NEC

User's Manual

μPD780318, 780328, 780338 Subseries

8-Bit Single-Chip Microcontrollers

μPD780316
μPD780318
μPD780326
μPD780328
μPD780336
μPD780338
μPD78F0338

Document No. U14701EJ3V0UD00 (3rd edition)
Date Published September 2002 N CP(K)

[MEMO]

NOTES FOR CMOS DEVICES

① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

Note:

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

② HANDLING OF UNUSED INPUT PINS FOR CMOS

Note:

No connection for CMOS device inputs can be cause of malfunction. If no connection is provided to the input pins, it is possible that an internal input level may be generated due to noise, etc., hence causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using a pull-up or pull-down circuitry. Each unused pin should be connected to V_{DD} or GND with a resistor, if it is considered to have a possibility of being an output pin. All handling related to the unused pins must be judged device by device and related specifications governing the devices.

③ STATUS BEFORE INITIALIZATION OF MOS DEVICES

Note:

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

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

License not needed: μPD78F0338GC-9EB, 78F0338GC-9EV

The customer must judge the need for a license for the following products:

μPD780316GC-xxx-9EB, 780316GC-xxx-9EV, 780318GC-xxx-9EB, 780318GC-xxx-9EV,
μPD780326GC-xxx-9EB, 780326GC-xxx-9EV, 780328GC-xxx-9EB, 780328GC-xxx-9EV,
μPD780336GC-xxx-9EB, 780336GC-xxx-9EV, 780338GC-xxx-9EB, 780338GC-xxx-9EV

- **The information in this document is current as of July, 2002. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC's data sheets or data books, etc., for the most up-to-date specifications of NEC semiconductor products. Not all products and/or types are available in every country. Please check with an NEC 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 prior written consent of NEC. NEC assumes no responsibility for any errors that may appear in this document.
- NEC 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 semiconductor 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 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 customer's equipment shall be done under the full responsibility of customer. NEC assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC endeavours to enhance the quality, reliability and safety of NEC semiconductor 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 semiconductor products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment, and anti-failure features.
- NEC semiconductor products are classified into the following three quality grades:
  "Standard", "Special" and "Specific". The "Specific" quality grade applies only to semiconductor products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of a semiconductor product depend on its quality grade, as indicated below. Customers must check the quality grade of each semiconductor 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 semiconductor products is "Standard" unless otherwise expressly specified in NEC's data sheets or data books, etc. If customers wish to use NEC semiconductor products in applications not intended by NEC, they must contact an NEC sales representative in advance to determine NEC's willingness to support a given application.
  (Note)
  (1) "NEC" as used in this statement means NEC Corporation and also includes its majority-owned subsidiaries.
  (2) "NEC semiconductor products" means any semiconductor product developed or manufactured by or for NEC (as defined above).

M8E 00. 4

Regional Information

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

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

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

NEC Electronics Inc. (U.S.)
Santa Clara, California
Tel: 408-588-6000
800-366-9782
Fax: 408-588-6130
800-729-9288

NEC do Brasil S.A.
Electron Devices Division
Guarulhos-SP, Brasil
Tel: 11-6462-6810
Fax: 11-6462-6829

NEC Electronics (Europe) GmbH
Duesseldorf, Germany
Tel: 0211-65 03 01
Fax: 0211-65 03 327

- **Sucursal en España**
  Madrid, Spain
  Tel: 091-504 27 87
  Fax: 091-504 28 60

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

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

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

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

- **United Kingdom Branch**
  Milton Keynes, UK
  Tel: 01908-691-133
  Fax: 01908-670-290

NEC Electronics Hong Kong Ltd.
Hong Kong
Tel: 2886-9318
Fax: 2886-9022/9044

NEC Electronics Hong Kong Ltd.
Seoul Branch
Seoul, Korea
Tel: 02-528-0303
Fax: 02-528-4411

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

NEC Electronics Taiwan Ltd.
Taipei, Taiwan
Tel: 02-2719-2377
Fax: 02-2719-5951

NEC Electronics Singapore Pte. Ltd.
Novena Square, Singapore
Tel: 253-8311
Fax: 250-3583

J02.4

Major Revisions in This Edition

Page	Description
p.57	Change of Recommended Connection of Unused Pins for the following pins in **Table 2-1 Pin I/O Circuit Types** • P60 to P63 • P80/S32 to P87/S39 (for flash memory version) • P90/S24 to P97/S31 (for flash memory version)
p.66	Addition of description to **(1) Internal high-speed RAM** and **(2) Internal expansion RAM** in **3.1.2 Internal data memory space**
p.75	Change of Manipulatable Bit Unit for ports 8 and 9 in **Table 3-4 Special Function Register List**
p.112	Modification of **Figure 4-18 P80 to P87 and P90 to P97 Block Diagram (Flash Memory Version)**
p.113	Modification of Caution in **4.2.11 Port 12**
p.165	Modification of clear conditions in **7.3 (1) 16-bit timer counter 4 (TM4)**
p.165	Modification of **Figure 7-1 16-Bit Timer/Event Counter 4 Block Diagram**
p.284	Modification of **Note** in **Table 17-4 Frame Frequency**
p.286	Modification of **Figure 17-6 Static/Dynamic Display Switching Register 3 (SDSEL3) Format**
pp.288 and 289	Switch in order between **17.4 LCD Controller/Driver Settings** and **17.5 LCD Display RAM** of previous edition
p.291 in previous edition	Deletion of **Table 17-7 LCD Drive Voltages** of previous edition
pp.291 to 293 and 390	Standardization of abbreviations • Output voltage of V_{LC0} pin: V_{LCD0} • Output voltage of V_{LC1} pin: V_{LCD1} • Output voltage of V_{LC2} pin: V_{LCD2}
p.295	Addition of description to **17.8.1 Static display example**
p.296 p.299 p.302	Modification of LCD panel connection example • **Figure 17-13 Static LCD Panel Connection Example (SDSEL3n = 1: n = 0, 1)** • **Figure 17-16 3-Time-Division LCD Panel Connection Example (SDSEL3n = 0: n = 0 to 2)** • **Figure 17-19 4-Time-Division LCD Panel Connection Example (SDSEL3n = 0, n = 0 to 2)**
pp.406, 410, and 411	Change of emulation probe name SWEX-120SE → SWEX-120SE-1
p.410	Modification of **Figure D-1 Distance from In-Circuit Emulator to Conversion Socket**
p.411	Modification of **Figure D-2 Connection Condition of Target System**

The mark ★ shows major revised points.

INTRODUCTION

Readers

This manual has been prepared for user engineers who wish to understand the functions of the μPD780318, 780328, and 780338 Subseries and design and develop application systems and programs for these devices.

μPD780318 Subseries: μPD780316, 780318
μPD780328 Subseries: μPD780326, 780328
μPD780338 Subseries: μPD780336, 780338, 78F0338

Purpose

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

Organization

The μPD780318, 780328, and 780338 Subseries manual is separated into two parts: this manual and the instructions edition (common to the 78K/0 Series).

μPD780318, 780328, 780338 Subseries User's Manual (This Manual)	**78K/0 Series User's Manual Instructions**
• Pin functions • Internal block functions • Interrupt • Other on-chip peripheral functions • Electrical specifications	• CPU functions • Instruction set • Explanation of each instruction

How To Read This Manual

It is assumed that the reader of this manual has general knowledge in the fields of electrical engineering, logic circuits, and microcontrollers.

- To gain a general understanding of functions:
  → Read this manual in the order of the contents.
- How to interpret the register format:
  → For the bit number enclosed in square, the bit name is defined as a reserved word in RA78K0, and in CC78K0, already defined in the header file named sfrbit.h.
- To check the details of a register when you know the register name.
  → Refer to **APPENDIX E REGISTER INDEX.**

Conventions

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

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

Documents Related to Devices

Document Name	Document No.
μPD780318, 780328, 780338 Subseries User's Manual	This document
78K/0 Series Instructions User's Manual	U12326E

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

Document Name		Document No.
RA78K0 Assembler Package	Operation	U14445E
	Language	U14446E
	Structured Assembly Language	U11789E
CC78K0 C Compiler	Operation	U14297E
	Language	U14298E
SM78K0S, SM78K0 System Simulator Ver. 2.10 or Later	Operation (Windows™ Based)	U14611E
SM78K Series System Simulator Ver. 2.10 or Later	External Part User Open Interface Specifications	U15006E
★ ID78K Series Integrated Debugger Ver. 2.30 or Later	Operation (Windows Based)	U15185E
RX78K0 Real-time OS	Fundamentals	U11537E
	Installation	U11536E
Project Manager Ver. 3.12 or Later (Windows Based)		U14610E

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

Document Name	Document No.
IE-78K0-NS In-Circuit Emulator	U13731E
IE-78K0-NS-A In-Circuit Emulator	U14889E
IE-780338-NS-EM1 Emulation Board	To be prepared

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

Documents Related to Flash Memory Writing

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

Other Related Documents

Document Name	Document No.
SEMICONDUCTOR SELECTION GUIDE - Products & Packages -	X13769E
Semiconductor Device Mounting Technology Manual	C10535E
Quality Grades on NEC Semiconductor Devices	C11531E
NEC Semiconductor Device Reliability/Quality Control System	C10983E
Guide to Prevent Damage for Semiconductor Devices by Electrostatic Discharge (ESD)	C11892E

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

CONTENTS

LIST OF FIGURES (1/7)

LIST OF FIGURES (2/7)

LIST OF FIGURES (3/7)

LIST OF FIGURES (4/7)

LIST OF FIGURES (5/7)

LIST OF FIGURES (6/7)

LIST OF FIGURES (7/7)

LIST OF TABLES (1/3)

LIST OF TABLES (2/3)

LIST OF TABLES (3/3)

CHAPTER 1 OUTLINE

1.1 Features

- Internal memory

Type / Part Number	Program Memory (ROM/Flash Memory)	Data Memory		LCD Display RAM
		High-Speed RAM	Expansion RAM	
μPD780316, 780326, 780336	48 KB	1,024 bytes	1,536 bytes	40 × 8 bits
μPD780318, 780328, 780338	60 KB			
μPD78F0338	60 KB[Note]			

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

- Minimum instruction execution time changeable from high speed (0.2 μs: @10 MHz operation with main system clock) to ultra-low speed (122 μs: @32.768 kHz operation with subsystem clock)
- Instruction set suited to system control
  - Bit manipulation possible in all address spaces
  - Multiply and divide instructions
- I/O port
  - μPD780316, 780318, 78F0338: 70 (Medium voltage N-ch open-drain: 4)
  - μPD780326, 780328: 62 (Medium voltage N-ch open-drain: 4)
  - μPD780336, 780338: 54 (Medium voltage N-ch open-drain: 4)
- 10-bit resolution A/D converter: 10 channels
- 8-bit resolution D/A converter: 1 channel
- LCD controller/driver
  - Segment signal output
    - μPD780316, 780318: 24 max.
    - μPD780326, 780328: 32 max.
    - μPD780336, 780338, 78F0338: 40 max.
  - Common signal output
    - Dynamic display: 4 max.
    - Static display: 1
  - LCD reference voltage generator: booster type (×3 only)
  - Fine tuning of LCD reference voltage possible with external resistor
  - Blinking display possible (blinking interval can be selected: 0.5 s or 1 s)
  - Static display and dynamic display (1/3 bias only) can be used simultaneously (Static display up to 12)
- Serial interface
  - UART/3-wire serial I/O mode: 1 channel[Note]
  - 3-wire serial I/O mode: 1 channel
- Timer
  - 16-bit timer/event counter: 2 channels
  - 8-bit timer/event counter: 3 channels
  - Watch timer: 1 channel
  - Watchdog timer: 1 channel

- ROM correction
- Vectored interrupt sources: 25
- Two types of on-chip clock oscillators (main system clock and subsystem clock)
- Power supply voltage: V_{DD} = 1.8 to 5.5 V

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

1.2 Applications

Cordless telephones (handset), compact cameras, etc.

1.3 Ordering Information

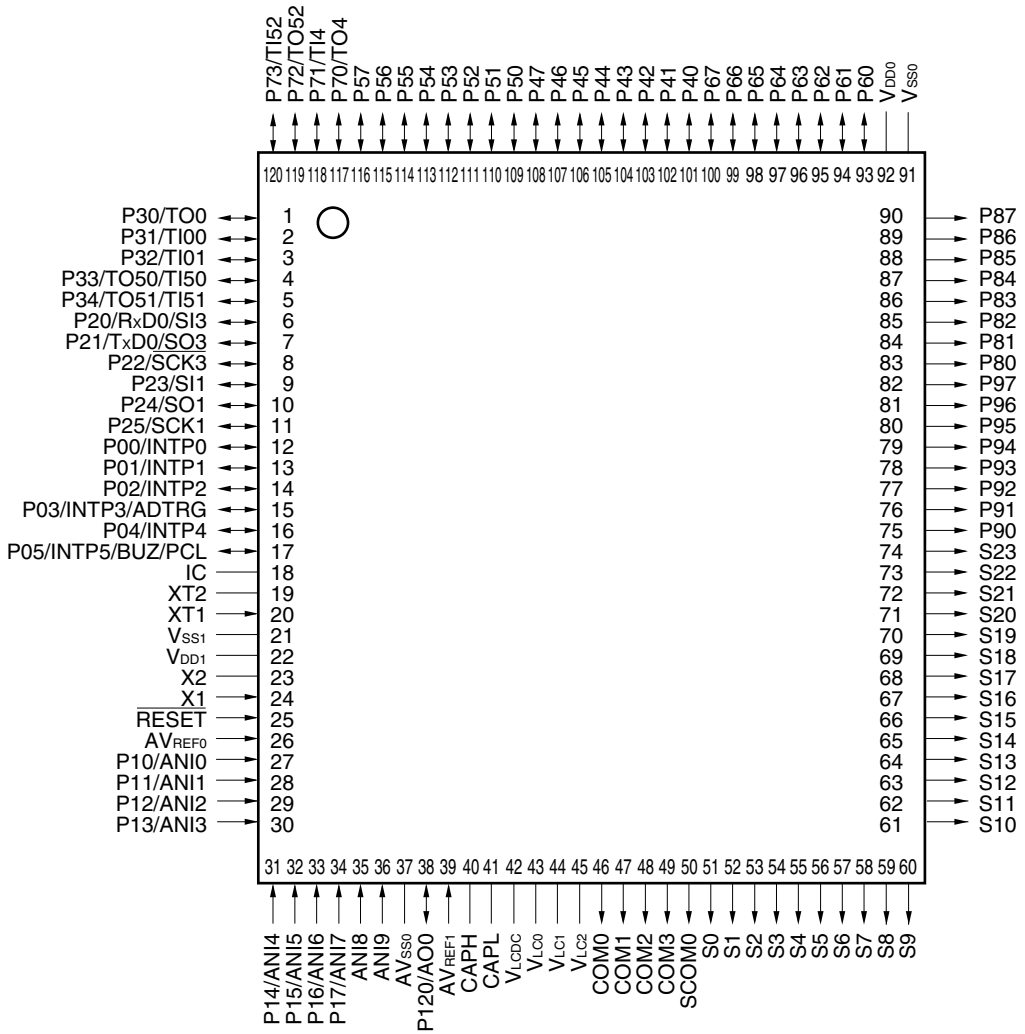
Part Number	Package	Internal ROM
μPD780316GC-×××-9EB	120-pin plastic TQFP (fine pitch) (14 × 14)	Mask ROM
μPD780316GC-×××-9EV	120-pin plastic TQFP (fine pitch) (14 × 14)	Mask ROM
μPD780318GC-×××-9EB	120-pin plastic TQFP (fine pitch) (14 × 14)	Mask ROM
μPD780318GC-×××-9EV	120-pin plastic TQFP (fine pitch) (14 × 14)	Mask ROM
μPD780326GC-×××-9EB	120-pin plastic TQFP (fine pitch) (14 × 14)	Mask ROM
μPD780326GC-×××-9EV	120-pin plastic TQFP (fine pitch) (14 × 14)	Mask ROM
μPD780328GC-×××-9EB	120-pin plastic TQFP (fine pitch) (14 × 14)	Mask ROM
μPD780328GC-×××-9EV	120-pin plastic TQFP (fine pitch) (14 × 14)	Mask ROM
μPD780336GC-×××-9EB	120-pin plastic TQFP (fine pitch) (14 × 14)	Mask ROM
μPD780336GC-×××-9EV	120-pin plastic TQFP (fine pitch) (14 × 14)	Mask ROM
μPD780338GC-×××-9EB	120-pin plastic TQFP (fine pitch) (14 × 14)	Mask ROM
μPD780338GC-×××-9EV	120-pin plastic TQFP (fine pitch) (14 × 14)	Mask ROM
μPD78F0338GC-9EB	120-pin plastic TQFP (fine pitch) (14 × 14)	Flash memory
μPD78F0338GC-9EV	120-pin plastic TQFP (fine pitch) (14 × 14)	Flash memory

Remark ××× indicates ROM code suffix.

1.4 Pin Configuration (Top View)

1.4.1 μPD780316, 780318

- **120-pin plastic TQFP (fine pitch) (14 × 14)**
  μPD780316GC-×××-9EB, 780316GC-×××-9EV, 780318GC-×××-9EB, 780318GC-×××-9EV



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

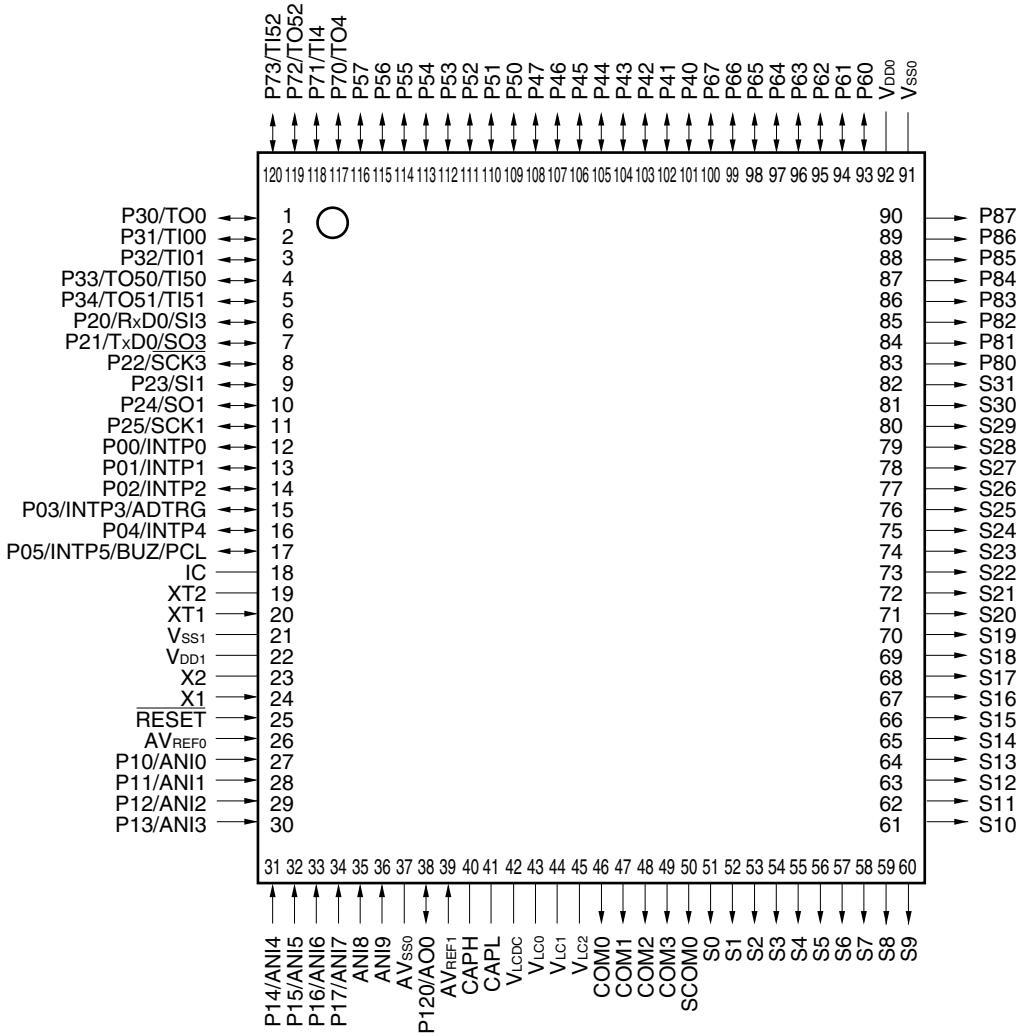
2. Connect the AV_{SS0} pin to V_{SS0}.

Remark When these devices are used in applications that require the reduction of noise generated from an on-chip microcontroller, the implementation of noise measures is recommended, such as supplying V_{DD0} and V_{DD1} independently, connecting V_{SS0} and V_{SS1} independently to ground lines, and so on.

ADTRG:	AD trigger input
ANI0 to ANI9:	Analog input
AO0:	Analog output
AV_{REF0}, AV_{REF1}:	Analog reference voltage
AV_{SS0}:	Analog ground
BUZ:	Buzzer output
CAPH, CAPL:	Capacitor for LCD
COM0 to COM3:	Common output for dynamic display
IC:	Internally connected
INTP0 to INTP5:	External interrupt input
P00 to P05:	Port 0
P10 to P17:	Port 1
P20 to P25:	Port 2
P30 to P34:	Port 3
P40 to P47:	Port 4
P50 to P57:	Port 5
P60 to P67:	Port 6
P70 to P73:	Port 7
P80 to P87:	Port 8
P90 to P97:	Port 9
P120:	Port 12
PCL:	Programmable clock
$\overline{\text{RESET}}$:	Reset
RxD0:	Receive data
S0 to S23:	Segment output
SCK1, $\overline{\text{SCK3}}$:	Serial clock
SCOM0:	Common output for static display
SI1, SI3:	Serial input
SO1, SO3:	Serial output
TI00, TI01, TI04, TI50, TI51, TI52:	Timer input
TO0, TO4, TO50, TO51, TO52:	Timer output
TxD0:	Transmit data
V_{DD0}, V_{DD1}:	Power supply
V_{LC0} to V_{LC2}:	LCD power supply
V_{LCDC}:	Reference voltage control for LCD driver
V_{SS0}, V_{SS1}:	Ground
X1, X2:	Crystal (main system clock)
XT1, XT2:	Crystal (subsystem clock)

1.4.2 μPD780326, 780328

- **120-pin plastic TQFP (fine pitch) (14 × 14)**
  μPD780326GC-×××-9EB, 780326GC-×××-9EV, 780328GC-×××-9EB, 780328GC-×××-9EV



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

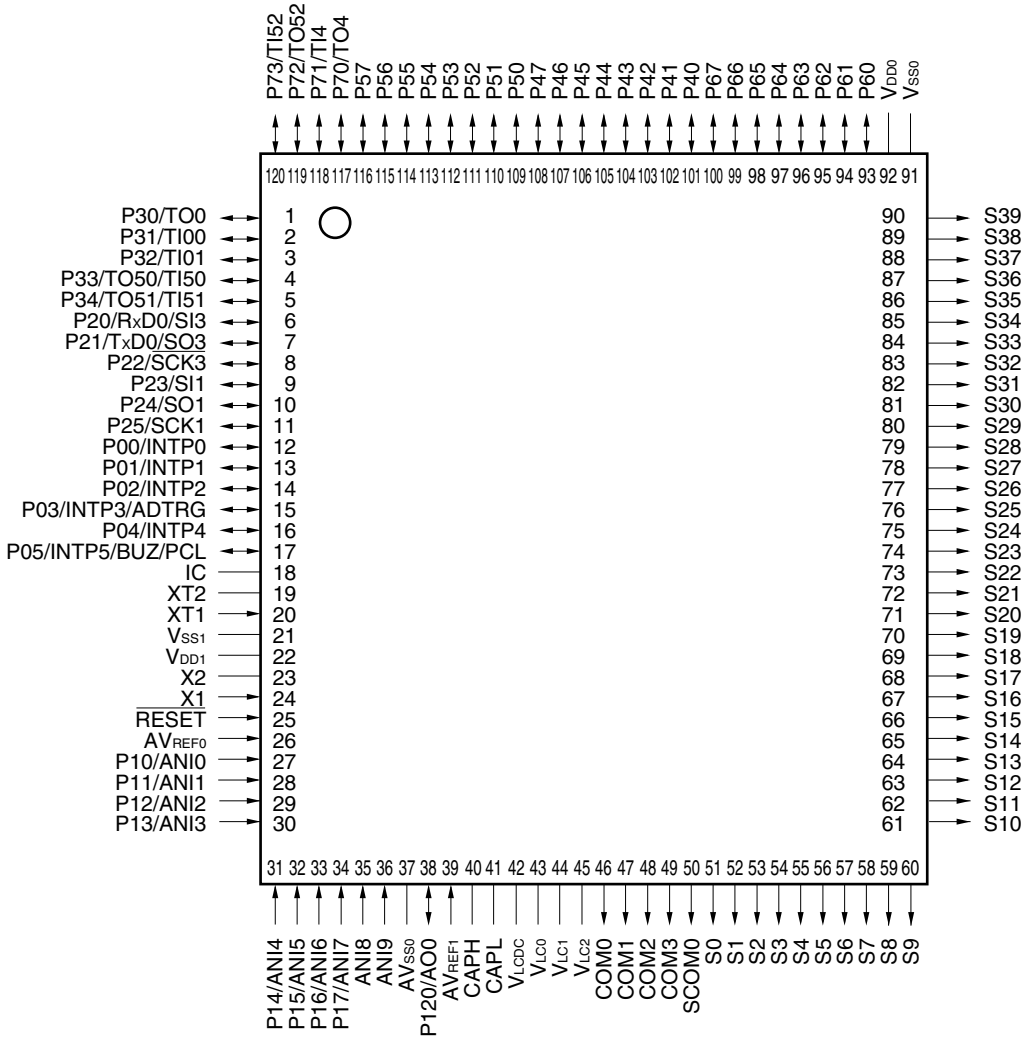
2. Connect the AV_{SS0} pin to V_{SS0}.

Remark When these devices are used in applications that require the reduction of noise generated from an on-chip microcontroller, the implementation of noise measures is recommended, such as supplying V_{DD0} and V_{DD1} independently, connecting V_{SS0} and V_{SS1} independently to ground lines, and so on.

ADTRG:	AD trigger input
ANI0 to ANI9:	Analog input
AO0:	Analog output
AV_{REF0}, AV_{REF1}:	Analog reference voltage
AV_{SS0}:	Analog ground
BUZ:	Buzzer output
CAPH, CAPL:	Capacitor for LCD
COM0 to COM3:	Common output for dynamic display
IC:	Internally connected
INTP0 to INTP5:	External interrupt input
P00 to P05:	Port 0
P10 to P17:	Port 1
P20 to P25:	Port 2
P30 to P34:	Port 3
P40 to P47:	Port 4
P50 to P57:	Port 5
P60 to P67:	Port 6
P70 to P73:	Port 7
P80 to P87:	Port 8
P120:	Port 12
PCL:	Programmable clock
$\overline{RESET}$:	Reset
RxD0:	Receive data
S0 to S31:	Segment output
SCK1, $\overline{SCK3}$:	Serial clock
SCOM0:	Common output for static display
SI1, SI3:	Serial input
SO1, SO3:	Serial output
TI00, TI01, TI04, TI50, TI51, TI52:	Timer input
TO0, TO4, TO50, TO51, TO52:	Timer output
TxD0:	Transmit data
V_{DD0}, V_{DD1}:	Power supply
V_{LC0} to V_{LC2}:	LCD power supply
V_{LCDC}:	Reference voltage control for LCD driver
V_{SS0}, V_{SS1}:	Ground
X1, X2:	Crystal (main system clock)
XT1, XT2:	Crystal (subsystem clock)

1.4.3 μPD780336, 780338

- **120-pin plastic TQFP (fine pitch) (14 × 14)**
  μPD780336GC-×××-9EB, 780336GC-×××-9EV, 780338GC-×××-9EB, 780338GC-×××-9EV



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

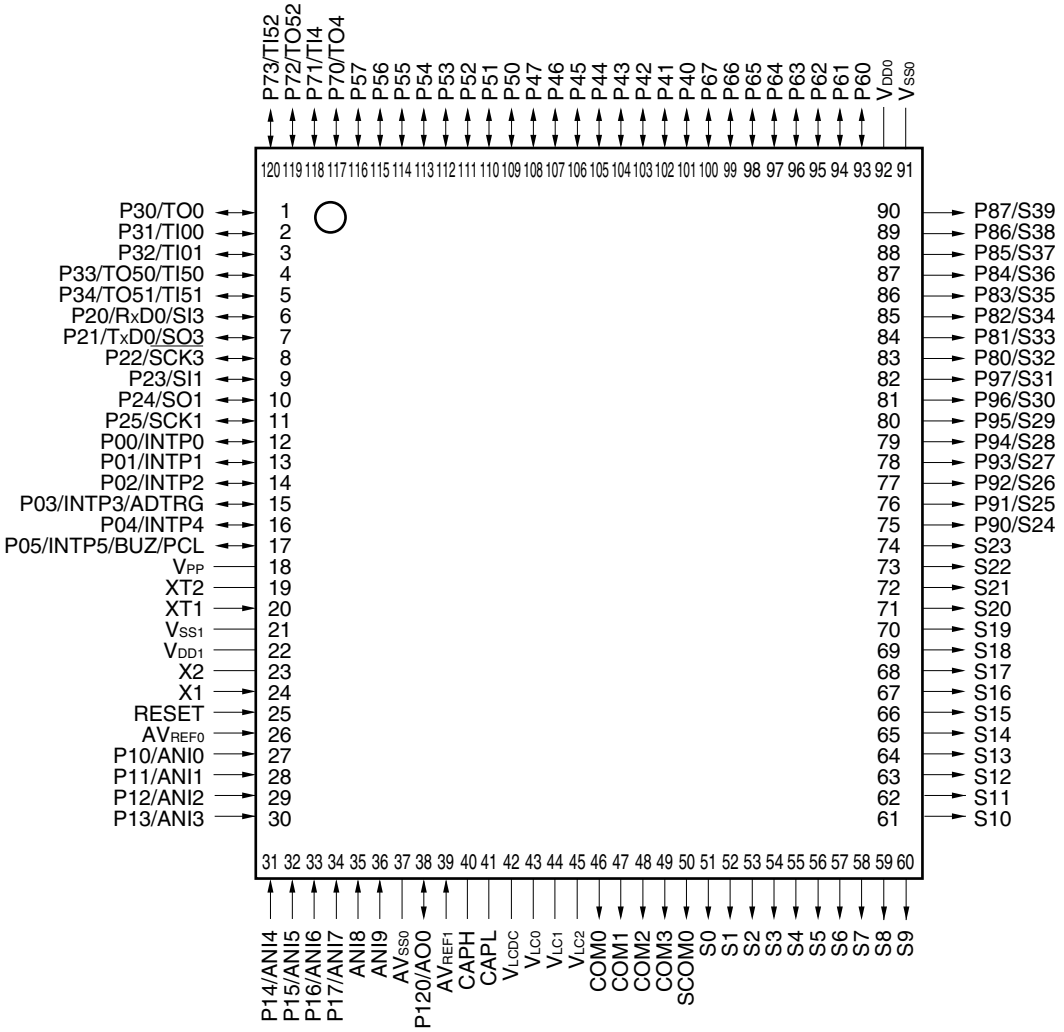
2. Connect the AV_{SS0} pin to V_{SS0}.

Remark When these devices are used in applications that require the reduction of noise generated from an on-chip microcontroller, the implementation of noise measures is recommended, such as supplying V_{DD0} and V_{DD1} independently, connecting V_{SS0} and V_{SS1} independently to ground lines, and so on.

ADTRG:	AD trigger input
ANI0 to ANI9:	Analog input
AO0:	Analog output
AV_{REF0}, AV_{REF1}:	Analog reference voltage
AV_{SS0}:	Analog ground
BUZ:	Buzzer output
CAPH, CAPL:	Capacitor for LCD
COM0 to COM3:	Common output for dynamic display
IC:	Internally connected
INTP0 to INTP5:	External interrupt input
P00 to P05:	Port 0
P10 to P17:	Port 1
P20 to P25:	Port 2
P30 to P34:	Port 3
P40 to P47:	Port 4
P50 to P57:	Port 5
P60 to P67:	Port 6
P70 to P73:	Port 7
P120:	Port 12
PCL:	Programmable clock
$\overline{\text{RESET}}$:	Reset
RxD0:	Receive data
S0 to S39:	Segment output
SCK1, $\overline{\text{SCK3}}$:	Serial clock
SCOM0:	Common output for static display
SI1, SI3:	Serial input
SO1, SO3:	Serial output
TI00, TI01, TI04, TI50, TI51, TI52:	Timer input
TO0, TO4, TO50, TO51, TO52:	Timer output
TxD0:	Transmit data
V_{DD0}, V_{DD1}:	Power supply
V_{LC0} to V_{LC2}:	LCD power supply
V_{LCDC}:	Reference voltage control for LCD driver
V_{SS0}, V_{SS1}:	Ground
X1, X2:	Crystal (main system clock)
XT1, XT2:	Crystal (subsystem clock)

1.4.4 μPD78F0338

- **120-pin plastic TQFP (fine pitch) (14 × 14)**
  μPD78F0338GC-9EB, 78F0338GC-9EV



Cautions 1. Connect the V_{PP} pin directly to V_{SS0} or V_{SS1}.

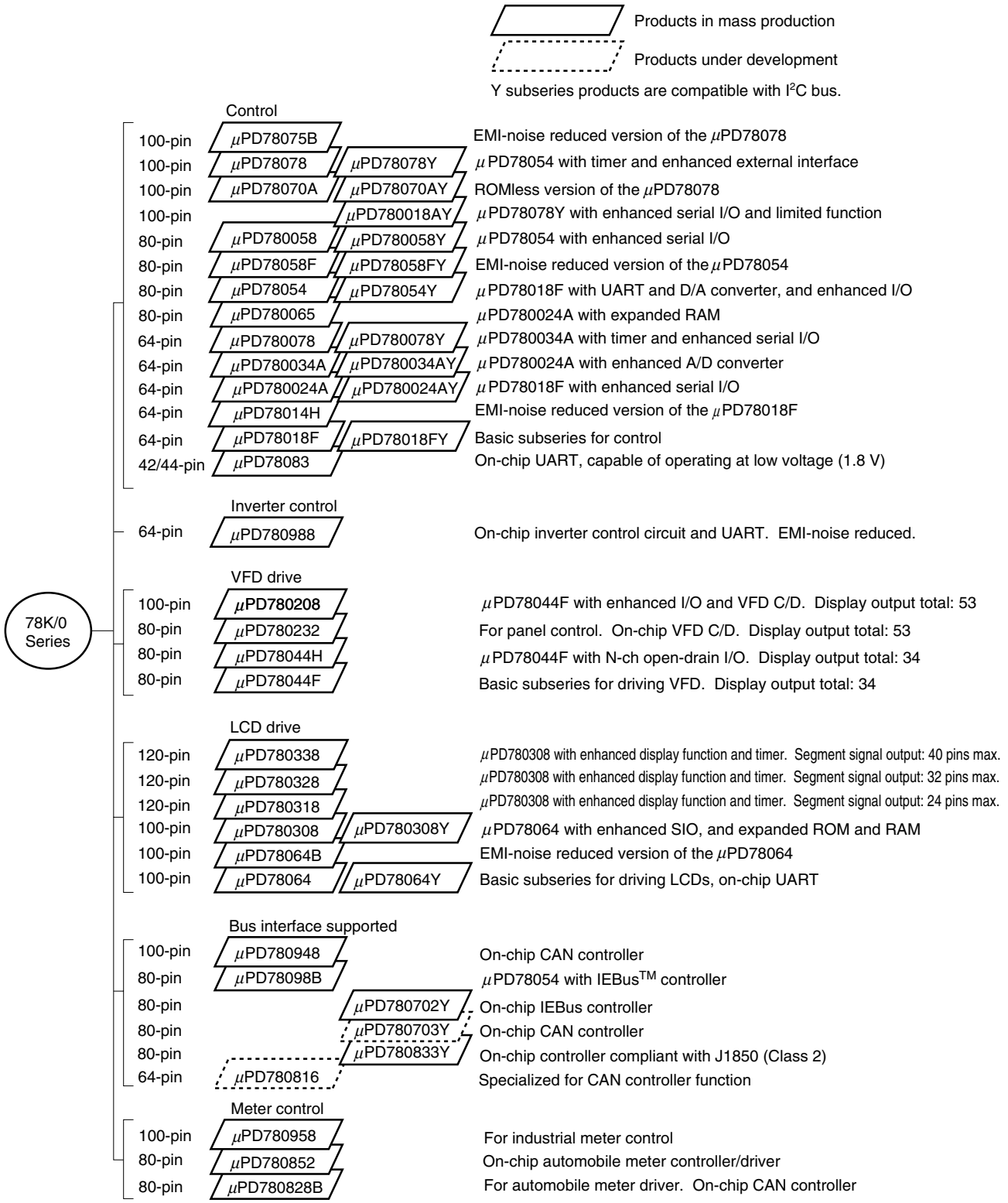
2. Connect the AV_{SS0} pin to V_{SS0}.

Remark When these devices are used in applications that require the reduction of noise generated from an on-chip microcontroller, the implementation of noise measures is recommended, such as supplying V_{DD0} and V_{DD1} independently, connecting V_{SS0} and V_{SS1} independently to ground lines, and so on.

ADTRG:	AD trigger input
ANI0 to ANI9:	Analog input
AO0:	Analog output
AV_{REF0}, AV_{REF1}:	Analog reference voltage
AV_{SS0}:	Analog ground
BUZ:	Buzzer output
CAPH, CAPL:	Capacitor for LCD
COM0 to COM3:	Common output for dynamic display
INTP0 to INTP5:	External interrupt input
P00 to P05:	Port 0
P10 to P17:	Port 1
P20 to P25:	Port 2
P30 to P34:	Port 3
P40 to P47:	Port 4
P50 to P57:	Port 5
P60 to P67:	Port 6
P70 to P73:	Port 7
P80 to P87:	Port 8
P90 to P97:	Port 9
P120:	Port 12
PCL:	Programmable clock
$\overline{RESET}$:	Reset
RxD0:	Receive data
S0 to S39:	Segment output
SCK1, $\overline{SCK3}$:	Serial clock
SCOM0:	Common output for static display
SI1, SI3:	Serial input
SO1, SO3:	Serial output
TI00, TI01, TI04, TI50, TI51, TI52:	Timer input
TO0, TO4, TO50, TO51, TO52:	Timer output
TxD0:	Transmit data
V_{DD0}, V_{DD1}:	Power supply
V_{LC0} to V_{LC2}:	LCD power supply
V_{LCDC}:	Reference voltage control for LCD driver
V_{PP}:	Programming power supply
V_{SS0}, V_{SS1}:	Ground
X1, X2:	Crystal (main system clock)
XT1, XT2:	Crystal (subsystem clock)

1.5 78K/0 Series Lineup

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



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

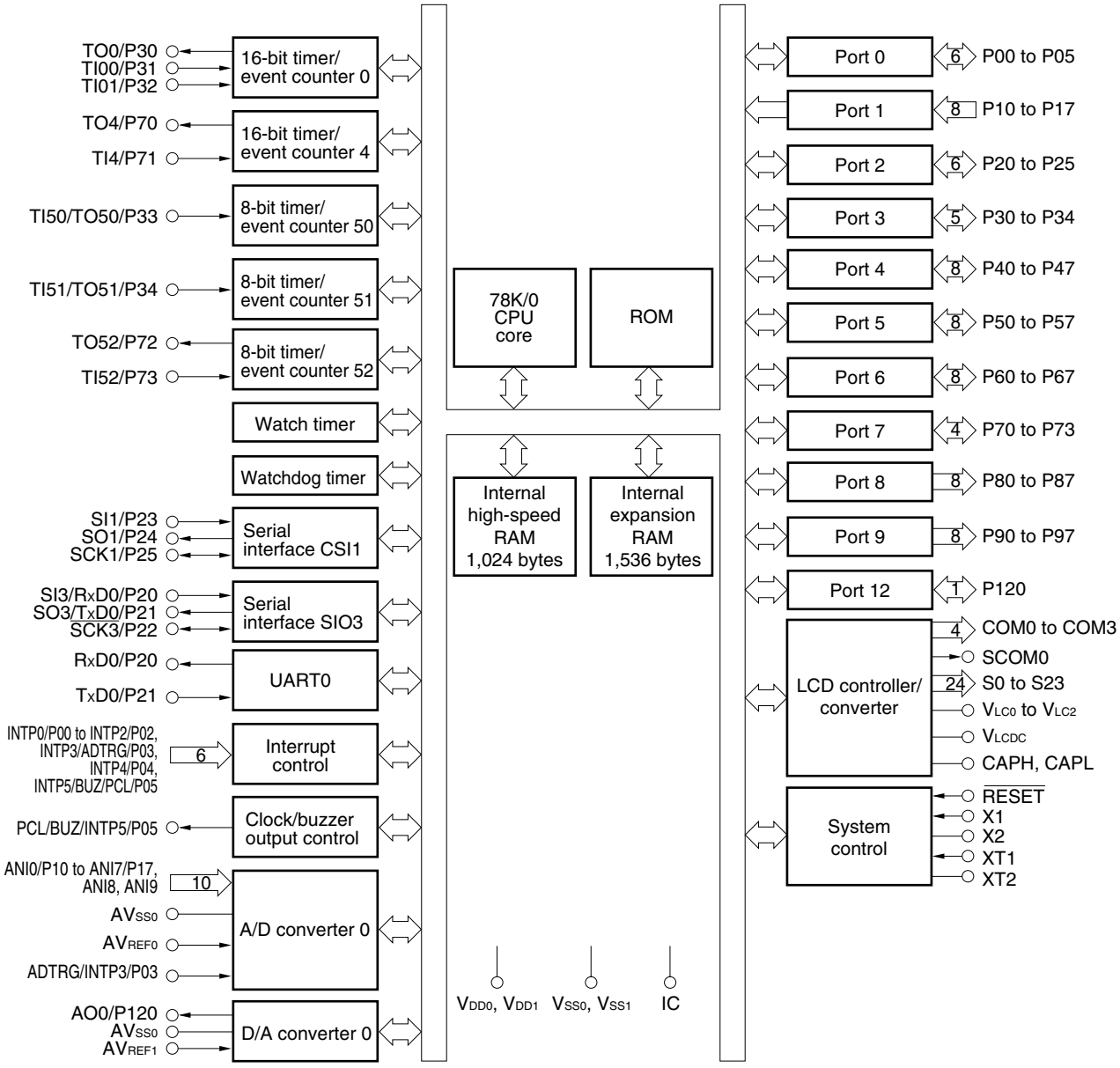
The major functional differences between the subseries are shown below.

