16-bit timer/event counter 0	Timer counter (TM0)	0000H
	Capture/compare registers (CR00, CR01)	0000H
	Capture/compare control register (CRC0)	00H
	Mode control register (TMC0)	00H
	Output control register (TOC0)	00H
	Prescaler mode register (PRM0)	00H
8-bit timer/event counters 50, 51	Timer counters (TM50, TM51)	00H
	Compare registers (CR50, CR51)	Undefined
	Mode control registers (TMC50, TMC51)	00H
	Timer clock select registers (TCL50, TCL51)	00H
Watchdog timer	Clock select register (WDCS)	00H
	Mode register (WDTM)	00H
UART	Transmit shift register (TXS0)	FFH
	Receive buffer register (RXB0)	FFH
	Asynchronous serial interface mode register (ASIM0)	01H
	Asynchronous serial interface status register (ASIS0)	00H
	Baud rate generator control register (BRGC0)	1FH
LCD controller/driver	Display mode register (LCDM0)	00H
	Clock control register (LCDC0)	00H
Multiplier	16-bit result storage register (MUL0)	Undefined
	Data registers (MRA0, MRB0)	Undefined
	Control register (MULC0)	00H
Interrupts	Request flag registers (IF0, IF1)	00H
	Mask flag registers (MK0, MK1)	FFH
	External interrupt mode registers (INTM0, INTM1)	00H

Notes 1. While a reset signal is being input, and during the oscillation stabilization period, only the contents of the PC will be undefined; the remainder of the hardware will be the same state as after reset.

2. In standby mode, RAM enters the hold state after reset.

CHAPTER 16 μPD78F9882

The μPD78F9882 is provided as the flash memory version of the μPD789881 Subseries.
The differences between the μPD78F9882 and the mask ROM version are shown in Table 16-1.

Table 16-1. Differences Between μPD78F9882 and Mask ROM Version

Item		Flash Memory Version	Mask ROM Version
		μPD78F9882	μPD789881
Internal memory	ROM	32 KB (flash memory)	16 KB
	High-speed RAM	512 bytes	
	LCD display RAM	26 × 4 bits	
Pull-up resistor of port 5		Not provided	Selectable by a mask option in 1-bit units
Voltage halver circuit		Not provided	Provided
IC0 pin		Not provided	Provided
V_{PP} pin		Provided	Not provided
Electrical specifications		Refer to **CHAPTER 19 ELECTRICAL SPECIFICATIONS**.	

Caution There are differences in current consumption, 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.
(The above parameters differ in parts between the development tool (ICE) and actual device. Also refer to the user's manual of the emulation board (IE-789882-NS-EM1).)

16.1 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 μPD78F9882 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 the products made by Naito Densei Machida Mfg. Co., Ltd. (TEL +81-45-475-4191).

Programming using flash memory has the following advantages.

- Software can be modified after the microcontroller is solder-mounted on the target system.
- Distinguishing software facilities low-quantity, varied model production
- Easy data adjustment when starting mass production

16.1.1 Programming environment

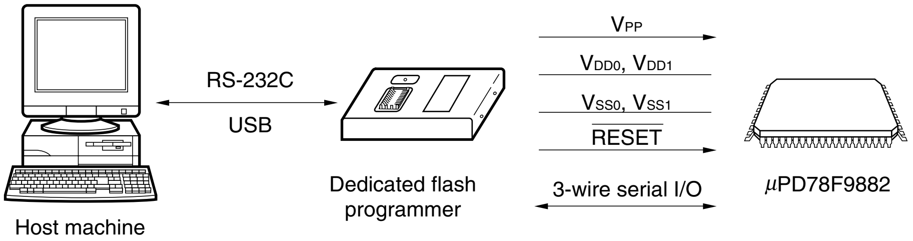
The following shows the environment required for μPD78F9882 flash memory programming.

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 the manuals for Flashpro III/Flashpro IV.

Remark USB is supported by Flashpro IV only.

Figure 16-1. Environment for Writing Program to Flash Memory



16.1.2 Communication mode

Use the communication mode shown in Table 16-2 to perform communication between the dedicated flash programmer and the μPD78F9882.

Table 16-2. Communication Mode List

Communication Mode	TYPE Setting[Note 1]					Pins Used	Number of V_{PP} Pulses
	COMM PORT	SIO Clock	CPU Clock	Flash Clock	Multiple Rate		
3-wire serial I/O	SIO ch-0 (3-wire, sync.)	100 Hz to 2 MHz[Note 2]	Optional	1 to 5 MHz	1.0	P22 (SI1) P21 (SO1) P20 (SCK1)	0

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. Use an appropriate clock after evaluation.

Figure 16-2. Communication Mode Selection Format

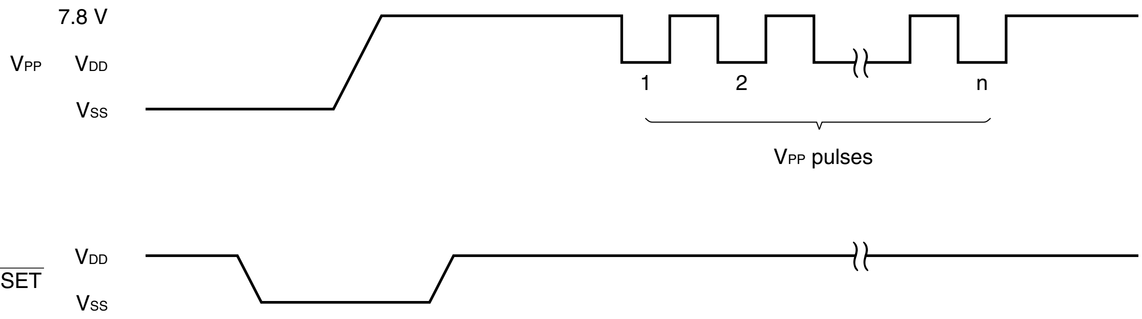
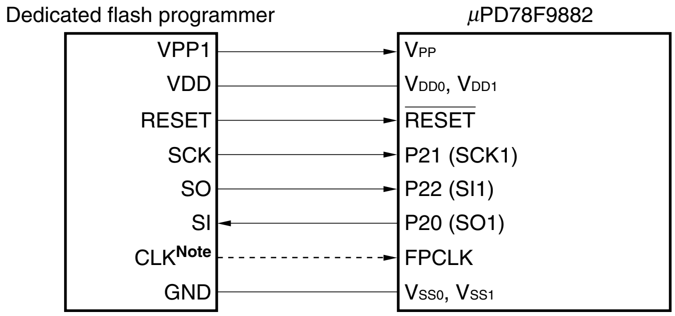


Figure 16-3. Example of Connection with Dedicated Flash Programmer (3-Wire Serial I/O)



Note Connect this pin when the system clock is supplied from the dedicated flash programmer. When the system clock is not supplied from the dedicated flash programmer, input an external clock of 1 to 5 MHz to the FPCLK pin.

Caution The V_{DD0} and V_{DD1} pins, if already connected to the external power supply, must be connected to the VDD pin of the dedicated flash programmer. Before using the external power supply connected to the V_{DD0} or V_{DD1} pin, supply voltage before starting programming.

If Flashpro III (part no. FL-PR3, PG-FP3)/Flashpro IV (part no. FL-PR4, PG-FP4) is used as a dedicated flash programmer, the following signals are generated for the μPD78F9882. For details, refer to the manual of Flashpro III/Flashpro IV.

Table 16-3. Pin Connection List

Signal Name	I/O	Pin Function	Pin Name	3-Wire Serial I/O
VPP1	Output	Write voltage	V_{PP}	◎
VPP2	–	–	–	×
VDD	I/O	V_{DD} voltage generation/ voltage monitoring	V_{DD0}, V_{DD1}	◎[Note]
GND	–	Ground	V_{SS0}, V_{SS1}	◎
CLK	Output	Clock output	FPCLK	○
RESET	Output	Reset signal	$\overline{\text{RESET}}$	◎
SI	Input	Receive signal	P20 (SO1)	◎
SO	Output	Transmit signal	P22 (SI1)	◎
SCK	Output	Transfer clock	P21 (SCK1)	◎
HS	Input	Handshake signal	–	×

Note V_{DD} voltage must be supplied before programming is started.