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

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

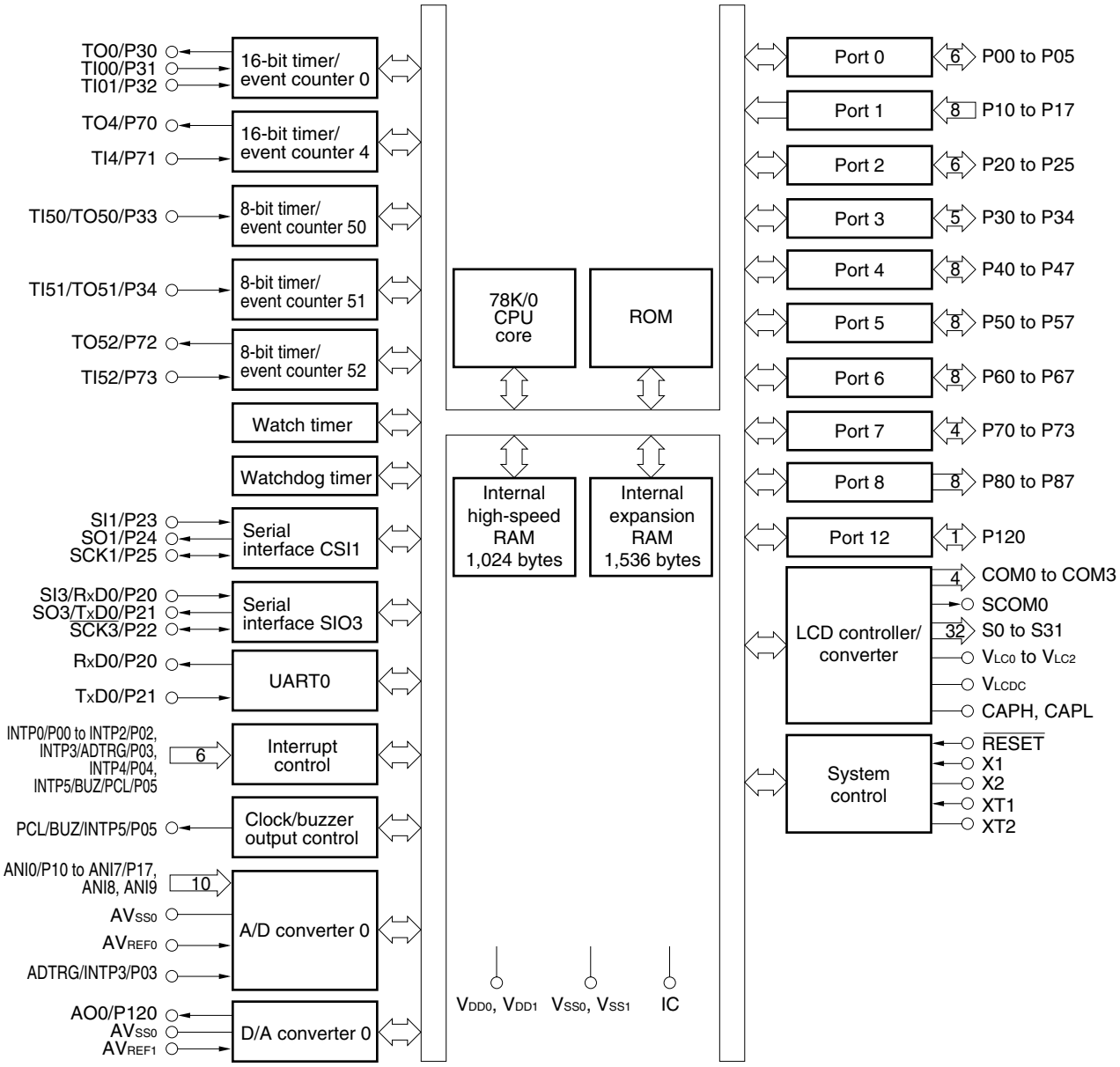
1.6 Block Diagram

1.6.1 μPD780316, 780318



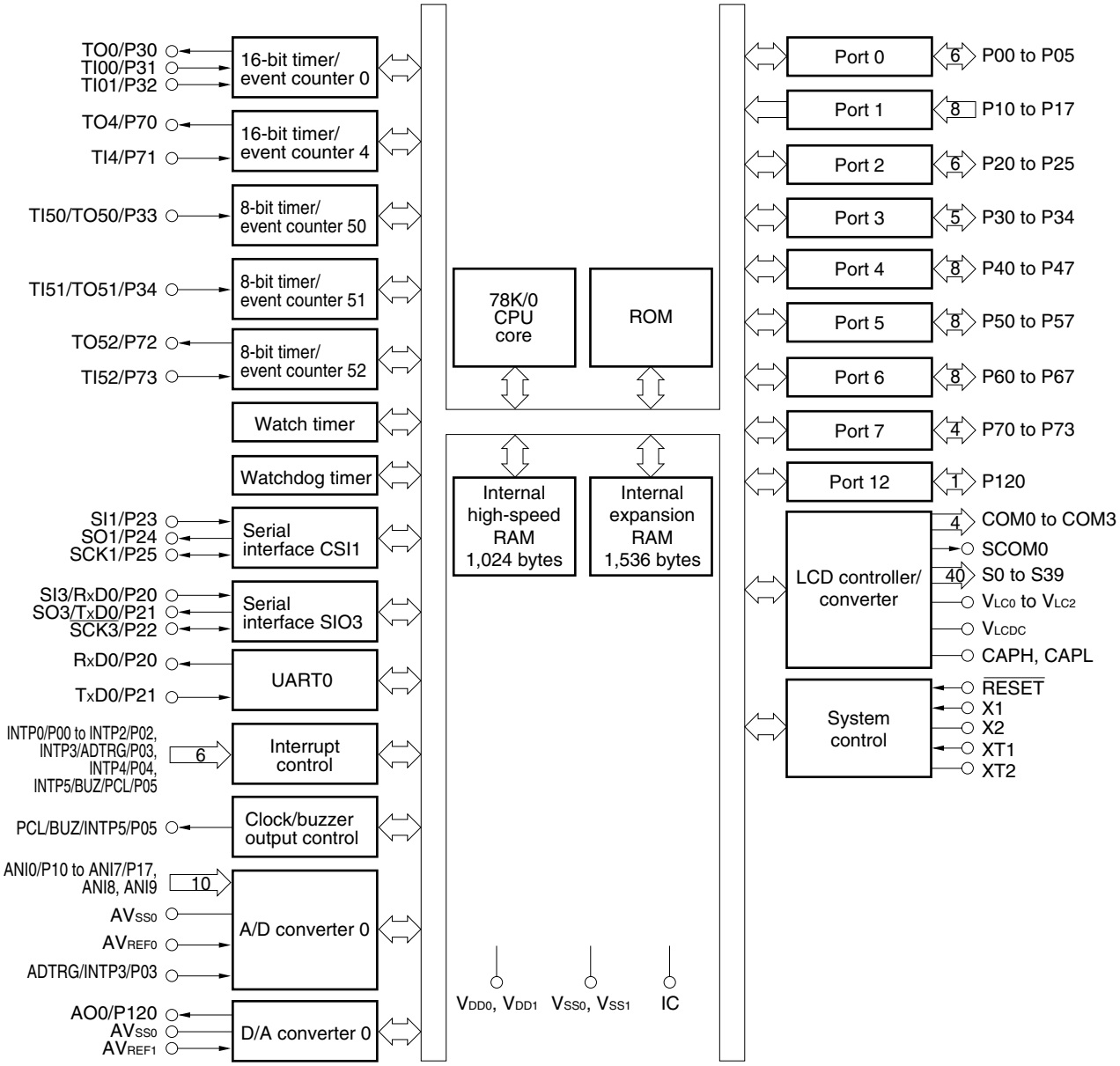
Remark The internal ROM capacity varies depending on the product.

1.6.2 μPD780326, 780328



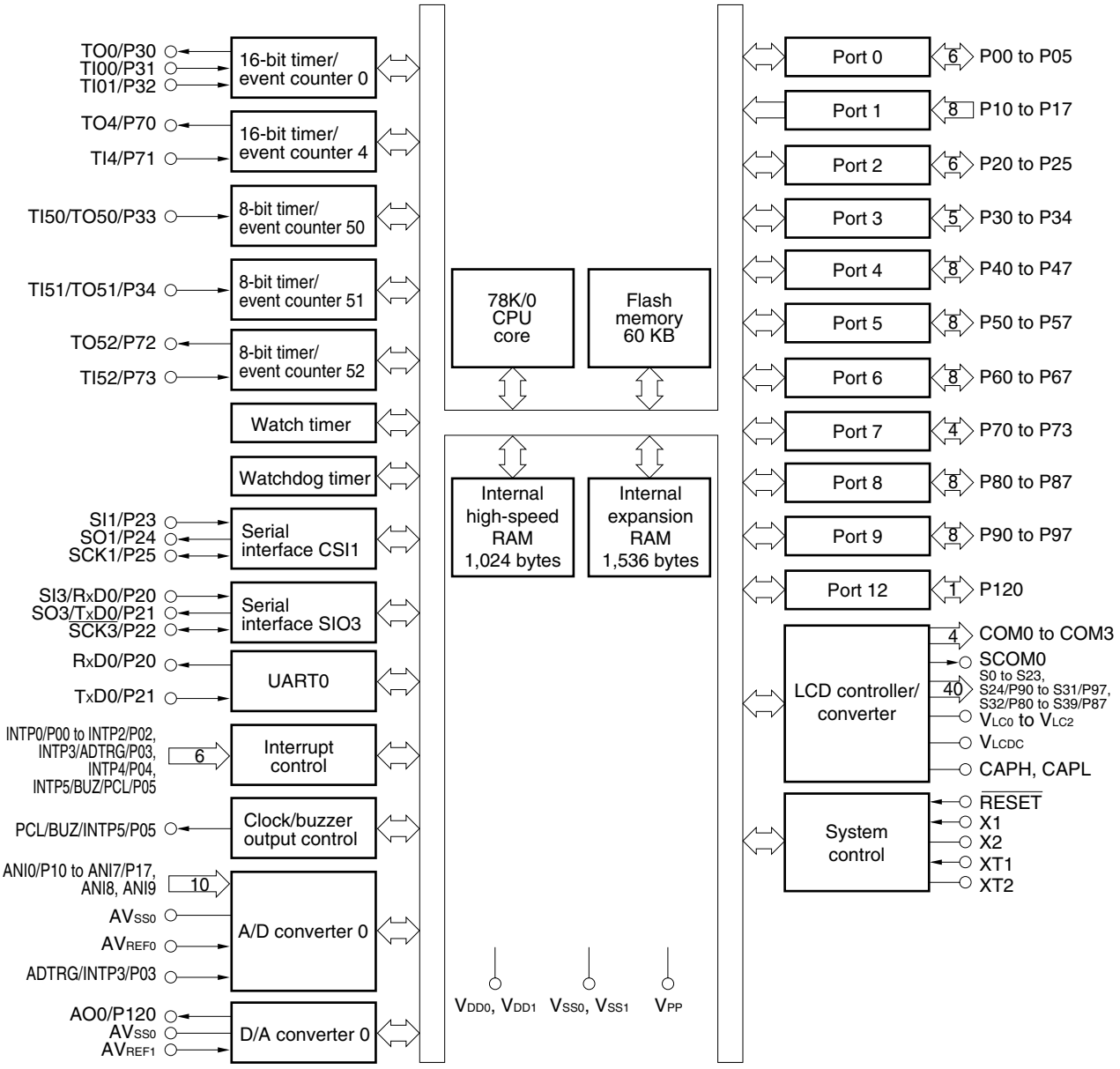
Remark The internal ROM capacity varies depending on the product.

1.6.3 μPD780336, 780338



Remark The internal ROM capacity varies depending on the product.

1.6.4 μPD78F0338



1.7 Outline of Functions

(1/2)

Item \ Part Number		μPD780316	μPD780318	μPD780326	μPD780328	μPD780336	μPD780338	μPD78F0338
Internal memory	ROM	48 KB (mask ROM)	60 KB (mask ROM)	48 KB (mask ROM)	60 KB (mask ROM)	48 KB (mask ROM)	60 KB (mask ROM)	60 KB[Note] (flash memory)
	High-speed RAM	1,024 bytes						
	Expansion RAM	1,536 bytes						
	LCD display RAM	40 × 8 bits						
Memory space		64 KB						
General-purpose registers		8 bits × 32 registers (8 bits × 8 registers × 4 banks)						
Minimum instruction execution time		Function to change minimum instruction execution time provided						
	When main system clock selected	0.2 μs/0.4 μs/0.8 μs/1.6 μs/3.2 μs (@V_{DD} = 5 V, f_X = 10 MHz)						
	When subsystem clock selected	122 μs (@f_{XT} = 32.768 kHz)						
Instruction set		• 16-bit operation • Multiply/divide (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulate (set, reset, test, and Boolean operation) • BCD adjust, etc.						
I/O ports		70		62		54		70
	CMOS input	8						
	CMOS output	16		8		None		16 (alternate with segment pin)
	CMOS I/O	42						
	N-ch open-drain I/O (15 V)	4						
A/D converter		• 10-bit resolution × 10 channels • Low-voltage operation: AV_{REF0} = 1.8 to 5.5 V						
D/A converter		8-bit resolution × 1 channel						
LCD controller/driver		• LCD reference voltage generator: booster type (×3 only) • Fine tuning of LCD reference voltage possible with external resistor • Blinking display possible (blinking interval can be selected: 0.5 s or 1 s) • Static display and dynamic display (1/3 bias only) can be used simultaneously (Static display up to 12 segments)						
	Segment signal output	24 max.		32 max.		40 max.		40 max. (when alternate with port pins: 16)
	Common signal output	4 max. (dynamic display), 1 (static display)						

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

(2/2)

Item \ Part Number	μPD780316	μPD780318	μPD780326	μPD780328	μPD780336	μPD780338	μPD78F0338
Serial interface	• 3-wire serial I/O mode/UART mode selectable[Note]: 1 channel • 3-wire serial I/O mode: 1 channel						
Timer	• 16-bit timer/event counter: 2 channels • 8-bit timer/event counter: 3 channels • Watch timer: 1 channel • Watchdog timer: 1 channel						
Timer outputs	5 (8-bit PWM output possible: 3)						
Clock output	• 78.1 kHz, 156 kHz, 313 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5 MHz, 10 MHz (@10 MHz operation with main system clock) • 32.768 kHz (@32.768 kHz operation with subsystem clock)						
Buzzer output	• 1.22 kHz, 2.44 kHz, 4.88 kHz, 9.77 kHz (@10 MHz operation with main system clock)						
Vectored interrupt sources – Maskable	Internal: 15, external: 7						
Vectored interrupt sources – Non-maskable	Internal: 1						
Vectored interrupt sources – Software	1						
ROM correction	Provided						
Power supply voltage	V_{DD} = 1.8 to 5.5 V						
Operating ambient temperature	T_A = –40 to +85°C						
Package	120-pin plastic TQFP (fine pitch) (14 × 14)						

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

The following table outlines the timers/event counters (for details, refer to **CHAPTER 6 16-BIT TIMER/EVENT COUNTER 0**, **CHAPTER 7 16-BIT TIMER/EVENT COUNTER 4**, **CHAPTER 8 8-BIT TIMER/EVENT COUNTERS 50**, **51**, **52**, **CHAPTER 9 WATCH TIMER**, and **CHAPTER 10 WATCHDOG TIMER**):

		16-Bit Timer/ Event Counter 0	16-Bit Timer/ Event Counter 4	8-Bit Timer/ Event Counters 50, 51, 52	Watch Timer	Watchdog Timer
Operation mode	Interval timer	1 channel	1 channel	3 channels	1 channel[Note 1]	1 channel[Note 2]
	External event counter	○	○	○	—	—
Function	Timer output	○	○	○	—	—
	PPG output	○	—	—	—	—
	PWM output	—	—	○	—	—
	Pulse width measurement	○	—	—	—	—
	Square wave output	○	○	○	—	—
	Interrupt request	○	○	○	○	○

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

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

1.8 Mask Options

The mask ROM versions (μPD780316, 780318, 780326, 780328, 780336, and 780338) provide pull-up resistor mask options which allow users to specify whether to connect a pull-up resistor to a specific port pin when the user places an order for device production. Using the mask option when pull-up resistors are required reduces the number of components to add to the device, resulting in board space saving.

The mask options provided in the μPD780318, 780328, and 780338 Subseries are shown in Table 1-1.

Table 1-1. Mask Options of Mask ROM Versions

Pin Names	Mask Option
P60 to P63	Pull-up resistor connection can be specified in 1-bit units.

CHAPTER 2 PIN FUNCTIONS

2.1 Pin Function List

(1) Port pins (1/2)

Pin Name	I/O	Function		After Reset	Alternate Function
P00	I/O	Port 0 6-bit I/O port Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be used by setting software.		Input	INTP0
P01					INTP1
P02					INTP2
P03					INTP3/ADTRG
P04					INTP4
P05					INTP5/BUZ/PCL
P10 to P17	Input	Port 1 8-bit input only port.		Input	ANI0 to ANI7
P20	I/O	Port 2 6-bit I/O port Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be used by setting software.		Input	RxD0/SI3
P21					TxD0/SO3
P22					$\overline{\text{SCK3}}$
P23					SI1
P24					SO1
P25					SCK1
P30	I/O	Port 3 5-bit I/O port Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be used by setting software.		Input	TO0
P31					TI00
P32					TI01
P33					TO50/TI50
P34					TO51/TI51
P40 to P47	I/O	Port 4 8-bit I/O port Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be used by setting software. Interrupt request flag (KRIF) is set to 1 by falling edge detection.		Input	—
P50 to P57	I/O	Port 5 8-bit I/O port Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be used by setting software.		Input	—
P60 to P63	I/O	Port 6 8-bit I/O port Input/output mode can be specified in 1-bit units. LEDs can be driven directly.	Medium-voltage N-ch open-drain I/O port On-chip pull-up resistor can be specified by mask option (mask ROM version only).	Input	—
P64 to P67			An on-chip pull-up resistor can be used by setting software.		

(1) Port pins (2/2)

Pin Name	I/O	Function	After Reset	Alternate Function
P70	I/O	Port 7 4-bit I/O port Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be used by setting software.	Input	TO4
P71				TI4
P72				TO52
P73				TI52
P80 to P87[Note]	Output	Port 8 8-bit output only port	Output	S32 to S39[Note]
P90 to P97[Note]	Output	Port 9 8-bit output only port	Output	S24 to S31[Note]
P120	I/O	Port 12 1-bit I/O port Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be used by setting software.	Input	AO0

Note Ports 8 and 9 vary depending on the product.

	Port 8	Port 9
μPD780316, 780318	P80 to P87 (without alternate pin)	P90 to P97 (without alternate pin)
μPD780326, 780328		None
μPD780336, 780338	None	
μPD78F0338	P80/S32 to P87/S39	P90/S24 to P97/S31

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

Pin Name	I/O	Function	After Reset	Alternate Function
INTP0	Input	External interrupt request input with specifiable valid edges (rising edge, falling edge, both rising and falling edges)	Input	P00
INTP1				P01
INTP2				P02
INTP3				P03/ADTRG
INTP4				P04
INTP5				P05/BUZ/PCL
SI1	Input	Serial interface serial data input	Input	P23
SI3				P20/RxD0
SO1	Output	Serial interface serial data output	Input	P24
SO3				P21/TxD0
SCK1	I/O	Serial interface serial clock input/output	Input	P25
$\overline{SCK3}$				P22
RxD0	Input	Asynchronous serial interface serial data input	Input	P20/SI3
TxD0	Output	Asynchronous serial interface serial data output	Input	P21/SO3
TI00	Input	External count clock input to 16-bit timer/event counter 0 Capture trigger input to capture registers (CR00, CR01) of 16-bit timer/event counter 0	Input	P31
TI01		Capture trigger input to capture register (CR00) of 16-bit timer/event counter 0		P32
TI4		External count clock input to 16-bit timer/event counter 4		P71
TO0	Output	16-bit timer/event counter 0 output	Input	P30
TO4		16-bit timer/event counter 4 output		P70
TI50	Input	External count clock input to 8-bit timer/event counter 50	Input	P33/TO50
TI51		External count clock input to 8-bit timer/event counter 51		P34/TO51
TI52		External count clock input to 8-bit timer/event counter 52		P73
TO50	Output	8-bit timer/event counter 50 output	Input	P33/TI50
TO51		8-bit timer/event counter 51 output		P34/TI51
TO52		8-bit timer/event counter 52 output		P72
PCL	Output	Clock output (for main system clock, subsystem clock trimming)	Input	P05/INTP5/BUZ
BUZ	Output	Buzzer output	Input	P05/INTP5/PCL
ANI0 to ANI7	Input	Analog input of A/D converter	Input	P10 to P17
ANI8, ANI9				—
ADTRG	Input	Trigger signal input of A/D converter	Input	P03/INTP3
AV_{REF0}	Input	Reference voltage input of A/D converter	—	—
AO0	Output	Analog output of D/A converter	Input	P120
AV_{REF1}	Input	Reference voltage input of D/A converter	—	—
AV_{SS0}	—	Ground potential for A/D converter and D/A converter. Supply the same potential as that of V_{SS0} or V_{SS1}.	—	—

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

Pin Name	I/O	Function	After Reset	Alternate Function
S0 to S11[Note]	Output	LCD controller/driver segment signal output (Static and dynamic display can be selected)	Output	—
S12 to S23[Note]		LCD controller/driver segment signal output (for dynamic display)		—
S24 to S31[Note]				P90 to P97[Note]
S32 to S39[Note]				P80 to P87[Note]
COM0 to COM3	Output	LCD controller/driver common signal output (for dynamic display)	Output	—
SCOM0	Output	LCD controller/driver common signal output (for static display)	Output	—
V_{LC0} to V_{LC2}	—	LCD driving voltage • V_{LC0}: Three times V_{LC2} output voltage • V_{LC1}: Two times V_{LC2} output voltage • V_{LC2}: Reference voltage	—	—
V_{LCDC}	—	LCD controller/driver reference voltage adjustment	—	—
CAPH, CAPL	—	Booster capacitor connection for LCD drive voltage	—	—
$\overline{\text{RESET}}$	Input	System reset input	—	—
X1	Input	Crystal connection for main system clock oscillation	—	—
X2	—		—	—
XT1	Input	Crystal connection for subsystem clock oscillation	—	—
XT2	—		—	—
V_{DD0}	—	Positive power supply for ports	—	—
V_{DD1}	—	Positive power supply other than ports	—	—
V_{SS0}	—	Ground potential for ports	—	—
V_{SS1}	—	Ground potential other than ports	—	—
IC	—	Internally connected. Connect directly to V_{SS0} or V_{SS1}.	—	—
V_{PP}	—	High-voltage application for program write/verify. Connect directly to V_{SS0} or V_{SS1} in normal operation mode.	—	—

Note Segment signal output pins vary depending on the product.

- μPD780316, 780318: S0 to S23 (without alternate pin)
- μPD780326, 780328: S0 to S31 (without alternate pin)
- μPD780336, 780338: S0 to S39 (without alternate pin)
- μPD78F0338: S0 to S39 (S24 to S31 and P90 to P97, and S32 to S39 and P80 to P87 are alternate-function pins. These functions can be switched with port functions in 8-bit units.)

2.2 Description of Pin Functions

2.2.1 P00 to P05 (Port 0)

These are 6-bit I/O ports. Besides serving as I/O ports, they function as an external interrupt request input, A/D converter external trigger input, clock output, and buzzer output function.

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

(1) Port mode

These ports function as 6-bit I/O ports. They can be specified as input or output ports in 1-bit units with port mode register 0 (PM0). On-chip pull-up resistors can be used by setting pull-up resistor option register 0 (PU0).

(2) Control mode

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

(a) INTP0 to INTP5

INTP0 to INTP5 are external interrupt request input pins which can specify valid edges (rising edge, falling edge, and both rising and falling edges).

(b) ADTRG

A/D converter external trigger input pin.

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

(c) PCL

Clock output pin.

(d) BUZ

Buzzer output pin.

2.2.2 P10 to P17 (Port 1)

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

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

(1) Port mode

These ports function as 8-bit input only ports.

(2) Control mode

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

2.2.3 P20 to P25 (Port 2)

These are 6-bit I/O ports. Besides serving as I/O ports, they function as serial interface data I/O and clock I/O. The following operation modes can be specified in 1-bit units.

(1) Port mode

These ports function as 6-bit I/O ports. They can be specified as input or output ports in 1-bit units with port mode register 2 (PM2). On-chip pull-up resistors can be used by setting pull-up resistor option register 2 (PU2).

(2) Control mode

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

(a) SI1, SI3, SO1, and SO3

Serial interface serial data I/O pins.

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

Serial interface serial clock I/O pins.

(c) RxD0 and TxD0

Asynchronous serial interface serial data I/O pins.

2.2.4 P30 to P34 (Port 3)

These are 5-bit I/O ports. Besides serving as I/O ports, they function as timer I/O.

(1) Port mode

These ports function as 5-bit I/O ports. They can be specified as input or output ports in 1-bit units with port mode register 3 (PM3). On-chip pull-up resistors can be used by setting pull-up resistor option register 3 (PU3). P31 and P32 are also capture trigger signal input pins to the capture registers (CR00 and CR01) of the 16-bit timer/event counter 0 with a valid edge input.

(2) Control mode

These ports function as timer I/O.

(a) TI00

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

(b) TI01

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

(c) TI50 and TI51

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

(d) TO0, TO50, and TO51

Timer output pins.

2.2.5 P40 to P47 (Port 4)

These are 8-bit I/O ports. They can be specified as input or output ports in 1-bit units with port mode register 4 (PM4). On-chip pull-up resistors can be used by setting pull-up resistor option register 4 (PU4).

Interrupt request flag (KRIF) can be set to 1 by detecting the falling edge. The number of ports to detect the falling edge can be selected at either four (P40 to P43) or eight (P40 to P47) by setting bit 0 (KRSEL0) of the key return switching register (KRSEL).

Cautions 1. Be sure to set memory expansion mode register (MEM) to 01H when using falling edge detection interrupt (INTKR).

2. If the number of key returns is set to four, the key return function cannot be evaluated with in-circuit emulator.

2.2.6 P50 to P57 (Port 5)

These are 8-bit I/O ports. They can be specified as input or output ports in 1-bit units with port mode register 5 (PM5). On-chip pull-up resistors can be used by setting pull-up resistor option register 5 (PU5).

2.2.7 P60 to P67 (Port 6)

These are 8-bit I/O ports. They can be specified as input or output ports in 1-bit units with port mode register 6 (PM6). Port 6 can drive LEDs directly.

P60 to P63 are medium-voltage N-ch open-drain. On-chip pull-up resistors can be used by a mask option with the mask ROM versions.

P64 to P67 can use on-chip pull-up resistors by setting pull-up resistor option register 6 (PU6).

2.2.8 P70 to P73 (Port 7)

These are 4-bit I/O ports. Besides serving as I/O ports, they function as timer I/O.

(1) Port mode

These ports function as 4-bit I/O ports. They can be specified as input or output ports in 1-bit units with port mode register 7 (PM7). On-chip pull-up resistors can be used by setting pull-up resistor option register 7 (PU7).

(2) Control mode

These ports function as timer I/O.

(a) TI4

External count clock input pin to 16-bit timer/event counter 4.

(b) TI52

External count clock input pin to 8-bit timer/event counter 52.

(c) TO4 and TO52

Timer output pins.

2.2.9 P80 to P87 (Port 8)[Notes 1, 2]

These are 8-bit output-only ports. Besides serving as output ports, they function as segment signal output (for dynamic display) of the LCD controller/driver. Either the output port or segment signal output function can be selected by setting the pin function switching register 8 (PF8)[Note 3].

(1) Port mode

These ports function as 8-bit output-only ports.

(2) Control mode

These ports function as segment signal output pins (S32 to S39) (for dynamic display) of the LCD controller/driver.

★ **Notes 1.** These ports are not provided in the μPD780336 and 780338.

★ **2.** Port 8 and segment signal output pins vary depending on the product.

	Pin Function
μPD780316, 780318	P80 to P87
μPD780326, 780328	
μPD780336, 780338	S32 to S39
μPD78F0338	P80/S32 to P87/S39

3. Pin function switching register 8 (PF8) is provided for the μPD78F0338 only.

2.2.10 P90 to P97 (Port 9)[Notes 1, 2]

These are 8-bit output-only ports. Besides serving as output ports, they function as segment signal output (for dynamic display) of the LCD controller/driver. Either the output port or segment signal output function can be selected by setting the pin function switching register 9 (PF9)[Note 3].

(1) Port mode

These ports function as 8-bit output-only ports.

(2) Control mode

These ports function as segment signal output pins (S24 to S31) (for dynamic display) of the LCD controller/driver.

★ **Notes 1.** These ports are not provided in the μPD780326, 780328, 780336, and 780338.

★ **2.** Port 9 and segment signal output pins vary depending on the product.

	Pin Function
μPD780316, 780318	P90 to P97
μPD780326, 780328	S24 to S31
μPD780336, 780338	
μPD78F0338	P90/S24 to P97/S31

3. Pin function switching register 9 (PF9) is provided for the μPD78F0338 only.

2.2.11 P120 (Port 12)

This is a 1-bit I/O port. Besides serving as an I/O port, this port functions as an analog output of the D/A converter.

(1) Port mode

This is a 1-bit I/O port. It can be specified as an input or output port in 1-bit units with port mode register 12 (PM12). An on-chip pull-up resistor can be used by setting pull-up resistor option register 12 (PU12).

(2) Control mode

This port functions as an analog output pin of the D/A converter (AO0).

Caution Set this port to input mode using the port mode register 12 and disconnect pull-up resistor before using the D/A converter.

2.2.12 ANI0 to ANI9

These are A/D converter analog input pins. ANI0 to ANI7 are also used with P10 to P17.

2.2.13 AV_{REF0}

This is an A/D converter reference voltage input pin. Supply power when using an A/D converter because this pin also functions as an analog power supply.

When A/D converter is not used, connect this pin to V_{SS0} or V_{SS1} pin.

2.2.14 AV_{REF1}

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

When D/A converter is not used, connect this pin to V_{DD0} or V_{DD1} pin.

2.2.15 AV_{SS0}

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

2.2.16 S0 to S39[Note]

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

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

Note Segment signal output pins vary depending on the product.

- μPD780316, 780318: S0 to S23 (without alternate pin)
- μPD780326, 780328: S0 to S31 (without alternate pin)
- μPD780336, 780338: S0 to S39 (without alternate pin)
- μPD78F0338: S0 to S39 (S24 to S31 and P90 to P97, and S32 to S39 and P80 to P87 are alternate-function pins. These functions can be switched with port functions in 8-bit units.)

2.2.17 COM0 to COM3

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

2.2.18 SCOM0

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

2.2.19 V_{LC0} to V_{LC2}

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

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

2.2.20 V_{LCDC}

This is an LCD controller/driver reference voltage adjustment pin. This pin is used for fine-tuning the LCD driving voltage by connecting resistors between V_{LC2} and V_{LCDC} externally.

2.2.21 CAPH and CAPL

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

2.2.22 $\overline{\text{RESET}}$

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

2.2.23 X1 and X2

Crystal resonator connect pins for main system clock oscillation.

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

2.2.24 XT1 and XT2

Crystal resonator connect pins for subsystem clock oscillation.

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

2.2.25 V_{DD0} and V_{DD1}

V_{DD0} is a positive power supply port pin.

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

2.2.26 V_{SS0} and V_{SS1}

V_{SS0} is a ground potential port pin.

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

2.2.27 V_{PP} (flash memory versions only)

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

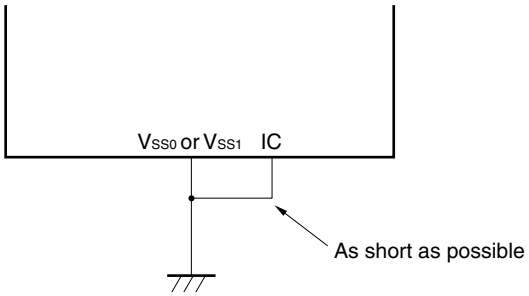
Connect directly to V_{SS0} or V_{SS1} in the normal operating mode.

2.2.28 IC (mask ROM version only)

The IC (Internally Connected) pin is provided to set the test mode to check the μPD780318, 780328, 780338 Subseries at delivery. Connect it directly to the V_{SS0} or V_{SS1} pin with the shortest possible wire in the normal operation mode.

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

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



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

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

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

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00/INTP0 to P02/INTP2	8-C	I/O	Input: Independently connect to V_{SS0} via a resistor. Output: Leave open.
P03/INTP3/ADTRG			
P04/INTP4			
P05/INTP5/BUZ/PCL			
P10/ANI0 to P17/ANI7	25	Input	Connect to V_{DD0} or V_{SS0}.
P20/RxD0/SI3	8-C	I/O	Input: Independently connect to V_{DD0} or V_{SS0} via a resistor. Output: Leave open.
P21/TxD0/SO3	5-H		
P22/$\overline{\text{SCK3}}$	8-C		
P23/SI1			
P24/SO1	5-H		
P25/SCK1	8-C		
P30/TO0	5-H		
P31/TI00	8-C		
P32/TI01			
P33/TO50/TI50			
P34/TO51/TI51			
P40 to P47	5-H		
P50 to P57			
★ P60 to P63 (for mask ROM version)	13-J		Input: Connect to V_{SS0}. Output: Set to low-level output and leave open.
P60 to P63 (for flash memory version)	13-K		
P64 to P67	5-H		Input: Independently connect to V_{DD0} or V_{SS0} via a resistor. Output: Leave open.
P70/TO4			
P71/TI4	8-C		
P72/TO52	5-H		
P73/TI52	8-C		

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

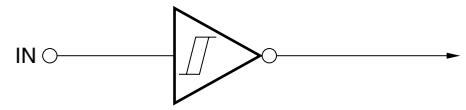
Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P80 to P87[Note] (for mask ROM version)	4-B	Output	Leave open.
★ P80/S32 to P87/S39 (for flash memory version)	31		Set to output and leave open.
P90 to P97[Note] (for mask ROM version)	4-B		Leave open.
★ P90/S24 to P97/S31 (for flash memory version)	31		Set to output and leave open.
P120/AO0	12-C	I/O	Input: Independently connect to V_{SS0} via a resistor. Output: Leave open.
ANI8, ANI9	25	Input	Connect to V_{DD0} or V_{SS0}.
S0 to S23[Note]	17-D	Output	Leave open.
S24 to S39[Note] (for mask ROM version)			
COM0 to COM3	18-B		
SCOM0			
V_{LC0} to V_{LC2}	—	—	
V_{LCDC}			
CAPH, CAPL			
$\overline{\text{RESET}}$	2	Input	—
XT1	16	Input	Connect to V_{DD0} or V_{DD1}.
XT2		—	Leave open.
AV_{REF0}	—	Input	Connect to V_{SS0} or V_{SS1}.
AV_{REF1}			Connect to V_{DD0} or V_{DD1}.
AV_{SS0}		—	Connect to V_{SS0} or V_{SS1}.
IC			Connect to V_{SS0} or V_{SS1} directly.
V_{PP}			

Note Ports 8 and 9 and segment signal output pins vary depending on the mask ROM version.

	Port 8	Port 9	Segment Signal Output
μPD780316, 780318	P80 to P87	P90 to P97	S0 to S23
μPD780326, 780328		None	S0 to S31
μPD780336, 780338	None		S0 to S39

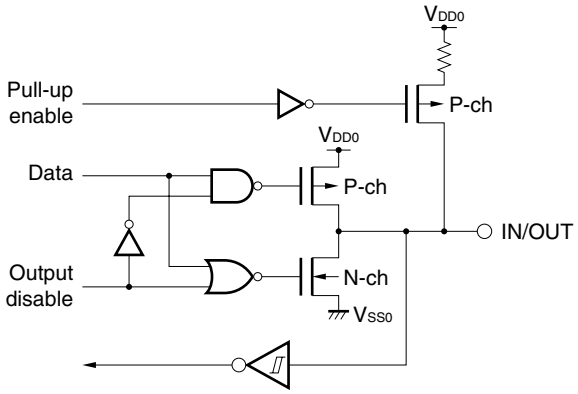
Figure 2-1. Pin I/O Circuit List (1/2)

Type 2

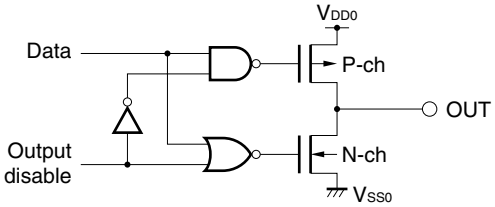


Schmitt-triggered input with hysteresis characteristics

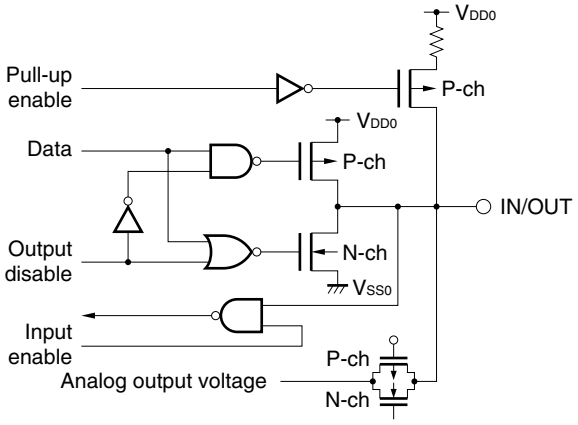
Type 8-C



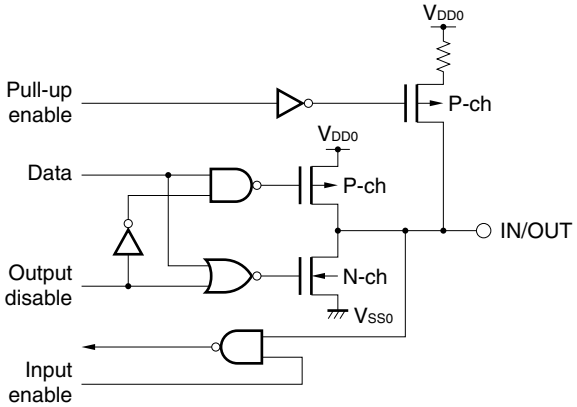
Type 4-B



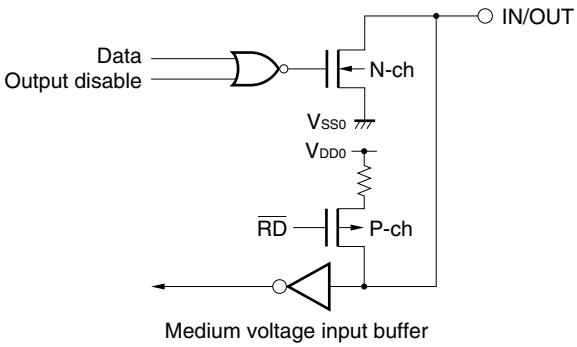
Type 12-C



Type 5-H

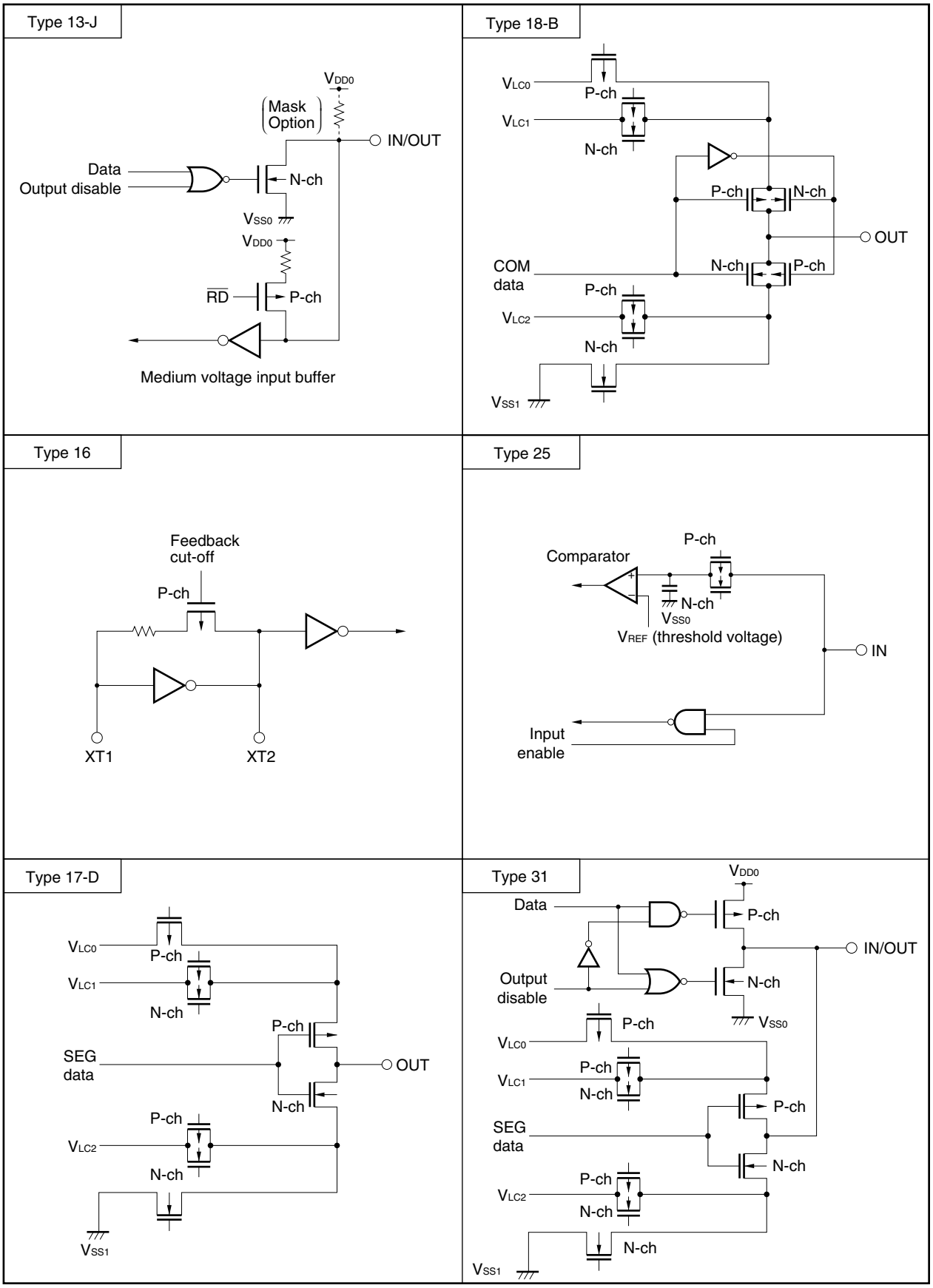


Type 13-K



Medium voltage input buffer

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



CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Spaces

The μPD780318, 780328, 780338 Subseries can each access a 64 KB memory space.
Figures 3-1 to 3-3 show the memory maps.

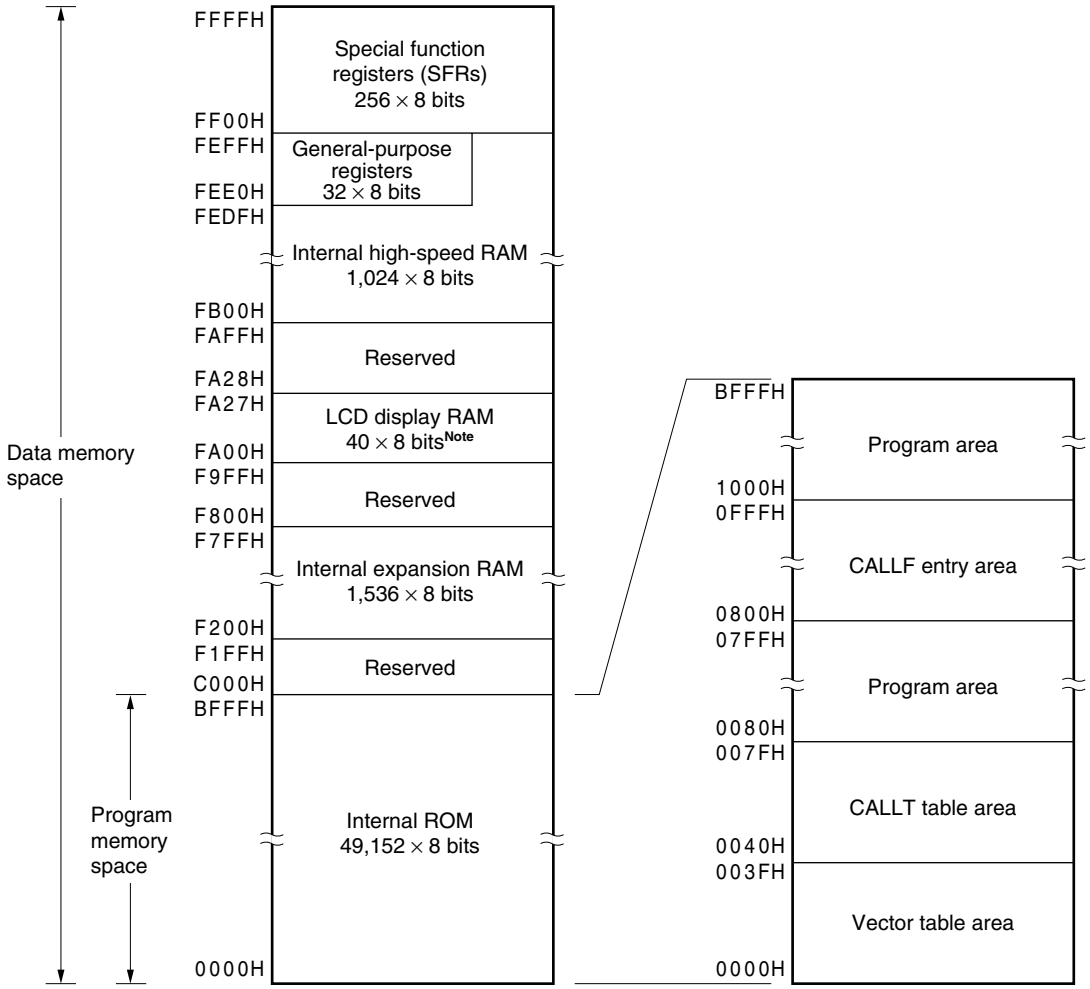
Caution In the case of the internal memory capacity, the initial values of the memory size switching register (IMS) and internal expansion RAM size switching register (IXS) of all products (μPD780318, 780328, and 780338 Subseries) are fixed (IMS = CFH, IXS = 0CH). Therefore, set the value corresponding to each product as indicated below.