Remark ◎: Pin must be connected.
○: If the external clock is supplied, pin need not be connected.
×: Pin need not be connected.

16.1.3 On-board pin processing

When performing programming on the target system, provide a connector on the target system to connect the dedicated flash programmer.

An on-board function that allows switching between normal operation mode and flash memory programming mode may be required in some cases.

<V_{PP} pin>

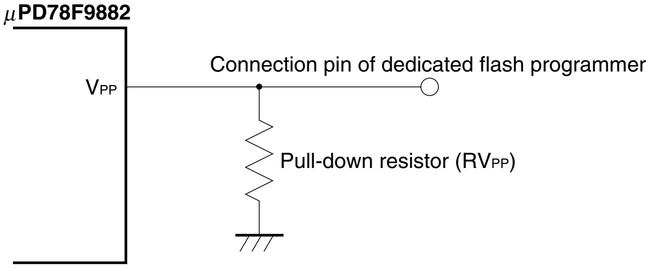
In normal operation mode, input 0 V to the V_{PP} pin. In flash memory programming mode, a write voltage of 7.8 V (TYP.) is supplied to the V_{PP} pin, so perform either of the following.

(1) Connect a pull-down resistor (RV_{PP} = 10 kΩ) to the V_{PP} pin.

(2) Use the jumper on the board to switch the V_{PP} pin input to either the writer or directly to GND.

A V_{PP} pin connection example is shown below.

Figure 16-4. V_{PP} Pin Connection Example


μPD78F9882
VPP
Connection pin of dedicated flash programmer
Pull-down resistor (RVPP)

<Serial interface pin>

The following shows the pins used by the serial interface.

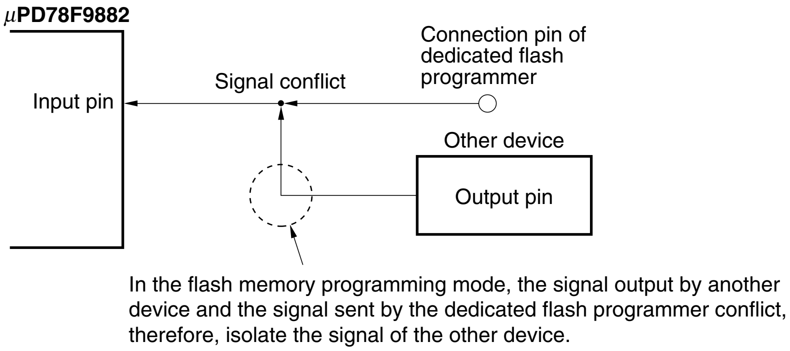
Serial Interface	Pins Used
3-wire serial I/O	P22 (SI1), P20 (SO1), P21 (SCK1)

When connecting the dedicated flash programmer to a serial interface pin that is connected to another device on-board, signal conflict or abnormal operation of the other devices may occur. Care must therefore be taken with such connections.

(1) Signal conflict

If the dedicated flash programmer (output) is connected to a serial interface pin (input) that is connected to another device (output), a signal conflict occurs. To prevent this, isolate the connection with the other device or set the other device to the output high impedance status.

Figure 16-5. Signal Conflict (Input Pin of Serial Interface)

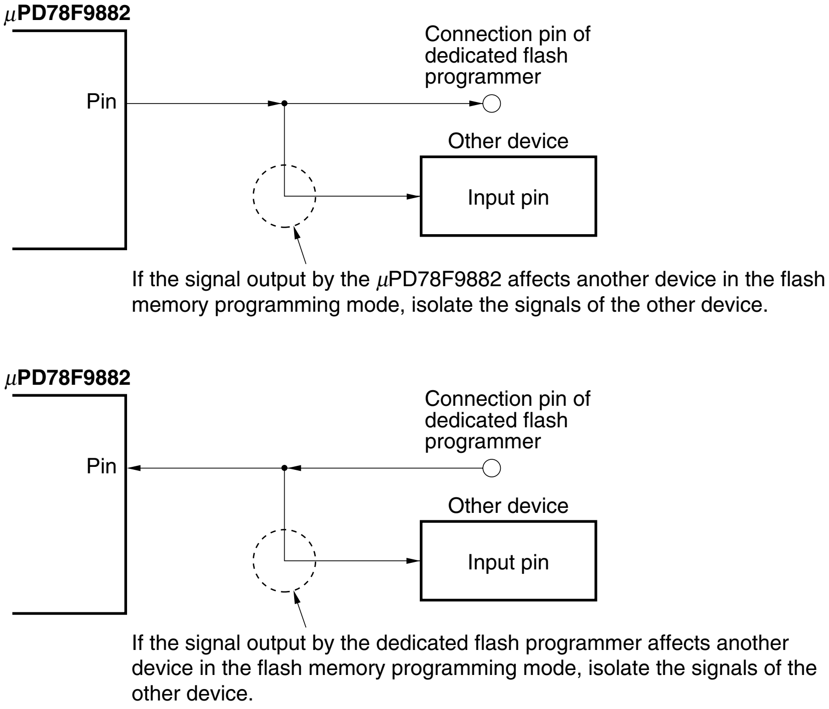


In the flash memory programming mode, the signal output by another device and the signal sent by the dedicated flash programmer conflict, therefore, isolate the signal of the other device.

(2) Abnormal operation of other device

If the dedicated flash programmer (output or input) is connected to a serial interface pin (input or output) that is connected to another device (input), a signal is output to the device, and this may cause an abnormal operation. To prevent this abnormal operation, isolate the connection with the other device or set so that the input signals to the other device are ignored.

Figure 16-6. Abnormal Operation of Other Device


μPD78F9882
Pin
Connection pin of dedicated flash programmer
Other device
Input pin
μPD78F9882
Pin
Connection pin of dedicated flash programmer
Other device
Input pin

If the signal output by the μPD78F9882 affects another device in the flash memory programming mode, isolate the signals of the other device.

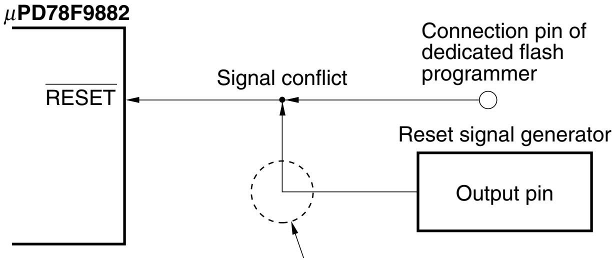
If the signal output by the dedicated flash programmer affects another device in the flash memory programming mode, isolate the signals of the other device.

<$\overline{\text{RESET}}$ pin>

If the reset signal of the dedicated flash programmer is connected to the $\overline{\text{RESET}}$ pin connected to the reset signal generator on-board, a signal conflict occurs. To prevent this, isolate the connection with the reset signal generator.

If the reset signal is input from the user system in the flash memory programming mode, a normal programming operation cannot be performed. Therefore, do not input reset signals from other than the dedicated flash programmer.

Figure 16-7. Signal Conflict ($\overline{\text{RESET}}$ Pin)


μPD78F9882
$\overline{\text{RESET}}$
Signal conflict
Connection pin of dedicated flash programmer
Reset signal generator
Output pin

The signal output by the reset signal generator and the signal output from the dedicated flash programmer conflict in the flash memory programming mode, so isolate the signal of the reset signal generator.

<Port pins>

When the μPD78F9882 enters the flash memory programming mode, all the pins other than those that communicate with flash programmer are in the same status as immediately after reset.

If the external device does not recognize initial statuses such as the output high impedance status, therefore, connect the external device to V_{DD0} or V_{SS0} via a resistor.

<Resonator>

To use the clock output of the flash programmer, directly connect it to the FPCLK pin.

Input an external clock to the FPCLK pin when the clock output of the flash programmer is not used.

<Power supply>

To use the power output from the flash programmer, connect the V_{DD0} or V_{DD1} pin to VDD of the flash programmer, and V_{SS0} or V_{SS1} pin to GND of the flash programmer, respectively.

To use the external power supply, make connection in accordance with the normal operation mode. However, because the voltage is monitored by the flash programmer, be sure to connect VDD of the flash programmer.

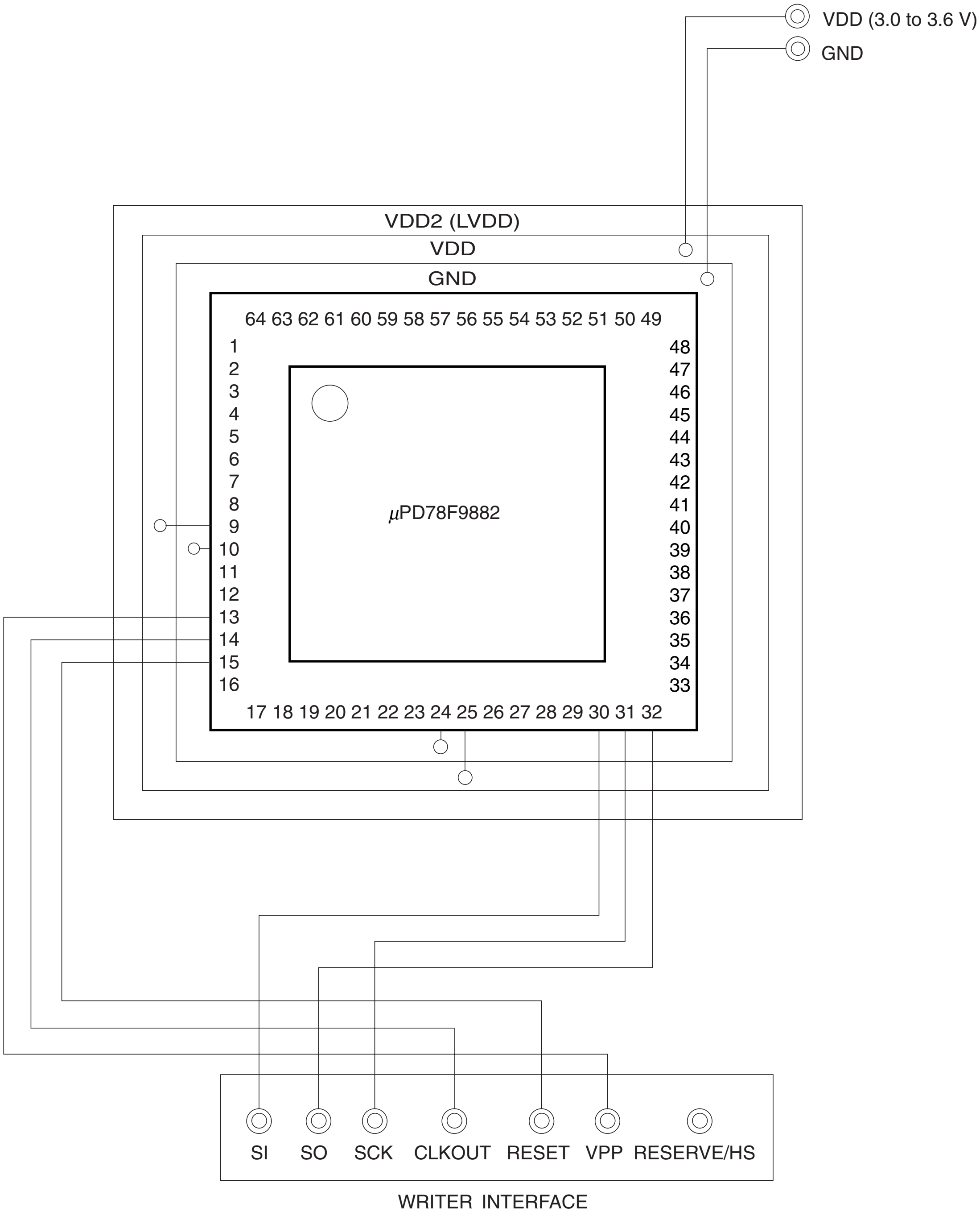
<Other pins>

Process the other pins (S0 to S14, COM0 to COM3, V_{LC1}, V_{LC2}, CAPH1, CAPL1, CAPH2, CAPL2, HV_{DD}, and VR_{OUT}) in the same manner as in the normal operation mode.

16.1.4 Connection of adapter for flash writing

The following figure shows an example of recommended connection when the adapter for flash writing is used.

Figure 16-8. Wiring Example for Flash Writing Adapter with 3-Wire Serial I/O



CHAPTER 17 MASK OPTIONS

The μPD789881 has the following mask options.

- Pull-up resistor

  The connection of on-chip pull-up resistors for port 5 (I/O port) can be switched in 1-bit units.

  <1> Pull-up resistor is connected

  <2> Pull-up resistor is not connected

Caution The mask option is not provided to the flash memory version (μPD78F9882).

CHAPTER 18 INSTRUCTION SET

This chapter lists the instruction set of the μPD789881 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)**.

18.1 Operation

18.1.1 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 18-1. 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 3-3 Special Function Registers** for symbols of special function registers.

18.1.2 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
∧:	Logical product (AND)
∨:	Logical sum (OR)
∀:	Exclusive logical sum (exclusive OR)
—:	Inverted data
addr16:	16-bit immediate data or label
jdisp8:	Signed 8-bit data (displacement value)

18.1.3 Description of “Flag” column

(Blank):	Unchanged
0:	Cleared to 0
1:	Set to 1
×:	Set/cleared according to the result
R:	Previously saved value is restored

18.2 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	×	×	×
	A, PSW	2	4	A ← PSW			
	PSW, A	2	4	PSW ← A	×	×	×
	A, [DE]	1	6	A ← (DE)			
	[DE], A	1	6	(DE) ← A			
	A, [HL]	1	6	A ← (HL)			
	[HL], A	1	6	(HL) ← A			
	A, [HL+byte]	2	6	A ← (HL + byte)			
	[HL+byte], A	2	6	(HL + byte) ← A			
XCH	A, X	1	4	A ↔ X			
	A, r Note 2	2	6	A ↔ r			
	A, saddr	2	6	A ↔ (saddr)			
	A, sfr	2	6	A ↔ sfr			
	A, [DE]	1	8	A ↔ (DE)			
	A, [HL]	1	8	A ↔ (HL)			
	A, [H, 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	×	×	×
	saddr, #byte	3	6	(saddr), CY ← (saddr) + byte	×	×	×
	A, r	2	4	A, CY ← A + r	×	×	×
	A, saddr	2	4	A, CY ← A + (saddr)	×	×	×
	A, !addr16	3	8	A, CY ← A + (addr16)	×	×	×
	A, [HL]	1	6	A, CY ← A + (HL)	×	×	×
	A, [HL+byte]	2	6	A, CY ← A + (HL + byte)	×	×	×
ADDC	A, #byte	2	4	A, CY ← A + byte + CY	×	×	×
	saddr, #byte	3	6	(saddr), CY ← (saddr) + byte + CY	×	×	×
	A, r	2	4	A, CY ← A + r + CY	×	×	×
	A, saddr	2	4	A, CY ← A + (saddr) + CY	×	×	×
	A, !addr16	3	8	A, CY ← A + (addr16) + CY	×	×	×
	A, [HL]	1	6	A, CY ← A + (HL) + CY	×	×	×
	A, [HL+byte]	2	6	A, CY ← A + (HL + byte) + CY	×	×	×
SUB	A, #byte	2	4	A, CY ← A – byte	×	×	×
	saddr, #byte	3	6	(saddr), CY ← (saddr) – byte	×	×	×
	A, r	2	4	A, CY ← A – r	×	×	×
	A, saddr	2	4	A, CY ← A – (saddr)	×	×	×
	A, !addr16	3	8	A, CY ← A – (addr16)	×	×	×
	A, [HL]	1	6	A, CY ← A – (HL)	×	×	×
	A, [HL+byte]	2	6	A, CY ← A – (HL + byte)	×	×	×

Note Only when rp = BC, DE, or HL.