	Set Value of IMS	Set Value of IXS
μPD780316, 780326, 780336	CCH	09H
μPD780318, 780328, 780338	CFH	
μPD78F0338	Value corresponding to mask ROM version	

(1) μPD780316, 780326, 780336

Set the value of the memory size switching register (IMS) to CCH, and the value of the internal expansion RAM size switching register (IXS) to 09H (default setting: IMS = CFH, IXS = 0CH).

Figure 3-1. Memory Map (μPD780316, 780326, 780336)



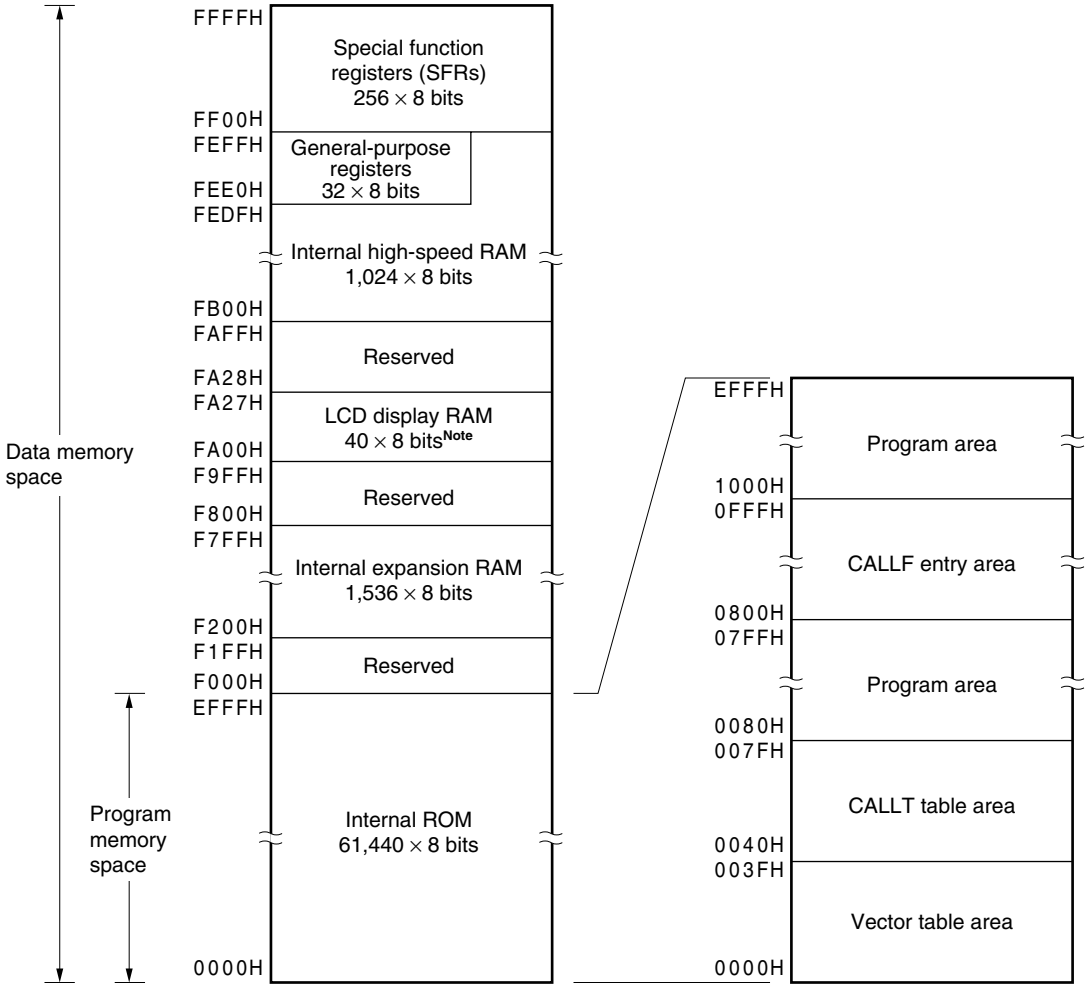
Note The area that can be used as LCD display data varies depending on the product. The area not used as LCD display data can be used as normal RAM.

- μPD780316: FA00H to FA17H (24 bytes)
- μPD780326: FA00H to FA1FH (32 bytes)
- μPD780336: FA00H to FA27H (40 bytes)

(2) μPD780318, 780328, 780338

Set the value of the memory size switching register (IMS) to CFH, and the value of the internal expansion RAM size switching register (IXS) to 09H (default setting: IMS = CFH, IXS = 0CH).

Figure 3-2. Memory Map (μPD780318, 780328, 780338)



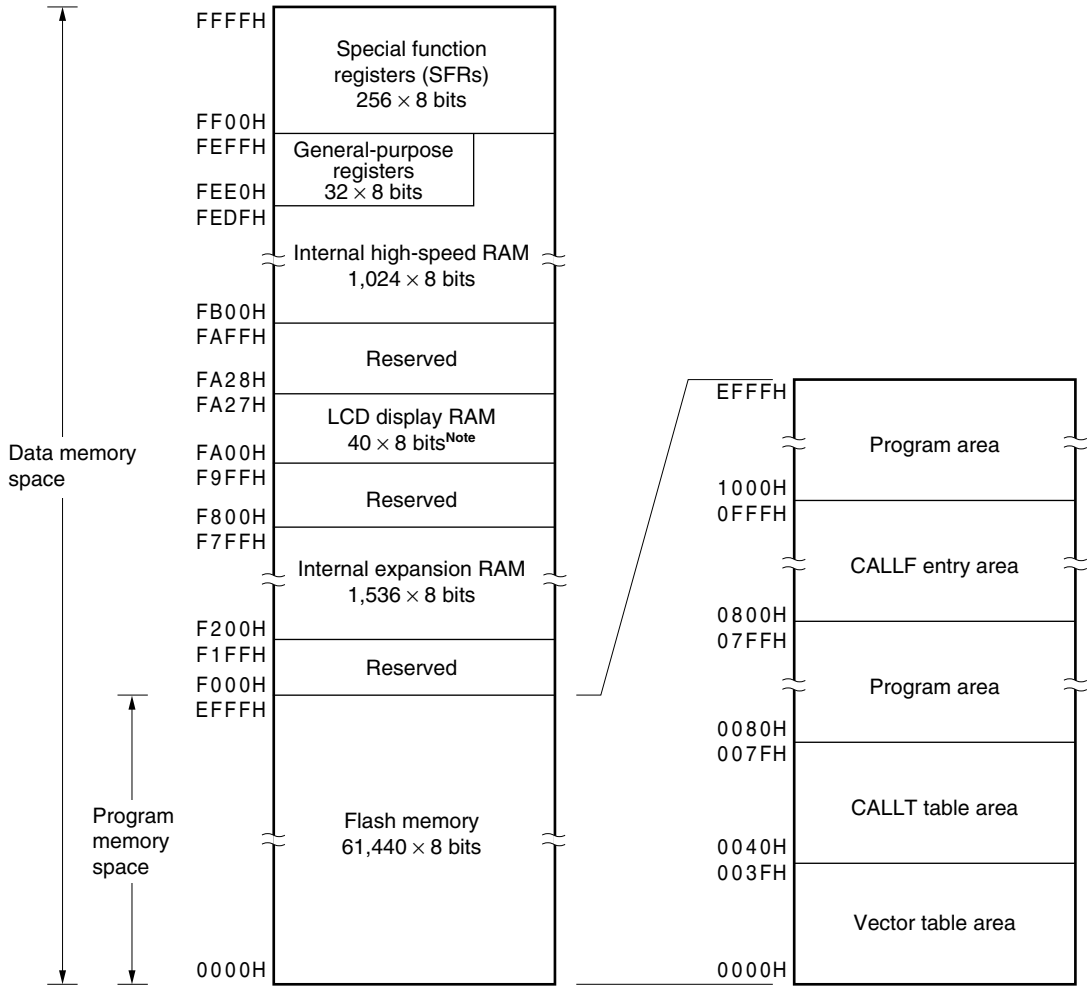
Note The area that can be used as LCD display data varies depending on the product. The area not used as LCD display data can be used as normal RAM.

- μPD780318: FA00H to FA17H (24 bytes)
- μPD780328: FA00H to FA1FH (32 bytes)
- μPD780338: FA00H to FA27H (40 bytes)

(3) μPD78F0338

Set the value of the memory size switching register (IMS) to value corresponding to mask ROM version, and the value of the internal expansion RAM size switching register (IXS) to 09H (default setting: IMS = CFH, IXS = 0CH).

Figure 3-3. Memory Map (μPD78F0338)



Note The area that can be used as LCD display data varies if P80/S32 to P87/S39 and P90/S24 to P97/S31 are used as port output or segment output. The area not used as LCD display data can be used as normal RAM.

- P80/S32 to P87/S39 and P90/S24 to P97/S31 are used as port output: FA00H to FA17H (24 bytes)
- P80/S32 to P87/S39 or P90/S24 to P97/S31 is used as port output: FA00H to FA1FH (32 bytes)
- P80/S32 to P87/S39 and P90/S24 to P97/S31 are used as segment output: FA00H to FA27H (40 bytes)

3.1.1 Internal program memory space

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

The μPD780318, 780328, and 780338 Subseries products incorporate an internal ROM (or flash memory), as listed below.

Table 3-1. Internal Memory Capacity

Part Number	Structure	Capacity
μPD780316, 780326, 780336	Mask ROM	49,152 × 8 bits (0000H to BFFFH)
μPD780318, 780328, 780338		61,440 × 8 bits (0000H to EFFFH)
μPD78F0338	Flash memory	

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

(1) Vector table area

The 64-byte area 0000H to 003FH is reserved as a vector table area. The $\overline{\text{RESET}}$ input and program start addresses for branch upon generation of each interrupt request are stored in the vector table area. Of the 16-bit address, lower 8 bits are stored at even addresses and higher 8 bits are stored at odd addresses.

Table 3-2. Vector Table

Vector Table Address	Interrupt Source	Vector Table Address	Interrupt Source
0000H	$\overline{\text{RESET}}$ input	001AH	INTCSI1
0004H	INTWDT	001CH	INTCSI3
0006H	INTP0	001EH	INTWTNI0
0008H	INTP1	0020H	INTTM00
000AH	INTP2	0022H	INTTM01
000CH	INTP3	0024H	INTTM4
000EH	INTP4	0026H	INTTM50
0010H	INTP5	0028H	INTTM51
0012H	INTKR	002AH	INTTM52
0014H	INTSER0	002CH	INTAD0
0016H	INTSR0	002EH	INTWTN0
0018H	INTST0	003EH	BRK

(2) CALLT instruction table area

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

(3) CALLF instruction entry area

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

3.1.2 Internal data memory space

The μPD780318, 780328, and 780338 Subseries products incorporate the following RAM.

(1) Internal high-speed RAM

This RAM is assigned to FB00H to FEFFH (1,024 bytes).

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

★ Instructions cannot be written and executed using this RAM as a program area.

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

(2) Internal expansion RAM

The area F200H to F7FFH (1,536 bytes) is assigned to the internal expansion RAM.

★ The internal expansion RAM can be used as a normal data area in the same way as the internal high-speed RAM. This RAM can also be used for writing and executing instructions as a program area.

(3) LCD display RAM

The area FA00H to FA27H (40 × 8 bits) is assigned to the LCD display RAM. Among this space, the area that can be used as LCD display data varies depending on the product, as described in Table 3-3.

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

Table 3-3. Area That Can Be Used as LCD Display Data

Part Number	Area That Can Be Used as LCD Display Data
μPD780316, 780318	FA00H to FA17H (24 bytes)
μPD780326, 780328	FA00H to FA1FH (32 bytes)
μPD780336, 780338	FA00H to FA27H (40 bytes)
μPD78F0338	• P80/S32 to P87/S39 and P90/S24 to P97/S31 are used as port output: FA00H to FA17H (24 bytes) • P80/S32 to P87/S39 or P90/S24 to P97/S31 is used as port output: FA00H to FA1FH (32 bytes) • P80/S32 to P87/S39 and P90/S24 to P97/S31 are used as segment output: FA00H to FA27H (40 bytes)

3.1.3 Special function register (SFR) area

On-chip peripheral hardware special function registers (SFRs) are allocated in the area FF00H to FFFFH (refer to **Table 3-4 Special Function Register List** in **3.2.3 Special function register (SFR)**).

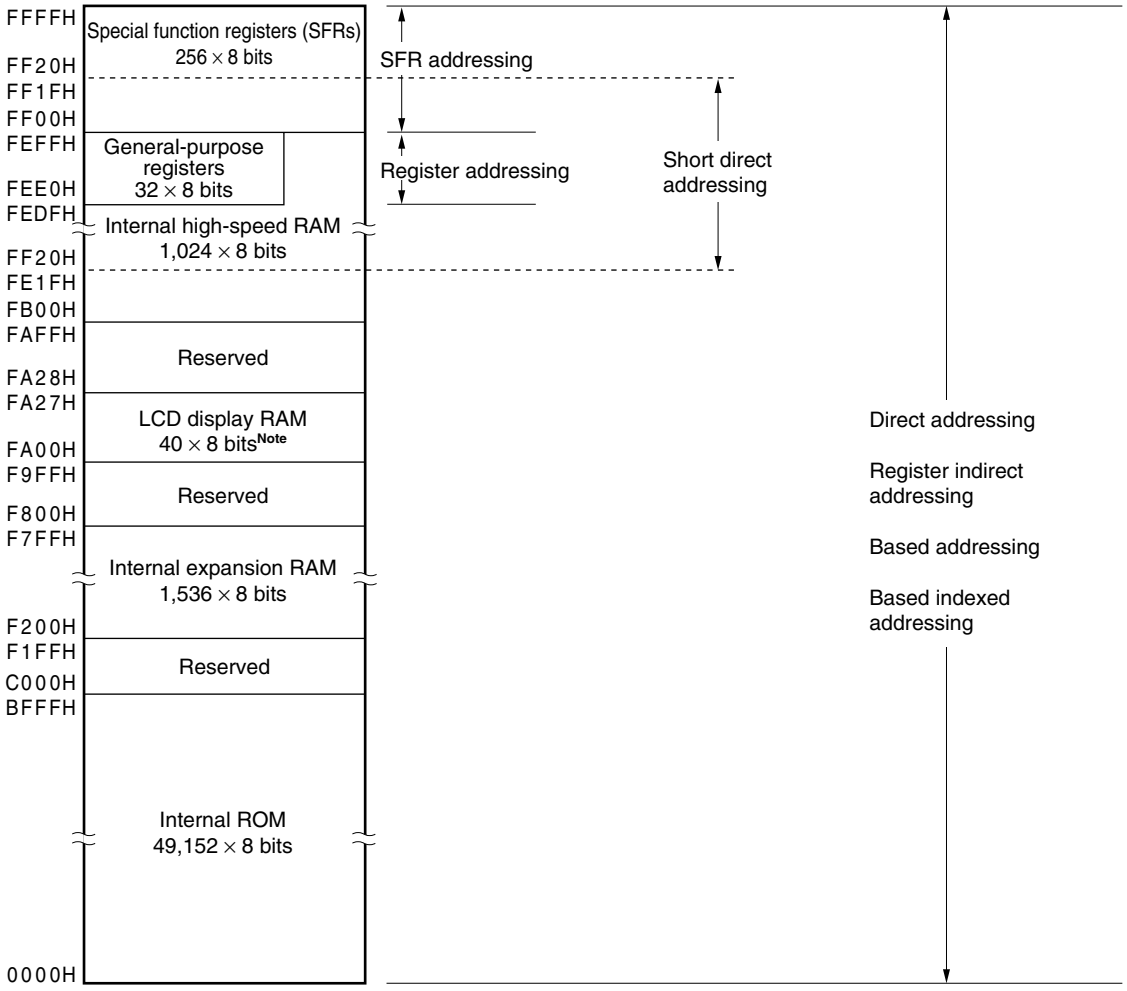
Caution Do not access addresses where an SFR is not assigned.

3.1.4 Data memory addressing

Addressing refers to the method of specifying the address of the instruction to be executed next or the address of the register or memory relevant to the execution of instructions.

Several addressing modes are provided for addressing the memory relevant to the execution of instructions for the μPD780318, 780328, and 780338 Subseries, based on operability and other considerations. For areas containing data memory in particular, special addressing methods designed for the functions of special function registers (SFR) and general-purpose registers are available for use. Data memory and its corresponding addressing are illustrated in Figures 3-4 to 3-6. For the details of each addressing mode, see **3.4 Operand Address Addressing**.

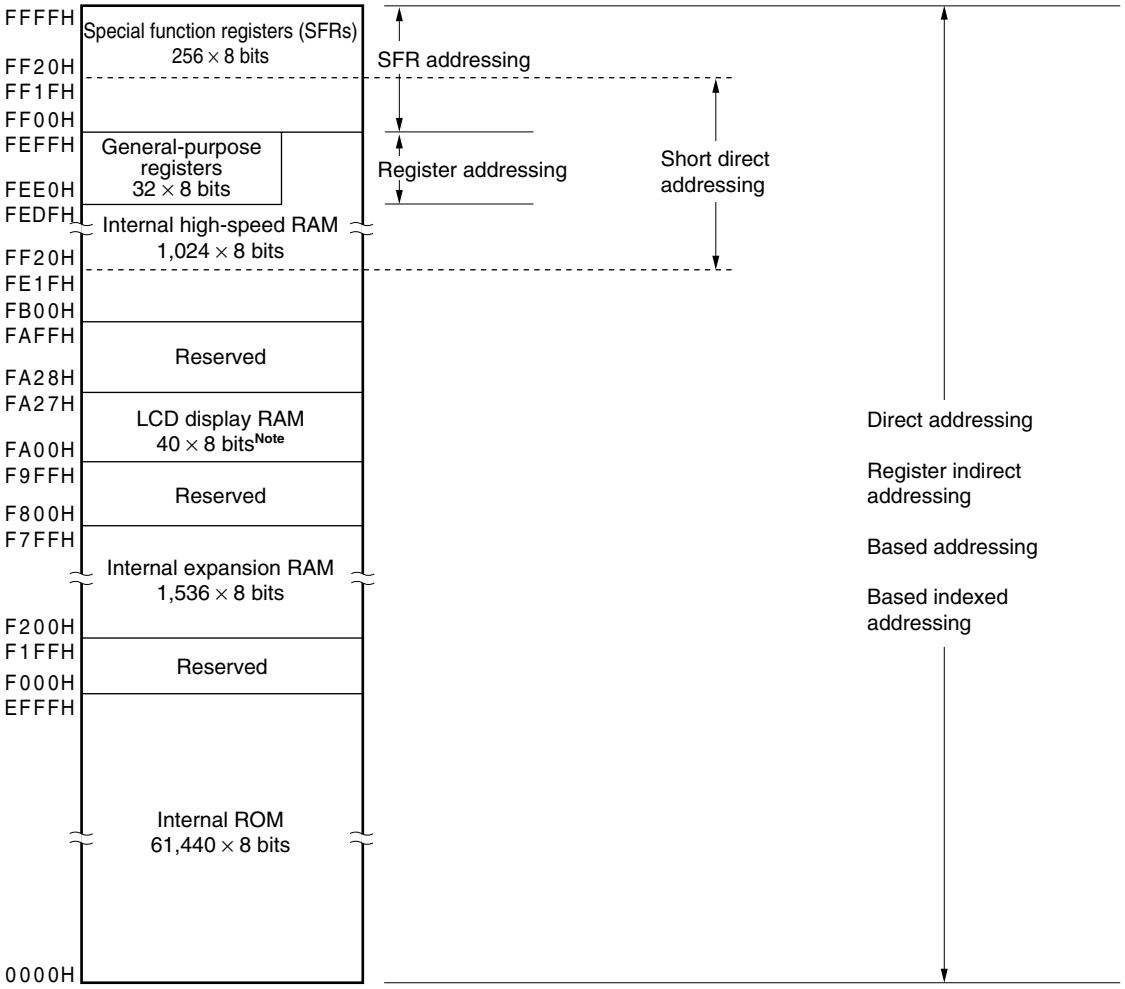
Figure 3-4. Data Memory Addressing (μPD780316, 780326, 780336)



Note The area that can be used as LCD display data varies depending on the product. The area not used as LCD display data can be used as normal RAM.

- μPD780316: FA00H to FA17H (24 bytes)
- μPD780326: FA00H to FA1FH (32 bytes)
- μPD780336: FA00H to FA27H (40 bytes)

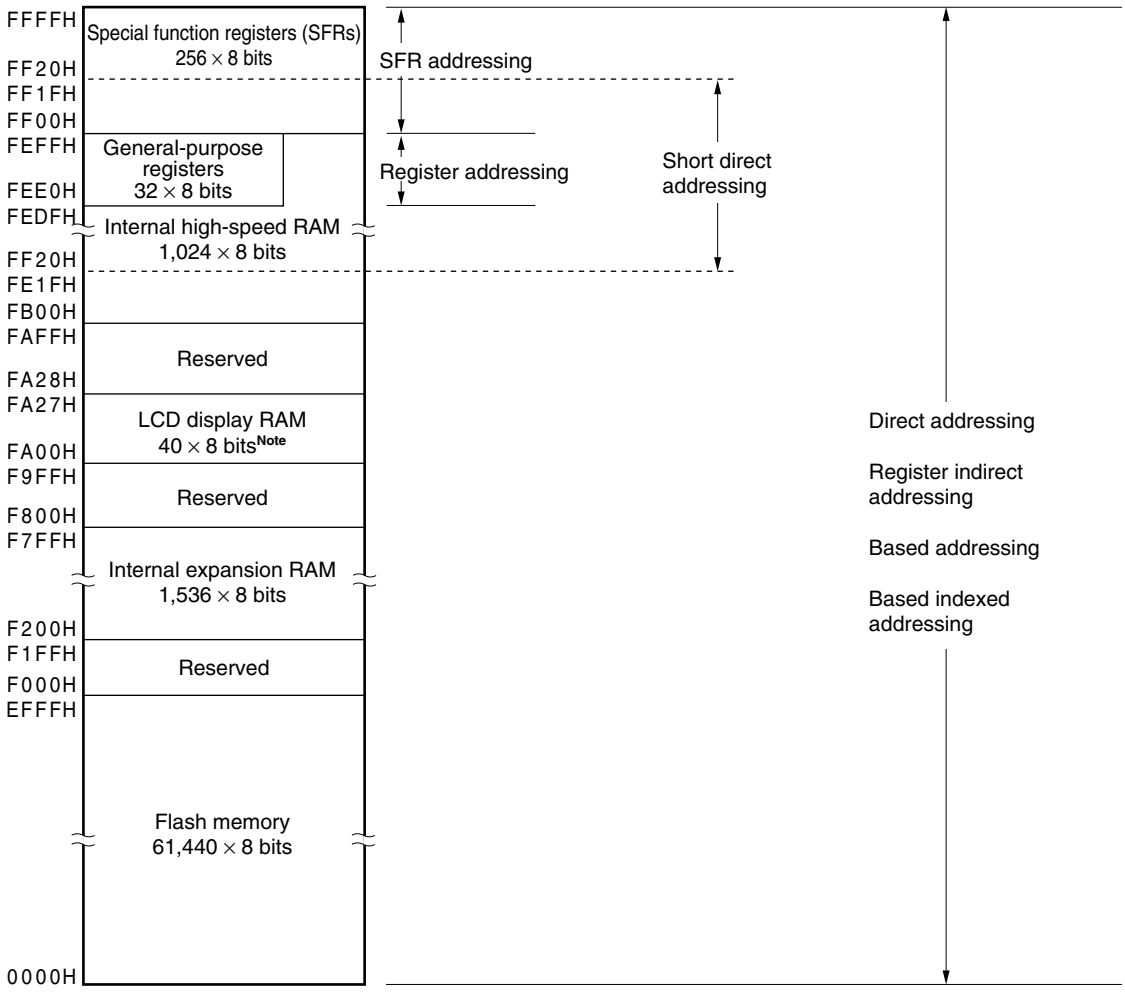
Figure 3-5. Data Memory Addressing (μPD780318, 780328, 780338)



Note The area that can be used as LCD display data varies depending on the product. The area not used as LCD display data can be used as normal RAM.

- μPD780318: FA00H to FA17H (24 bytes)
- μPD780328: FA00H to FA1FH (32 bytes)
- μPD780338: FA00H to FA27H (40 bytes)

Figure 3-6. Data Memory Addressing (μPD78F0338)



Note The area that can be used as LCD display data varies if P80/S32 to P87/S39 and P90/S24 to P97/S31 are used as port output or segment output. The area not used as LCD display data can be used as normal RAM.

- P80/S32 to P87/S39 and P90/S24 to P97/S31 are used as port output: FA00H to FA17H (24 bytes)
- P80/S32 to P87/S39 or P90/S24 to P97/S31 is used as port output: FA00H to FA1FH (32 bytes)
- P80/S32 to P87/S39 and P90/S24 to P97/S31 are used as segment output: FA00H to FA27H (40 bytes)

3.2 Processor Registers

The μPD780318, 780328, and 780338 Subseries products incorporate the following processor registers.

3.2.1 Control registers

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

(1) Program counter (PC)

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

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

Figure 3-7. Program Counter Format

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

(2) Program status word (PSW)

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

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

Figure 3-8. Program Status Word Format

	7							0
PSW	IE	Z	RBS1	AC	RBS0	0	ISP	CY

(a) Interrupt enable flag (IE)

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

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

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

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

(b) Zero flag (Z)

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

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

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

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

(d) Auxiliary carry flag (AC)

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

(e) In-service priority flag (ISP)

This flag manages the priority of acknowledgeable maskable vectored interrupts. When this flag is 0, low-level vectored interrupt requests specified with a priority specification flag register (PR0L, PR0H, PR1L) (refer to **18.3 (3) Priority specification flag registers (PR0L, PR0H, PR1L)**) are disabled for acknowledgement. Actual request acknowledgement is controlled with the interrupt enable flag (IE).

(f) Carry flag (CY)

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

(3) Stack pointer (SP)

This is a 16-bit register to hold the start address of the memory stack area. Only the internal high-speed RAM area (FB00H to FEFFH) can be set as the stack area.

Figure 3-9. Stack Pointer Format

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

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

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

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

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

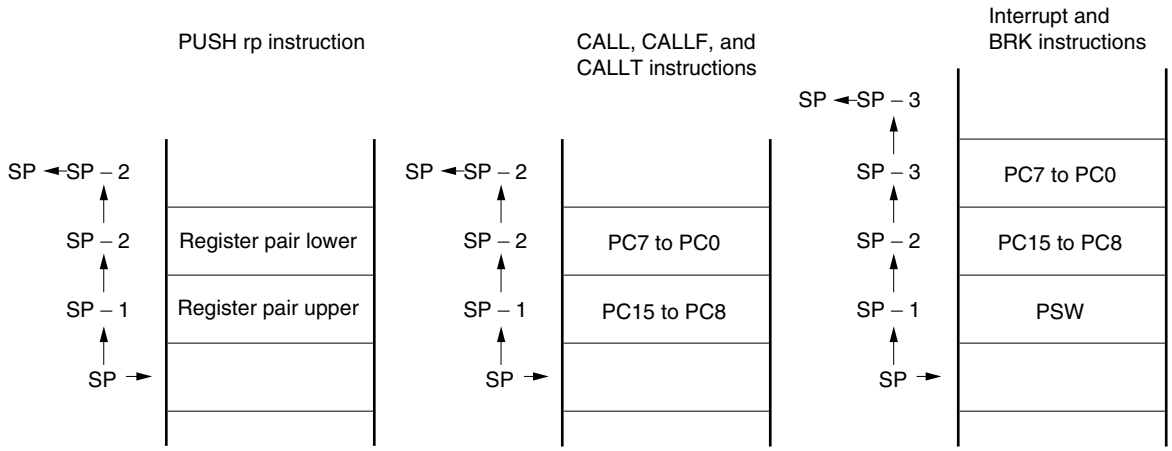
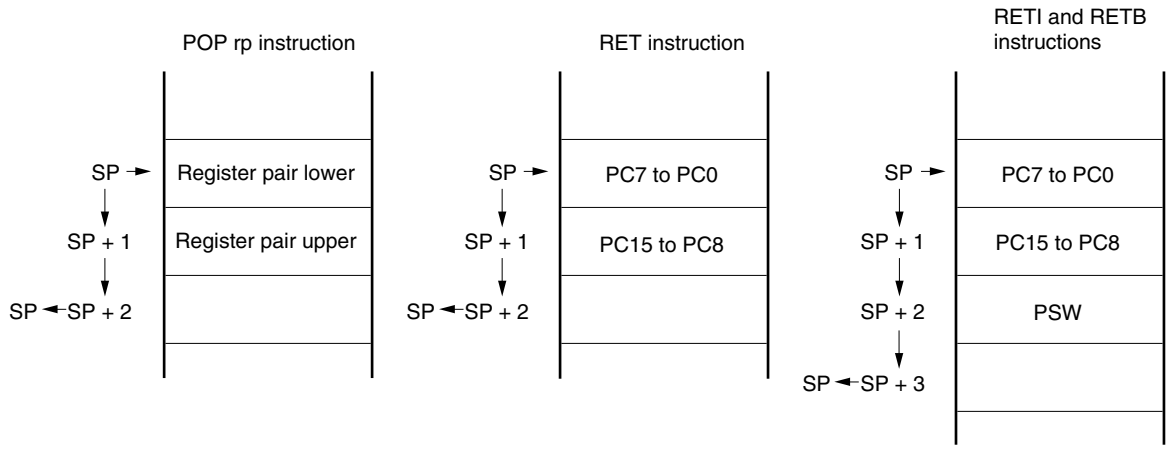


Figure 3-11. Data to Be Restored from Stack Memory



3.2.2 General-purpose registers

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

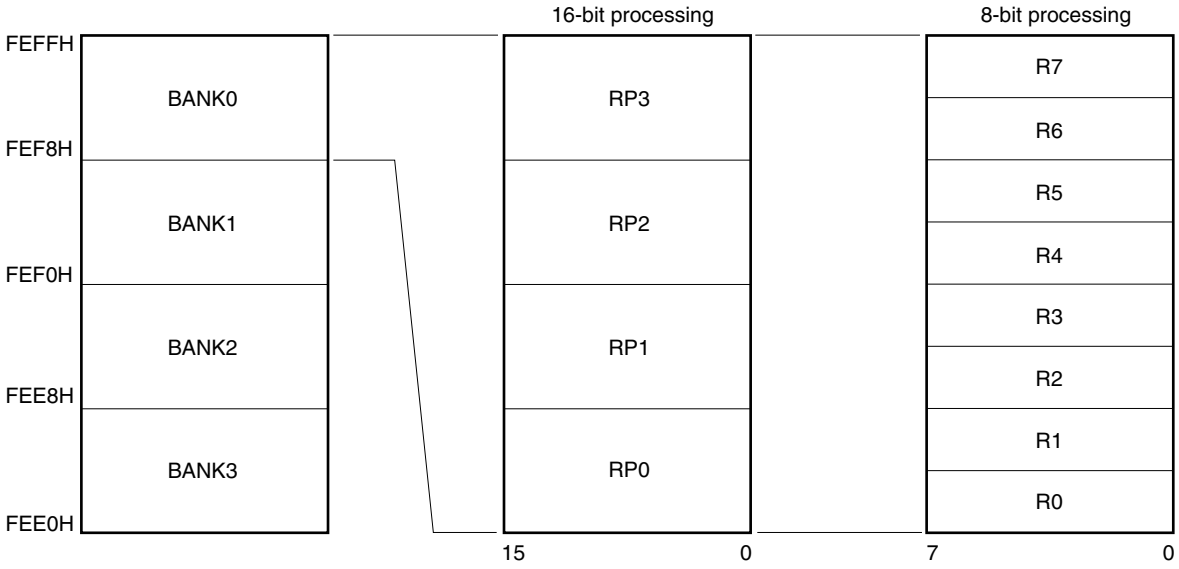
Each register can be used as an 8-bit register, and two 8-bit registers can be used in pairs as a 16-bit register (AX, BC, DE, and HL).

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

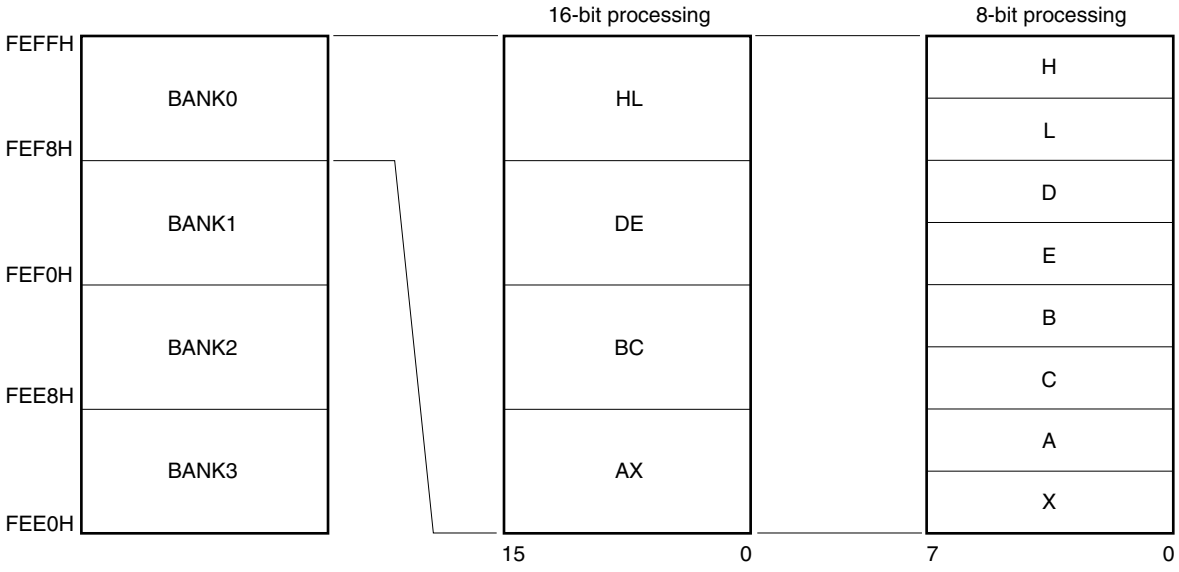
Register banks to be used for instruction execution are set with the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interrupts for each bank.

Figure 3-12. General-Purpose Register Configuration

(a) Absolute name



(b) Function name



3.2.3 Special function register (SFR)

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

The special function registers are allocated in the FF00H to FFFFH area.

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

Each manipulation bit unit can be specified as follows.

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

Table 3-4 gives a list of special function registers. The meaning of items in the table is as follows.

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

Table 3-4. Special Function Register List (1/4)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset
				1 Bit	8 Bits	16 Bits	
FF00H	Port 0	P0	R/W	√	√	—	00H
FF01H	Port 1	P1	R	√	√	—	00H
FF02H	Port 2	P2	R/W	√	√	—	00H
FF03H	Port 3	P3	R/W	√	√	—	00H
FF04H	Port 4	P4	R/W	√	√	—	00H
FF05H	Port 5	P5	R/W	√	√	—	00H
FF06H	Port 6	P6	R/W	√	√	—	00H
FF07H	Port 7	P7	R/W	√	√	—	00H
★ FF08H	Port 8 Note 1	P8	R/W	√	√	—	00H
★ FF09H	Port 9 Note 2	P9	R/W	√	√	—	00H
FF0CH	Port 12	P12	R/W	√	√	—	00H
FF0EH	A/D conversion result register 0	ADCR0	R	—	—	√	0000H
FF0FH							
FF10H	16-bit timer capture/compare register 00	CR00	R/W	—	—	√	Undefined
FF11H							
FF12H	16-bit timer capture/compare register 01	CR01	R/W	—	—	√	Undefined
FF13H							
FF14H	16-bit timer counter 0	TM0	R	—	—	√	0000H
FF15H							
FF16H	8-bit timer compare register 50	CR50	R/W	—	√	—	Undefined
FF17H	8-bit timer compare register 51	CR51	R/W	—	√	—	Undefined
FF18H	8-bit timer counter 50	TM50	R	—	√	—	00H
FF19H	8-bit timer counter 51	TM51	R	—	√	—	00H
FF1AH	Serial I/O shift register 3	SIO3	R/W	—	√	—	Undefined
FF1BH	Transmit shift register 0	TXS0	W	—	√	—	FFH
	Receive buffer register 0	RXB0	R	—	√	—	FFH

Notes 1. μPD780316, 780318, 780326, 780328, 78F0338 only

2. μPD780316, 780318, 78F0338 only

Table 3-4. Special Function Register List (2/4)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset
				1 Bit	8 Bits	16 Bits	
FF20H	Port mode register 0	PM0	R/W	√	√	—	FFH
FF22H	Port mode register 2	PM2	R/W	√	√	—	FFH
FF23H	Port mode register 3	PM3	R/W	√	√	—	FFH
FF24H	Port mode register 4	PM4	R/W	√	√	—	FFH
FF25H	Port mode register 5	PM5	R/W	√	√	—	FFH
FF26H	Port mode register 6	PM6	R/W	√	√	—	FFH
FF27H	Port mode register 7	PM7	R/W	√	√	—	FFH
FF28H	Port mode register 8**Note 1**	PM8	W	—	√	—	FFH
FF29H	Port mode register 9**Note 1**	PM9	W	—	√	—	FFH
FF2CH	Port mode register 12	PM12	R/W	√	√	—	FFH
FF30H	Pull-up resistor option register 0	PU0	R/W	√	√	—	00H
FF32H	Pull-up resistor option register 2	PU2	R/W	√	√	—	00H
FF33H	Pull-up resistor option register 3	PU3	R/W	√	√	—	00H
FF34H	Pull-up resistor option register 4	PU4	R/W	√	√	—	00H
FF35H	Pull-up resistor option register 5	PU5	R/W	√	√	—	00H
FF36H	Pull-up resistor option register 6	PU6	R/W	√	√	—	00H
FF37H	Pull-up resistor option register 7	PU7	R/W	√	√	—	00H
FF38H	Correction address register 0	CORAD0	R/W	—	—	√	0000H
FF39H							
FF3AH	Correction address register 1	CORAD1	R/W	—	—	√	0000H
FF3BH							
FF3CH	Pull-up resistor option register 12	PU12	R/W	√	√	—	00H
FF40H	Clock output select register	CKS	R/W	√	√	—	00H
FF41H	Watch timer operation mode register 0	WTNM0	R/W	√	√	—	00H
FF42H	Watchdog timer clock select register	WDCS	R/W	—	√	—	00H
FF47H	Memory expansion mode register	MEM	R/W	√	√	—	00H
FF48H	External interrupt rising edge enable register	EGP	R/W	√	√	—	00H
FF49H	External interrupt falling edge enable register	EGN	R/W	√	√	—	00H
FF58H	Pin function switching register 8**Note 2**	PF8	W	—	√	—	00H
FF59H	Pin function switching register 9**Note 2**	PF9	W	—	√	—	00H

Notes 1. μPD78F0338 only.

2. μPD78F0338 only. PF8 and PF9 can only be set once after reset. To change the value, reset the register.

Table 3-4. Special Function Register List (3/4)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset
				1 Bit	8 Bits	16 Bits	
FF60H	16-bit timer mode control register 0	TMC0	R/W	√	√	—	00H
FF61H	Prescaler mode register 0	PRM0	R/W	—	√	—	00H
FF62H	Capture/compare control register 0	CRC0	R/W	√	√	—	00H
FF63H	16-bit timer output control register 0	TOC0	R/W	√	√	—	00H
FF64H	16-bit timer compare register 4	CR4	R/W	—	—	√	Undefined
FF65H							
FF66H	16-bit timer counter 4	TM4	—	—	—	—	Undefined
FF67H							
FF68H	16-bit timer mode control register 4	TMC4	R/W	√	√	—	00H
FF70H	8-bit timer mode control register 50	TMC50	R/W	√	√	—	00H
FF71H	Timer clock select register 50	TCL50	R/W	—	√	—	00H
FF73H	8-bit timer mode control register 51	TMC51	R/W	√	√	—	00H
FF74H	Timer clock select register 51	TCL51	R/W	—	√	—	00H
FF76H	8-bit timer mode control register 52	TMC52	R/W	√	√	—	00H
FF77H	Timer clock select register 52	TCL52	R/W	—	√	—	00H
FF79H	8-bit timer compare register 52	CR52	R/W	—	√	—	Undefined
FF7AH	8-bit timer counter 52	TM52	R	—	√	—	00H
FF80H	A/D converter mode register 0	ADM0	R/W	√	√	—	00H
FF81H	Analog input channel specification register 0	ADS0	R/W	—	√	—	00H
FF82H	D/A converter mode register 0	DAM0	R/W	√	√	—	00H
FF83H	D/A conversion value setting register 0	DA0	R/W	—	√	—	00H
FF8AH	Correction control register	CORCN	R/W	√	√	—	00H
FF8FH	Key return switching register	KRSEL	R/W[Note]	√	√	—	00H
FF90H	LCD display mode register 3	LCDM3	R/W	√	√	—	00H
FF91H	LCD clock control register 3	LCDC3	R/W	—	√	—	00H
FF92H	Static/dynamic display switching register 3	SDSEL3	R/W	—	√	—	00H
FFA0H	Asynchronous serial interface mode register 0	ASIM0	R/W	√	√	—	00H
FFA1H	Asynchronous serial interface status register 0	ASIS0	R	—	√	—	00H
FFA2H	Baud rate generator control register 0	BRGC0	R/W	—	√	—	00H
FFAFH	Serial operation mode register 3	CSIM3	R/W	√	√	—	00H
FFB0H	Serial operation mode register 1	CSIM1	R/W	√	√	—	00H
FFB1H	Serial clock select register 1	CSIC1	R/W	√	√	—	10H
FFB2H	Serial I/O shift register 1	SIO1	R	—	√	—	Undefined
FFB4H	Transmit buffer register 1	SOTB1	R/W	—	√	—	Undefined

Note KRSEL can be accessed but its read value is not guaranteed.

Table 3-4. Special Function Register List (4/4)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset
					1 Bit	8 Bits	16 Bits	
FFE0H	Interrupt request flag register 0L	IF0	IF0L	R/W	√	√	√	00H
FFE1H	Interrupt request flag register 0H		IF0H	R/W	√	√		00H
FFE2H	Interrupt request flag register 1L	IF1L		R/W	√	√	—	00H
FFE4H	Interrupt mask flag register 0L	MK0	MK0L	R/W	√	√	√	FFH
FFE5H	Interrupt mask flag register 0H		MK0H	R/W	√	√		FFH
FFE6H	Interrupt mask flag register 1L	MK1L		R/W	√	√	—	FFH
FFE8H	Priority specification flag register 0L	PR0	PR0L	R/W	√	√	√	FFH
FFE9H	Priority specification flag register 0H		PR0H	R/W	√	√		FFH
FFEAH	Priority specification flag register 1L	PR1L		R/W	√	√	—	FFH
FFF0H	Memory size switching register[Note 1]	IMS		R/W	—	√	—	CFH
FFF4H	Internal expansion RAM size switching register[Note 2]	IXS		R/W	√	√	—	0CH
FFF9H	Watchdog timer mode register	WDTM		R/W	√	√	—	00H
FFFAH	Oscillation stabilization time select register	OSTS		R/W	—	√	—	04H
FFFBH	Processor clock control register	PCC		R/W	√	√	—	04H

Notes 1. Although the default value of this register is CFH, set the value corresponding to each product as indicated below.