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

Mnemonic	Operands	Byte	Clock	Operation	Flag		
					Z	AC	CY
SUBC	A, #byte	2	4	A, CY ← A – byte – CY	×	×	×
	saddr, #byte	3	6	(saddr), CY ← (saddr) – byte – CY	×	×	×
	A, r	2	4	A, CY ← A – r – CY	×	×	×
	A, saddr	2	4	A, CY ← A – (saddr) – CY	×	×	×
	A, !addr16	3	8	A, CY ← A – (addr16) – CY	×	×	×
	A, [HL]	1	6	A, CY ← A – (HL) – CY	×	×	×
	A, [HL+byte]	2	6	A, CY ← A – (HL + byte) – CY	×	×	×
AND	A, #byte	2	4	A ← A ∧ byte	×		
	saddr, #byte	3	6	(saddr) ← (saddr) ∧ byte	×		
	A, r	2	4	A ← A ∧ r	×		
	A, saddr	2	4	A ← A ∧ (saddr)	×		
	A, !addr16	3	8	A ← A ∧ (addr16)	×		
	A, [HL]	1	6	A ← A ∧ (HL)	×		
	A, [HL+byte]	2	6	A ← A ∧ (HL + byte)	×		
OR	A, #byte	2	4	A ← A ∨ byte	×		
	saddr, #byte	3	6	(saddr) ← (saddr) ∨ byte	×		
	A, r	2	4	A ← A ∨ r	×		
	A, saddr	2	4	A ← A ∨ (saddr)	×		
	A, !addr16	3	8	A ← A ∨ (addr16)	×		
	A, [HL]	1	6	A ← A ∨ (HL)	×		
	A, [HL+byte]	2	6	A ← A ∨ (HL + byte)	×		
XOR	A, #byte	2	4	A ← A ∀ byte	×		
	saddr, #byte	3	6	(saddr) ← (saddr) ∀ byte	×		
	A, r	2	4	A ← A ∀ r	×		
	A, saddr	2	4	A ← A ∀ (saddr)	×		
	A, !addr16	3	8	A ← A ∀ (addr16)	×		
	A, [HL]	1	6	A ← A ∀ (HL)	×		
	A, [HL+byte]	2	6	A ← A ∀ (HL + byte)	×		

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

Mnemonic	Operands	Byte	Clock	Operation	Flag		
					Z	AC	CY
CMP	A, #byte	2	4	A – byte	×	×	×
	saddr, #byte	3	6	(saddr) – byte	×	×	×
	A, r	2	4	A – r	×	×	×
	A, saddr	2	4	A – (saddr)	×	×	×
	A, !addr16	3	8	A – (addr16)	×	×	×
	A, [HL]	1	6	A – (HL)	×	×	×
	A, [HL+byte]	2	6	A – (HL + byte)	×	×	×
ADDW	AX, #word	3	6	AX, CY ← AX + word	×	×	×
SUBW	AX, #word	3	6	AX, CY ← AX – word	×	×	×
CMPW	AX, #word	3	6	AX – word	×	×	×
INC	r	2	4	r ← r + 1	×	×	
	saddr	2	4	(saddr) ← (saddr) + 1	×	×	
DEC	r	2	4	r ← r – 1	×	×	
	saddr	2	4	(saddr) ← (saddr) – 1	×	×	
INCW	rp	1	4	rp ← rp + 1			
DECW	rp	1	4	rp ← rp – 1			
ROR	A, 1	1	2	(CY, A_7 ← A_0, A_{m-1} ← A_m) × 1			×
ROL	A, 1	1	2	(CY, A_0 ← A_7, A_{m+1} ← A_m) × 1			×
RORC	A, 1	1	2	(CY ← A_0, A_7 ← CY, A_{m-1} ← A_m) × 1			×
ROLC	A, 1	1	2	(CY ← A_7, A_0 ← CY, A_{m+1} ← A_m) × 1			×
SET1	saddr.bit	3	6	(saddr.bit) ← 1			
	sfr.bit	3	6	sfr.bit ← 1			
	A.bit	2	4	A.bit ← 1			
	PSW.bit	3	6	PSW.bit ← 1	×	×	×
	[HL].bit	2	10	(HL).bit ← 1			
CLR1	saddr.bit	3	6	(saddr.bit) ← 0			
	sfr.bit	3	6	sfr.bit ← 0			
	A.bit	2	4	A.bit ← 0			
	PSW.bit	3	6	PSW.bit ← 0	×	×	×
	[HL].bit	2	10	(HL).bit ← 0			
SET1	CY	1	2	CY ← 1			1
CLR1	CY	1	2	CY ← 0			0
NOT1	CY	1	2	CY ← $\overline{CY}$			×
CALL	!addr16	3	6	(SP – 1) ← $(PC + 3)_H$, (SP – 2) ← $(PC + 3)_L$, PC ← addr16, SP ← SP – 2			

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
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).

18.3 Instructions Listed by Addressing Type

(1) 8-bit instructions

MOV, XCH, ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP, INC, DEC, ROR, ROL, RORC, ROLC, PUSH, POP, DBNZ

2nd Operand 1st Operand	#byte	A	r	sfr	saddr	!addr16	PSW	[DE]	[HL]	[HL+byte]	$addr16	1	None
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV[Note] XCH[Note] ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP		ROR ROL RORC ROLC	
r	MOV	MOV											INC DEC
B, C											DBNZ		
sfr	MOV	MOV											
saddr	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV									DBNZ		INC DEC
!addr16		MOV											
PSW	MOV	MOV											PUSH POP
[DE]		MOV											
[HL]		MOV											
[HL+byte]		MOV											

Note Except r = A.

(2) 16-bit instructions

MOVW, XCHW, ADDW, SUBW, CMPW, PUSH, POP, INCW, DECW

2nd Operand 1st Operand	#word	AX	rp[Note]	saddrp	SP	None
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	
rp	MOVW	MOVW[Note]				INCW DECW PUSH POP
saddrp		MOVW				
sp		MOVW				

Note Only when rp = BC, DE, or HL.

(3) Bit manipulation instructions

SET1, CLR1, NOT1, BT, BF

2nd Operand 1st Operand	$addr16	None
A.bit	BT BF	SET1 CLR1
sfr.bit	BT BF	SET1 CLR1
saddr.bit	BT BF	SET1 CLR1
PSW.bit	BT BF	SET1 CLR1
[HL].bit		SET1 CLR1
CY		SET1 CLR1 NOT1

(4) Call instructions/branch instructions

CALL, CALLT, BR, BC, BNC, BZ, BNZ, DBNZ

2nd Operand 1st Operand	AX	!addr16	[addr5]	$addr16
Basic Instructions	BR	CALL BR	CALLT	BR BC BNC BZ BNZ
Compound Instructions				DBNZ

(5) Other instructions

RET, RETI, NOP, EI, DI, HALT, STOP

CHAPTER 19 ELECTRICAL SPECIFICATIONS

Absolute Maximum Ratings (T_A = 25°C)