μPD780316, 780326, 780336: CCH

μPD780318, 780328, 780338: CFH

μPD78F0338: Value for mask ROM version

2. Although the default value of this register is 0CH, use this register with a setting of 09H.

3.3 Instruction Address Addressing

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

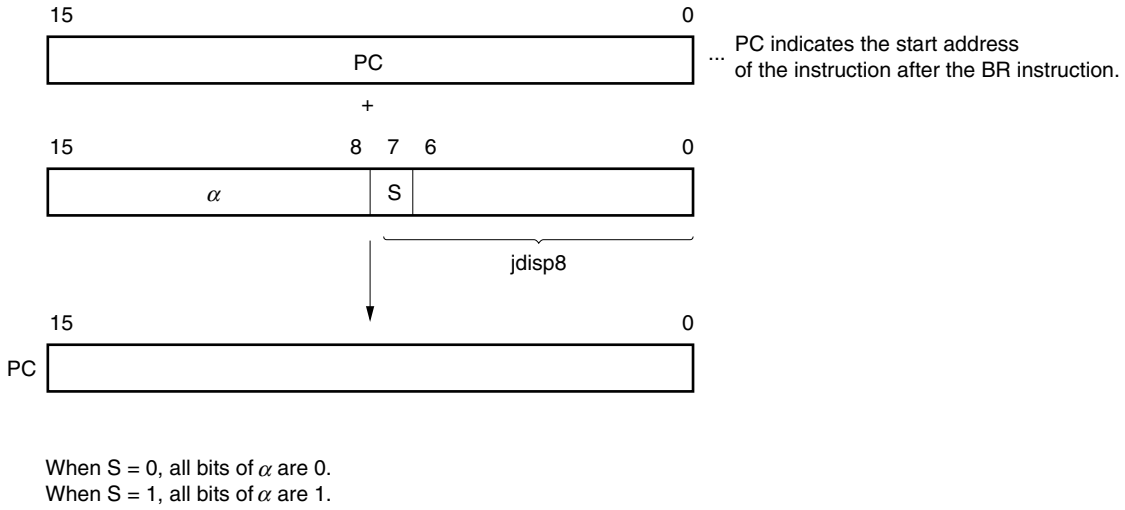
3.3.1 Relative addressing

[Function]

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

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

[Illustration]



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

3.3.2 Immediate addressing

[Function]

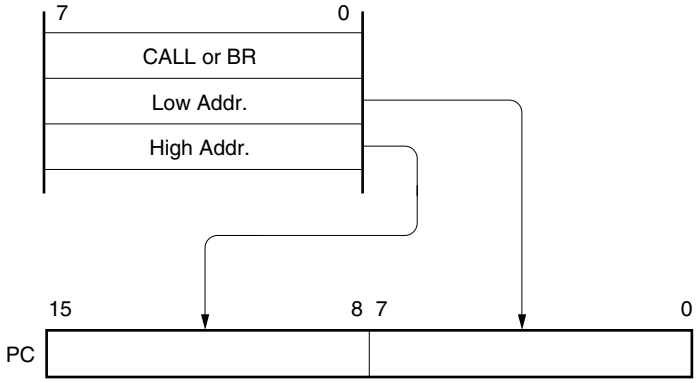
Immediate data in the instruction word is transferred to the program counter (PC) and branched.

This function is carried out when the CALL !addr16 or BR !addr16 or CALLF !addr11 instruction is executed.

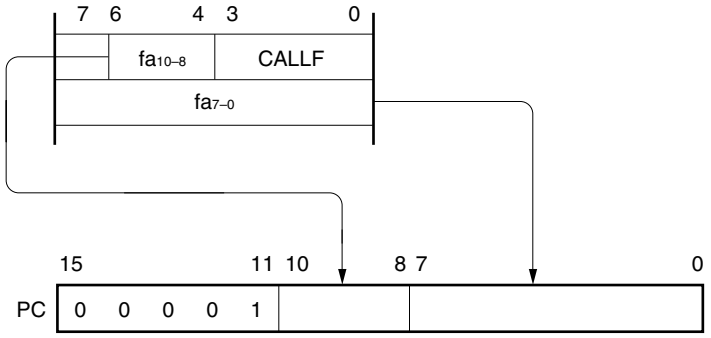
CALL !addr16 and BR !addr16 instructions can be branched to the entire memory space. The CALLF !addr11 instruction is branched to the 0800H to 0FFFH area.

[Illustration]

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



In the case of CALLF !addr11 instruction



3.3.3 Table indirect addressing

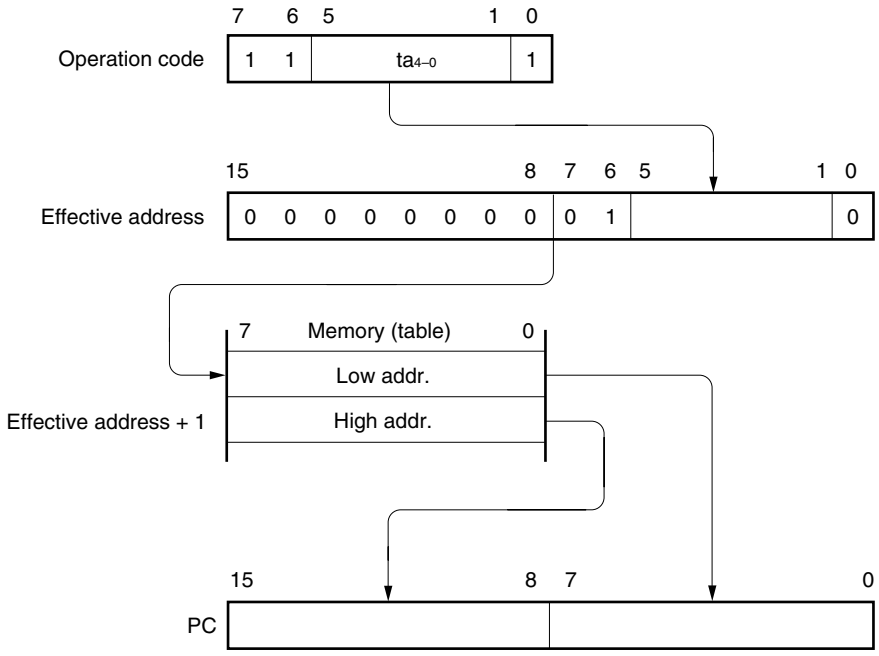
[Function]

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

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

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

[Illustration]



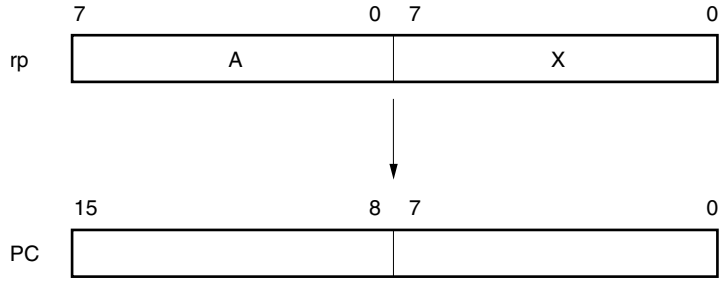
3.3.4 Register addressing

[Function]

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

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

[Illustration]



3.4 Operand Address Addressing

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

3.4.1 Implied addressing

[Function]

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

Of the μPD780318, 780328, and 780338 Subseries instruction words, the following instructions employ implied addressing.

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

[Operand format]

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

[Description example]

In the case of MULU X

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

3.4.2 Register addressing

[Function]

The general-purpose register to be specified is accessed as an operand with the register specify code (Rn and RPn) of an instruction word in the registered bank specified with the register bank select flag (RBS0 and RBS1). Register addressing is carried out when an instruction with the following operand format is executed. When an 8-bit register is specified, one of the eight registers is specified with 3 bits in the operation code.

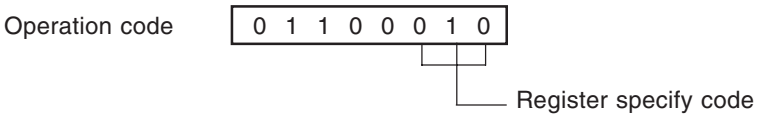
[Operand format]

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

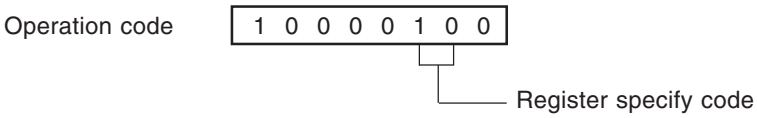
'r' and 'rp' can be described with absolute names (R0 to R7 and RP0 to RP3) as well as function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL).

[Description example]

MOV A, C; when selecting C register as r



INCW DE; when selecting DE register pair as rp



3.4.3 Direct addressing

[Function]

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

[Operand format]

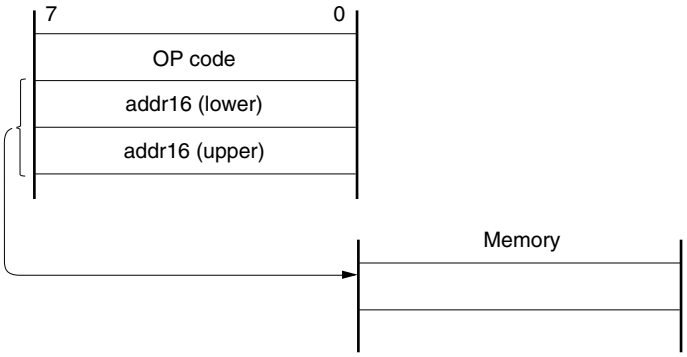
Identifier	Description
addr16	Label or 16-bit immediate data

[Description example]

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

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

[Illustration]



3.4.4 Short direct addressing

[Function]

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

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

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

[Operand format]

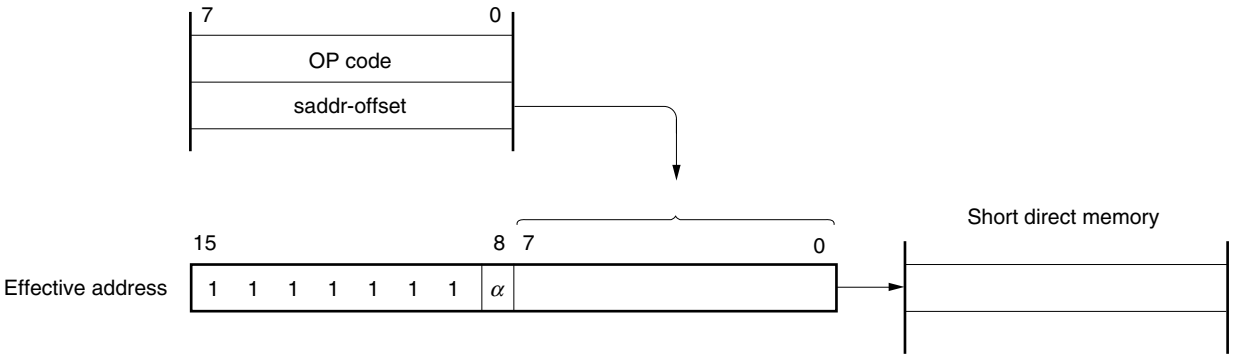
Identifier	Description
saddr	Label or FE20H to FF1FH immediate data
saddrp	Label or FE20H to FF1FH immediate data (even address only)

[Description example]

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

Operation code	0 0 0 1 0 0 0 1	OP code
	0 0 1 1 0 0 0 0	30H (saddr-offset)
	0 1 0 1 0 0 0 0	50H (immediate data)

[Illustration]



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

3.4.5 Special function register (SFR) addressing

[Function]

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

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

[Operand format]

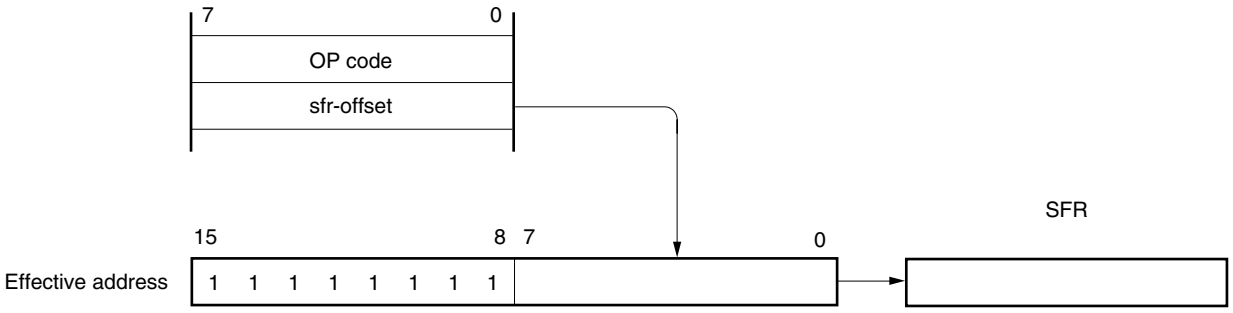
Identifier	Description
sfr	Special function register name
sfrp	16-bit manipulatable special function register name (even address only)

[Description example]

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

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

[Illustration]



3.4.6 Register indirect addressing

[Function]

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

[Operand format]

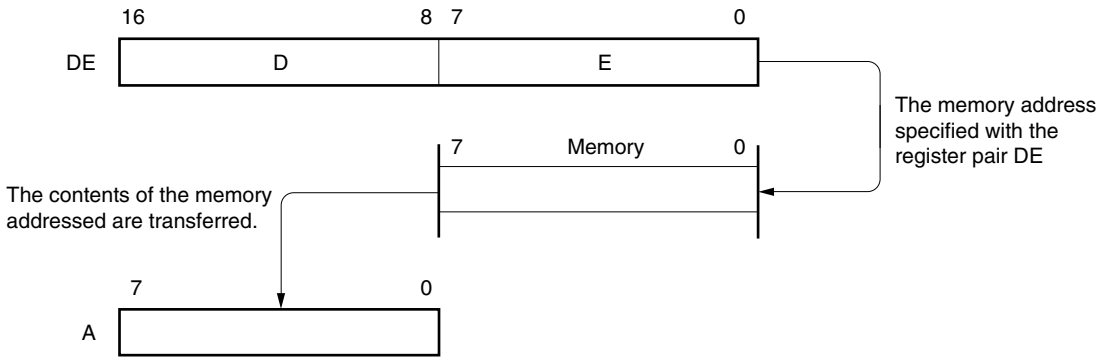
Identifier	Description
—	[DE], [HL]

[Description example]

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

Operation code	1	0	0	0	0	1	0	1

[Illustration]



3.4.7 Based addressing

[Function]

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

[Operand format]

Identifier	Description
—	[HL + byte]

[Description example]

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

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

3.4.8 Based indexed addressing

[Function]

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

[Operand format]

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

[Description example]

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

Operation code	1	0	1	0	1	0	1	1

3.4.9 Stack addressing

[Function]

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

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

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

[Description example]

In the case of PUSH DE

Operation code	1	0	1	1	0	1	0	1

CHAPTER 4 PORT FUNCTIONS

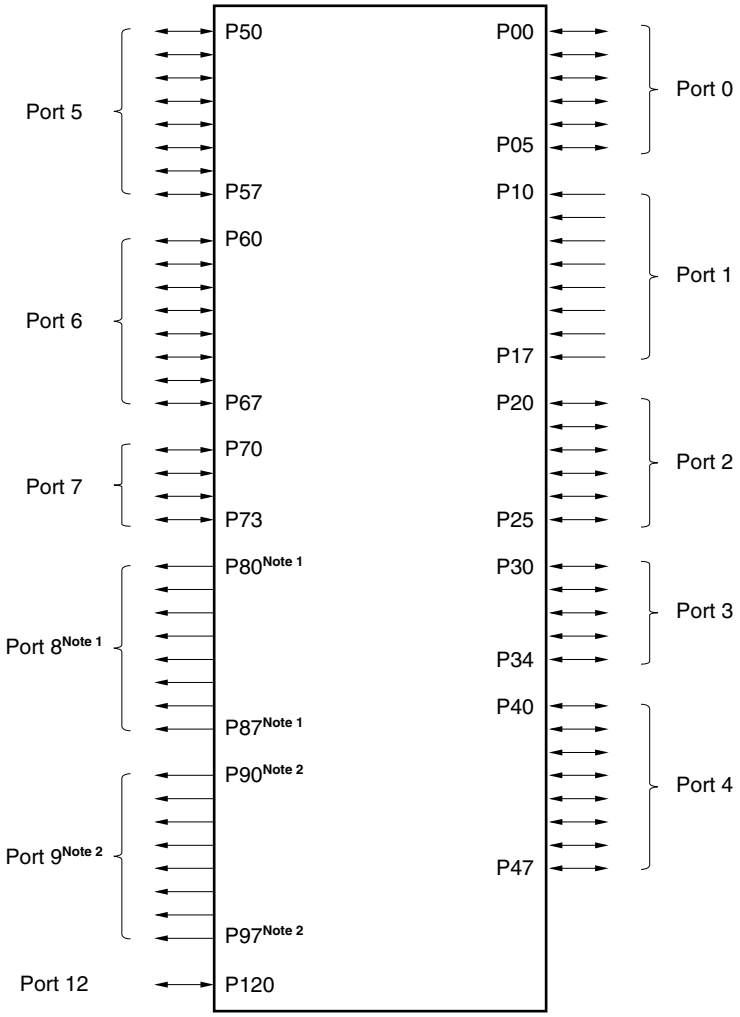
4.1 Port Functions

The μPD780318, 780328, and 780338 Subseries products incorporate input port, output port, and I/O port as listed in Table 4-1. Figure 4-1 shows the port configuration. Every port is capable of 1-bit and 8-bit manipulations and can carry out considerably varied control operations. Besides port functions, the ports can also serve as on-chip hardware I/O pins.

Table 4-1. Port Types

	Input Pin	Output Pin	I/O Pin
μPD780316, 780318, 78F0338	8	16	46
μPD780326, 780328		8	
μPD780336, 780338		None	

Figure 4-1. Port Types



Notes 1. The μPD780336 and 780338 do not incorporate port 8.

2. The μPD780326, 780328, 780336, and 780338 do not incorporate port 9.

Table 4-2. Port Functions (1/2)

Pin Name	I/O	Function		After Reset	Alternate Function
P00	I/O	Port 0 6-bit I/O port Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be used by setting software.		Input	INTP0
P01					INTP1
P02					INTP2
P03					INTP3/ADTRG
P04					INTP4
P05					INTP5/BUZ/PCL
P10 to P17	Input	Port 1 8-bit input only port.		Input	ANI0 to ANI7
P20	I/O	Port 2 6-bit I/O port Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be used by setting software.		Input	RxD0/SI3
P21					TxD0/SO3
P22					$\overline{\text{SCK3}}$
P23					SI1
P24					SO1
P25					SCK1
P30	I/O	Port 3 5-bit I/O port Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be used by setting software.		Input	TO0
P31					TI00
P32					TI01
P33					TO50/TI50
P34					TO51/TI51
P40 to P47	I/O	Port 4 8-bit I/O port Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be used by setting software. Interrupt request flag (KRIF) is set to 1 by falling edge detection.		Input	—
P50 to P57	I/O	Port 5 8-bit I/O port Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be used by setting software.		Input	—
P60 to P63	I/O	Port 6 8-bit I/O port Input/output mode can be specified in 1-bit units. LEDs can be driven directly.	Medium-voltage N-ch open-drain I/O port On-chip pull-up resistor can be specified by mask option.	Input	—
P64 to P67			An on-chip pull-up resistor can be used by setting software.		

Table 4-2. Port Functions (2/2)

Pin Name	I/O	Function	After Reset	Alternate Function
P70	I/O	Port 7 4-bit I/O port Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be used by setting software.	Input	TO4
P71				TI4
P72				TO52
P73				TI52
P80 to P87[Note]	Output	Port 8 8-bit output only port	Output	S32 to S39[Note]
P90 to P97[Note]	Output	Port 9 8-bit output only port	Output	S24 to S31[Note]
P120	I/O	Port 12 1-bit I/O port Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be used by setting software.	Input	AO0

Note Ports 8 and 9 vary depending on the product.

	Port 8	Port 9
μPD780316, 780318	P80 to P87 (without alternate pin)	P90 to P97 (without alternate pin)
μPD780326, 780328		None
μPD780336, 780338	None	
μPD78F0338	P80/S32 to P87/S39	P90/S24 to P97/S31

4.2 Port Configuration

A port consists of the following hardware.

Table 4-3. Port Configuration

Item	Configuration
Control registers	Port mode register (PMm: m = 0, 2 to 7, 8[Note], 9[Note], 12) Pull-up resistor option register (PUm: m = 0, 2 to 7, 12) Memory expansion register (MEM) Key return switching register (KRSEL) Pin function switching registers 8 and 9 (PF8 and PF9)[Note]
Ports	• μPD780316, 780318, 78F0338 Total: 70 (input: 8, output: 16, I/O: 46) • μPD780326, 780328 Total: 62 (input: 8, output: 8, I/O: 46) • μPD780336, 780338 Total: 54 (input: 8, I/O: 46)
Pull-up resistor	• Mask ROM version Total: 46 (software control: 42, mask option: 4) • Flash memory version Total: 42 (software control: 42)

Note μPD78F0338 only

4.2.1 Port 0

Port 0 is a 6-bit I/O port with output latch. Input/output mode can be specified for pins P00 to P05 in 1-bit units using port mode register 0 (PM0). An on-chip pull-up resistor can be used for the P00 to P05 pins in 1-bit units using pull-up resistor option register 0 (PU0).

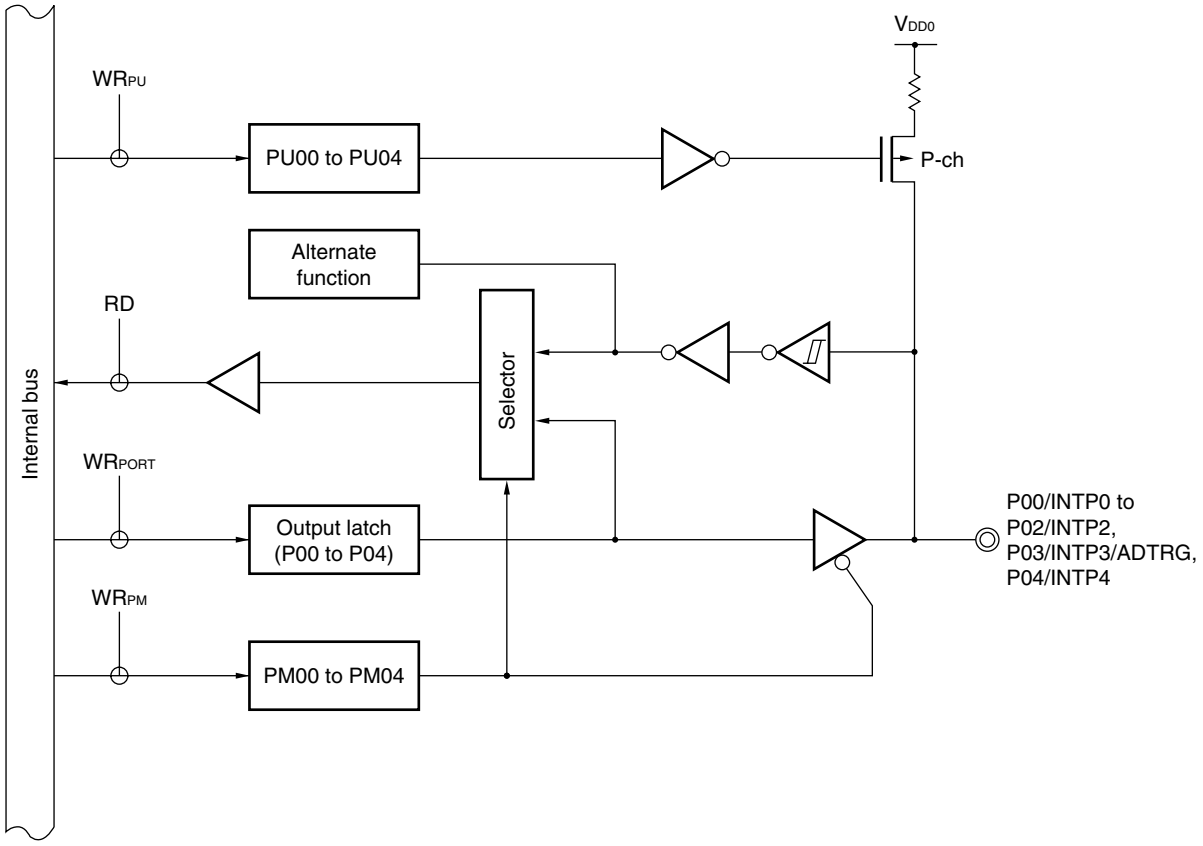
This port can also be used as an external interrupt request input, A/D converter external trigger input, clock output, and buzzer output.

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

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

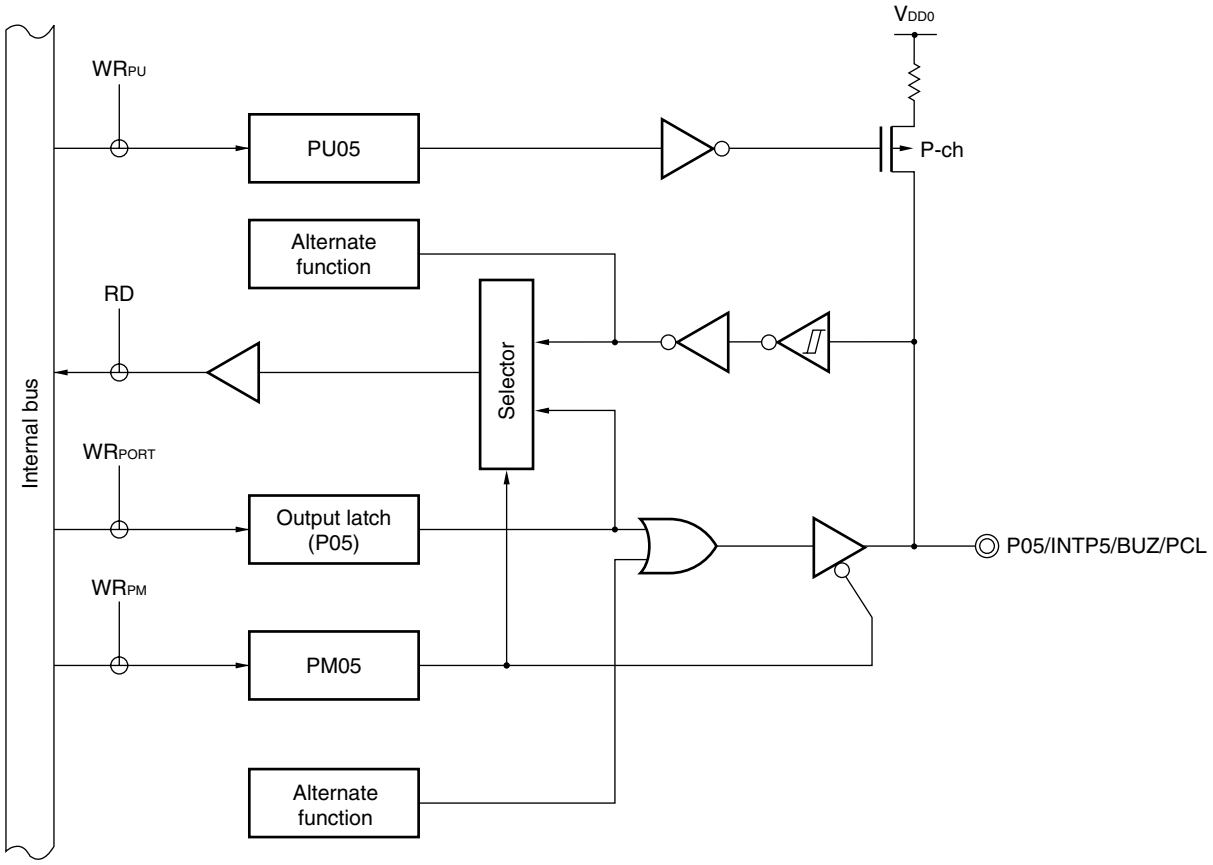
Caution Because port 0 also serves as an external interrupt request input, when the port function output mode is specified and the output level is changed, the interrupt request flag is set. Thus, when the output mode is used, set the interrupt mask flag to 1.

Figure 4-2. P00 to P04 Block Diagram



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

Figure 4-3. P05 Block Diagram



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

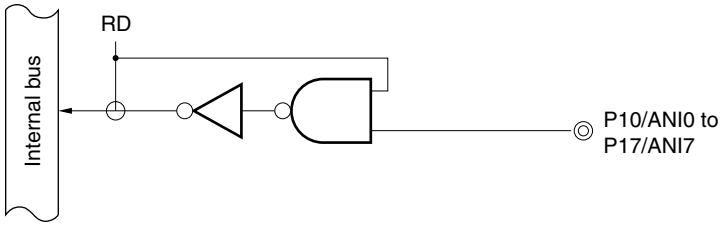
4.2.2 Port 1

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

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

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

Figure 4-4. P10 to P17 Block Diagram



RD: Port 1 read signal

4.2.3 Port 2

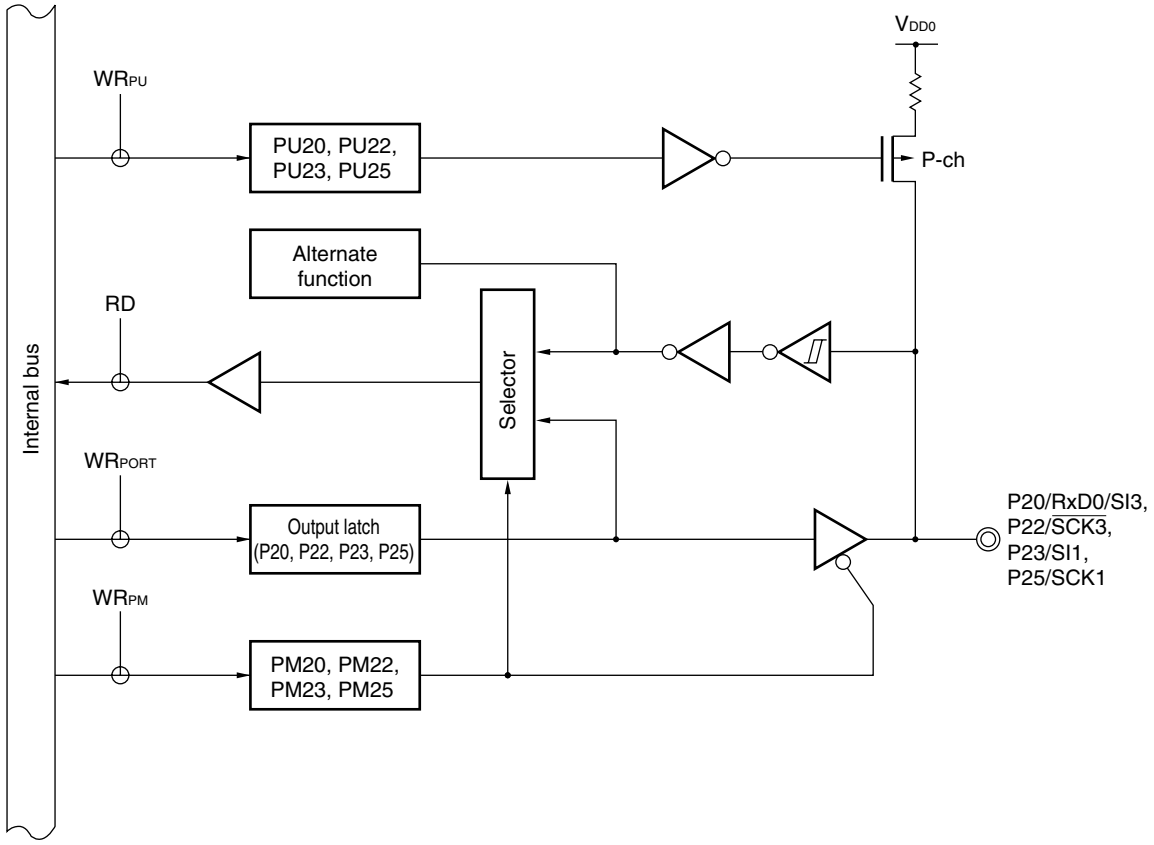
Port 2 is a 6-bit I/O port with output latch. Input/output mode can be specified for pins P20 to P25 in 1-bit units using port mode register 2 (PM2). An on-chip pull-up resistor can be used for the P20 to P25 pins in 1-bit units using pull-up resistor option register 2 (PU2).

This port has also alternate functions as serial interface data I/O and clock I/O.

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

Figures 4-5 and 4-6 show block diagrams of port 2.

Figure 4-5. P20, P22, P23, P25 Block Diagram



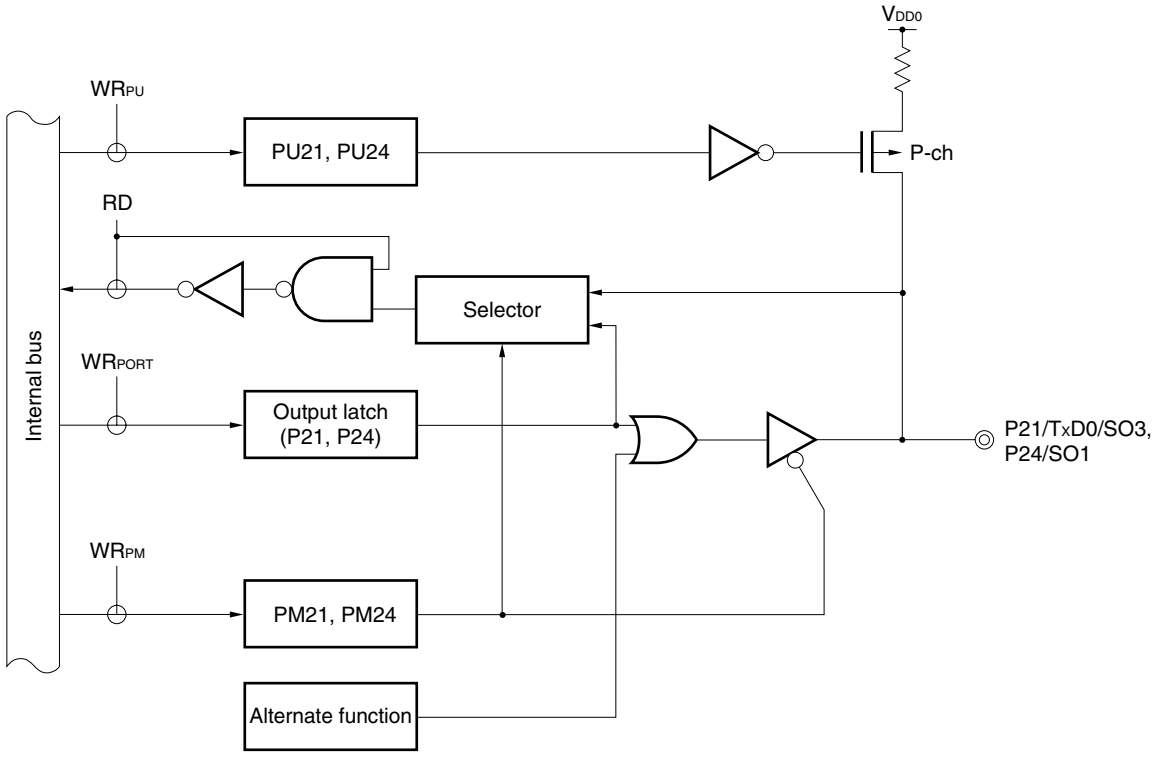
PU: Pull-up resistor option register

PM: Port mode register

RD: Port 2 read signal

WR: Port 2 write signal

Figure 4-6. P21, P24 Block Diagram



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

4.2.4 Port 3

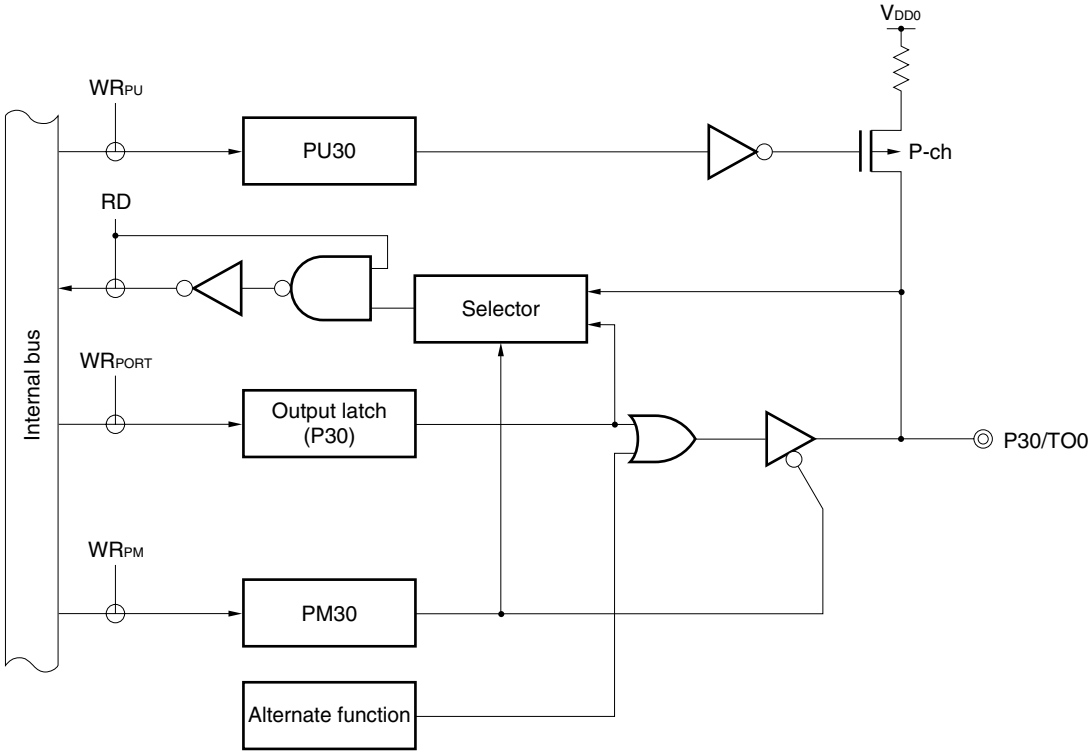
Port 3 is a 5-bit I/O port with output latch. Input/output mode can be specified for pins P30 to P34 in 1-bit units using port mode register 3 (PM3). An on-chip pull-up resistor can be used for the P30 to P34 pins in 1-bit units using pull-up resistor option register 3 (PU3).

This port has also alternate functions as timer I/O.

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

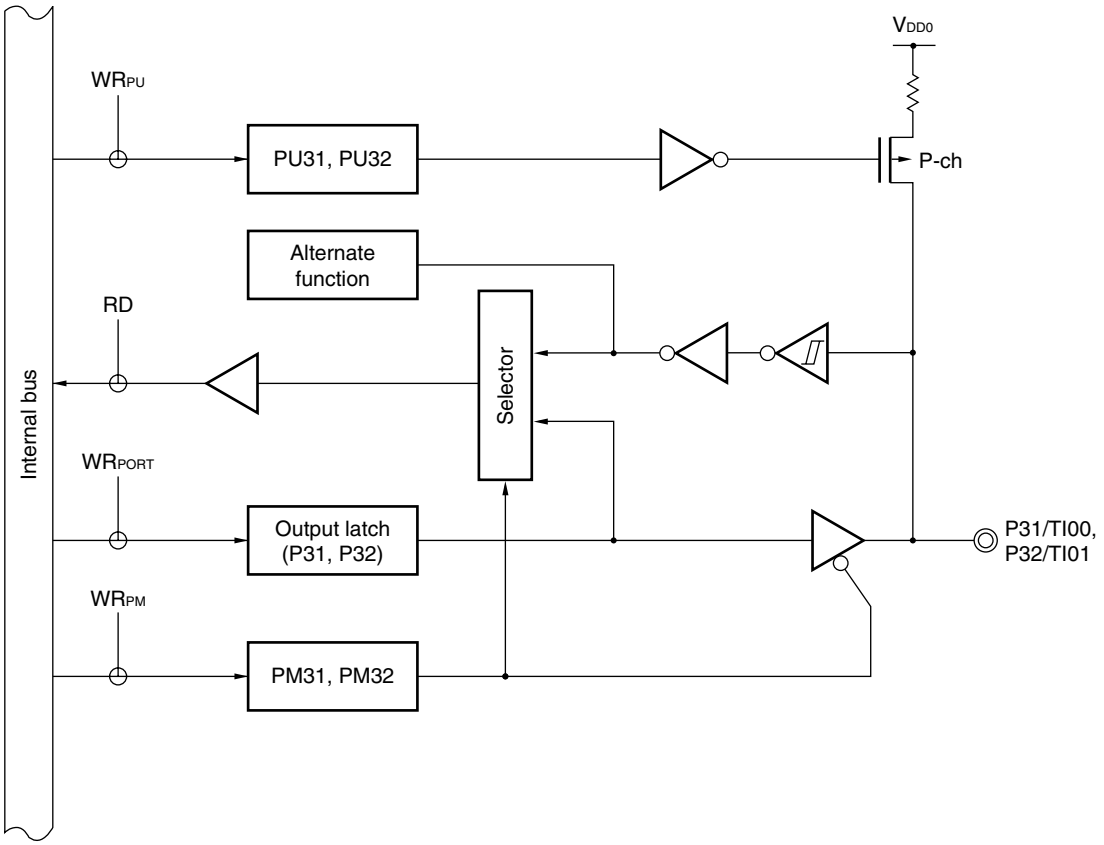
Figures 4-7 to 4-9 show block diagrams of port 3.

Figure 4-7. P30 Block Diagram



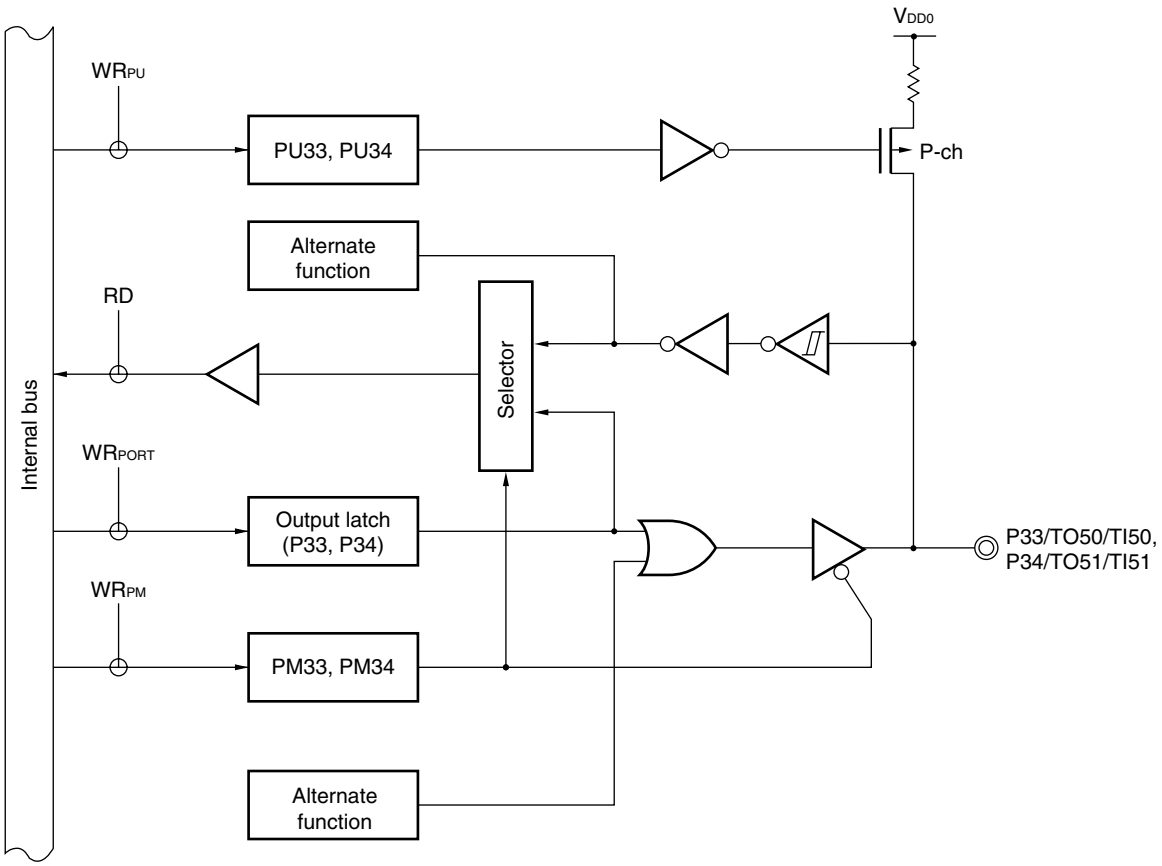
PU: Pull-up resistor option register
PM: Port mode register
RD: Port 3 read signal
WR: Port 3 write signal

Figure 4-8. P31, P32 Block Diagram



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

Figure 4-9. P33, P34 Block Diagram



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

4.2.5 Port 4

Port 4 is an 8-bit I/O port with output latch. Input/output mode can be specified for pins P40 to P47 in 1-bit units using port mode register 4 (PM4). An on-chip pull-up resistor can be used for the P40 to P47 pins in 1-bit units using pull-up resistor option register 4 (PU4).

The interrupt request flag (KRIF) can be set to 1 by falling edge detection.

The number of ports to detect the falling edge can be selected as either four (P40 to P43) or eight (P40 to P47) by setting bit 0 (KRSEL0) of the key return switching register (KRSEL).

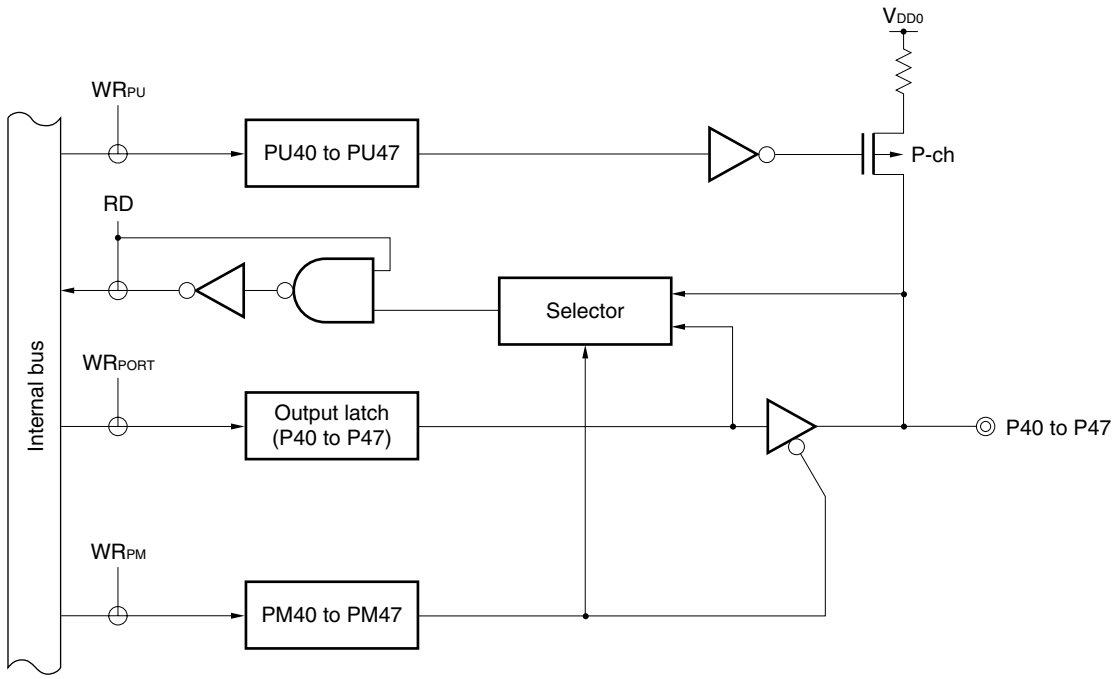
$\overline{\text{RESET}}$ input sets port 4 to input mode.

Figure 4-10 shows a block diagram of port 4 and Figure 4-11 shows a block diagram of the falling edge detector.

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

2. If the number of key returns is set to four, the key return function cannot be evaluated with an in-circuit emulator.

Figure 4-10. P40 to P47 Block Diagram



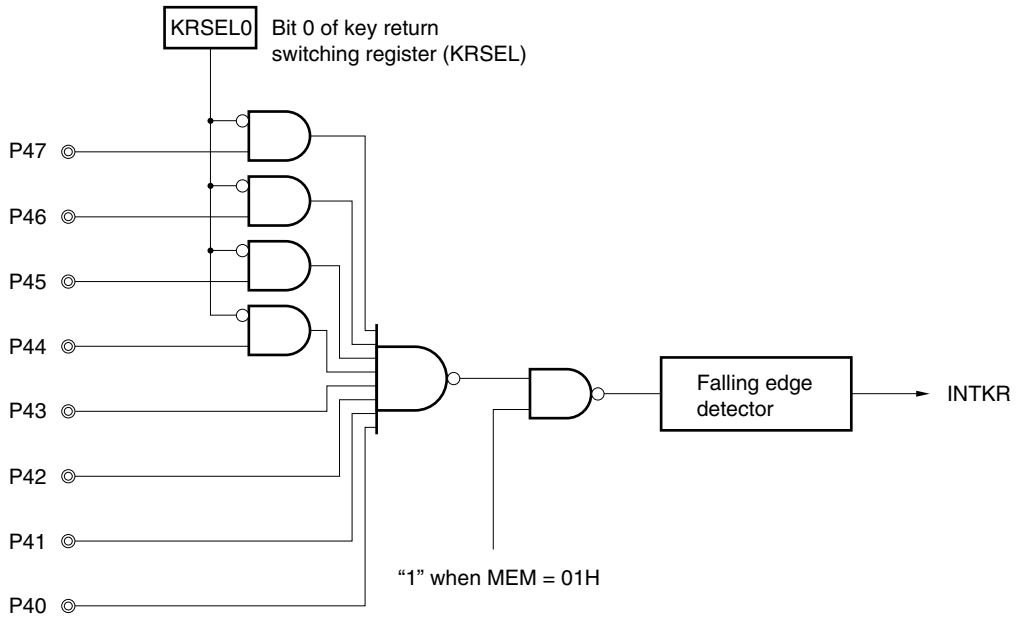
PU: Pull-up resistor option register

PM: Port mode register

RD: Port 4 read signal

WR: Port 4 write signal

Figure 4-11. Falling Edge Detector Block Diagram



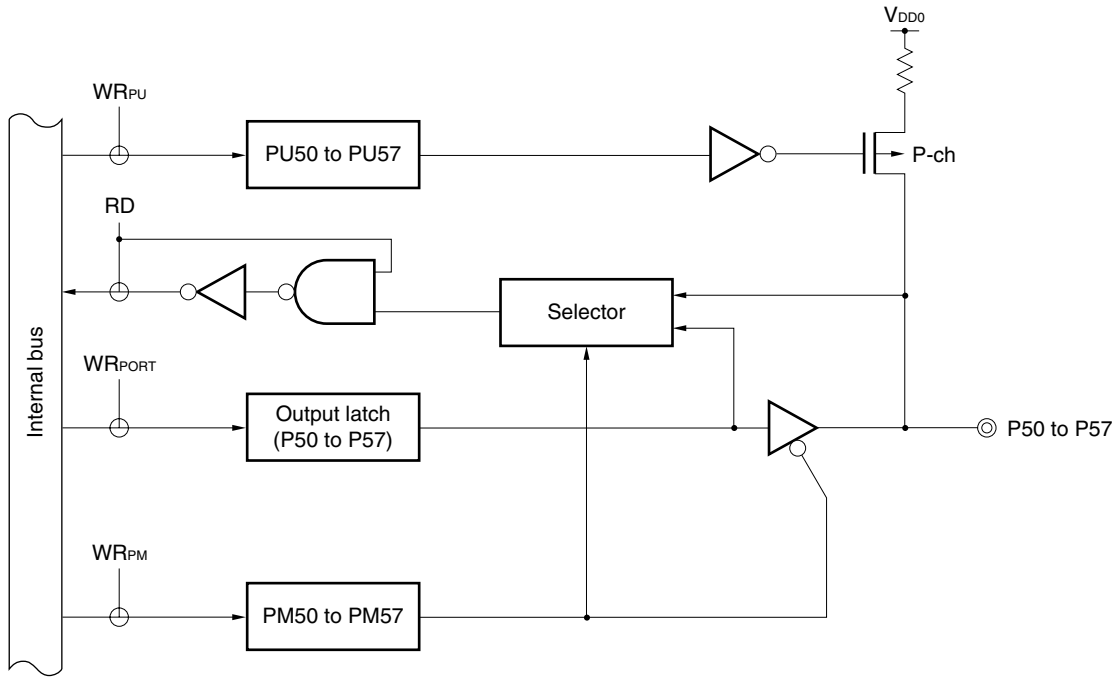
4.2.6 Port 5

Port 5 is an 8-bit I/O port with output latch. Input/output mode can be specified for pins P50 to P57 in 1-bit units using port mode register 5 (PM5). An on-chip pull-up resistor can be used for the P50 to P57 pins in 1-bit units using pull-up resistor option register 5 (PU5).

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

Figure 4-12 shows a block diagram of port 5.

Figure 4-12. P50 to P57 Block Diagram



PU: Pull-up resistor option register
PM: Port mode register
RD: Port 5 read signal
WR: Port 5 write signal

4.2.7 Port 6

Port 6 is an 8-bit I/O port with output latch. Input/output mode can be specified for pins P60 to P67 in 1-bit units using port mode register 6 (PM6).

This port has the following functions for pull-up resistors. These functions differ depending on the port's higher 4 bits/lower 4 bits, and whether the product is a mask ROM version or a flash memory version.

Table 4-4. Pull-Up Resistor of Port 6

	Higher 4 Bits (P64 to P67 Pins)	Lower 4 Bits (P60 to P63 Pins)
Mask ROM version	An on-chip pull-up resistor can be used in 1-bit units by PU6	On-chip pull-up resistor can be specified in 1-bit units by mask option
Flash memory version		On-chip pull-up resistor is not provided

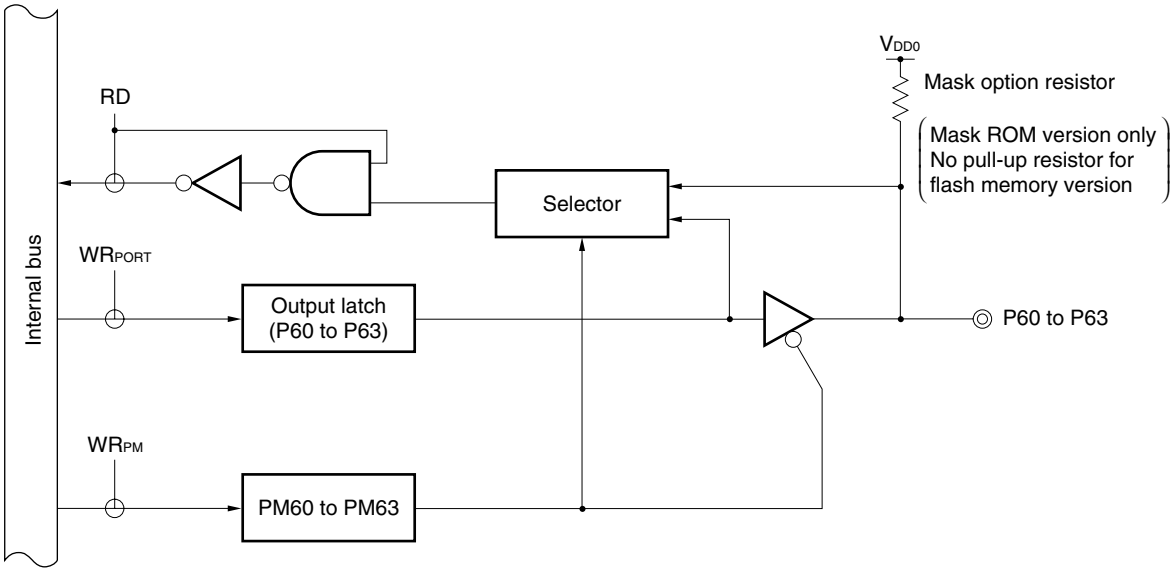
PU6: Pull-up resistor option register 6

The P60 to P67 pins can drive LEDs directly.

$\overline{\text{RESET}}$ input sets port 6 to input mode.

Figures 4-13 and 4-14 show block diagrams of port 6.

Figure 4-13. P60 to P63 Block Diagram

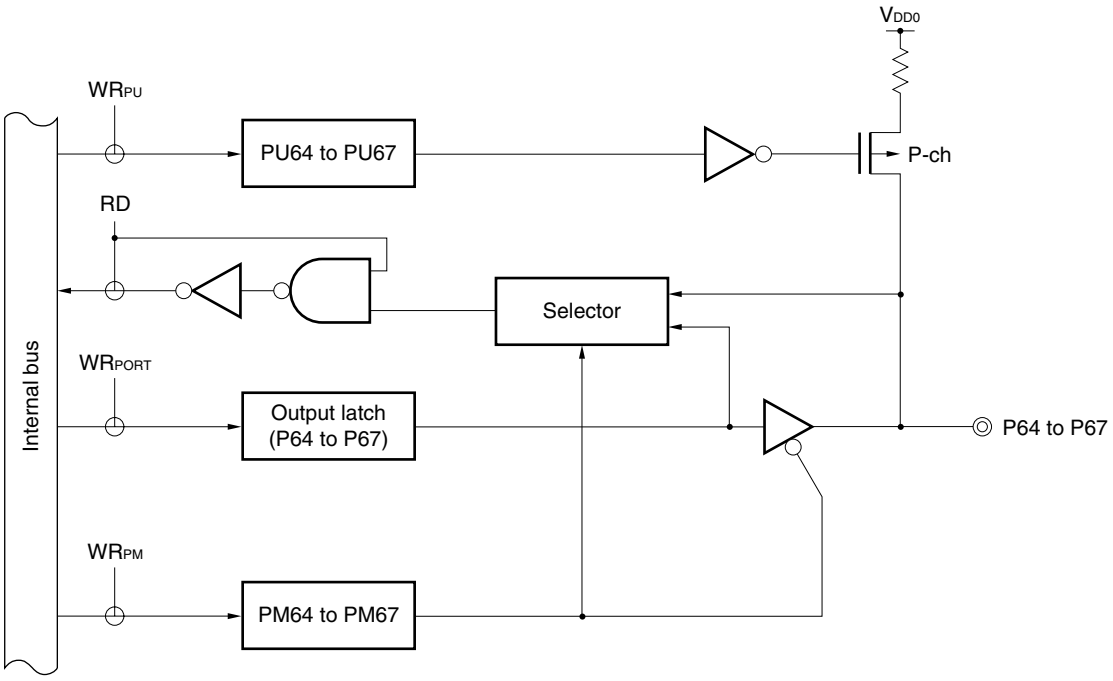


PM: Port mode register

RD: Port 6 read signal

WR: Port 6 write signal

Figure 4-14. P64 to P67 Block Diagram



PU: Pull-up resistor option register
PM: Port mode register
RD: Port 6 read signal
WR: Port 6 write signal

4.2.8 Port 7

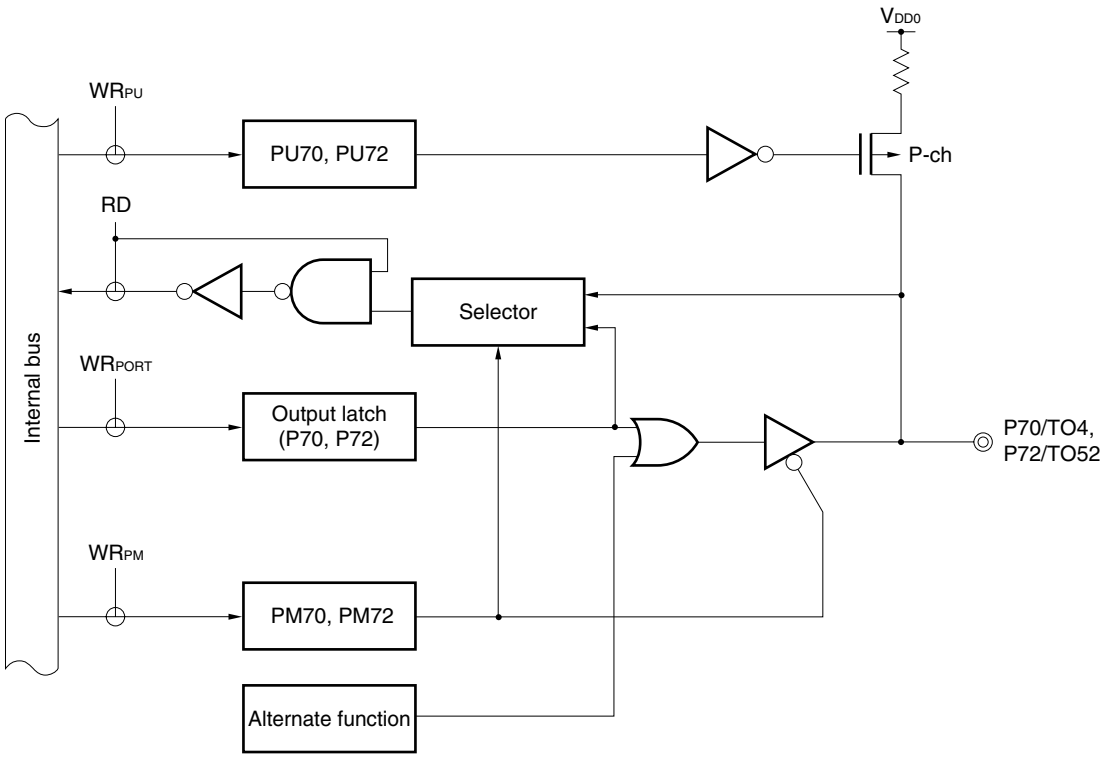
Port 7 is a 4-bit I/O port with output latches. Input/output mode can be specified in 1-bit units using port mode register 7 (PM7). An on-chip pull-up resistor can be used for the P70 to P73 pins in 1-bit units using pull-up resistor option register 7 (PU7).

This port can also be used as a timer I/O.

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

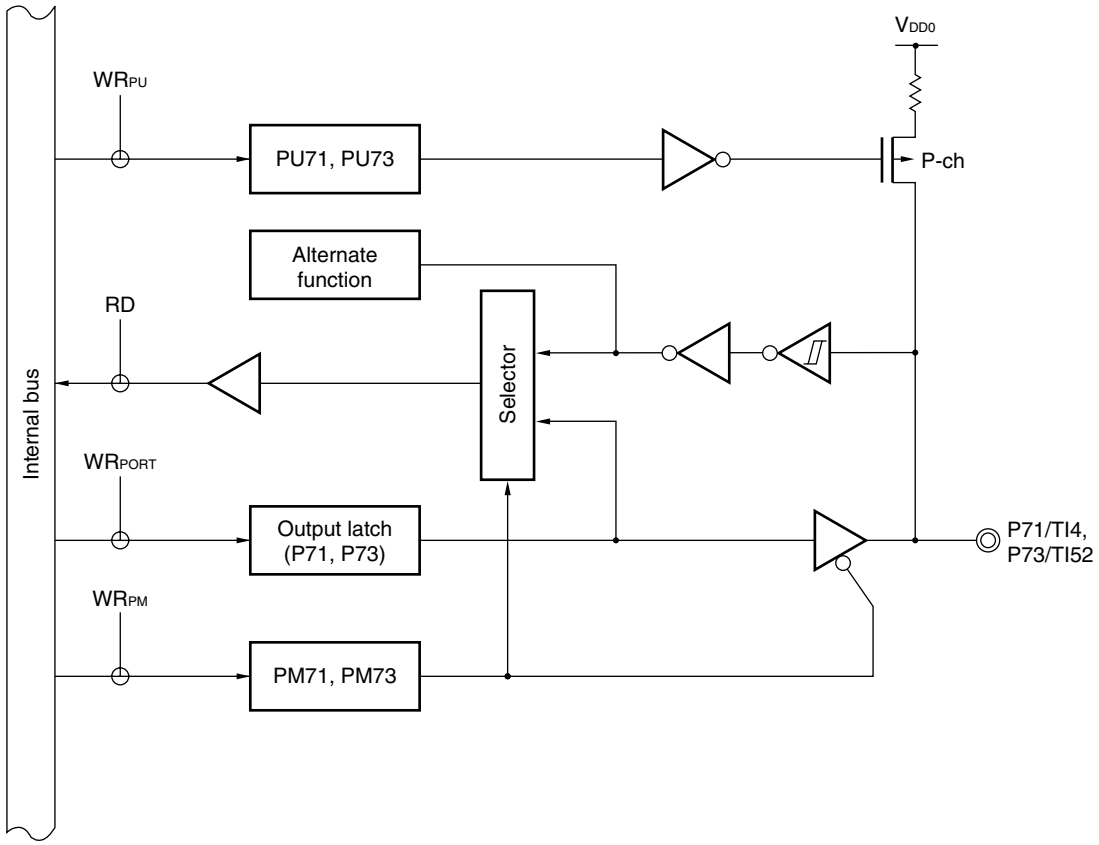
Figures 4-15 and 4-16 show block diagrams of port 7.

Figure 4-15. P70, P72 Block Diagram



PU: Pull-up resistor option register
PM: Port mode register
RD: Port 7 read signal
WR: Port 7 write signal

Figure 4-16. P71, P73 Block Diagram



PU: Pull-up resistor option register
PM: Port mode register
RD: Port 7 read signal
WR: Port 7 write signal

4.2.9 Ports 8 and 9 (mask ROM version)

Ports 8 and 9 are 8-bit output-only ports.

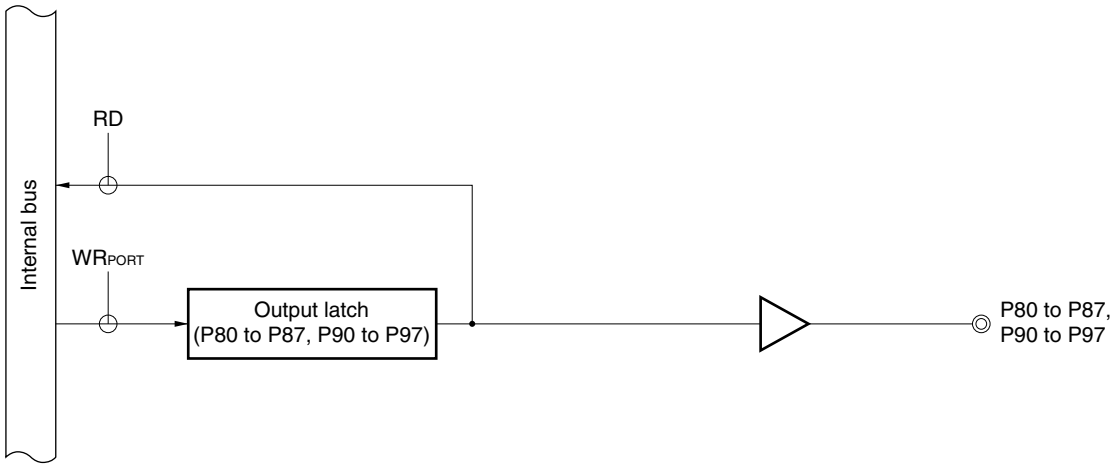
Ports 8 and 9 vary depending on the product.

Table 4-5. Ports 8 and 9 of Mask ROM Version

	Port 8	Port 9
μPD780316, 780318	P80 to P87 (without alternate pin)	P90 to P97 (without alternate pin)
μPD780326, 780328		None
μPD780336, 780338	None	

Figure 4-17 shows a block diagram of ports 8 and 9.

Figure 4-17. P80 to P87 and P90 to P97 Block Diagram (Mask ROM Version)



RD: Ports 8 and 9 read signal

WR: Ports 8 and 9 write signal

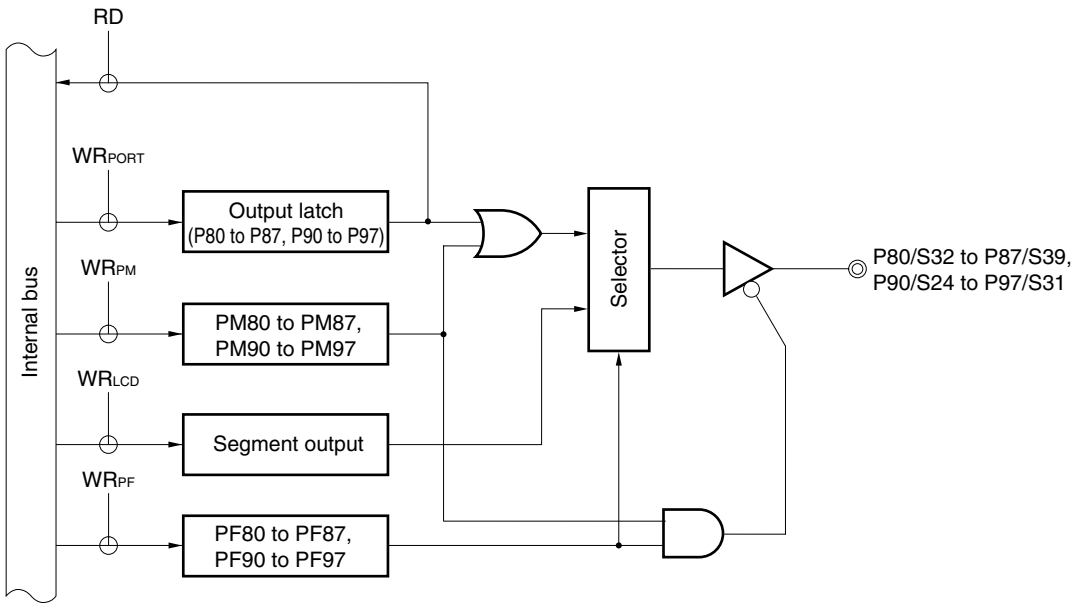
4.2.10 Ports 8 and 9 (flash memory version)

Ports 8 and 9 are 8-bit output-only ports.

These ports can also be used as an LCD controller/driver segment output.

Figure 4-18 shows a block diagram of ports 8 and 9.

★ **Figure 4-18. P80 to P87 and P90 to P97 Block Diagram (Flash Memory Version)**



PF: Pin function switching register

PM: Port mode register

RD: Ports 8 and 9 read signal

WR: Ports 8 and 9 write signal

Caution When ports 8 and 9 are used as dedicated output ports, set the pin function switching registers (PF8, PF9) of the port used to FFH and set the port mode registers (PM8, PM9) to 00H.

4.2.11 Port 12

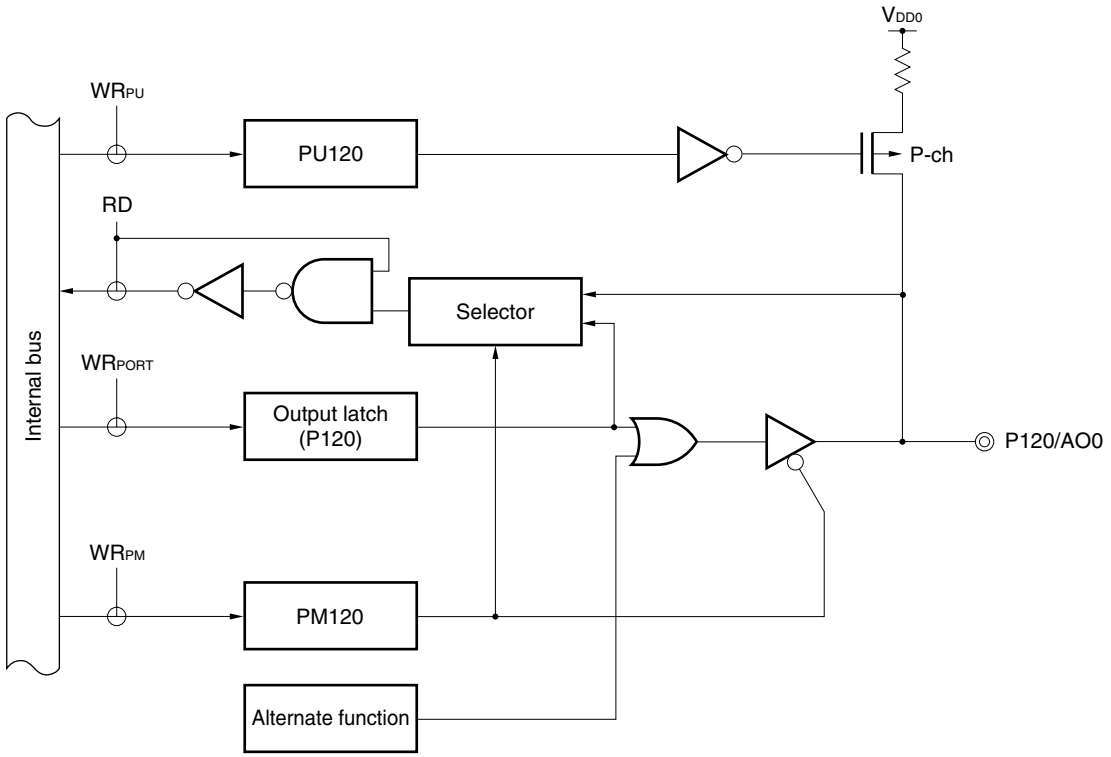
Port 12 is a 1-bit I/O port with output latch. Input/output mode can be specified for P120 in 1-bit units using port mode register 12 (PM12). An on-chip pull-up resistor can be used for the P120 pin in 1-bit units using pull-up resistor option register 12 (PU12).

This port also has an alternate function as the D/A converter analog output.

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

Figure 4-19 shows a block diagram of port 12.

Figure 4-19. P120 Block Diagram



PU: Pull-up resistor option register
PM: Port mode register
RD: Port 12 read signal
WR: Port 12 write signal

Caution Set port mode register 12 (PM12) and pull-up resistor option register 12 (PU12) as follows when used as D/A converter analog output.

- **Bit 0 (PM120) of PM12 to 1, input mode**
- **Bit 0 (PU120) of PU12 to 0, disconnect pull-up resistor** ★

4.3 Port Function Control Registers

The following five types of registers control the ports.

- Port mode registers (PM0, PM2 to PM7, PM8[Note], PM9[Note], PM12)
- Pull-up resistor option registers (PU0, PU2 to PU7, PU12)
- Memory expansion register (MEM)
- Key return switching register (KRSEL)
- Pin function switching registers 8 and 9 (PF8, PF9)[Note]

Note μPD78F0338 only

(1) Port mode registers (PM0, PM2 to PM7, PM8[Note], PM9[Note], PM12)

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

PM0, PM2 to PM7, and PM12 are independently set by a 1-bit or 8-bit memory manipulation instruction. PM8 and PM9 are independently set by an 8-bit memory manipulation instruction.

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

Note μPD78F0338 only.

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

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

3. If a port has an alternate function pin and it is used as an alternate output function, set the output latches (P0, P2 to P7, and P12) to 0.

Figure 4-20. Port Mode Registers (PM0, PM2 to PM9, PM12) Format

Address: FF20H After reset: FFH R/W

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

Address: FF22H After reset: FFH R/W

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

Address: FF23H After reset: FFH R/W

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

Address: FF24H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40

Address: FF25H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50

Address: FF26H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	PM67	PM66	PM65	PM64	PM63	PM62	PM61	PM60

Address: FF27H After reset: FFH R/W

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

Address: FF28H After reset: FFH W

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

Address: FF29H After reset: FFH W

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

Address: FF2CH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM12	1	1	1	1	1	1	1	PM120

PMmn	Pmn pin I/O mode selection (m = 0, 2 to 9, 12: n = 0 to 7)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

Note μPD78F0338 only. When ports 8 and 9 of the μPD78F0338 are used as dedicated output ports, set the pin function switching registers (PF8, PF9) of the port used to FFH and set the port mode registers (PM8, PM9) to 00H.

(2) Pull-up resistor option registers (PU0, PU2 to PU7, PU12)

These registers are used to set whether to use an on-chip pull-up resistor at each port or not. By setting PU0, PU2 to PU7, and PU12, the on-chip pull-up resistors of the port pins corresponding to the bits in PU0, PU2 to PU7, and PU12 can be used.

PU0, PU2 to PU7, and PU12 are set by a 1-bit or 8-bit memory manipulation instruction.

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

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

2. Pins P60 to P63 can be used with pull-up resistor by mask option only for mask ROM version.

3. When PUm is set to 1, the on-chip pull-up resistor is connected irrespective of the input/output mode. When using in output mode, therefore, set the bit of PUm to 0 (m = 0, 2 to 7, 12).

Figure 4-21. Pull-Up Resistor Option Registers (PU0, PU2 to PU7, PU12) Format

Address: FF30H After reset: 00H R/W

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

Address: FF32H After reset: 00H R/W

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

Address: FF33H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU3	0	0	0	PU34	PU33	PU32	PU31	PU30

Address: FF34H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU4	PU47	PU46	PU45	PU44	PU43	PU42	PU41	PU40

Address: FF35H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU5	PU57	PU56	PU55	PU54	PU53	PU52	PU51	PU50

Address: FF36H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU6	PU67	PU66	PU65	PU64	0	0	0	0

Address: FF37H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU7	0	0	0	0	PU73	PU72	PU71	PU70

Address: FF3CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU12	0	0	0	0	0	0	0	PU120

PUmn	Pmn pin internal pull-up resistor selection (m = 0, 2 to 7, 12: n = 0 to 7)
0	On-chip pull-up resistor not used
1	On-chip pull-up resistor used

(3) Memory expansion mode register (MEM)

This register is used to set the mode of port 4.

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

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

Figure 4-22. Memory Expansion Mode Register (MEM) Format

Address: FF47H After reset: 00H R/W

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

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

Note P44 to P47 pins can be used as port pins if bit 0 (KRSEL0) of key return switching register (KRSEL) is set to 1. At this time, key return function cannot be evaluated with in-circuit emulator.

Caution Be sure to set MM1 and MM2 to 0.

(4) Key return switching register (KRSEL)

This register is used to set the pins used as key return signals (port 4 falling edge detection).

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

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

Figure 4-23. Key Return Switching Register (KRSEL) Format

Address: FF8FH After reset: 00H R/W[Note 1]

Symbol	7	6	5	4	3	2	1	0
KRSEL	0	0	0	0	0	0	0	KRSEL0

KRSEL0	Setting the pin used for port 4 falling edge detection
0	P40 to P47 are used as key return signal (port 4 falling edge detection)
1	P40 to P43 are used as key return signal (port 4 falling edge detection)[Note 2]

Notes 1. KRSEL can be accessed but its read value is not guaranteed.

2. P44 to P47 can be used as port pins.

Caution KRSEL0 can only be set once after reset. To change the value, reset the register.

(5) Pin function switching registers 8 and 9 (PF8, PF9)[Note]

These registers are used to select if ports 8 and 9 are used as port pins or segment pins.

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

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

Note μPD78F0338 only

Figure 4-24. Pin Function Switching Registers 8 and 9 (PF8, PF9) Format

Address: FF58H After reset: 00H W

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

Address: FF59H After reset: 00H W

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

PFn7	PFn6	PFn5	PFn4	PFn3	PFn2	PFn1	PFn0	Pin settings
0	0	0	0	0	0	0	0	Segment output (n = 8: S32 to S39, n = 9: S24 to S31)
1	1	1	1	1	1	1	1	Output-only port (n = 8: P87 to P80, n = 9: P97 to P90)
Other than above								Setting prohibited

Cautions 1. PF8 and PF9 can only be set to 00H or FFH once after reset. Do not set any values other than 00H and FFH. To change values, reset the register.

2. When ports 8 and 9 are used as dedicated output ports, set the pin function switching registers (PF8, PF9) of the port used to FFH and set the port mode registers (PM8, PM9) to 00H.

4.4 Port Function Operations

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

4.4.1 Writing to I/O port

(1) Output mode

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

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

(2) Input mode

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

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

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

4.4.2 Reading from I/O port

(1) Output mode

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

(2) Input mode

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

4.4.3 Operations on I/O port

(1) Output mode

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

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

(2) Input mode

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

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

4.5 Selection of Mask Option

The following mask option is provided in the mask ROM version. The flash memory versions have no mask options.

Table 4-6. Comparison Between Mask ROM Version and Flash Memory Version

Pin Name	Mask ROM Version	Flash Memory Version
Mask option for pins P60 to P63	On-chip pull-up resistors can be specified in 1-bit units	Cannot specify an on-chip pull-up resistor

CHAPTER 5 CLOCK GENERATOR

5.1 Clock Generator Functions

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

(1) Main system clock oscillator

This circuit oscillates at frequencies of 1 to 10 MHz. Oscillation can be stopped by executing the STOP instruction or setting the processor clock control register (PCC).

(2) Subsystem clock oscillator

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

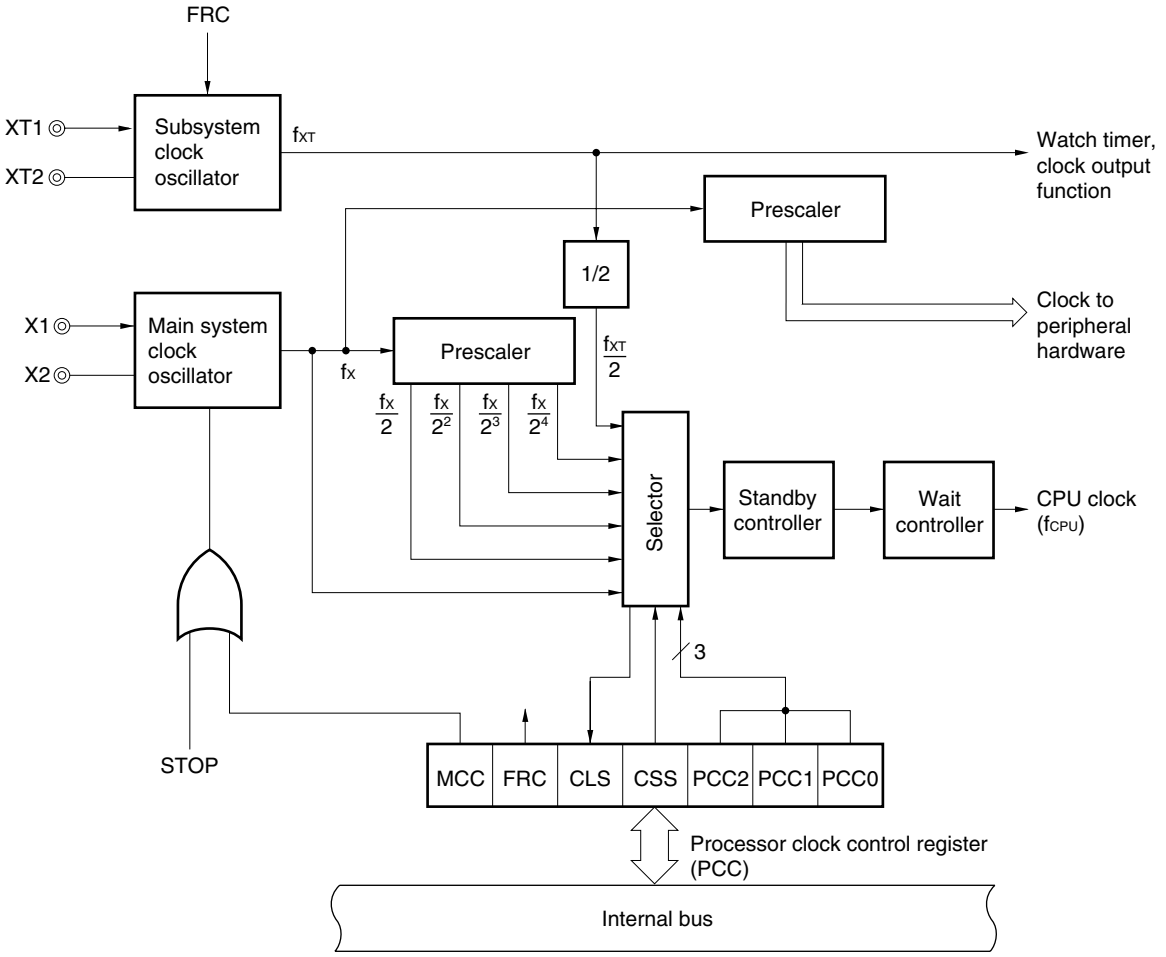
5.2 Clock Generator Configuration

The clock generator consists of the following hardware.

Table 5-1. Clock Generator Configuration

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

Figure 5-1. Clock Generator Block Diagram



5.3 Clock Generator Control Register

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

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

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

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

Figure 5-2. Subsystem Clock Feedback Resistor

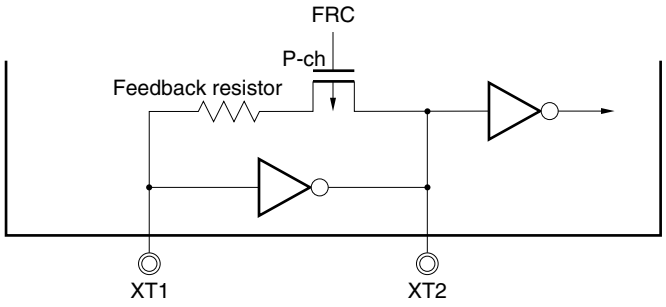


Figure 5-3. Processor Clock Control Register (PCC) Format

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

Symbol	7	6	5	4	3	2	1	0
PCC	MCC	FRC	CLS	CSS	0	PCC2	PCC1	PCC0

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

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

CLS	CPU clock status
0	Main system clock
1	Subsystem clock

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

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

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

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

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

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

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

The fastest instructions of μPD780318, 780328, and 780338 Subseries are carried out in two CPU clocks. The relationship of CPU clock (f_{CPU}) and minimum instruction execution time is shown in Table 5-2.

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

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

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

f_X: Main system clock oscillation frequency

f_{XT}: Subsystem clock oscillation frequency

5.4 System Clock Oscillator

5.4.1 Main system clock oscillator

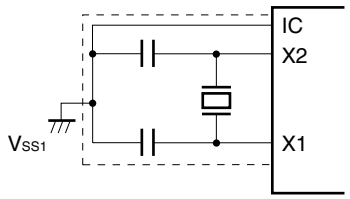
The main system clock oscillator oscillates with a crystal resonator or a ceramic resonator (10 MHz TYP.) connected to the X1 and X2 pins.

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

Figure 5-4 shows an external circuit of the main system clock oscillator.

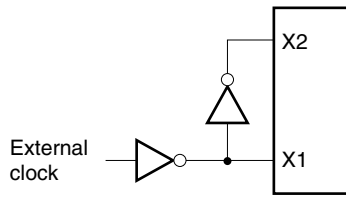
Figure 5-4. External Circuit of Main System Clock Oscillator

(a) Crystal and ceramic oscillation



Crystal resonator or ceramic resonator

(b) External clock



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

5.4.2 Subsystem clock oscillator

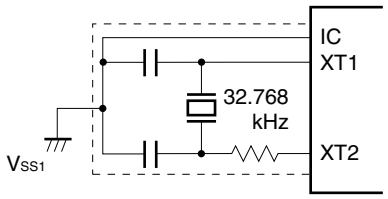
The subsystem clock oscillator oscillates with a crystal resonator (32.768 kHz TYP.) connected to the XT1 and XT2 pins.

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

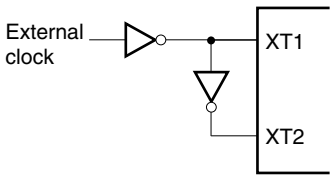
Figure 5-5 shows an external circuit of the subsystem clock oscillator.

Figure 5-5. External Circuit of Subsystem Clock Oscillator

(a) Crystal oscillation



(b) External clock



Cautions are listed on the next page.

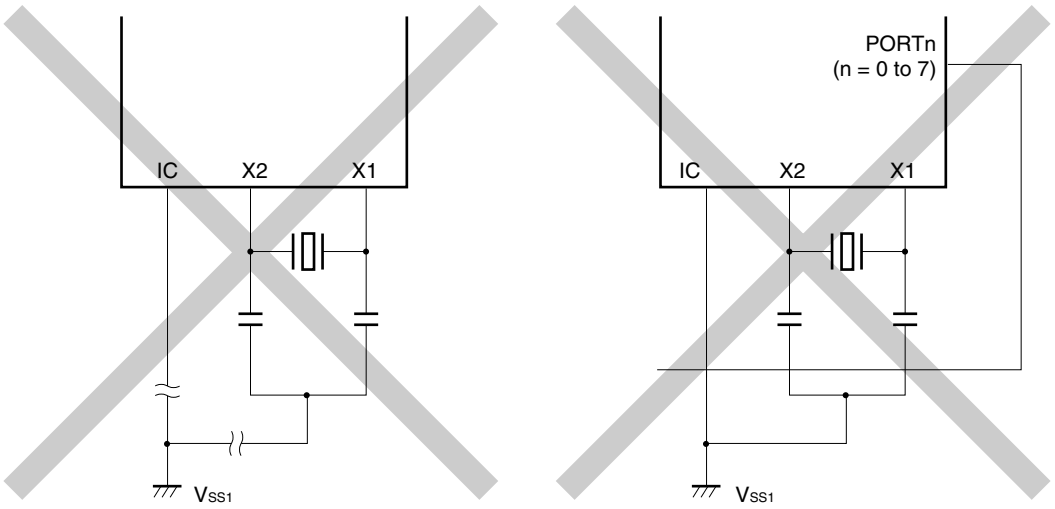
Cautions 1. When using the main system clock oscillator and a subsystem clock oscillator, carry out wiring in the broken-line area in Figures 5-4 and 5-5 to prevent any effects from wiring capacitance.

- **Minimize the wiring length.**
- **Do not allow wiring to intersect with other signal lines. Do not route the wiring in the vicinity of a line through which a high-fluctuating current flows.**
- **Always keep the ground of the capacitor of the oscillator at the same potential as V_{SS1}. Do not ground a capacitor to a ground pattern where high-current flows.**
- **Do not fetch signals from the oscillator.**

Take special note of the fact that the subsystem clock oscillator is a circuit with low-level amplification so that current consumption is maintained at low levels.
Figure 5-6 shows examples of incorrect resonator connection.