Parameter	Symbol	Conditions		Ratings	Unit
Power supply voltage	V_{DD}			–0.3 to +4.6	V
	V_{PP}	μPD78F9882 only **Note 1**		–0.3 to +8.5	V
Input voltage	V_{I1}	P00 to P03, P10, P11, P20 to P22, P30 to P33, P80 to P84[Note 2], P90 to P95[Note 2], XT1, XT2, $\overline{\text{RESET}}$		–0.3 to V_{DD} + 0.3[Note 3]	V
	V_{I2}	P50 to P53	N-ch open drain	–0.3 to +4.6	V
			On-chip pull-up resistor	–0.3 to V_{DD} + 0.3[Note 2]	V
Output voltage	V_O	P00 to P03, P10, P11, P20 to P22, P30 to P33, P80 to P84[Note 1], P90 to P95[Note 1]		–0.3 to V_{DD} + 0.3[Note 2]	V
Output current, high	I_{OH}	Per pin		–10	mA
		Total for all pins		–30	mA
Output current, low	I_{OL}	Per pin		30	mA
		Total for all pins		160	mA
Operating ambient temperature	T_A	Normal operation		–40 to +85	°C
		Flash memory programming		–20 to +85	°C
Storage temperature	T_{stg}	μPD789881		–65 to +150	°C
		μPD78F9882		–55 to +125	°C

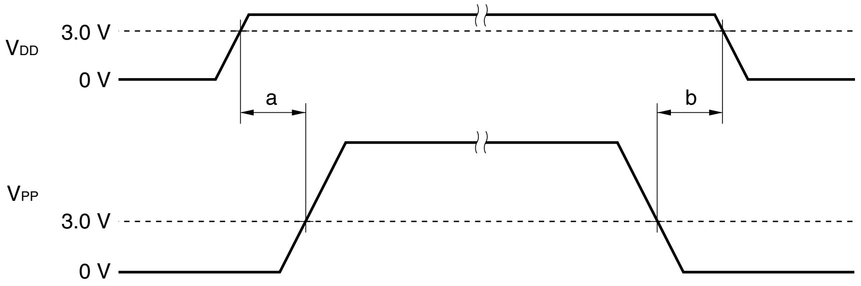
Notes 1. Make sure that the following conditions of the V_{PP} voltage application timing are satisfied when the flash memory is written.

- **When supply voltage rises**

  V_{PP} must exceed V_{DD} 10 μs or more after V_{DD} has reached the lower-limit value (3.0 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 (3.0 V) of the operating voltage range of V_{DD} (see b in the figure below).



2. When used as I/O port by port function register.

3. Must be 4.6 V or lower.

Caution **Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.**

Remark Unless otherwise specified, the characteristics of alternate-function pins are the same as those of port pins.

Main System Clock Oscillator Characteristics

(T_A = –40 to +85°C, V_{DD} = 2.7 to 3.6 V (Mask ROM Version), V_{DD} = 3.0 to 3.6 V (μPD78F9882))

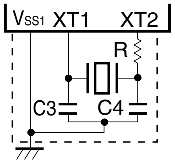
Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Internal ring oscillator	–	Oscillation frequency (f_X)[Note]		0.25	0.5	1.1	MHz

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

Caution When the main system clock is stopped and the device is operating on the subsystem clock, it is not necessary to secure the oscillation stabilization time by the program before switching back to the main system clock.

Subsystem Clock Oscillator Characteristics

(T_A = –40 to +85°C, V_{DD} = 2.7 to 3.6 V (Mask ROM Version), V_{DD} = 3.0 to 3.6 V (μPD78F9882))

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Crystal resonator		Oscillation frequency (f_{XT})[Note 1]		32	32.768	39	kHz
		Oscillation stabilization time[Note 2]			3	10	s

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, users are required to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

DC Characteristics (1/3)

(T_A = –40 to +85°C, V_{DD} = 2.7 to 3.6 V (Mask ROM Version), V_{DD} = 3.0 to 3.6 V (μPD78F9882))

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Output current, low	I_{OL}	Per pin					10	mA
		All pins					80	mA
Output current, high	I_{OH}	Per pin					–1	mA
		All pins					–15	mA
Input voltage, high	V_{IH1}	P00 to P03, P10, P11, P80 to P84, P90 to P95			$0.7V_{DD}$		V_{DD}	V
	V_{IH2}	P50 to P53	N-ch open drain		$0.7V_{DD}$		3.6	V
			On-chip pull-up resistor		$0.7V_{DD}$		V_{DD}	V
	V_{IH3}	$\overline{\text{RESET}}$, P20 to P22, P30 to P33			$0.8V_{DD}$		V_{DD}	V
Input voltage, low	V_{IL1}	P00 to P03, P10, P11, P80 to P84, P90 to P95			0		$0.3V_{DD}$	V
	V_{IL2}	P50 to P53			0		$0.3V_{DD}$	V
	V_{IL3}	$\overline{\text{RESET}}$, P20 to P22, P30 to P33			0		$0.2V_{DD}$	V
Output voltage, high	V_{OH}	P00 to P03, P10, P11, P20 to P22, P30 to P33, P50 to P53, P80 to P84, P90 to P95	I_{OH} = –400 μA		V_{DD} – 0.5			V
Output voltage, low	V_{OL1}	P00 to P03, P10, P11, P20 to P22, P30 to P33, P80 to P84, P90 to P95	I_{OL} = 400 μA				0.5	V
	V_{OL2}	P50 to P53	I_{OL} = 5 mA				0.5	V
Input leakage current, high	I_{LIH1}	V_I = V_{DD}	P00 to P03, P10, P11, P20 to P22, P30 to P33, P80 to P84, P90 to P95, $\overline{\text{RESET}}$				3	μA
	I_{LIH2}	V_I = 3.6 V	P50 to P53				3	μA
Input leakage current, low	I_{LIL1}	V_I = 0 V	P00 to P03, P10, P11, P20 to P22, P30 to P33, P80 to P84, P90 to P95, $\overline{\text{RESET}}$				–3	μA
	I_{LIL2}		P50 to P53	Other than read			–10	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, P11, P20 to P22, P30 to P33		50	100	200	kΩ
			P80 to P84, P90 to P95		100	200	400	kΩ
Mask option pull-up resistor[Note]	R_2	V_I = 0 V	P50 to P53		10	30	60	kΩ

Note Mask ROM version only

Remark Unless otherwise specified, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (2/3)

(T_A = –40 to +85°C, V_{DD} = 2.7 to 3.6 V (Mask ROM Version))

Parameter	Symbol	Conditions			MIN.	TYP.[Note 5]	MAX.	Unit
Power supply current[Note 1] (mask ROM version)	I_{DD1}	0.5 MHz main clock operation mode[Note 2]	V_{DD} = 3.6 V	T_A = –40 to +60°C		25		µA
			V_{DD} = 3.0 V	T_A = –40 to +60°C		21	45	µA
				T_A = –40 to +85°C			90	µA
			V_{DD} = 2.7 V	T_A = –40 to +60°C		18		µA
	I_{DD2}	0.5 MHz main clock HALT mode[Note 2]	V_{DD} = 3.6 V	T_A = –40 to +60°C		8		µA
			V_{DD} = 3.0 V	T_A = –40 to +60°C		6	16	µA
				T_A = –40 to +85°C			32	µA
			V_{DD} = 2.7 V	T_A = –40 to +60°C		5		µA
	I_{DD3}	32.768 kHz crystal oscillation ×4 multiplication operation mode[Note 3]	V_{DD} = 3.6 V	T_A = –40 to +60°C		7		µA
			V_{DD} = 3.0 V	T_A = –40 to +60°C		6.1	12.5	µA
				T_A = –40 to +85°C			25	µA
			V_{DD} = 2.7 V	T_A = –40 to +60°C		5		µA
	I_{DD4}	32.768 kHz crystal oscillation HALT mode[Note 4]	V_{DD} = 3.6 V	T_A = –40 to +60°C		1		µA
			V_{DD} = 3.0 V	T_A = –40 to +60°C		0.9	1.9	µA
				T_A = –40 to +85°C			4.0	µA
			V_{DD} = 2.7 V	T_A = –40 to +60°C		0.87		µA

Notes 1. The current that flows through the on-chip pull-up resistors is not included.