Figure 5-6. Examples of Incorrect Resonator Connection (1/2)

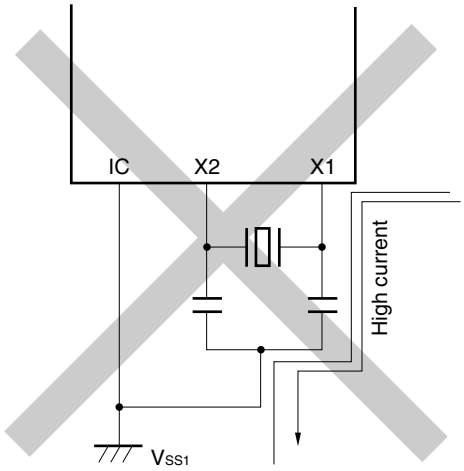
(a) Too long wiring **(b) Crossed signal line**



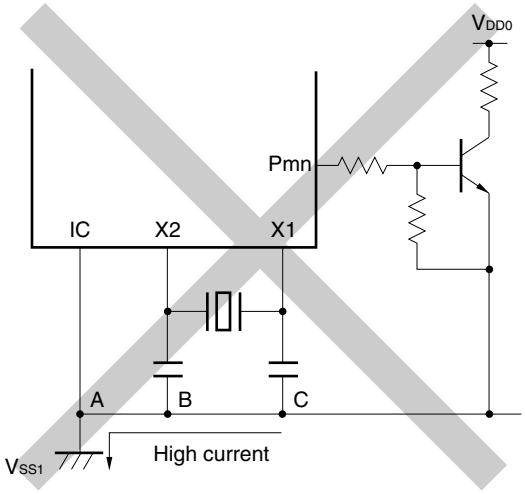
Remark When using a subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Further, insert resistors in series on the side of XT2.

Figure 5-6. Examples of Incorrect Resonator Connection (2/2)

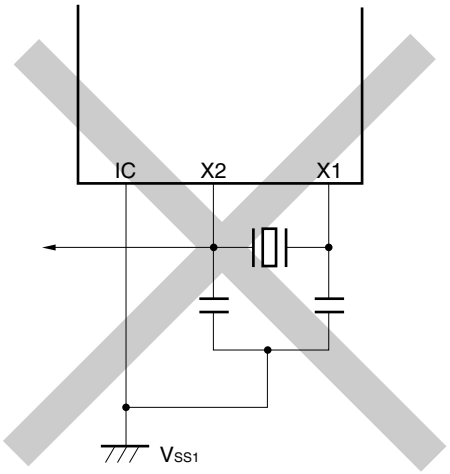
(c) Wiring near high alternating current



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



(e) Signals are fetched



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

Cautions 2. When X2 and XT1 are wired in parallel, the crosstalk noise of X2 may increase with XT1, resulting in malfunctioning.
To prevent that from occurring, it is recommended to wire X2 and XT1 so that they are not in parallel.

5.4.3 Divider

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

5.4.4 When no subsystem clocks are used

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

XT1: Connect to V_{DD0}
XT2: Open

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

5.5 Clock Generator Operations

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

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

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

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

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

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

(d) The PCC can be used to select the subsystem clock and to operate the system with low-current consumption (122 μs @32.768 kHz operation).

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

(f) The main system clock is divided and supplied to the peripheral hardware. The subsystem clock is supplied to the watch timer and clock output functions only. Thus the watch function and the clock output function can also be continued in the standby state. However, since all other peripheral hardware operate with the main system clock, the peripheral hardware also stops if the main system clock is stopped (except external input clock operation).

5.5.1 Main system clock operations

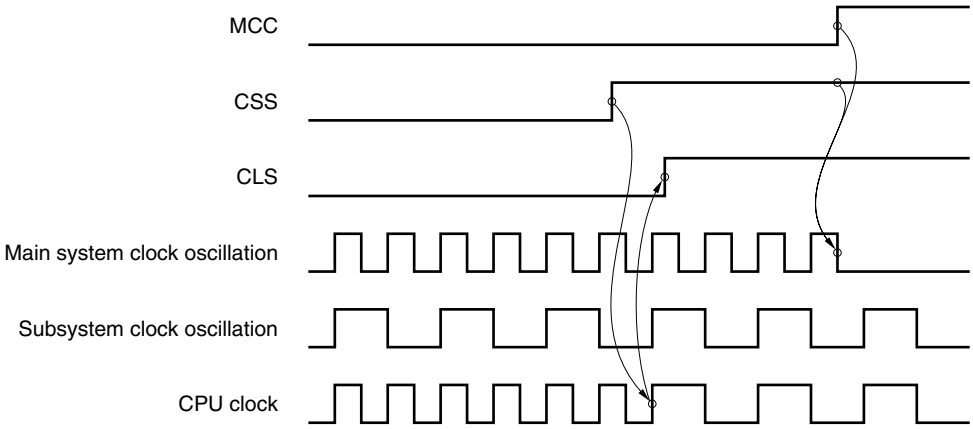
When operated with the main system clock (with bit 5 (CLS) of the processor clock control register (PCC) set to 0), the following operations are carried out by PCC setting.

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

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

Figure 5-7. Main System Clock Stop Function (1/2)

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



(b) Operation when MCC is set in case of main system clock operation

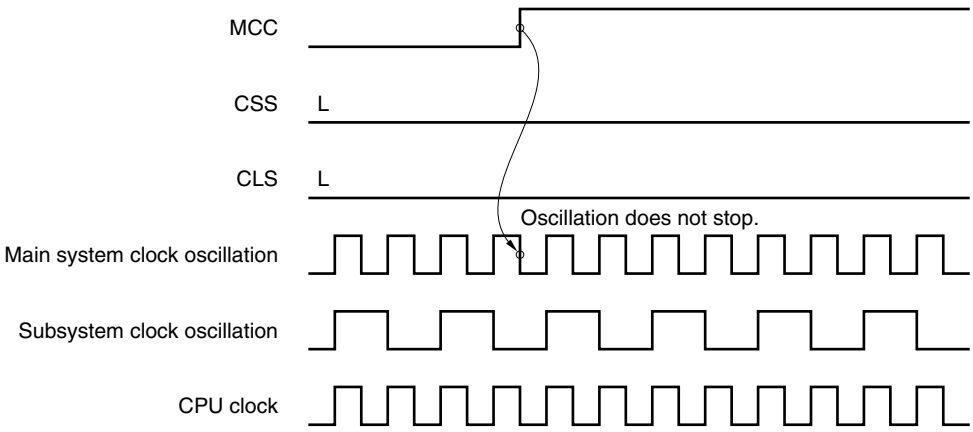
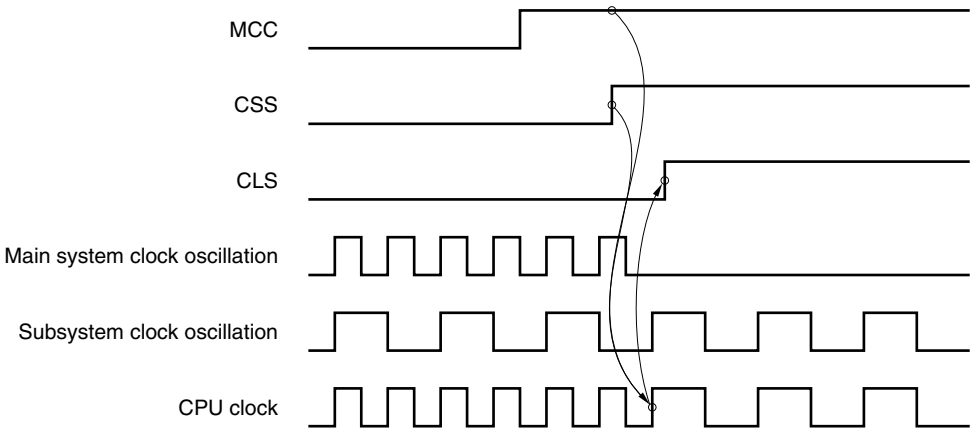


Figure 5-7. Main System Clock Stop Function (2/2)

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



5.5.2 Subsystem clock operations

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

(a) The minimum instruction execution time remains constant (122 μs @32.768 kHz operation) irrespective of bits 0 to 2 (PCC0 to PCC2) of the PCC.

(b) Watchdog timer counting stops.

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

5.6 Changing System Clock and CPU Clock Settings

5.6.1 Time required for switchover between system clock and CPU clock

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

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

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

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

Set Value Before Switchover				Set Value After Switchover																							
CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0
				0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	1	×	×	×
0	0	0	0					16 instructions				16 instructions				16 instructions				16 instructions				$f_X/2f_{XT}$ instruction (153 instructions)			
	0	0	1	8 instructions								8 instructions				8 instructions				8 instructions				$f_X/4f_{XT}$ instruction (77 instructions)			
	0	1	0	4 instructions				4 instructions								4 instructions				4 instructions				$f_X/8f_{XT}$ instruction (39 instructions)			
	0	1	1	2 instructions				2 instructions				2 instructions								2 instructions				$f_X/16f_{XT}$ instruction (20 instructions)			
	1	0	0	1 instruction				1 instruction				1 instruction				1 instruction								$f_X/32f_{XT}$ instruction (10 instructions)			
1	×	×	×	1 instruction				1 instruction				1 instruction				1 instruction				1 instruction							

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

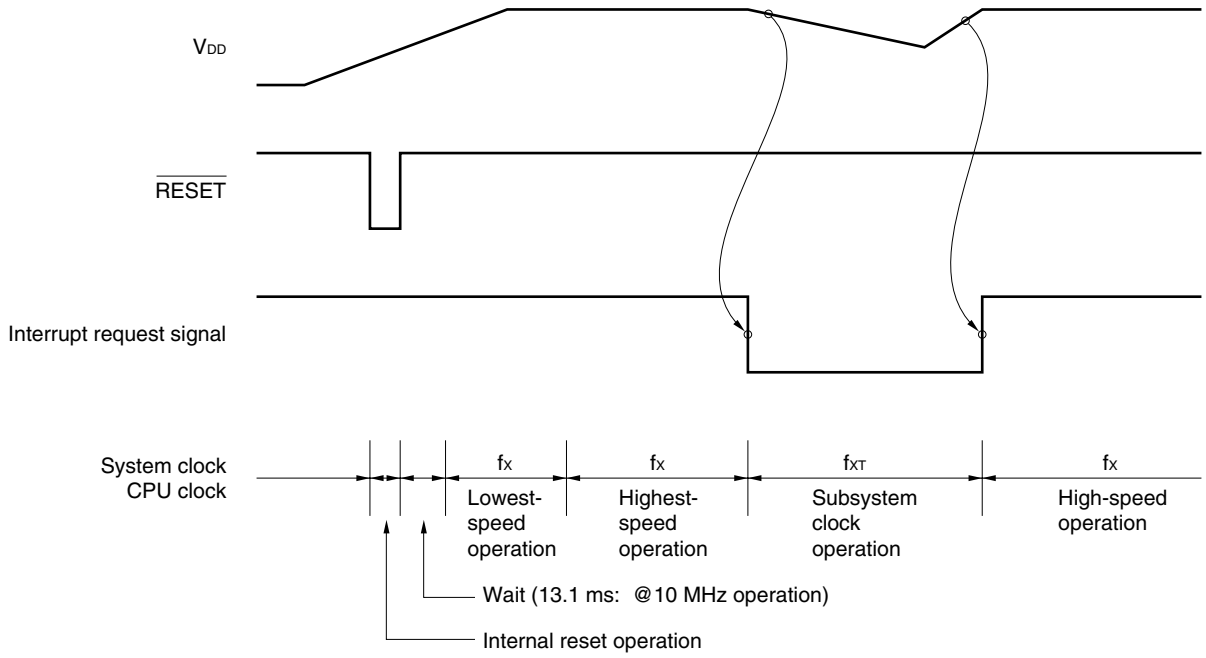
2. Figures in parentheses are for operation with f_X = 10 MHz and f_{XT} = 32.768 kHz.

Caution Selection of the CPU clock cycle division rate (PCC0 to PCC2) and switchover from the main system clock to the subsystem clock (changing CSS from 0 to 1) should not be set simultaneously. Simultaneous setting is possible, however, for selection of the CPU clock cycle division rate (PCC0 to PCC2) and switch over from the subsystem clock to the main system clock (changing CSS from 1 to 0).

5.6.2 System clock and CPU clock switching procedure

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

Figure 5-8. System Clock and CPU Clock Switching



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

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

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

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

Caution When subsystem clock is being operated while the main system clock is stopped, if switching to the main system clock is done again, be sure to switch after securing oscillation stabilization time by program.

CHAPTER 6 16-BIT TIMER/EVENT COUNTER 0

6.1 Outline of 16-Bit Timer/Event Counter 0

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

6.2 16-Bit Timer/Event Counter 0 Functions

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

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

(1) Interval timer

Generates an interrupt request at the preset time interval.

(2) PPG output

Can output a square wave whose frequency and output pulse can be set freely.

(3) Pulse width measurement

Can measure the pulse width of an externally input signal.

(4) External event counter

Can measure the number of pulses of an externally input signal.

(5) Square-wave output

Can output a square wave with any selected frequency.

6.3 16-Bit Timer/Event Counter 0 Configuration

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

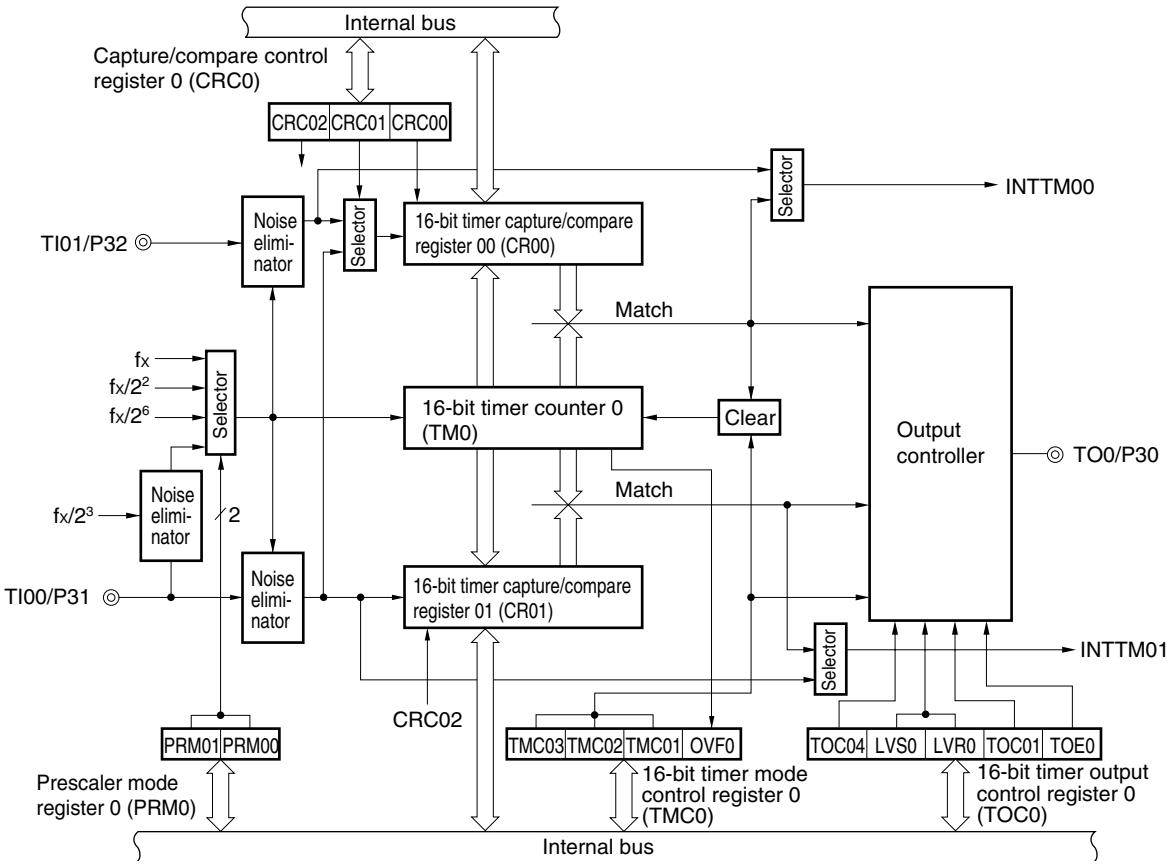
Table 6-1. 16-Bit Timer/Event Counter 0 Configuration

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

Note Refer to **Figure 4-7 P30 Block Diagram** and **Figure 4-8 P31, P32 Block Diagram.**

Figure 6-1 shows a block diagram.

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



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

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

The counter is incremented in synchronization with the rising edge of 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 TMC03 and TMC02 are cleared
<3> If valid edge of TI00 is input in the clear & start mode by inputting valid edge of TI00
<4> If TM0 and CR00 match with each other in the clear & start mode on match between TM0 and CR00

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

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

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

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

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

  It is possible to select the valid edge of the TI00/P31 pin or the TI01/P32 pin as the capture trigger. Setting of the TI00 or TI01 valid edge is performed by means of prescaler mode register 0 (PRM0).

  If CR00 is specified as a capture register and capture trigger is specified to be the valid edge of the TI00/P31 pin, the situation is as shown in Table 6-2. On the other hand, when capture trigger is specified to be the valid edge of the TI01/P32 pin, the situation is as shown in Table 6-3.

Table 6-2. TI00/P31 Pin Valid Edge and CR00, CR01 Capture Trigger

ES01	ES00	TI00/P31 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/P32 Pin Valid Edge and CR00 Capture Trigger

ES11	ES10	TI01/P32 Pin Valid Edge	CR00 Capture Trigger
0	0	Falling edge	Falling edge
0	1	Rising edge	Rising edge
1	0	Setting prohibited	Setting prohibited
1	1	Both rising and falling edges	Both rising and falling edges

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

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

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

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

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

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

- **When CR01 is used as a capture register**
  It is possible to select the valid edge of the TI00/P31 pin as the capture trigger. The TI00/P31 valid edge is set by means of prescaler mode register 0 (PRM0). Table 6-2 shows the setting when the valid edge of the TI00/P31 pin is specified as the capture trigger.

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

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

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

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

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

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

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

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

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

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

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

Address: FF60H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TMC0	0	0	0	0	TMC03	TMC02	TMC01	OVF0

TMC03	TMC02	TMC01	Operation mode and clear mode selection	TO0 output timing selection	Interrupt request generation
0	0	0	Operation stop (TM0 cleared to 0)	No change	Not generated
0	0	1			
0	1	0	Free-running mode	Match between TM0 and CR00 or match between TM0 and CR01	Generated on match between TM0 and CR00, or match between TM0 and CR01
0	1	1		Match between TM0 and CR00, match between TM0 and CR01 or TI00 valid edge	
1	0	0	Clear & start on TI00 valid edge	—	
1	0	1			
1	1	0	Clear & start on match between TM0 and CR00	Match between TM0 and CR00 or match between TM0 and CR01	
1	1	1		Match between TM0 and CR00, match between TM0 and CR01 or TI00 valid edge	

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

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

2. Set the valid edge of the TI00/P31 pin with prescaler mode register 0 (PRM0).

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

Remarks 1. TO0: 16-bit timer/event counter 0 output pin

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

3. TM0: 16-bit timer counter 0

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

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

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

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

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

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

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

Address: FF62H After reset: 00H R/W

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

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

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

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

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

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

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

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

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

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

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

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

Figure 6-4 shows the TOC0 format.

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

Address: FF63H 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	No change
0	1	Timer output F/F reset (0)
1	0	Timer output F/F set (1)
1	1	Setting prohibited

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

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

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

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

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

(4) Prescaler mode register 0 (PRM0)

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

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

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

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

Address: FF61H After reset: 00H R/W

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

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

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

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

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

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

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

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

Remarks 1. f_X: Main system clock oscillation frequency

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

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

(5) Port mode register 3 (PM3)

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

When using the P30/TO0 pin for timer output, set PM30 and the output latch of P30 to 0.

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

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

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

Address: FF23H After reset: FFH R/W

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

PM3n	P3n pin I/O mode selection (n = 0 to 4)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

6.5 16-Bit Timer/Event Counter 0 Operations

6.5.1 Interval timer operations

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

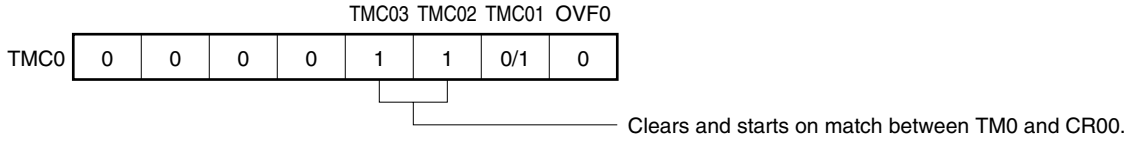
When the count value of 16-bit timer counter 0 (TM0) matches the value set to CR00, counting continues with the TM0 value cleared to 0 and the interrupt request signal (INTTM00) is generated.

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

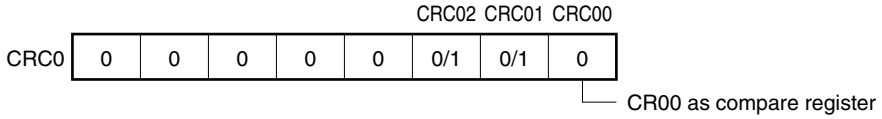
See **6.6 16-Bit Timer/Event Counter 0 Cautions (2)** about the operation when the compare register value is changed during timer count operation.

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

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



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



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

Figure 6-8. Interval Timer Configuration Diagram

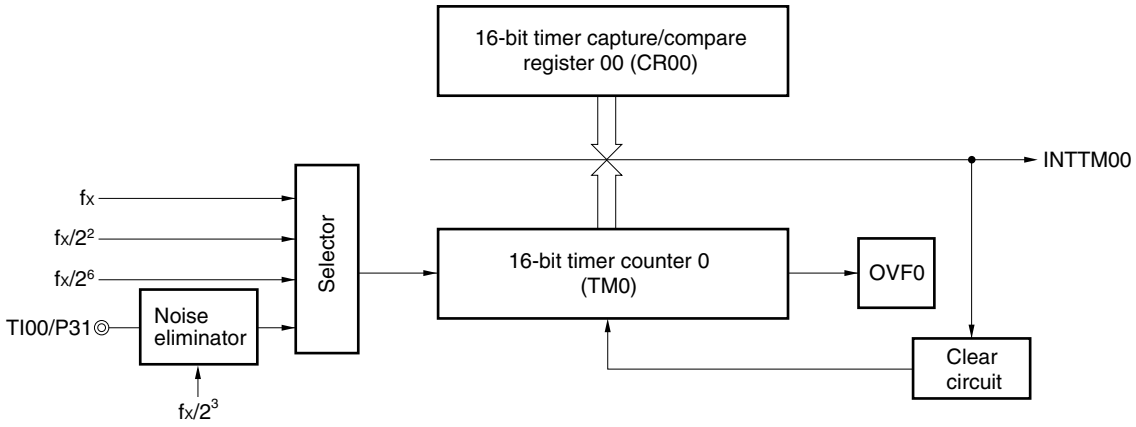
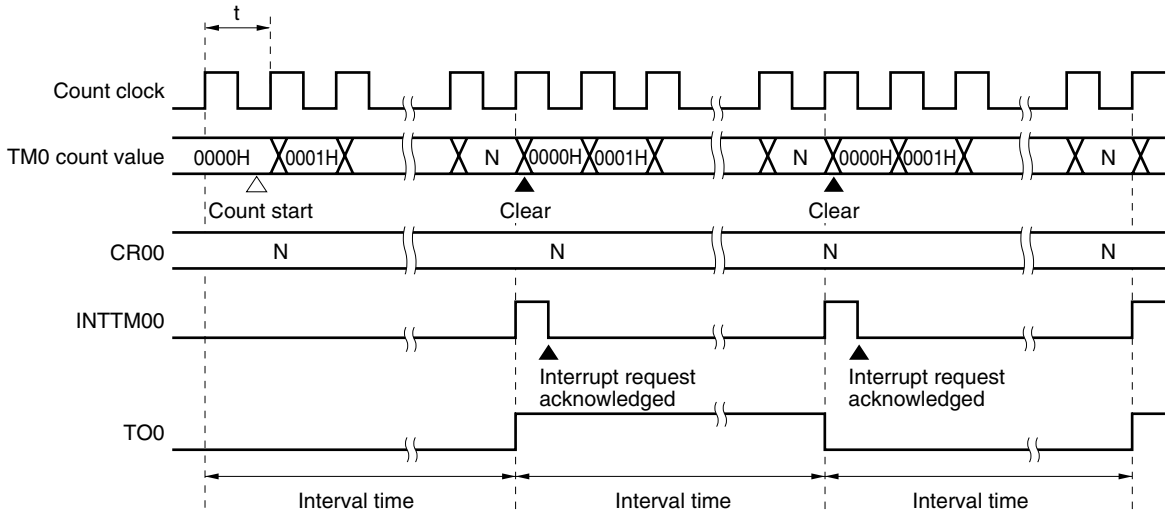


Figure 6-9. Timing of Interval Timer Operation



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

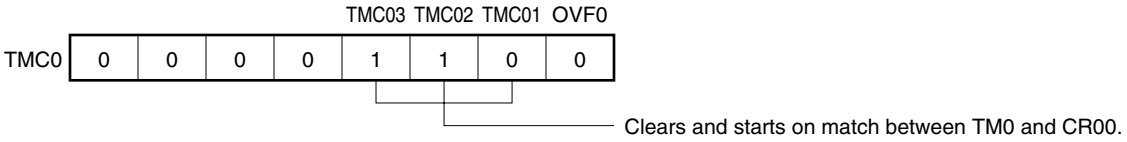
6.5.2 PPG output operations

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

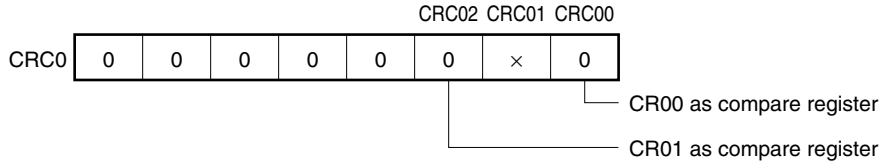
In the PPG output operation, square waves are output from the TO0/P30 pin with the pulse width and the cycle that correspond to the count values set beforehand in 16-bit timer capture/compare register 01 (CR01) and in 16-bit timer capture/compare register 00 (CR00), respectively.

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

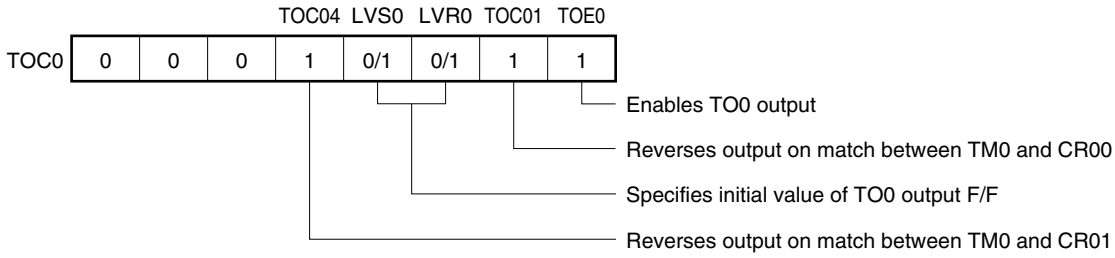
(a) 16-bit timer mode control register 0 (TMC0)



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



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



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

6.5.3 Pulse width measurement operations

It is possible to measure the pulse width of the signals input to the TI00/P31 pin and TI01/P32 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/P31 pin.

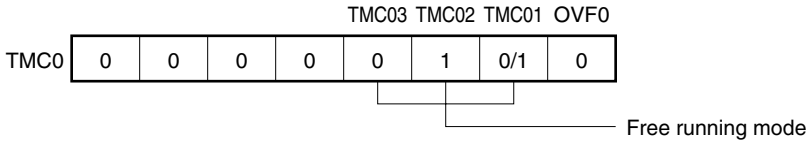
(1) Pulse width measurement with free-running counter and one capture register

When 16-bit timer counter 0 (TM0) is operated in free-running mode (see register settings in **Figure 6-11**), and the edge specified by prescaler mode register 0 (PRM0) is input to the TI00/P31 pin, the value of TM0 is taken into 16-bit timer capture/compare register 01 (CR01) and an external interrupt request signal (INTTM01) is set. Any of three edges can be selected—rising, falling, or both edges—specified by means of bits 4 and 5 (ES00 and ES01) of PRM0.

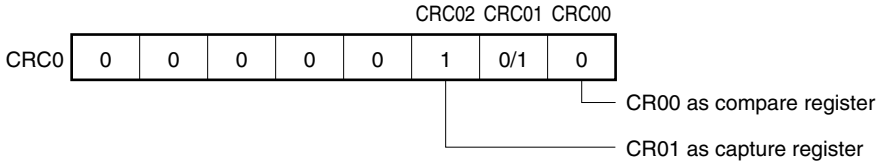
Sampling is performed at the count clock selected by PRM0, and a capture operation is only performed when a valid level of the TI00/P31 pin or TI01/P32 pin is detected twice, thus eliminating noise with a short pulse width.

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

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



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



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

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

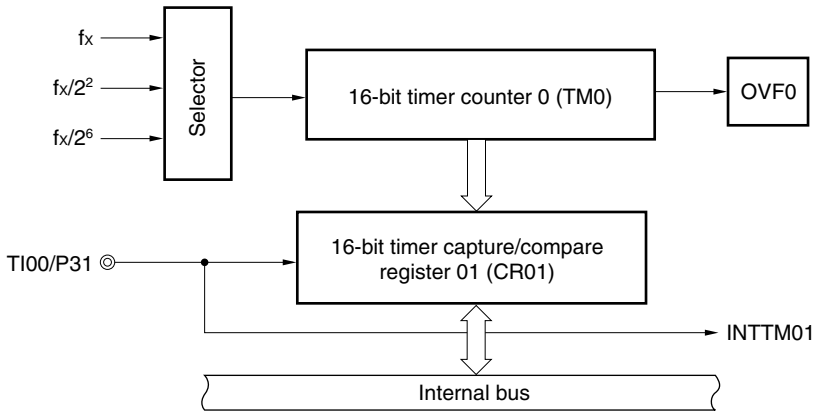
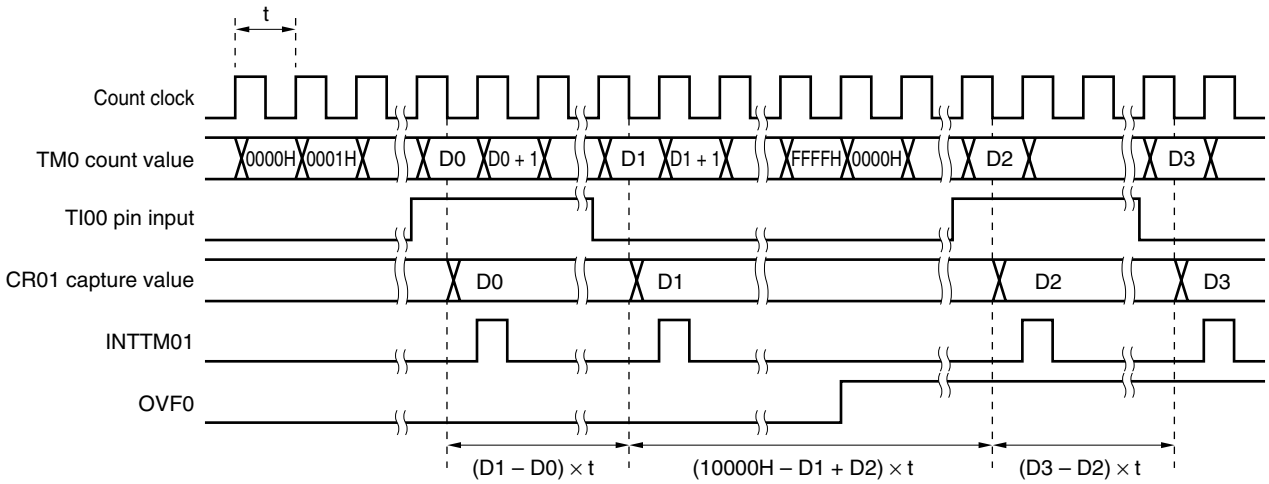


Figure 6-13. Timing of Pulse Width Measurement Operation by Free-Running Counter and One Capture Register (with Both Edges Specified)



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

When 16-bit timer counter 0 (TM0) is operated in free-running mode (see register settings in **Figure 6-14**), it is possible to simultaneously measure the pulse widths of the two signals input to the TI00/P31 pin and the TI01/P32 pin.

When the edge specified by bits 4 and 5 (ES00 and ES01) of prescaler mode register 0 (PRM0) is input to the TI00/P31 pin, the value of TM0 is taken into 16-bit timer capture/compare register 01 (CR01) and an interrupt request signal (INTTM01) is set.

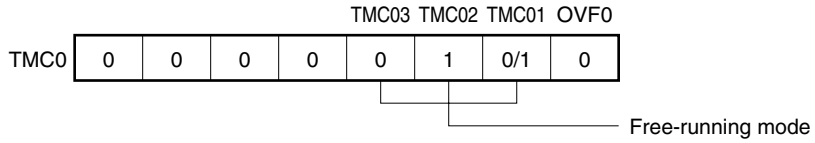
Also, when the edge specified by bits 6 and 7 (ES10 and ES11) of PRM0 is input to the TI01/P32 pin, the value of TM0 is taken into 16-bit timer capture/compare register 00 (CR00) and an interrupt request signal (INTTM00) is set.

Any of three edges can be selected—rising, falling, or both edges—as the valid edges for the TI00/P31 pin and the TI01/P32 pin specified by means of bits 4 and 5 (ES00 and ES01) and bits 6 and 7 (ES10 and ES11) of PRM0, respectively.

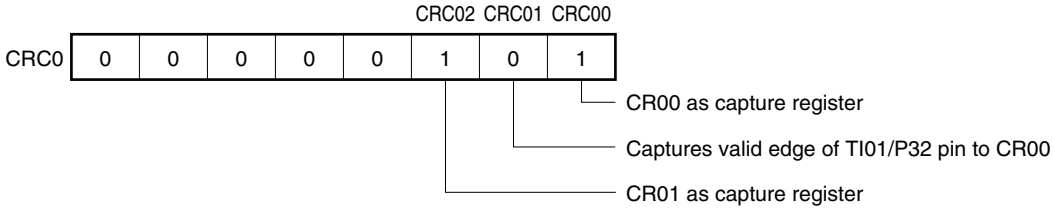
Sampling is performed at the interval selected by means of prescaler mode register 0 (PRM0), and a capture operation is only performed when a valid level of the TI00/P31 pin or TI01/P32 pin is detected twice, thus eliminating noise with a short pulse width.

Figure 6-14. Control Register Settings for Measurement of Two Pulse Widths with Free-Running Counter

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



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



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

- **Capture operation (free-running mode)**
  Capture register operation in capture trigger input is shown.

Figure 6-15. Capture Operation of CR01 with Rising Edge Specified

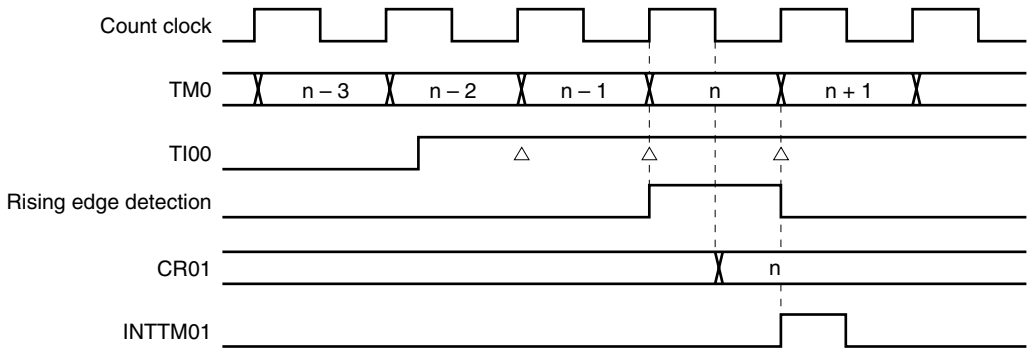
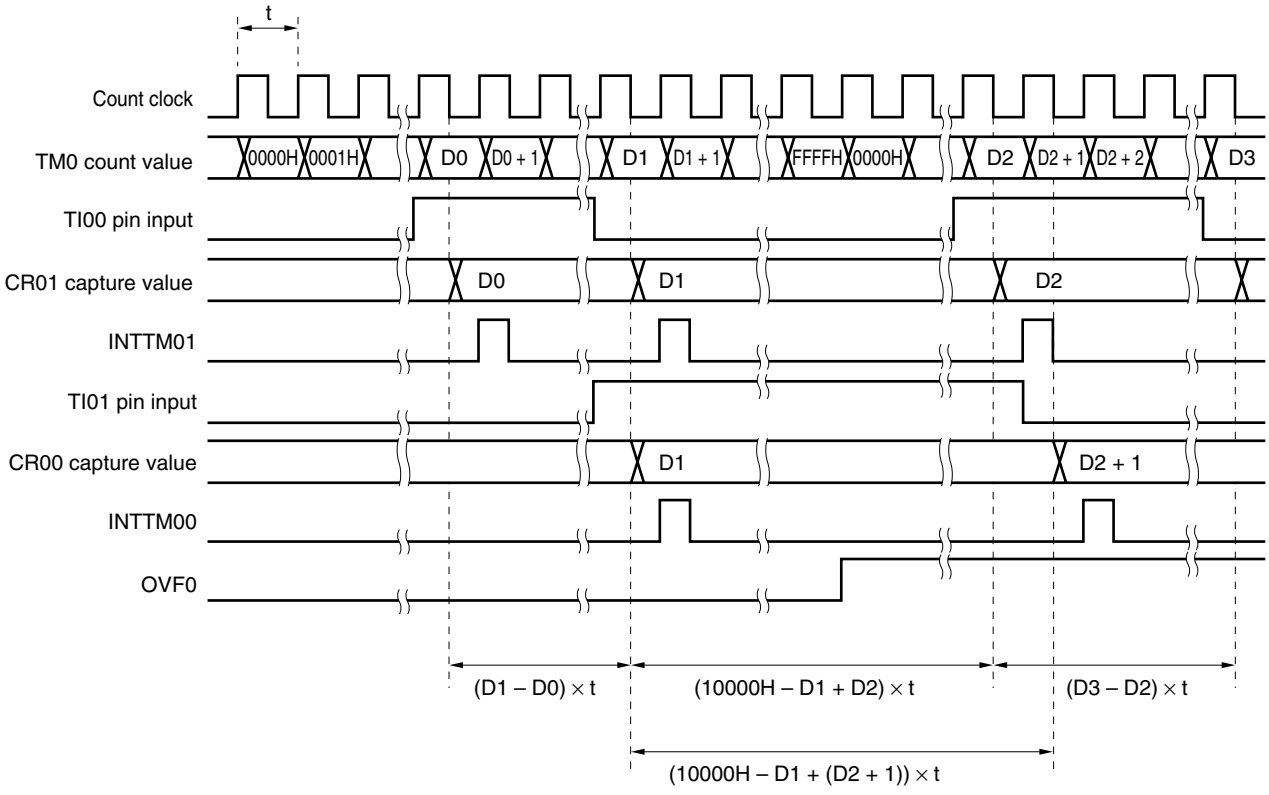


Figure 6-16. Timing of Pulse Width Measurement Operation with Free-Running Counter (with Both Edges Specified)



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

When 16-bit timer counter 0 (TM0) is operated in free-running mode (see register settings in **Figure 6-17**), it is possible to measure the pulse width of the signal input to the TI00/P31 pin.

When the edge specified by bits 4 and 5 (ES00 and ES01) of prescaler mode register 0 (PRM0) is input to the TI00/P31 pin, the value of TM0 is taken into 16-bit timer capture/compare register 01 (CR01) and an interrupt request signal (INTTM01) is set.

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

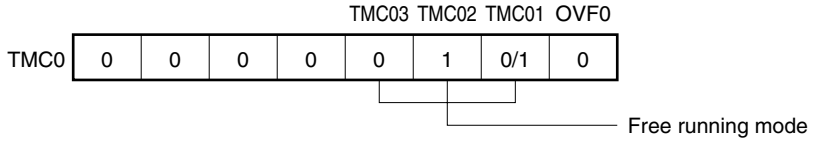
Either of two edges can be selected—rising or falling—as the valid edges for the TI00/P31 pin specified by means of bits 4 and 5 (ES00 and ES01) of prescaler mode register 0 (PRM0).

Sampling is performed at the interval selected by means of prescaler mode register 0 (PRM0), and a capture operation is only performed when a valid level of the TI00/P31 pin is detected twice, thus eliminating noise with a short pulse width.

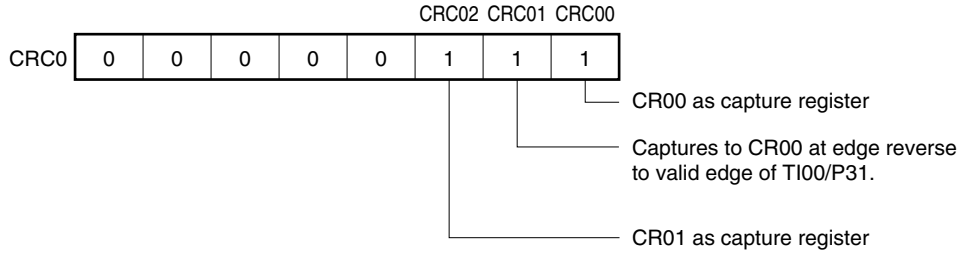
Caution If the valid edge of TI00/P31 pin is specified to be both rising and falling edges, 16-bit timer capture/compare register 00 (CR00) cannot perform the capture operation.

Figure 6-17. Control Register Settings for Pulse Width Measurement with Free-Running Counter and Two Capture Registers

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

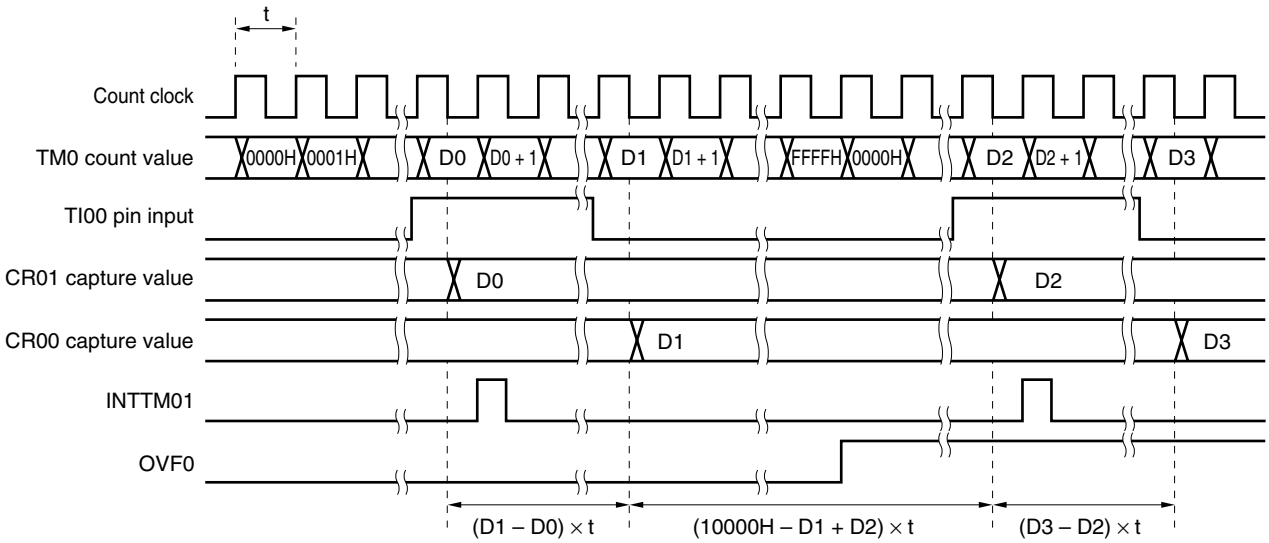


(b) Capture/compare control register 0 (CRC0)



Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. See the description of the respective control registers for details.

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



(4) Pulse width measurement by means of restart

When input of a valid edge to the TI00/P31 pin is detected, the count value of 16-bit timer counter 0 (TM0) is taken into 16-bit timer capture/compare register 01 (CR01), and then the pulse width of the signal input to the TI00/P31 pin is measured by clearing TM0 and restarting the count (see register settings in **Figure 6-19**).