2. High-speed mode operation (when the processor clock control register (PCC) is cleared to 00H)

3. The CPU operates on the subsystem clock multiplied by four, with the main system clock stopped.

4. When timer 50, the watchdog timer, and the LCD are operating (when LCDON0 = 1, LIPS0 = 1)

5. The TYP. value is at T_A = 25°C and when all the peripheral functions are OFF.

Remark Unless otherwise specified, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (3/3)
(T_A = –40 to +85°C, V_{DD} = 3.0 to 3.6 V (μPD78F9882))

Parameter	Symbol	Conditions			MIN.	TYP.[Note 5]	MAX.	Unit
Power supply current[Note 1] (μPD78F9882)	I_{DD1}	0.5 MHz main clock operation mode[Note 2]	V_{DD} = 3.6 V	T_A = –40 to +60°C		9.5		mA
			V_{DD} = 3.0 V	T_A = –40 to +60°C		8.2		mA
				T_A = –40 to +85°C			18	mA
	I_{DD2}	0.5 MHz main clock HALT mode[Note 2]	V_{DD} = 3.6 V	T_A = –40 to +60°C		9.4		mA
			V_{DD} = 3.0 V	T_A = –40 to +60°C		8.15		mA
				T_A = –40 to +85°C			18	mA
	I_{DD3}	32.768 kHz crystal oscillation ×4 multiplication operation mode[Note 3]	V_{DD} = 3.6 V	T_A = –40 to +60°C		9.4		mA
			V_{DD} = 3.0 V	T_A = –40 to +60°C		8.15		mA
				T_A = –40 to +85°C			18	mA
	I_{DD4}	32.768 kHz crystal oscillation HALT mode[Note 4]	V_{DD} = 3.6 V	T_A = –40 to +60°C		9.3		mA
			V_{DD} = 3.0 V	T_A = –40 to +60°C		8.1		mA
				T_A = –40 to +85°C			18	mA

Notes 1. The current that flows through the on-chip pull-up resistors is not included.

2. High-speed mode operation (when the processor clock control register (PCC) is cleared to 00H)

3. The CPU operates on the subsystem clock multiplied by four, with the main system clock stopped.

4. When timer 50, the watchdog timer, and the LCD are operating (when LCDON0 = 1, LIPS0 = 1)

5. The TYP. value is at T_A = 25°C and when all the peripheral functions are OFF.

Remark Unless otherwise specified, the characteristics of alternate-function pins are the same as those of port pins.

AC Characteristics

(1) Basic operation
(T_A = –40 to +85°C, V_{DD} = 2.7 to 3.6 V (mask ROM version), V_{DD} = 3.0 to 3.6 V (μPD78F9882))

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Cycle time (minimum instruction execution time)	T_{CY}	Operating with main system clock	1.81	4	32	μs
		Operating with subsystem clock		30.5		μs
TI00, TI01 high-/low-level width	t_{TIH0}, t_{TIL0}		2/f_{sam} + 0.2^{Note}			μs
TI50, TI51 input frequency	f_{TI5}		0		250	kHz
TI50, TI51 input high-/low-level width	t_{TIH5}, t_{TIL5}		1.8			μs
Interrupt input high-/low-level width	t_{INTH}, t_{INTL}	INTP0 to INTP3	10			μs
$\overline{\text{RESET}}$ low-level width	t_{RSL}		10			μs

Note f_{sam}: Count clock frequency of 16-bit timer/event counter 0.
Selection of f_{sam} = 4f_{XT}, f_{XT}, or f_{XT}/2 is possible using bits 0 and 1 (PRM00, PRM01) of prescaler mode register 0 (PRM0). Note that when selecting the TI00 valid edge as the count clock, f_{sam} = f_X.

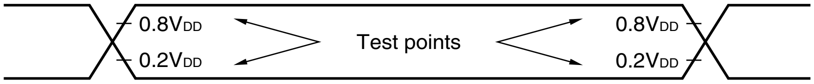
Remarks 1. f_X: Main system clock frequency
2. f_{XT}: Subsystem clock frequency

(2) Serial interface UART0
(T_A = –40 to +85°C, V_{DD} = 2.7 to 3.6 V (mask ROM version), V_{DD} = 3.0 to 3.6 V (μPD78F9882))

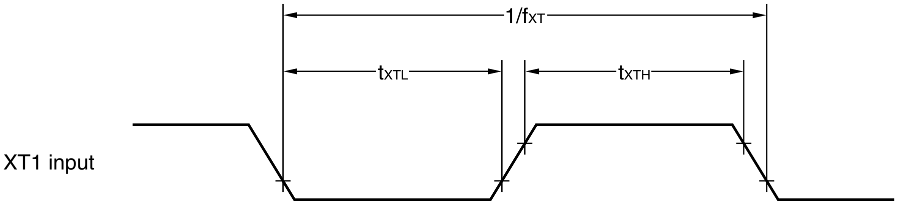
(a) UART mode (dedicated baud rate generator output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		f_{XT} = 32.768 kHz			4800	bps
		f_{XT} = 38.4 kHz			9600	bps

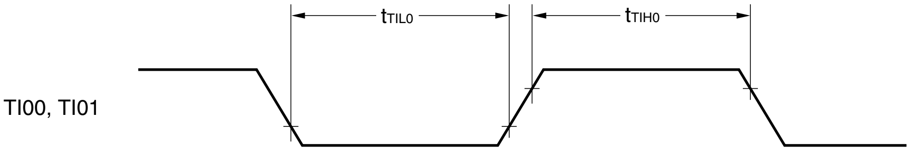
AC Timing Test Points (Excluding X1 and XT1 Inputs)

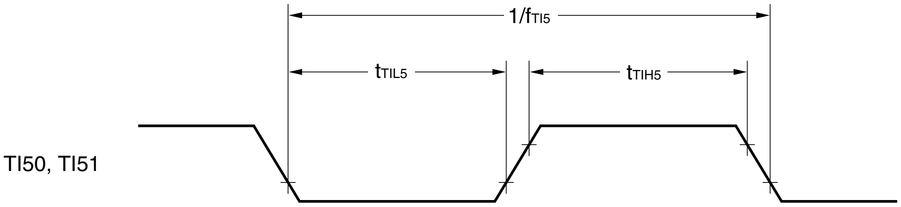


Clock Timing

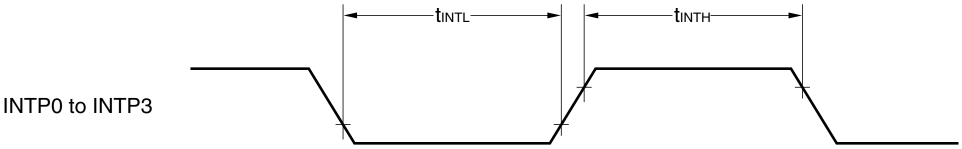


Timer Input Timing

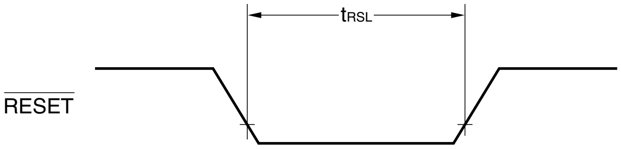




Interrupt Input Timing



$\overline{\text{RESET}}$ Input Timing



LCD Characteristics

(T_A = –40 to +85°C, V_{DD} = 2.7 to 3.6 V (Mask ROM Version), V_{DD} = 3.0 to 3.6 V (μPD78F9882))

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
LCD drive voltage	V_{LCD}	$V_{DD} = V_{LCD}$	Mask ROM version	2.7		3.6	V
			μPD78F9882	3.0		3.6	V
Capacitor drive power boosting wait time[Note 1]	t_{VLCD}	C1 to C3[Note 2] = 0.47 μF		0.5			s
LCD output voltage differential[Note 3] (common)	V_{ODC}	I_O = ±5 μA	Static or 1/3 bias	0		±0.2	V
LCD output voltage differential[Note 3] (segment)	V_{ODS}	I_O = ±1 μA	Static or 1/3 bias	0		±0.2	V

Notes 1. This is the wait time from when LCD drive power is supplied (VAON0 = 1) until capacitor boosting (LCD drive power supply is stabilized).