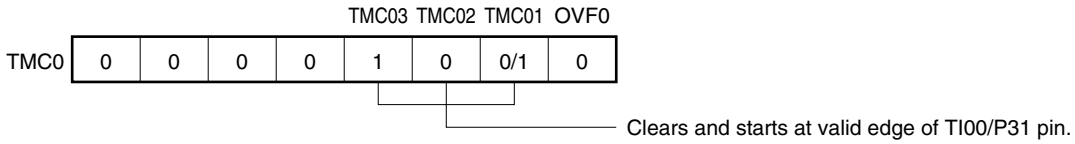
The edge specification can be selected from two types, rising and falling edges, by bits 4 and 5 (ES00 and ES01) of prescaler mode register 0 (PRM0).

In a valid edge detection, the sampling is performed by a cycle selected by prescaler mode register 0 (PRM0) and a capture operation is only performed when a valid level is detected twice, thus eliminating noise with a short pulse width.

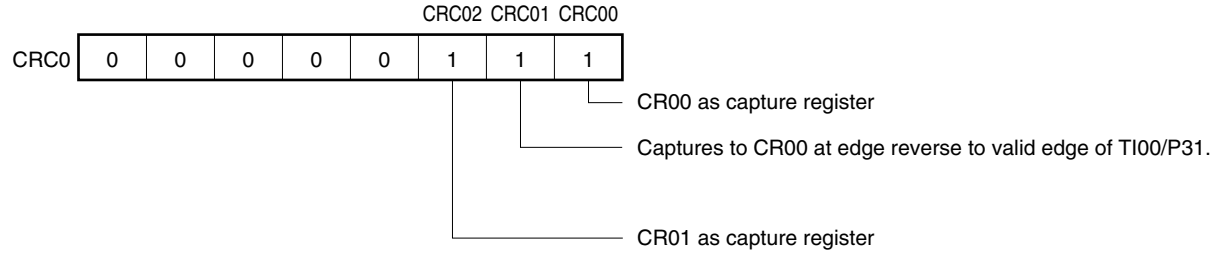
Caution If the valid edge of TI00/P31 pin is specified to be both rising and falling edges, 16-bit timer capture/compare register 00 (CR00) cannot perform the capture operation.

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

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

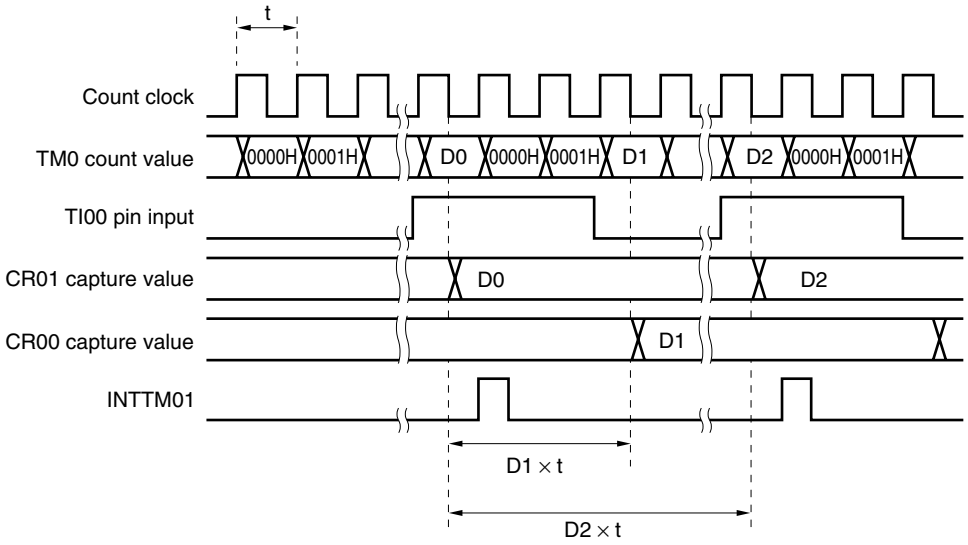


(b) Capture/compare control register 0 (CRC0)



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

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



6.5.4 External event counter operation

The external event counter counts the number of external clock pulses to be input to the TI00/P31 pin with 16-bit timer counter 0 (TM0).

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

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

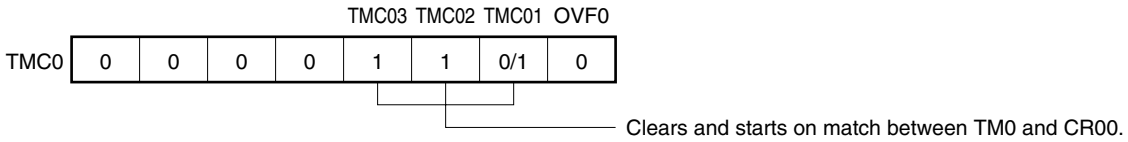
Input a value other than 0000H to CR00. (A count operation with a pulse cannot be carried out.)

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

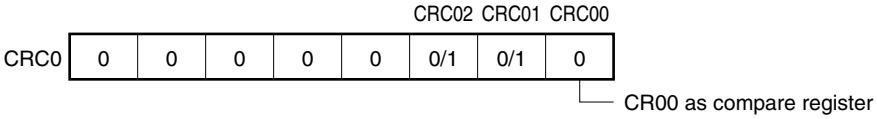
Because capture operation is carried out only after the valid edge of the TI00/P31 pin is detected twice by sampling with the internal clock ($f_X/2^3$), noise with short pulse widths can be eliminated.

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

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



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



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

Figure 6-22. External Event Counter Configuration Diagram

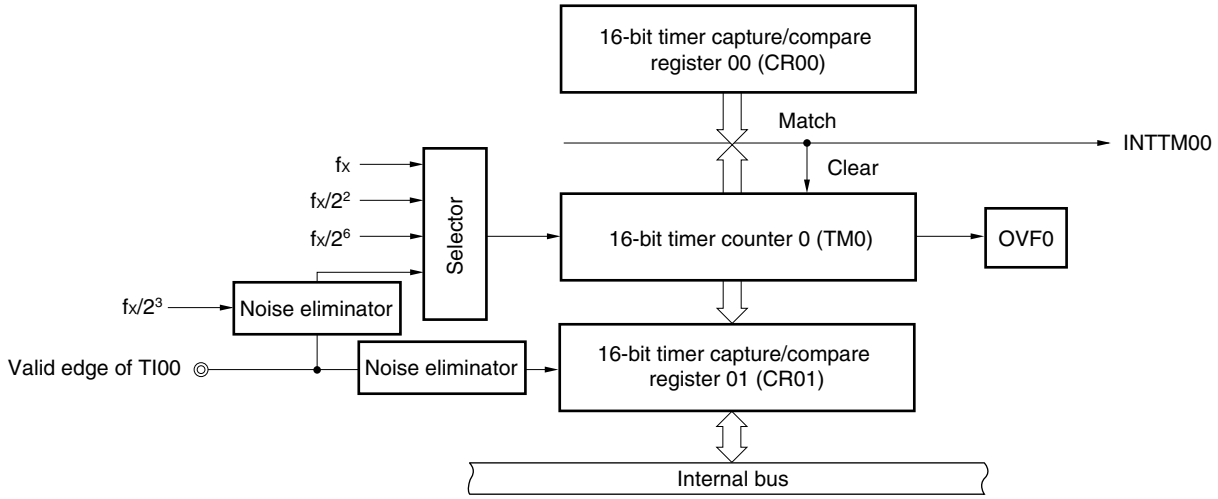
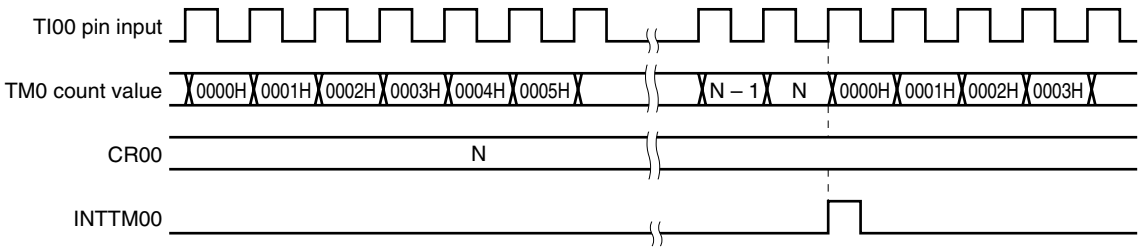


Figure 6-23. External Event Counter Operation Timings (with Rising Edge Specified)



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

6.5.5 Square-wave output operation

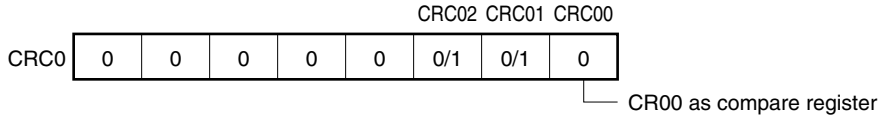
A square wave with any selected frequency can be output at intervals of the count value preset to 16-bit timer capture/compare register 00 (CR00).

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

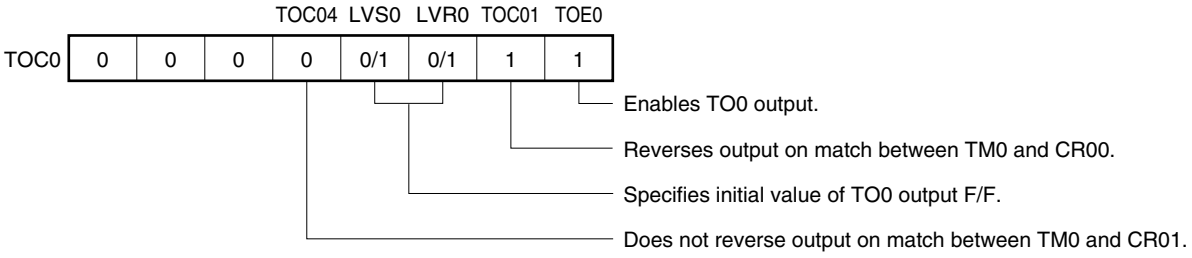
Figure 6-24. Control Register Settings in Square-Wave Output Mode

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

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

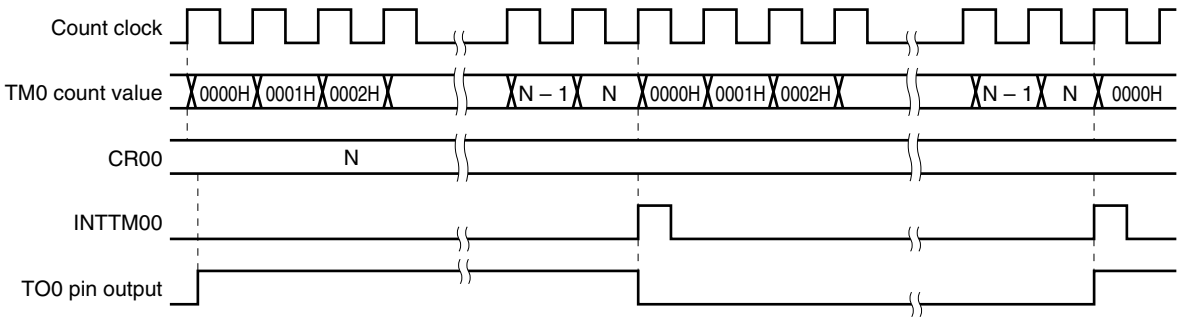


(c) 16-bit timer output control register 0 (TOC0)



Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with square-wave output. See **Figures 6-2**, **6-3**, and **6-4**.

Figure 6-25. Square-Wave Output Operation Timing

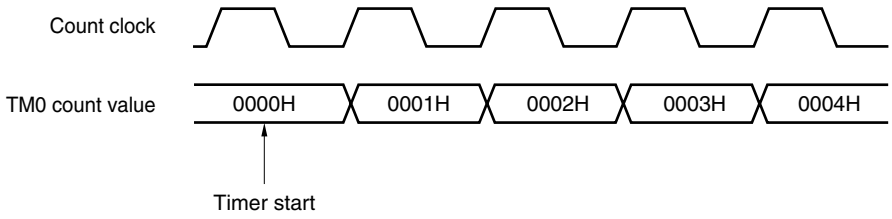


6.6 16-Bit Timer/Event Counter 0 Cautions

(1) Timer start errors

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

Figure 6-26. 16-Bit Timer Counter 0 (TM0) Start Timing



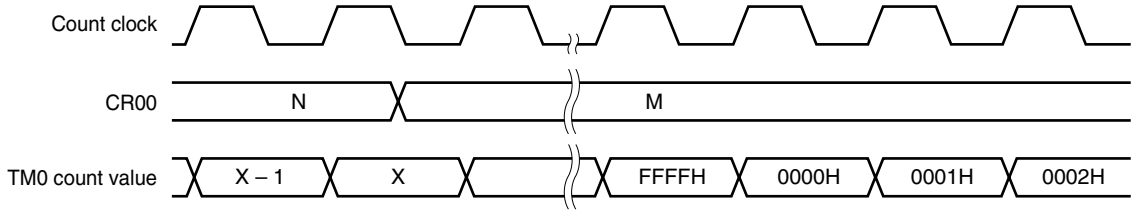
(2) 16-bit timer compare register setting (in the clear & start mode on match between TM0 and CR00)

Set other than 0000H to 16-bit timer capture/compare registers 00, 01 (CR00, CR01). This means 1-pulse count operation cannot be performed when it is used as the event counter.

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

If the value after 16-bit timer capture/compare register 00 (CR00) is changed is smaller than that of 16-bit timer counter 0 (TM0), TM0 continues counting, overflows and then restarts counting from 0. Thus, if the value (M) after the CR00 change is smaller than that (N) before the change, it is necessary to reset and restart the timer after changing CR00.

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

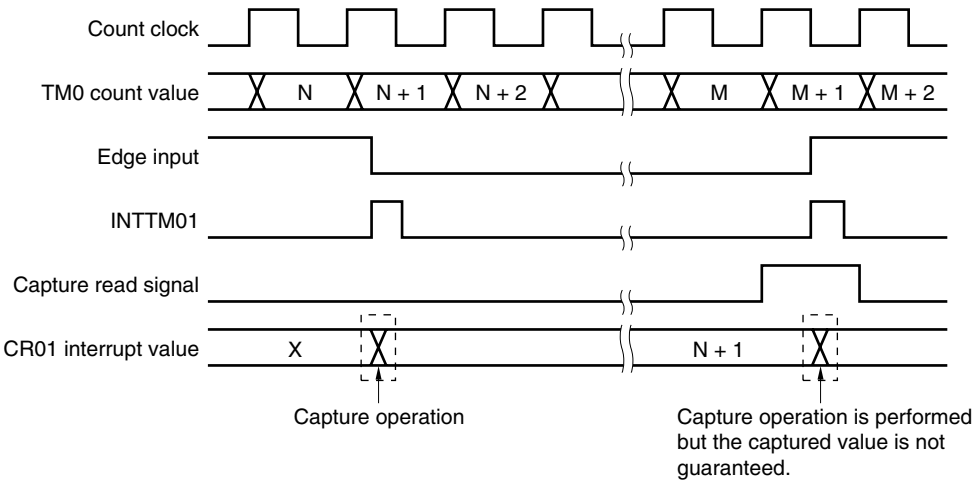


Remark N > X > M

(4) Capture register data retention timings

If the valid edge of the TI00/P31 pin is input during 16-bit timer capture/compare register 01 (CR01) read, CR01 carries out capture operation but the captured value at this time is not guaranteed. However, the interrupt request signal (TMIF01) is set upon detection of the valid edge.

Figure 6-28. Capture Register Data Retention Timing



(5) Valid edge setting

Set the valid edge of the TI00/P31 pin after setting bits 2 and 3 (TMC02 and TMC03) of 16-bit timer mode control register 0 (TMC0) to 0, 0, respectively, and then stopping timer operation. The valid edge is set with bits 4 and 5 (ES00 and ES01) of prescaler mode register 0 (PRM0).

(6) Operation of OVF0 flag

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

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

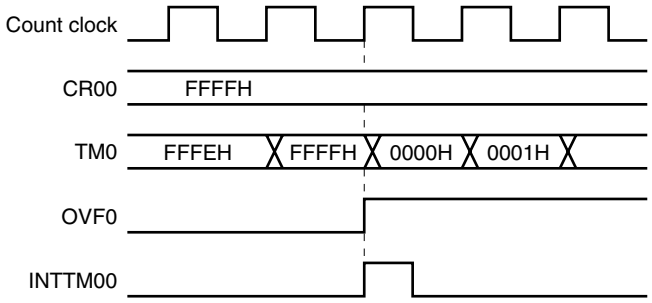
↓

CR00 is set to FFFFH.

↓

When TM0 is counted up from FFFFH to 0000H.

Figure 6-29. Operation Timing of OVF0 Flag



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

(7) Contending operations

<1> The contending operations between the read time of the 16-bit timer capture/compare register (CR00/CR01) and capture trigger input (CR00/CR01 used as capture register)
Capture trigger input is prior to the other. The data read from CR00/CR01 is not defined.

<2> The match timing of contending operations between the write period of the 16-bit timer capture/compare register (CR00/CR01) and 16-bit timer counter 0 (TM0) (CR00/CR01 used as a compare register)
The match discriminant is not performed normally. Do not write any data to CR00/CR01 near the match timing.

(8) Timer operation

<1> Even if 16-bit timer counter 0 (TM0) is read, the value is not captured by 16-bit timer capture/compare register 01 (CR01).
<2> Regardless of the CPU's operation mode, when the timer stops, the signals input to pins TI00/TI01 are not acknowledged.

(9) Capture operation

<1> If TI00 is specified as the valid edge of the count clock, capture operation by the capture register specified as the trigger for TI00 is not possible.
<2> If both the rising and falling edges are selected as the valid edges of TI00, capture is not performed.
<3> To ensure the reliability of the capture operation, the capture trigger requires a pulse two times longer than the count clock selected by prescaler mode register 0 (PRM0).
<4> The capture operation is performed at the fall of the count clock. An interrupt request input (INTTM0n), however, is generated at the rise of the next count clock.

(10) Compare operation

<1> The INTTM0 may not be generated if the set value of 16-bit timer capture registers 00, 01 (CR00, CR01) and the count value of 16-bit timer counter 0 (TM0) match and CR00 and CR01 are overwritten at the timing of INTTM0 generation. Therefore, do not overwrite CR00 and CR01 frequently even if overwriting the same value.
<2> Capture operation may not be performed for CR00/CR01 set in compare mode even if a capture trigger has been input.

(11) Edge detection

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

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

CHAPTER 7 16-BIT TIMER/EVENT COUNTER 4

7.1 Outline of 16-Bit Timer/Event Counter 4

16-bit timer/event counter 4 can be used to serve as an interval timer, square wave output with any selected frequency, and an external event counter.

7.2 16-Bit Timer/Event Counter 4 Functions

16-bit timer/event counter 4 has the following functions.

- Interval timer
- Square wave output
- External event counter

(1) Interval timer

Generates an interrupt request at the preset time interval.

(2) Square wave output

Can output a square wave with any selected frequency.

(3) External event counter

Can measure the number of pulses (TI4) of an externally input signal.

7.3 16-Bit Timer/Event Counter 4 Configuration

16-bit timer/event counter 4 consists of the following hardware.

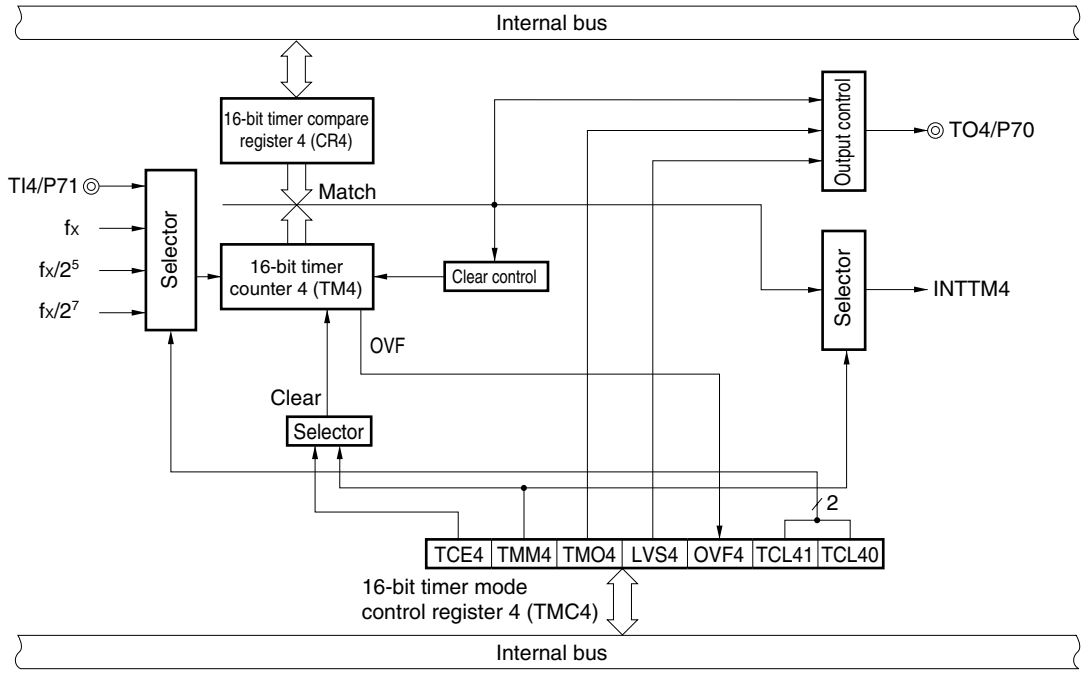
Table 7-1. 16-Bit Timer/Event Counter 4 Configuration

Item	Configuration
Timer/counter	16 bits × 1 (TM4)
Register	16-bit timer compare register 4: 16 bits × 1 (CR4)
Timer output	1 (TO4)
Control registers	16-bit timer mode control register 4 (TMC4) Port mode register 7 (PM7)[Note]

Note Refer to **Figure 4-15 P70, P72 Block Diagram** and **Figure 4-16 P71, P73 Block Diagram**.

Figure 7-1 shows a block diagram.

★ **Figure 7-1. 16-Bit Timer/Event Counter 4 Block Diagram**



(1) 16-bit timer counter 4 (TM4)

TM4 is a 16-bit register that counts count pulses.

The counter is incremented in synchronization with the rising edge of an input clock. TM4 cannot be read or written to.

The value of this register is undefined when $\overline{\text{RESET}}$ is input.

The count value is reset to 0000H in the following cases.

<1> If TCE4 is cleared

<2> If TM4 and CR4 match with each other in clear & start mode on match between TM4 and CR4

<3> Immediately after TM4 overflowed in free-running mode

(2) 16-bit timer compare register 4 (CR4)

This register always compares the value set in CR4 and the count value of 16-bit timer counter 4 (TM4). If the two values match, CR4 generates an interrupt request (INTTM4). If TM4 is specified as an interval timer, this register can be used to hold the interval time.

CR4 is set by a 16-bit memory manipulation instruction.

The value of this register is undefined when $\overline{\text{RESET}}$ is input.

Caution Do not write values to CR4 during TM4 count operation. Stop the count operation first if overwriting the same value.

7.4 Registers to Control 16-Bit Timer/Event Counter 4

The following two registers are used to control 16-bit timer/event counter 4.

- 16-bit timer mode control register 4 (TMC4)
- Port mode register 7 (PM7)

(1) 16-bit timer mode control register 4 (TMC4)

This register controls the 16-bit timer counter 4 (TM4) count operation and timer output (TO4), selects the operation mode, specifies the TO4 initial value, and sets the TM4 count clock and valid edge of TI4 input.

TMC4 is set by a 1-bit or 8-bit memory manipulation instruction.

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

Figure 7-2. 16-Bit Timer Mode Control Register 4 (TMC4) Format

Address: FF68H After reset: 00H R/W[Note 1]

Symbol	7	6	5	4	3	2	1	0
TMC4	TCE4	TMM4	TMO4	LVS4	OVF4	0	TCL41	TCL40

TCE4	TM4 count operation control
0	Count operation stop (TM4 cleared to 0)
1	Count operation start

TMM4	TM4 operation mode selection	INTTM4 generation timing
0	Clear & start on match between TM4 and CR4[Note 2]	Match between TM4 and CR4
1	Free-running mode	INTTM4 not generated

TMO4	Timer output (TO4) control
0	Output disabled (output level is fixed at 0)
1	Output enabled

LVS4	Timer output (TO4) initial value setting
0	Low level
1	High level

OVF4	The value of OVF4 reversed each time an overflow occurs (reset value: OVF4 = 0).

TCL41	TCL40	Count clock selection
0	0	f_X (10 MHz)
0	1	$f_X/2^5$ (312.5 kHz)
1	0	$f_X/2^7$ (78.125 kHz)
1	1	Rising edge of TI4

Notes 1. Bit 3 is a read-only bit.

2. Overflow is not detected if clear & start mode is selected by match between TM4 and CR4.

Caution Be sure to stop timer operation (TCE4 = 0) before setting TMC4.

Remarks 1. The initial value of TO4 is the timer output value of TO4 when timer output is enabled (TMO4 = 1) and the count operation is stopped (TCE4 = 0).

2. f_X: Main system clock oscillation frequency

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

(2) Port mode register 7 (PM7)

This register sets port 7 input/output in 1-bit units.

When using the TO4/P70 pin for timer output, set PM70 and the output latch of P70 to 0.

PM7 is set by a 1-bit or 8-bit memory manipulation instruction.

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

Figure 7-3. Port Mode Register 7 (PM7) Format

Address: FF27H After reset: FFH R/W

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

PM7n	PM7n pin I/O mode selection (n = 0 to 3)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

7.5 16-Bit Timer/Event Counter 4 Operations

7.5.1 Interval timer operation

16-bit timer/event counter 4 operates as an interval timer which generates interrupt requests repeatedly at intervals of the count value preset to 16-bit timer compare register 4 (CR4).

When the count value of 16-bit timer counter 4 (TM4) matches the value set to CR4, counting continues with the TM4 value cleared to 0 and an interrupt request signal (INTTM4) is generated.

The count clock of TM4 can be selected with bits 0 and 1 (TCL40 and TCL41) of 16-bit timer mode control register 4 (TMC4).

[Setting]

<1> Set the registers.
- TCL41, TCL40: Set count clock.
- CR4: Compare value
- TMM4: Clear & start mode by match of TM4 and CR4 (TMM4 = 0)
- TMO4: Timer output is set to disable (TMO4 = 0).

<2> After TCE4 = 1 is set, count operation starts.

<3> If the values of TM4 and CR4 match, INTTM4 is generated (TM4 is cleared to 0000H).

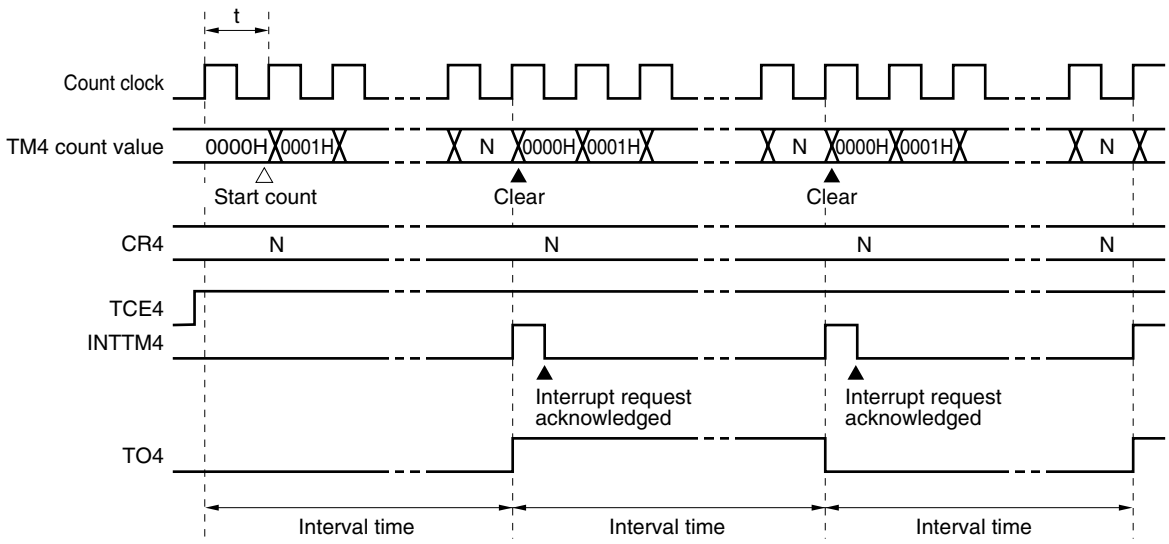
<4> INTTM4 generates repeatedly at the same interval. Set TCE4 to 0 to stop count operation.

Cautions 1. INTTM4 is fixed to high level after count start when CR4 = 0000H is set. Therefore, only the first rising edge is valid for INTTM4.

2. The rising edge of the first clock immediately after setting TCE4 to 1 is not counted. Count operation is started from the rising edge of the second clock.

Figure 7-4. Interval Timer Operation Timings (1/2)

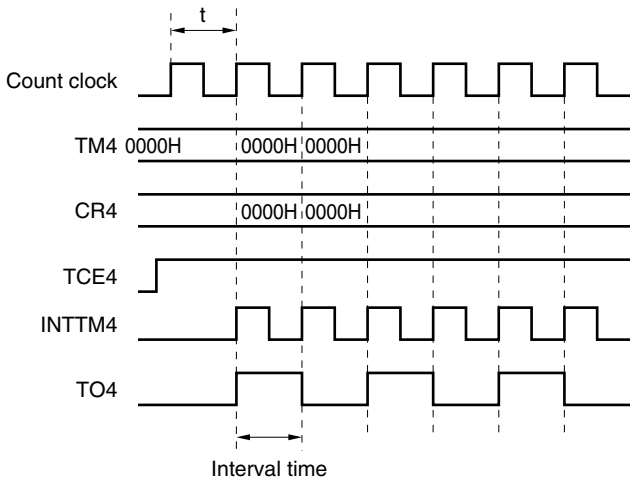
(a) Basic operation



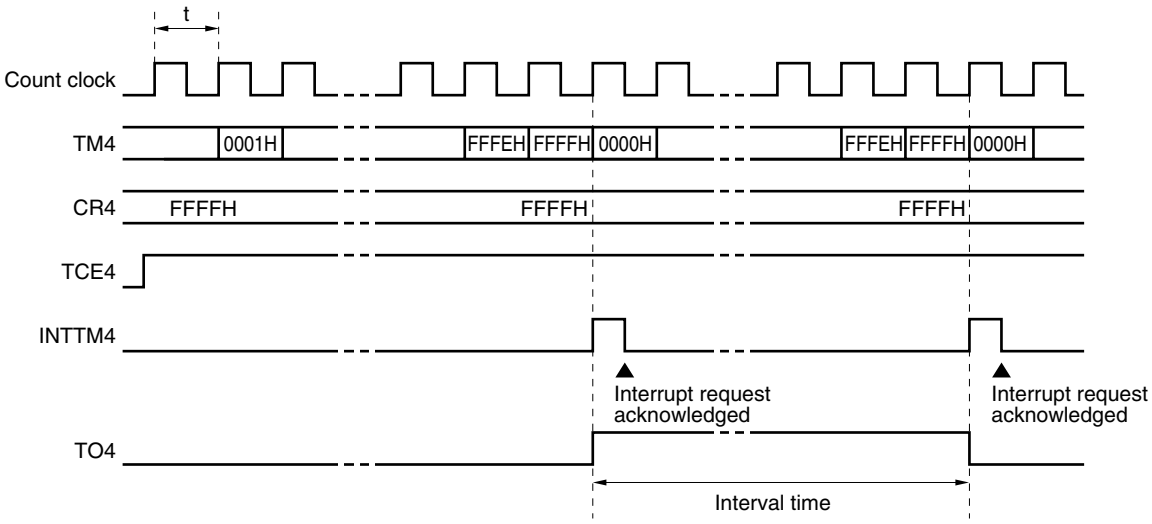
Remark Interval time = (N + 1) × t
N = 0000H to FFFFH

Figure 7-4. Interval Timer Operation Timings (2/2)

(b) When CR4 = 0000H



(c) When CR4 = FFFFH



7.5.2 Square-wave output operation

A square wave with any selected frequency is output at intervals of the value preset to 16-bit timer compare register 4 (CR4).

TO4 pin output status is reversed at intervals of the count value preset to CR4 by setting bit 7 (TCE4) of 16-bit timer mode control register 4 (TMC4) to 1. This enables a square wave with any selected frequency to be output (duty = 50%).

[Setting]

<1> Set each register.

- Set port latch and port mode register to 0.
- TCL41, 40: Select count clock
- CR4: Compare value
- TMM4: Clear & start mode by match of TM4 and CR4 (TMM4 = 0)
- LVS4: Set initial status of timer output (TO4)
  - Initial output = 1 ← LVS4 = 1
  - Initial output = 0 ← LVS4 = 0
- TMO4: Timer output is set to enable (TMO4 = 1)

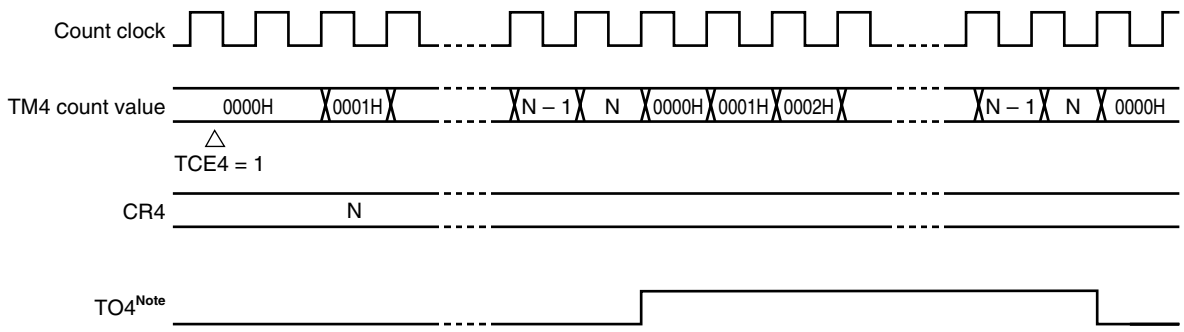
<2> After TCE4 = 1 is set, count operation starts.

<3> Timer output F/F is reversed by match of TM4 and CR4. After INTTM4 is generated, TM4 is cleared to 00H.

<4> Timer output F/F is reversed at the same interval and square wave is output from TO4.

Caution The rising edge of the first clock immediately after setting TCE4 to 1 is not counted. Count operation is started from the rising edge of the second clock.

Figure 7-5. Square-Wave Output Operation Timing



Note TO4 output initial value can be set by bit 4 (LVS4) of 16-bit timer mode control register 4 (TMC4).

7.5.3 External event counter operation

The external event counter counts the number of external clock pulses to be input to TI4 by 16-bit timer counter 4 (TM4).

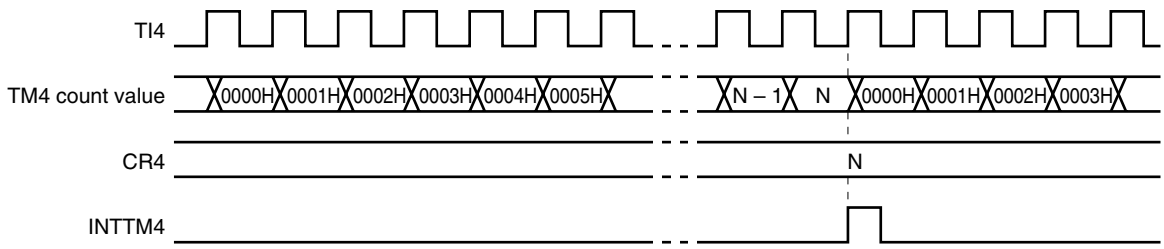
TM4 is incremented each time the rising edge of TI4 specified with 16-bit timer mode control register 4 (TMC4) is input.

When the TM4 counted values match the values of 16-bit timer compare register 4 (CR4), TM4 is cleared to 0 and the interrupt request signal (INTTM4) is generated.

Whenever the TM4 counted value matches the value of CR4, INTTM4 is generated.

Caution The rising edge of the first clock immediately after setting TCE4 to 1 is not counted. Count operation is started from the rising edge of the second clock.

Figure 7-6. External Event Counter Operation Timing

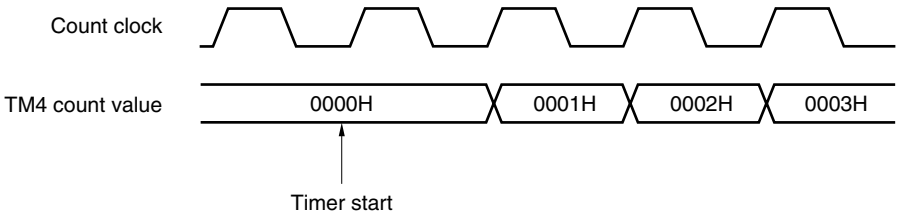


7.6 16-Bit Timer/Event Counter 4 Cautions

(1) Timer start errors

An error of up to one clock may occur in the time required for a match signal to be generated after timer start. This is because 16-bit timer counter 4 (TM4) is started asynchronously to the count clock.

Figure 7-7. 16-Bit Timer Counter 4 (TM4) Start Timing



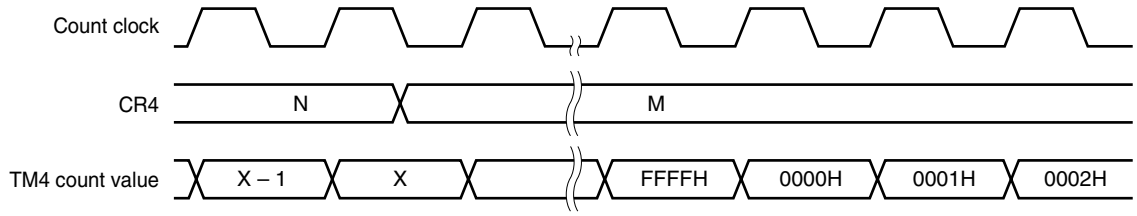
(2) 16-bit timer compare register setting

Set other than 0000H to 16-bit timer compare register 4 (CR4). This means a 1-pulse count operation cannot be performed when it is used as the event counter.

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

If the value after 16-bit timer compare register 4 (CR4) is changed is smaller than that of 16-bit timer counter 4 (TM4), TM4 continues counting, overflows and then restarts counting from 0. Thus, if the value (M) after the CR4 change is smaller than that (N) before the change, it is necessary to reset and restart the timer after changing CR4.

Figure 7-8. Timings After Change of Compare Register During Timer Count Operation



Remark N > X > M

(4) Contending operations

If the match timing between the write period of 16-bit timer compare register 4 (CR4) and 16-bit timer counter 4 (TM4) conflicts, the match discriminant is not performed normally. Do not write any data to CR4 near the match timing.

(5) Timer operation

Regardless of the CPU's operation mode, when the timer stops, the input signal to pin TI4 is not acknowledged.

CHAPTER 8 8-BIT TIMER/EVENT COUNTERS 50, 51, 52

8.1 Outline of 8-Bit Timer/Event Counters 50, 51, and 52

8-bit timer/event counters 50, 51, and 52 can be used to serve as an interval timer, an external event counter, to output square wave output with any selected frequency, and PWM output.

8.2 8-Bit Timer/Event Counters 50, 51, and 52 Functions

8-bit timer/event counters 50, 51, and 52 have the following functions.

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

(1) Interval timer

These counters generate interrupt requests at the preset time interval.

(2) External event counter

These counters can measure the number of pulses of an externally input signal.

(3) Square wave output

These counters can output a square wave with any selected frequency.

(4) PWM output

These counters can output PWM.

Figures 8-1 to 8-3 show 8-bit timer/event counter block diagrams.

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

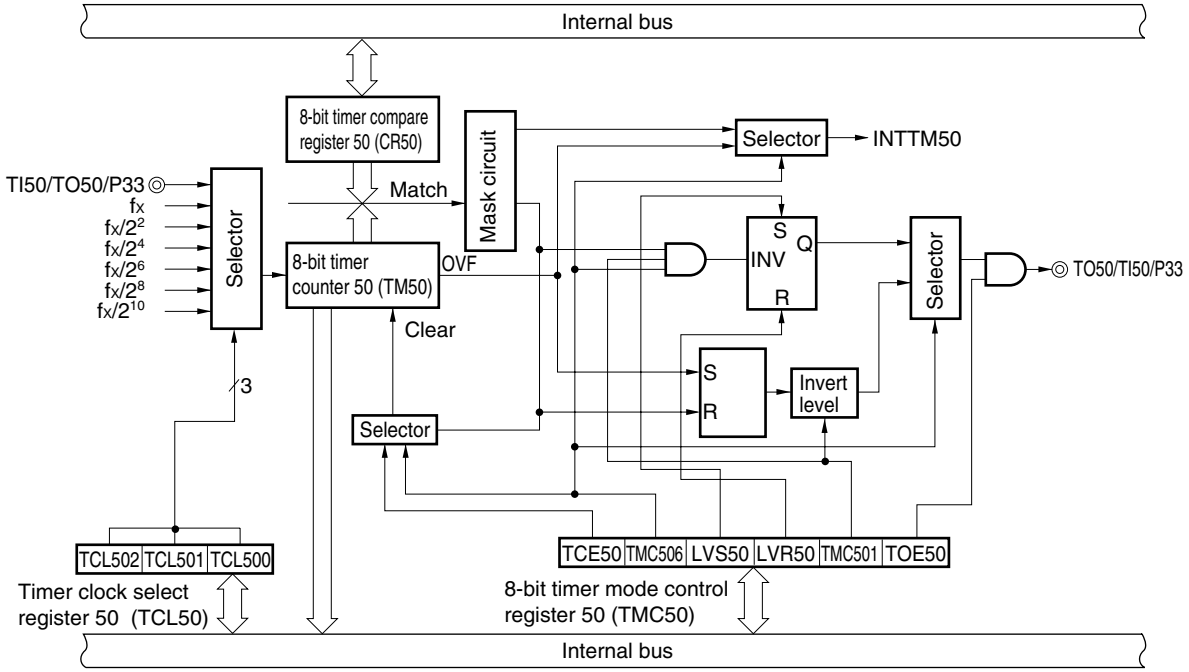


Figure 8-2. 8-Bit Timer/Event Counter 51 Block Diagram

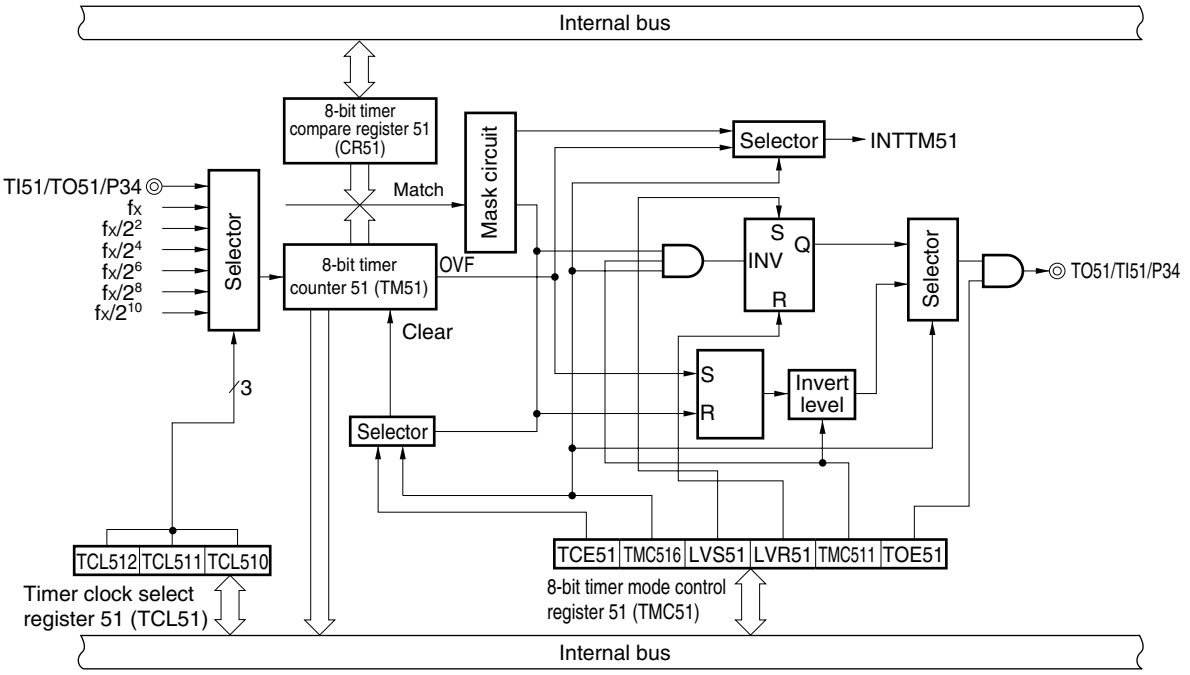
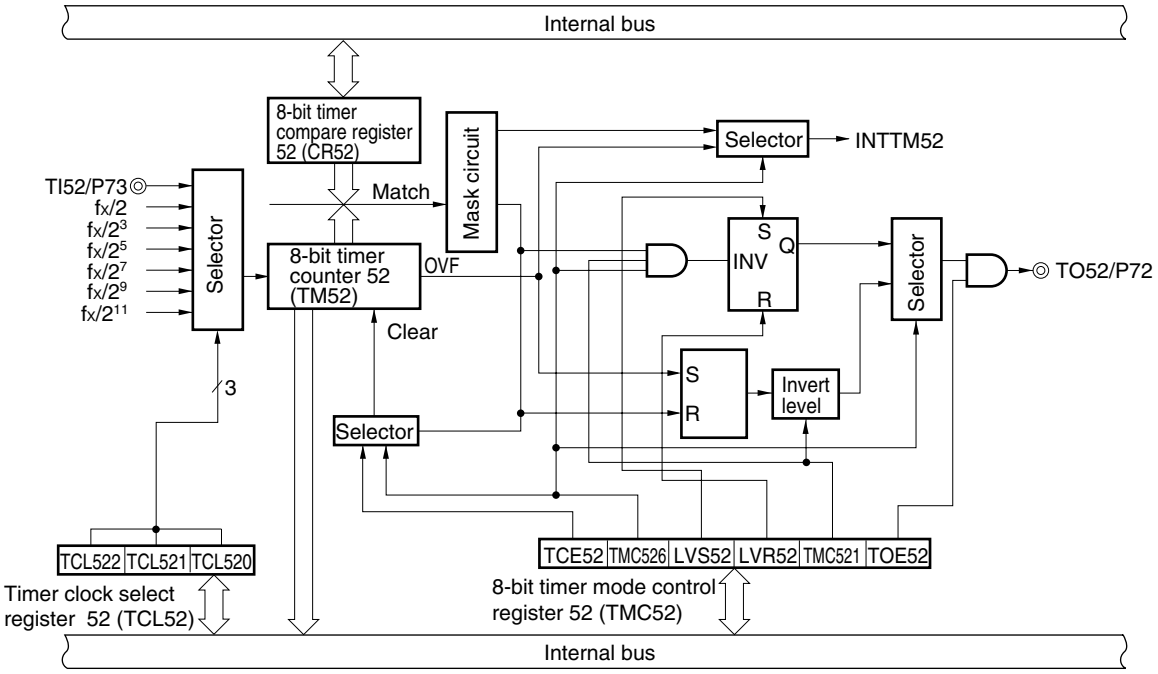


Figure 8-3. 8-Bit Timer/Event Counter 52 Block Diagram



8.3 8-Bit Timer/Event Counters 50, 51, and 52 Configurations

8-bit timer/event counters 50, 51, and 52 consist of the following hardware.