2. This is a capacitor that is connected between voltage pins used to drive the LCD.
C1: A capacitor connected between CAPH2 and CAPL2
C2: A capacitor connected between V_{LC1} and V_{SS}
C3: A capacitor connected between V_{LC2} and V_{SS}

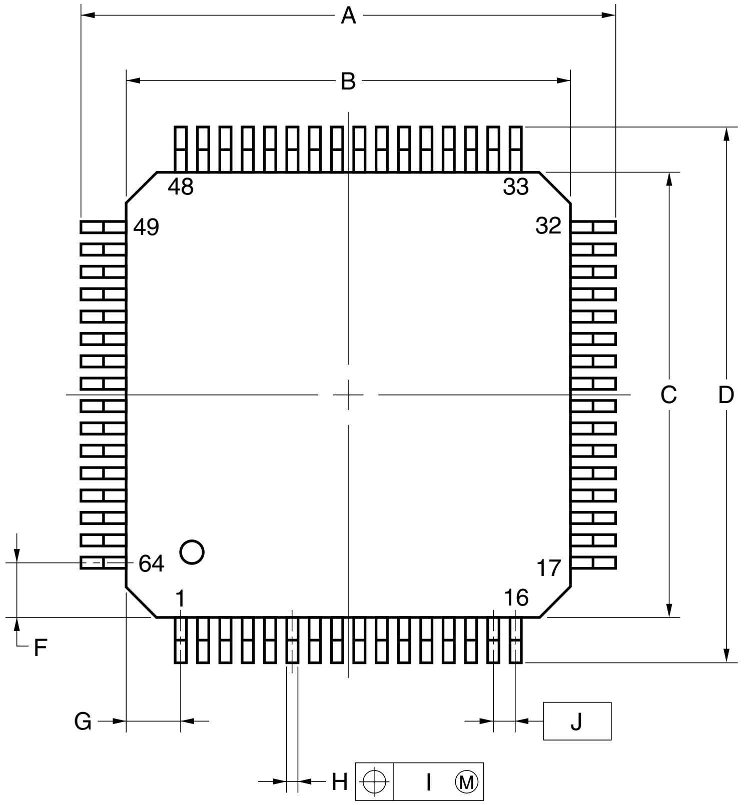
3. The voltage differential is the difference between the segment and common signal output's actual and ideal output voltages with no load.

Flash Memory Writing and Erasing Characteristics (μPD78F9882) (T_A = 10 to 40°C, V_{DD} = 3.0 to 3.6 V)

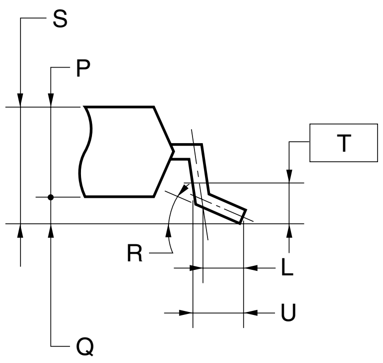
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Write/erase operating frequency	f_X	FPCLK pin external clock input	1.0		5	MHz
SCK cycle time	t_{KCY}		500			ns
Write current (V_{DD} pin)[Note]	I_{DDW}	When V_{PP} supply voltage = V_{PP1} (at 5.0 MHz operation)			21	mA
Write current (V_{PP} pin)[Note]	I_{PPW}	When V_{PP} supply voltage = V_{PP1}			50	mA
Erase current (V_{DD} pin)[Note]	I_{DDE}	When V_{PP} supply voltage = V_{PP1} (at 5.0 MHz operation)			21	mA
Erase current (V_{PP} pin)[Note]	I_{PPE}	When V_{PP} supply voltage = V_{PP1}			100	mA
Total erase time	t_{era}	100 times of retry of 0.2 s			20	s
Number of overwrites		Erase and write is considered as 1 cycle			20	Times
V_{PP} supply voltage	V_{PP0}	Normal operation	0		$0.2V_{DD}$	V
	V_{PP1}	Flash memory programming	7.5	7.8	8.1	V

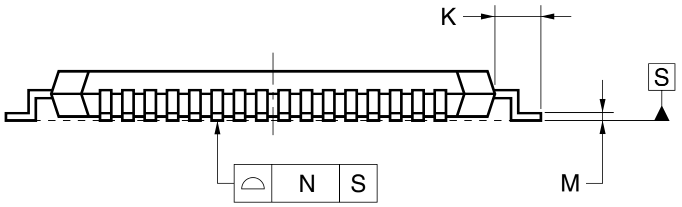
Note Excludes current flowing through ports (including on-chip pull-up resistors)

64-PIN PLASTIC LQFP (10x10)



detail of lead end





NOTE

Each lead centerline is located within 0.08 mm of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS
A	12.0±0.2
B	10.0±0.2
C	10.0±0.2
D	12.0±0.2
F	1.25
G	1.25
H	0.22±0.05
I	0.08
J	0.5 (T.P.)
K	1.0±0.2
L	0.5
M	$0.17^{+0.03}_{-0.07}$
N	0.08
P	1.4
Q	0.1±0.05
R	$3^{\circ}{}^{+4^{\circ}}_{-3^{\circ}}$
S	1.5±0.10
T	0.25
U	0.6±0.15

S64GB-50-8EU-2

CHAPTER 21 RECOMMENDED SOLDERING CONDITIONS

The μPD789881 Subseries should be soldered and mounted under the following recommended conditions.
For soldering methods and conditions other than those recommended below, contact an NEC sales representative.
For technical information, see the following website.

Semiconductor Device Mount Manual (http://www.necel.com/pkg/en/mount/index.html)

Table 21-1. Surface Mounting Type Soldering Conditions (1/2)

μPD789881GB-×××-8EU: 64-pin plastic LQFP (10 × 10)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 235°C, Time: 30 seconds max. (at 210°C or higher), Count: Twice or less	IR35-00-2
VPS	Package peak temperature: 215°C, Time: 40 seconds max. (at 200°C or higher), Count: Twice or less	VP15-00-2
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).

μPD78F9882GB-8EU: 64-pin plastic LQFP (10 × 10)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 235°C, Time: 30 seconds max. (at 210°C or higher), Count: Twice or less, Exposure limit: 3 days[Note] (after that, prebake at 125°C for 10 hours)	IR35-103-2
VPS	Package peak temperature: 215°C, Time: 40 seconds max. (at 200°C or higher), Count: Twice or less, Exposure limit: 3 days[Note] (after that, prebake at 125°C for 10 hours)	VP15-103-2
Partial heating	Pin temperature: 350°C max., Time: 3 seconds max. (per pin row)	–

Note After opening the dry pack, store it at 25°C or less and 65% RH or less for the allowable storage period.

Caution Do not use different soldering methods together (except for partial heating).

Table 21-1. Surface Mounting Type Soldering Conditions (2/2)

★ **μPD789881GB-×××-8EU-A: 64-pin plastic LQFP (10 × 10)**
μPD789882GB-8EU-A: 64-pin plastic LQFP (10 × 10)

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).

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

APPENDIX A DEVELOPMENT TOOLS

The following development tools are available for development of systems using the μPD789881 Subseries. Figure A-1 shows development tools.

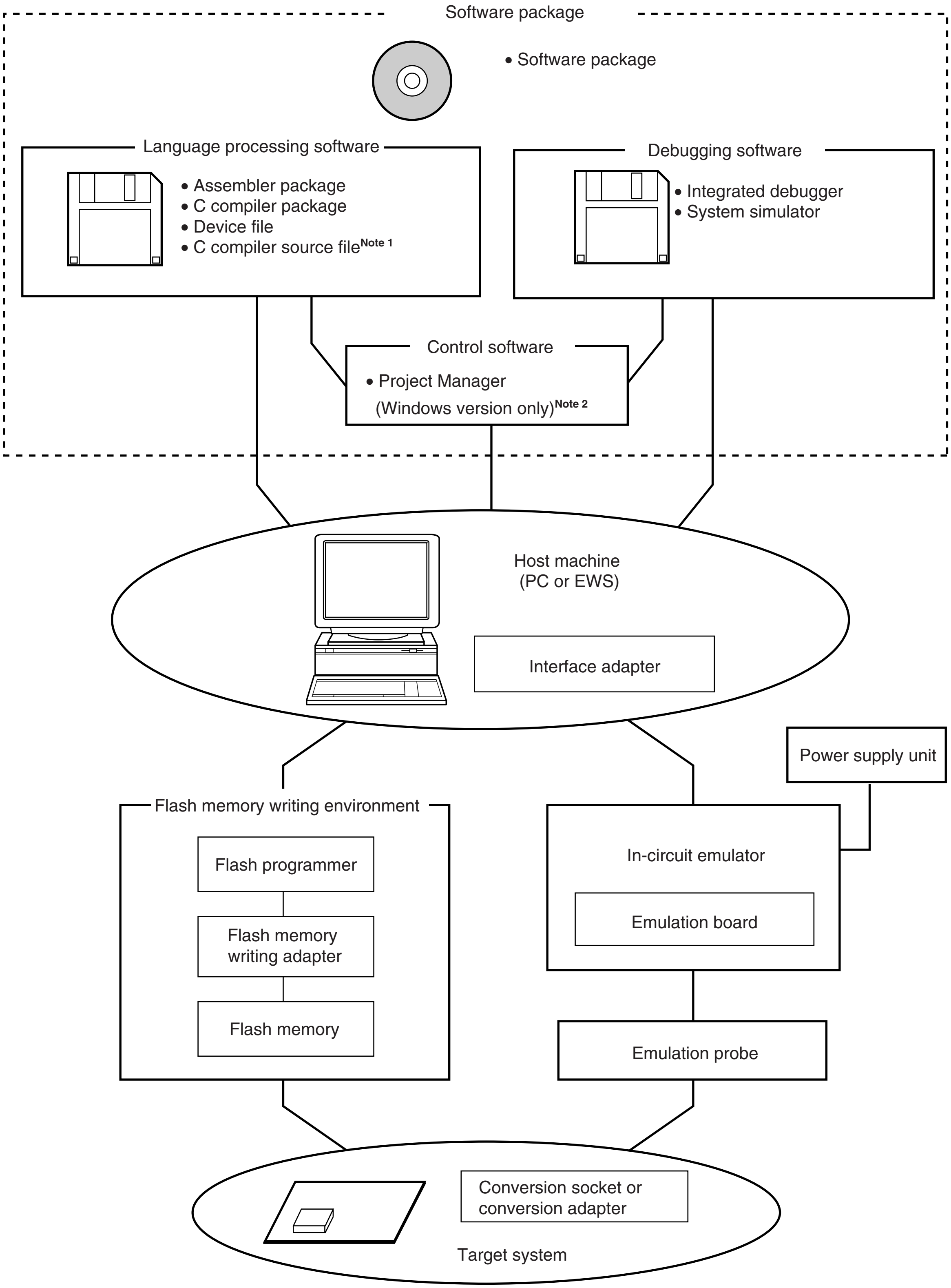
- Support to PC98-NX Series

  Unless specified otherwise, the products supported by IBM PC/AT™ compatibles can be used in 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, 98, 2000
  - Windows NT™ Ver.4.0

Figure A-1. Development Tools



Notes 1. C compiler 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 systems to be used.

μS××××SP78K0S

××××	Host Machine	OS	Supply Media
AB17	PC-9800 series, IBM PC/AT compatible	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 symbol table and optimize branch instructions are also provided. Used in combination with optional device file (DF789882). **<Caution when used under PC environment>** The assembler package is a DOS-based application but may be used under the Windows environment by using 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 optional assembler package (RA78K0S) and device file (DF789882). **<Caution when used under PC environment>** The C compiler package is a DOS-based application but may be used under the Windows environment by using Project Manager of Windows (included in the assembler package).
	Part number: μS××××CC78K0S
DF789882[Note 1] Device file	File containing the information inherent to the device. Used in combination with optional RA78K0S, CC78K0S, ID78K0S-NS, and SM78K0S.
	Part number: μS××××DF789882
CC78K0S-L[Note 2] C compiler 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. DF789882 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 machines and operating systems to be used.

μS××××RA78K0S
μS××××CC78K0S

××××	Host Machine	OS	Supply Media
AB13	PC-9800 series, IBM PC/AT compatible	Japanese Windows	3.5" 2HD FD
BB13		English Windows	
AB17		Japanese Windows	CD-ROM
BB17		English Windows	
3P17	HP9000 series 700™	HP-UX™ (Rel. 10.10)	
3K17	SPARCstation™	SunOS™ (Rel. 4.1.4), Solaris™ (Rel. 2.5.1)	

μS××××DF789882
μS××××CC78K0S-L

××××	Host Machine	OS	Supply Media
AB13	PC-9800 series, IBM PC/AT compatible	Japanese Windows	3.5" 2HD FD
BB13		English Windows	
3P16	HP9000 series 700	HP-UX (Rel. 10.10)	DAT
3K13	SPARCstation	SunOS (Rel. 4.1.4), Solaris (Rel. 2.5.1)	3.5" 2HD FD
3K15			1/4-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 (Part No. FL-PR3, PG-FP3) Flash programmer	Dedicated flash programmer for microcontrollers incorporating flash memory
FA-64GB-8EU Flash memory writing adapter	Adapter for writing to flash memory and connected to Flashpro III. • FA-64GB-8EU: For 64-pin plastic QFP (GB-8EU type)

Remark The FL-PR3 and FA-64GB-8EU 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 hardware and software of application system using 78K/0S Series. Supports integrated debugger (ID78K0S-NS). Used in combination with 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 PC-9800 series PC (except notebook type) as host machine (C bus supported)
IE-70000-CD-IF-A PC card interface	PC card and interface cable necessary when using notebook PC as host machine (PCMCIA socket supported)
IE-70000-PC-IF-C Interface adapter	Interface adapter necessary when using IBM PC/AT compatible as host machine (ISA bus supported)
IE-70000-PCI-IF-A Interface adapter	Adapter necessary when using personal computer incorporating PCI bus as host machine
IE-789882-NS-EM1 Emulation board	Board for emulating the peripheral hardware inherent to the device. Used in combination with in-circuit emulator.
NP-64GB-TQ Emulation probe	Cable to connect the in-circuit emulator and target system. Used in combination with the TGB-064SDP.
TGB-064SDP Conversion adapter	Conversion adapter to connect the NP-64GB-TQ and a target system board on which a 64-pin plastic TQFP (GB-8EU type) can be mounted

Remarks 1. The NP-64BG-TQ is a product made by Naito Densei Machida Mfg. Co., Ltd. (TEL +81-45-475-4191).

2. The TGB-064SDP is a product 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 optional device file (DF789882).
	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 optional device file (DF789882).
	Part number: μS××××SM78K0S
DF789882[Note] Device file	File containing the information inherent to the device. Used in combination with the optional RA78K0S, CC78K0S, ID78K0S-NS, and SM78K0S.
	Part number: μS××××DF789882

Note DF789882 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 systems and supply medium to be used.

μS××××ID78K0S-NS
μS××××SM78K0S

××××	Host Machine	OS	Supply Media
AB13	PC-9800 series IBM PC/AT compatibles	Japanese Windows	3.5" 2HD FD
BB13		English Windows	
AB17		Japanese Windows	CD-ROM
BB17		English Windows	

APPENDIX B REGISTER INDEX

B.1 Register Index (Alphabetic Order of Register Name)

B.2 Register Index (Alphabetic Order of Register Symbol)