Table 8-1. 8-Bit Timer/Event Counters 50, 51, and 52 Configuration

Item	Configuration
Timer register	8-bit timer counter 5n (TM5n)
Register	8-bit timer compare register 5n (CR5n)
Timer output	3 (TO5n)
Control registers	Timer clock select register 5n (TCL5n) 8-bit timer mode control register 5n (TMC5n) Port mode registers 3, 7 (PM3, PM7)**Note**

Note See **Figure 4-9 P33, P34 Block Diagram**, **Figure 4-15 P70, P72 Block Diagram**, and **Figure 4-16 P71, P73 Block Diagram.**

Remark n = 0 to 2

(1) 8-bit timer counter 5n (TM5n: n = 0 to 2)

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

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

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

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

<2> When TCE5n is cleared

<3> When TM5n and CR5n match in clear & start mode if this mode was entered upon match of TM5n and CR5n values.

Remark n = 0 to 2

(2) 8-bit timer compare register 5n (CR5n: n = 0 to 2)

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

It is possible to rewrite the value of CR5n within 00H to FFH during count operation.

Remark n = 0 to 2

8.4 Registers to Control 8-Bit Timer/Event Counters 50, 51, and 52

The following three types of registers are used to control 8-bit timer/event counters 50, 51, and 52.

- Timer clock select register 5n (TCL5n)
- 8-bit timer mode control register 5n (TMC5n)
- Port mode registers 3, 7 (PM3, PM7)

Remark n = 0 to 2

(1) Timer clock select register 5n (TCL5n: n = 0 to 2)

This register sets count clocks of 8-bit timer/event counter 5n and the valid edge of TI5n input.

TCL5n is set by an 8-bit memory manipulation instruction.

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

Figure 8-4. Timer Clock Select Register 50 (TCL50) Format

Address: FF71H After reset: 00H R/W

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

TCL502	TCL501	TCL500	Count clock selection
0	0	0	TI50 falling edge
0	0	1	TI50 rising edge
0	1	0	f_X (10 MHz)
0	1	1	$f_X/2^2$ (2.5 MHz)
1	0	0	$f_X/2^4$ (625 kHz)
1	0	1	$f_X/2^6$ (156.2 kHz)
1	1	0	$f_X/2^8$ (39.1 kHz)
1	1	1	$f_X/2^{10}$ (9.77 kHz)

Cautions 1. When rewriting TCL50 to other data, stop the timer operation beforehand.

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

Remarks 1. f_X: Main system clock oscillation frequency

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

Figure 8-5. Timer Clock Select Register 51 (TCL51) Format

Address: FF74H After reset: 00H R/W

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

TCL512	TCL511	TCL510	Count clock selection
0	0	0	TI51 falling edge
0	0	1	TI51 rising edge
0	1	0	f_X (10 MHz)
0	1	1	$f_X/2^2$ (2.5 MHz)
1	0	0	$f_X/2^4$ (625 kHz)
1	0	1	$f_X/2^6$ (156.2 kHz)
1	1	0	$f_X/2^8$ (39.1 kHz)
1	1	1	$f_X/2^{10}$ (9.77 kHz)

Cautions 1. When rewriting TCL51 to other data, stop the timer operation beforehand.

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

Remarks 1. f_X: Main system clock oscillation frequency

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

Figure 8-6. Timer Clock Select Register 52 (TCL52) Format

Address: FF77H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TCL52	0	0	0	0	0	TCL522	TCL521	TCL520

TCL522	TCL521	TCL520	Count clock selection
0	0	0	TI52 falling edge
0	0	1	TI52 rising edge
0	1	0	$f_X/2$ (5 MHz)
0	1	1	$f_X/2^3$ (1.25 MHz)
1	0	0	$f_X/2^5$ (312.5 kHz)
1	0	1	$f_X/2^7$ (78.1 kHz)
1	1	0	$f_X/2^9$ (19.5 kHz)
1	1	1	$f_X/2^{11}$ (4.88 kHz)

Cautions 1. When rewriting TCL52 to other data, stop the timer operation beforehand.

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

Remarks 1. fx: Main system clock oscillation frequency

2. Figures in parentheses are for operation with fx = 10 MHz

(2) 8-bit timer mode control register 5n (TMC5n: n = 0 to 2)

TMC5n is a register which sets the following five types.

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

TMC5n is set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets the value of this register to 00H.
Figure 8-7 shows the TMC5n format.

Figure 8-7. 8-Bit Timer Mode Control Register 5n (TMC5n) Format

Address: FF70H (TMC50) FF73H (TMC51) FF76H (TMC52) 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	After clearing to 0, count operation disabled (prescaler disabled)
1	Count operation start

TMC5n6	TM5n operation mode selection
0	Clear and start mode by matching between TM5n and CR5n
1	PWM (free-running) mode

LVS5n	LVR5n	Timer output F/F status setting
0	0	No change
0	1	Timer output F/F reset (0)
1	0	Timer output F/F set (1)
1	1	Setting prohibited

TMC5n1	In other modes (TMC5n6 = 0)	In PWM mode (TMC5n6 = 1)
	Timer F/F control	Active level selection
0	Inversion operation disabled	Active high
1	Inversion operation enabled	Active low

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

Remarks 1. In PWM mode, PWM output will be inactive because of TCE5n = 0.

2. If LVS5n and LVR5n are read after data is set, 0 is read.

3. n = 0 to 2

(3) Port mode registers 3 and 7 (PM3, PM7)

These registers set ports 3 and 7 input/output in 1-bit units.

When using the P33/TO50/TI50, P34/TO51/TI51, and P72/TO52 pins for timer output, set PM33, PM34, PM72 and the output latches of P33, P34, and P72 to 0.

PM3 and PM7 are set by a 1-bit or 8-bit memory manipulation instruction.

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

Figure 8-8. Port Mode Registers 3, 7 (PM3, PM7) Format

Address: FF23H After reset: FFH R/W

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

Address: FF27H After reset: FFH R/W

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

PMmn	Pmn pin I/O mode selection (m = 3: n = 0 to 4, m = 7: n = 0 to 3)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

8.5 8-Bit Timer/Event Counters 50, 51, and 52 Operations

8.5.1 Interval timer operation

The 8-bit timer/event counters operate as interval timers which generate interrupt requests repeatedly at intervals of the count value preset to 8-bit timer compare register 5n (CR5n).

When the count value of 8-bit timer counter 5n (TM5n) matches the value set to CR5n, counting continues with the TM5n value cleared to 0 and an interrupt request signal (INTTM5n) is generated.

The count clock of TM5n can be selected with bits 0 to 2 (TCL5n0 to TCL5n2) of timer clock select register 5n (TCL5n).

See **8.6 8-Bit Timer/Event Counters 50, 51, and 52 Cautions (2)** about the operation when the compare register value is changed during timer count operation.

[Setting]

<1> Set the registers.

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

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

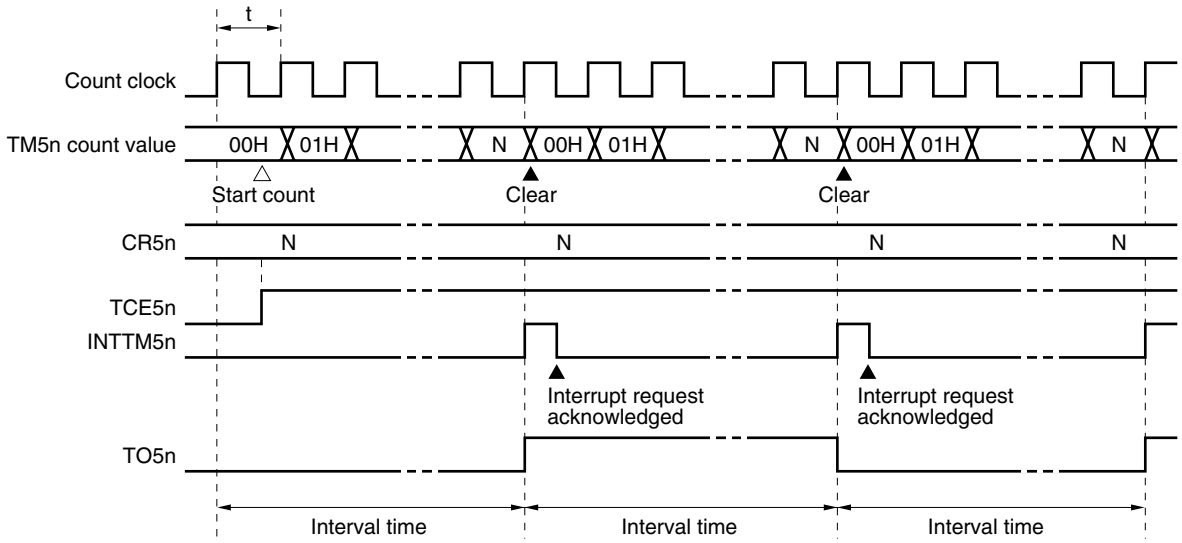
<3> If the values of TM5n and CR5n match, INTTM5n is generated (TM5n is cleared to 00H).

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

Remark n = 0 to 2

Figure 8-9. Interval Timer Operation Timings (1/3)

(a) Basic operation

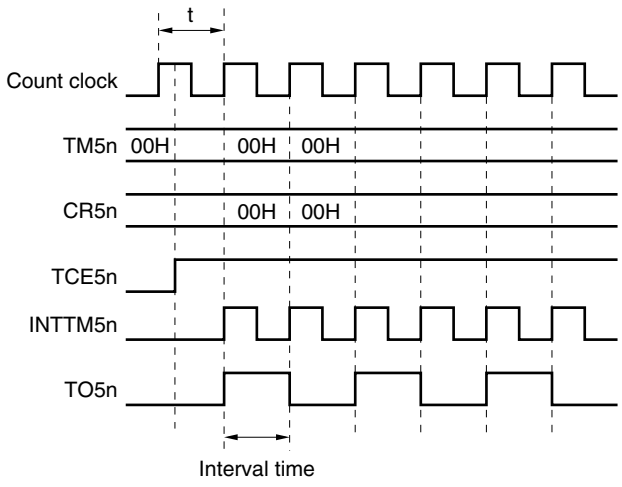


Remarks 1. Interval time = (N + 1) × t
N = 00H to FFH

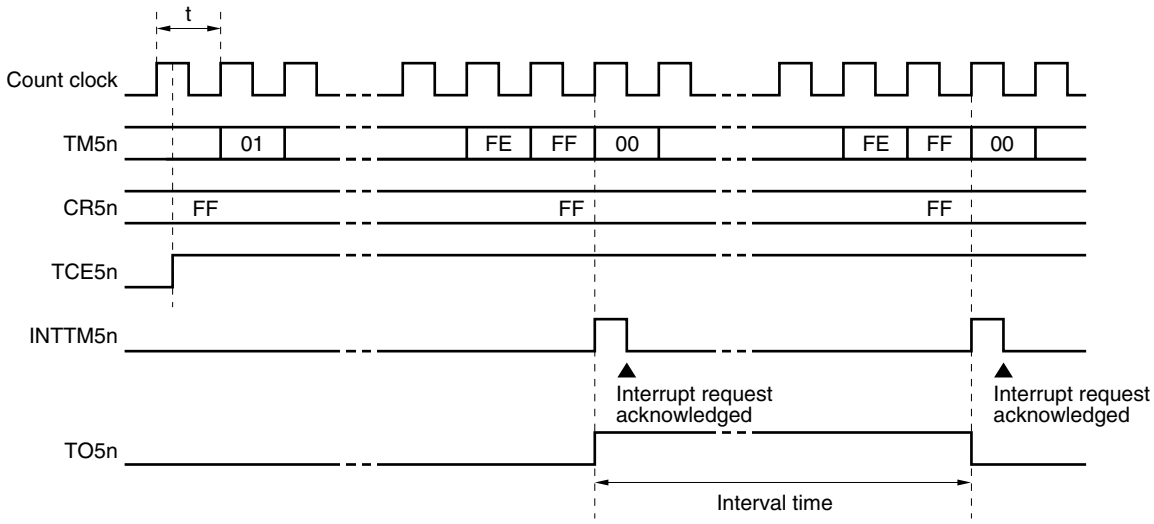
2. n = 0 to 2

Figure 8-9. Interval Timer Operation Timings (2/3)

(b) When CR5n = 00H



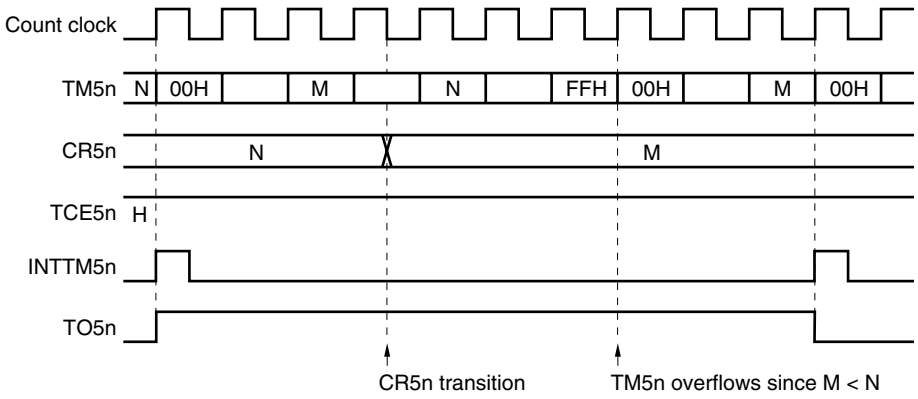
(c) When CR5n = FFH



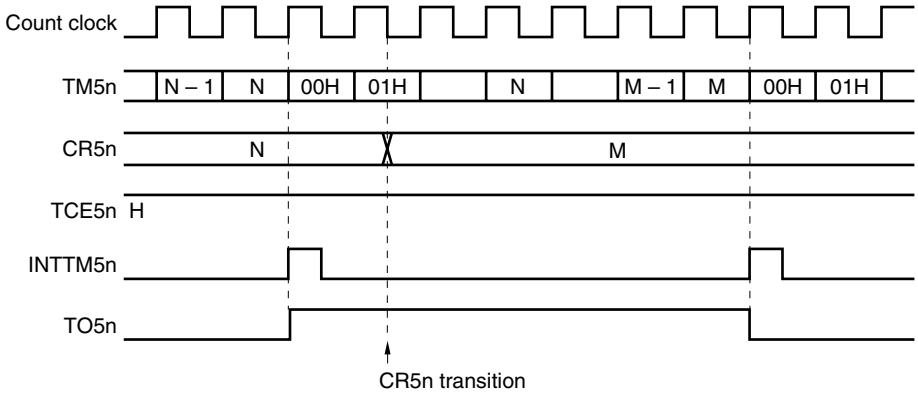
Remark n = 0 to 2

Figure 8-9. Interval Timer Operation Timings (3/3)

(d) Operated by CR5n transition (M < N)



(e) Operated by CR5n transition (M > N)



Remark n = 0 to 2

8.5.2 External event counter operation

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

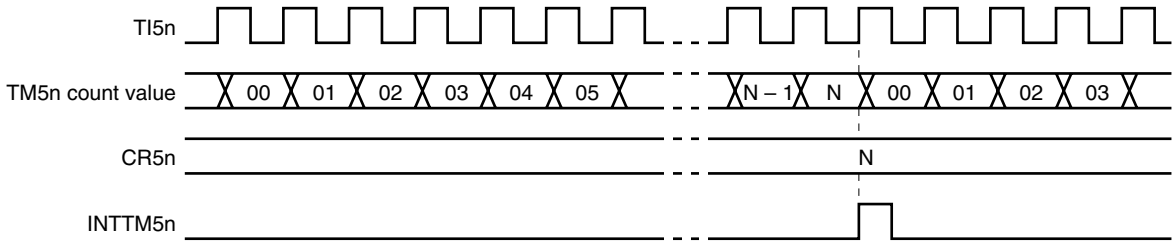
TM5n is incremented each time the valid edge specified with timer clock select register 5n (TCL5n) is input. Either the rising or falling edge can be selected.

When the TM5n count value matches the value of 8-bit timer compare register 5n (CR5n), TM5n is cleared to 0 and an interrupt request signal (INTTM5n) is generated.

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

Remark n = 0 to 2

Figure 8-10. External Event Counter Operation Timing (with Rising Edge Specified)



Remark n = 0 to 2

8.5.3 Square-wave output operation

A square wave with any selected frequency is output at intervals of the value preset to 8-bit timer compare register 5n (CR5n).

The TO5n pin output status is reversed at intervals of the count value preset to CR5n by setting bit 0 (TOE5n) of 8-bit timer mode control register 5n (TMC5n) to 1. This enables a square wave with any selected frequency to be output (duty = 50%).

[Setting]

<1> Set each register.

- Set port 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

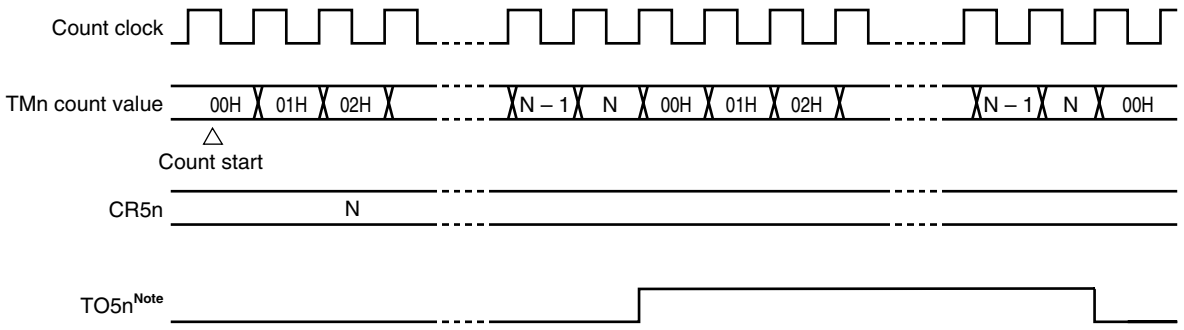
<2> After TCE5n = 1 is set, count operation starts.

<3> Timer output F/F is reversed by a 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 a square wave is output from TO5n.

Remark n = 0 to 2

Figure 8-11. Square-Wave Output Operation Timing



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

Remark n = 0 to 2

8.5.4 PWM output operation

The 8-bit timer/event counter operates as PWM output when bit 6 (TMC5n6) of 8-bit timer mode control register 5n (TMC5n) is set to 1.

The duty rate pulse determined by the value set to 8-bit timer compare register 5n (CR5n) is output from TO5n.

Set the active level width of the PWM pulse to CR5n, and the active level can be selected with bit 1 (TMC5n1) of TMC5n.

The count clock can be selected with bits 0 to 2 (TCL5n0 to TCL5n2) of timer clock select register 5n (TCL5n).

Enable/disable for PWM output can be selected with bit 0 (TOE5n) of TMC5n.

Caution CR5n can be rewritten only once a cycle in PWM mode.

Remark n = 0 to 2

(1) PWM output basic operation

[Setting]

<1> Set the port latches (P33, P34, and P72) and port mode registers 3, 7 (PM33, PM34, and PM72) to 0.
<2> Set the active level width with the 8-bit timer compare register (CR5n).
<3> Select the count clock with timer clock select register 5n (TCL5n).
<4> Set the active level with bit 1 (TMC5n1) of TMC5n.
<5> The count operation starts when bit 7 (TCE5n) of TMC5n is set to 1.
Set TCE5n to 0 to stop the count operation.

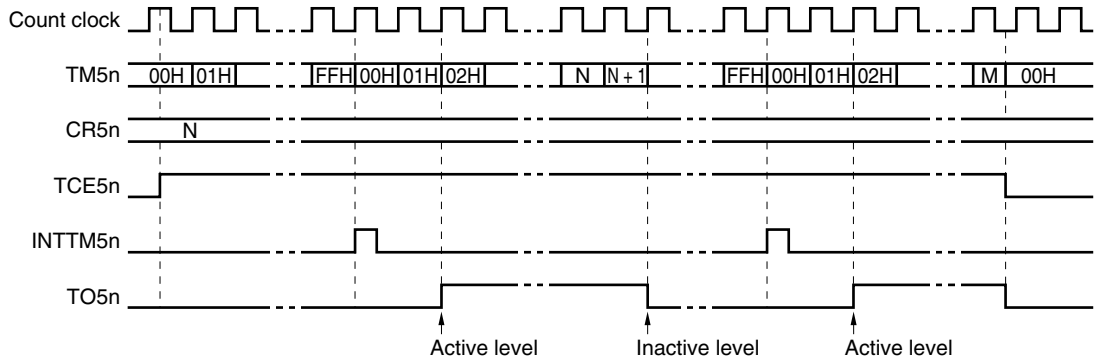
[PWM output operation]

<1> PWM output (output from TO5n) outputs an inactive level after the count operation starts until overflow is generated.
<2> When overflow is generated, the active level set in <1> of [Setting] is output.
The active level is output until CR5n matches the count value of 8-bit timer counter 5n (TM5n).
<3> After CR5n matches the count value, PWM output outputs the inactive level again until 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 changes to the inactive level.

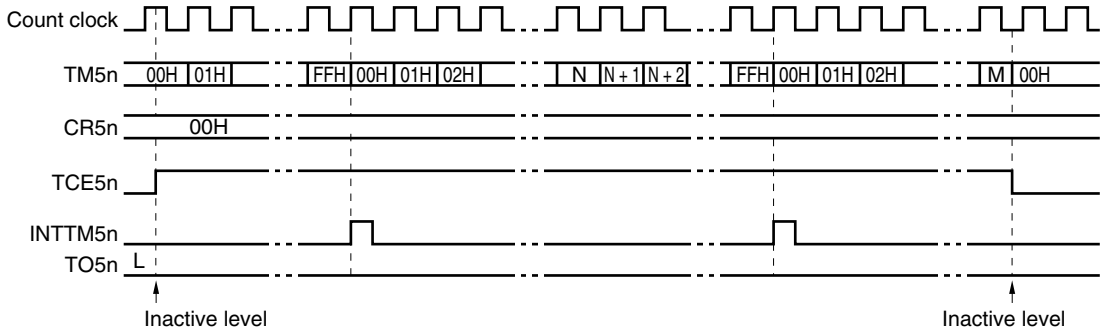
Remark n = 0 to 2

Figure 8-12. PWM Output Operation Timing

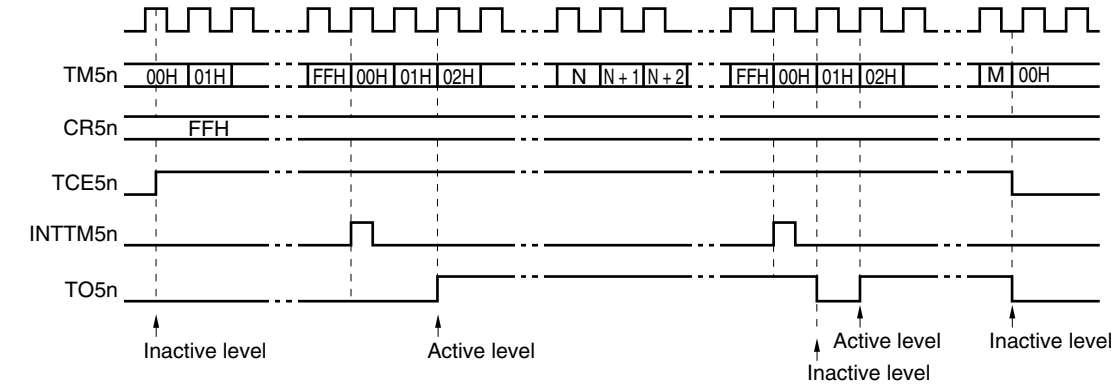
(a) Basic operation (active level = H)



(b) CR5n = 0



(c) CR5n = FFH

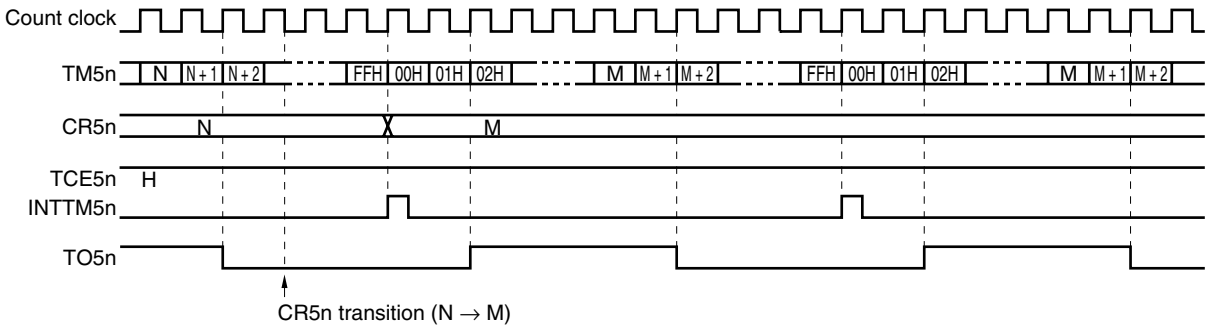


Remark n = 0 to 2

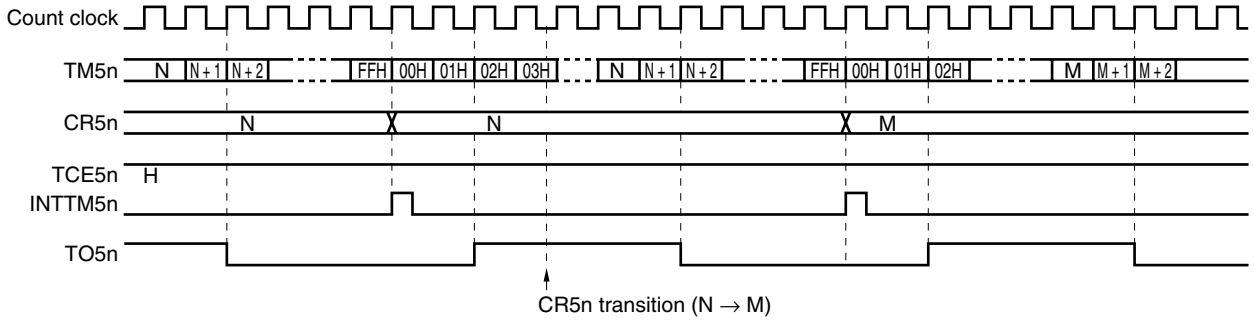
(2) Operated by CR5n transition

Figure 8-13. Timing of Operation by CR5n Transition

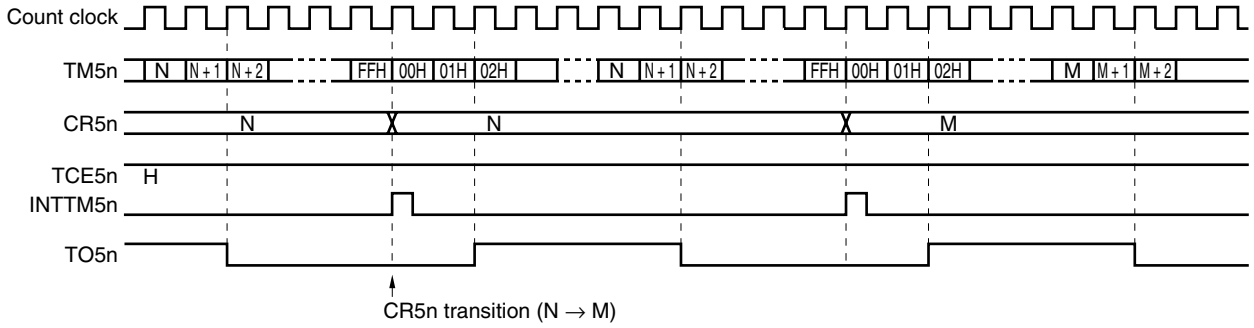
(a) CR5n value transits from N to M before overflow of TM5n



(b) CR5n value transits from N to M after overflow of TM5n



(c) CR5n value transits from N to M between two clocks (00H and 01H) after overflow of TM5n



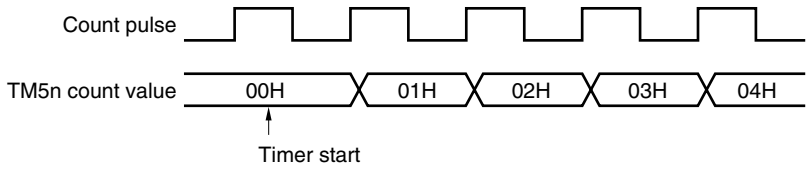
Remark n = 0 to 2

8.6 8-Bit Timer/Event Counters 50, 51, and 52 Cautions

(1) Timer start errors

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

Figure 8-14. 8-Bit Timer Counter Start Timing

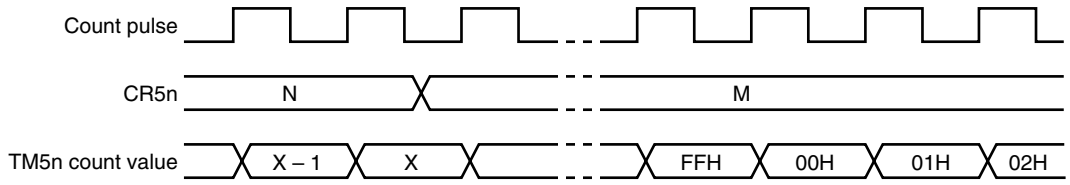


Remark n = 0 to 2

(2) Operation after compare register transition during timer count operation

If the value after 8-bit timer compare register 5n (CR5n) is changed is smaller than the value of 8-bit timer counter 5n (TM5n), TM5n continues counting, overflows and then restarts counting from 0. Thus, if the value (M) after the CR5n change is smaller than value (N) before the change, it is necessary to restart the timer after changing CR5n.

Figure 8-15. Timing After Compare Register Change During Timer Count Operation



Caution Except when the TI5n input is selected, always set TCE5n = 0 before setting the stop state.

Remarks 1. N > X > M

2. n = 0 to 2

(3) TM5n (n = 0 to 2) reading during timer operation

When reading TM5n during operation, select a count clock with a high/low level waveform longer than two cycles of the CPU clock because the count clock stops temporarily. For example, in the case where the CPU clock (f_{CPU}) is f_X, when the selected count clock is $f_X/4$ or below, it can be read.

Remark n = 0 to 2

CHAPTER 9 WATCH TIMER

9.1 Outline of Watch Timer

The watch timer generates interrupt requests (INTWTN0 and INTWTNI0) at the preset time interval.

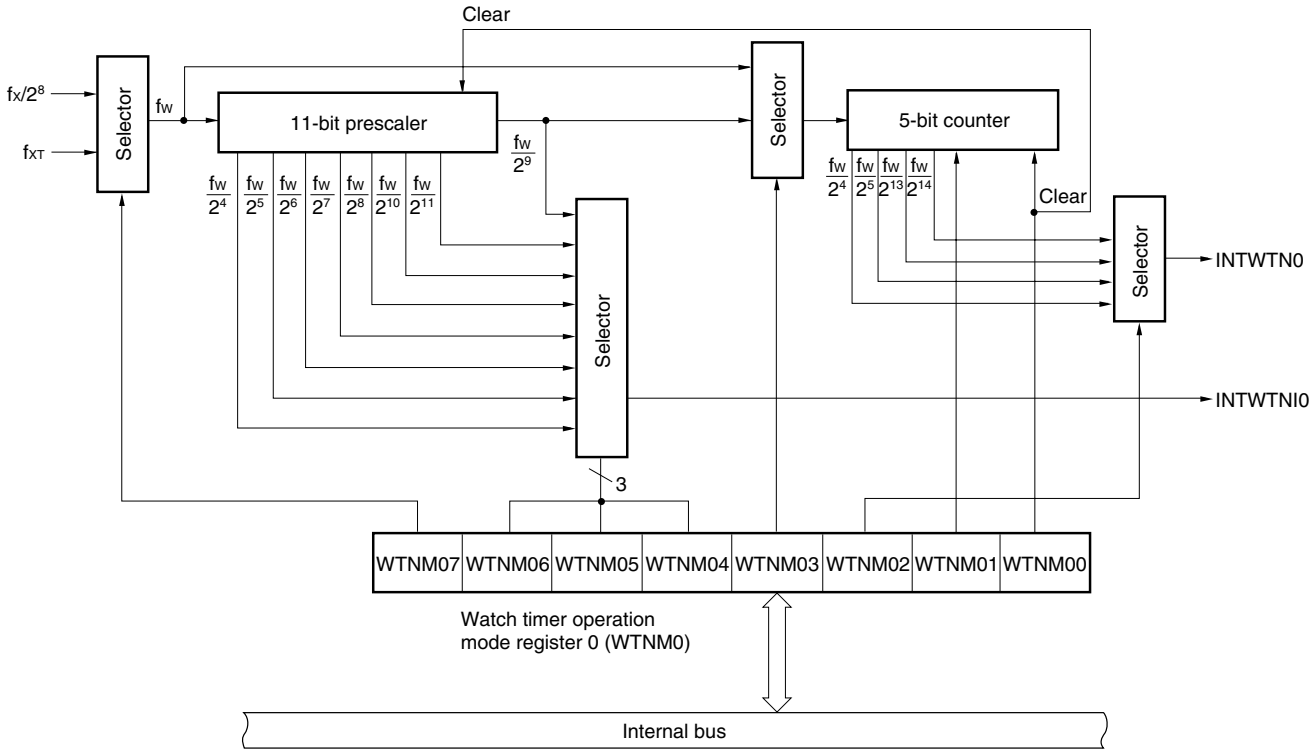
9.2 Watch Timer Functions

The watch timer has the following functions.

- Watch timer
- Interval timer

The watch timer and the interval timer can be used simultaneously.
Figure 9-1 shows the watch timer block diagram.

Figure 9-1. Watch Timer Block Diagram



Remark fx: Main system clock oscillation frequency
fXT: Subsystem clock oscillation frequency
fw: Watch timer clock frequency

(1) Watch timer

By using the main system clock or subsystem clock, interrupt requests (INTWTN0) are generated at preset intervals.

Table 9-1. Watch Timer Interrupt Request Time

Interrupt Request Time	When Operated at f_X = 10 MHz	When Operated at f_{XT} = 32.768 kHz
$2^4/f_W$	409.6 μs	488 μs
$2^5/f_W$	819.2 μs	977 μs
$2^{13}/f_W$	0.2 s	0.25 s
$2^{14}/f_W$	0.41 s	0.5 s

Remark f_W: Watch timer clock frequency ($f_X/2^8$ or f_{XT})
f_X: Main system clock oscillation frequency
f_{XT}: Subsystem clock oscillation frequency

(2) Interval timer

By using the main system clock or subsystem clock, interrupt requests (INTWTNI0) are generated at preset intervals.

Table 9-2. Interval Timer Interval Time

Interrupt Request Time	When Operated at f_X = 10 MHz	When Operated at f_{XT} = 32.768 kHz
$2^4/f_W$	409.6 μs	488 μs
$2^5/f_W$	819.2 μs	977 μs
$2^6/f_W$	1.64 ms	1.95 ms
$2^7/f_W$	3.28 ms	3.91 ms
$2^8/f_W$	6.56 ms	7.81 ms
$2^9/f_W$	13.1 ms	15.6 ms
$2^{10}/f_W$	26.2 ms	31.2 ms
$2^{11}/f_W$	52.4 ms	62.4 ms

Remark f_W: Watch timer clock frequency ($f_X/2^8$ or f_{XT})
f_X: Main system clock oscillation frequency
f_{XT}: Subsystem clock oscillation frequency

9.3 Watch Timer Configuration

The watch timer consists of the following hardware.

Table 9-3. Watch Timer Configuration

Item	Configuration
Prescaler	11 bits × 1, 5 bits × 1
Control register	Watch timer operation mode register 0 (WTNM0)

9.4 Register to Control Watch Timer

Watch timer operation mode register 0 (WTNM0) is a register to control the watch timer.

- **Watch timer operation mode register 0 (WTNM0)**
  This register sets the watch timer enable/disable operation, 11-bit prescaler interval time, and 5-bit counter operation control.
  WTNM0 is set by a 1-bit or 8-bit memory manipulation instruction.
  $\overline{\text{RESET}}$ input sets the value of this register to 00H.

Figure 9-2. Watch Timer Operation Mode Register 0 (WTNM0) Format

Address: FF41H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
WTNM0	WTNM07	WTNM06	WTNM05	WTNM04	WTNM03	WTNM02	WTNM01	WTNM00

WTNM07	Watch timer count clock selection
0	$f_X/2^8$ (39.1 kHz)
1	f_{XT} (32.768 kHz)

WTNM06	WTNM05	WTNM04	11-bit prescaler interval time selection		
				WTNM07 = 0	WTNM07 = 1
0	0	0	$2^4/f_W$	$2^{12}/f_X$ (409.6 μs)	$2^4/f_{XT}$ (488 μs)
0	0	1	$2^5/f_W$	$2^{13}/f_X$ (819.2 μs)	$2^5/f_{XT}$ (977 μs)
0	1	0	$2^6/f_W$	$2^{14}/f_X$ (1.64 ms)	$2^6/f_{XT}$ (1.95 ms)
0	1	1	$2^7/f_W$	$2^{15}/f_X$ (3.28 ms)	$2^7/f_{XT}$ (3.91 ms)
1	0	0	$2^8/f_W$	$2^{16}/f_X$ (6.56 ms)	$2^8/f_{XT}$ (7.81 ms)
1	0	1	$2^9/f_W$	$2^{17}/f_X$ (13.1 ms)	$2^9/f_{XT}$ (15.6 ms)
1	1	0	$2^{10}/f_W$	$2^{18}/f_X$ (26.2 ms)	$2^{10}/f_{XT}$ (31.2 ms)
1	1	1	$2^{11}/f_W$	$2^{19}/f_X$ (52.4 ms)	$2^{11}/f_{XT}$ (62.4 ms)

WTNM03	WTNM02	Selection of interrupt request time of the watch timer		
			WTNM07 = 0	WTNM07 = 1
0	0	$2^{14}/f_W$	$2^{22}/f_X$ (0.41 s)	$2^{14}/f_{XT}$ (0.5 s)
0	1	$2^{13}/f_W$	$2^{21}/f_X$ (0.2 s)	$2^{13}/f_{XT}$ (0.25 s)
1	0	$2^5/f_W$	$2^{13}/f_X$ (819.2 μs)	$2^5/f_{XT}$ (977 μs)
1	1	$2^4/f_W$	$2^{12}/f_X$ (409.6 μs)	$2^6/f_{XT}$ (488 μs)

WTNM01	5-bit counter operation control
0	Clear after operation stop
1	Start

WTNM00	Watch timer operation enable
0	Operation stop (clear both 11-bit prescaler and 5-bit counter)
1	Operation enable

Caution Do not change the count clock, interval time, and interrupt request time (by using bits 2 to 7 (WTNM02 to WTNM07) of WTNM0) while the watch timer is operating.

Remarks 1. f_W: Watch timer clock frequency ($f_X/2^8$ or f_{XT})
f_X: Main system clock oscillation frequency
f_{XT}: Subsystem clock oscillation frequency

2. Figures in parentheses apply to operation with f_X = 10 MHz, f_{XT} = 32.768 kHz.

9.5 Watch Timer Operations

9.5.1 Watch timer operation

By using the main system clock or subsystem clock, it operates as a watch timer with preset timing intervals.

Bits 2, 3, and 7 (WTNM02, WTNM03, and WTNM07) of watch timer operation mode register 0 (WTNM0) enable the selection of the timing for the watch timer.

The watch timer generates an interrupt request (INTWTN0) at a fixed time interval.

If bit 0 (WTNM00) and bit 1 (WTNM01) of watch timer operation mode register 0 (WTNM0) are set to 1, the count operation starts. If set to 0, the 5-bit counter is cleared and the count operation stops.

For simultaneous operation of the interval timer, a zero-second start can be achieved by setting WTNM01 to 0.

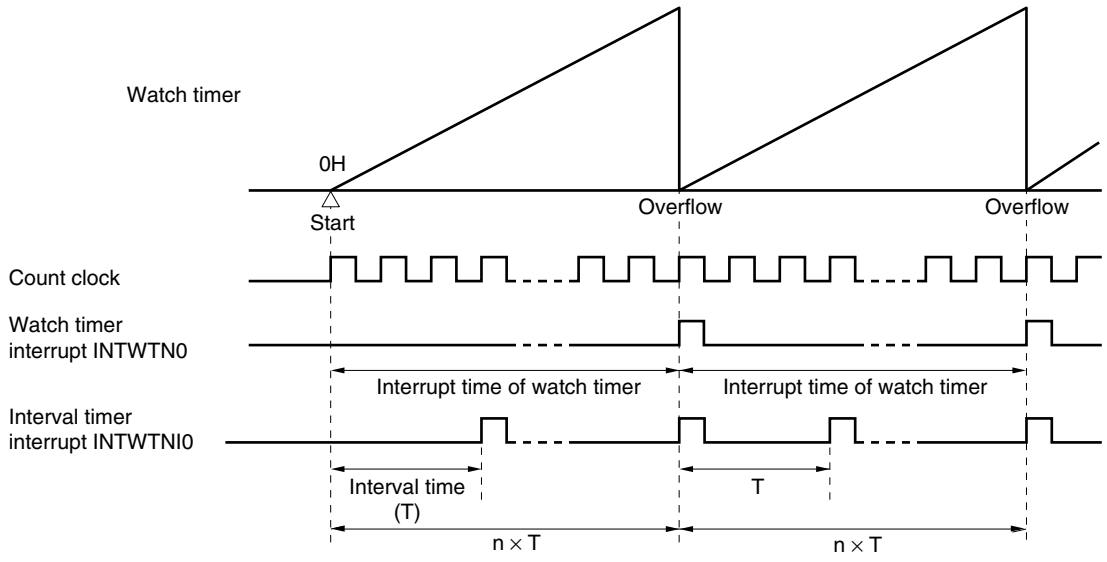
However, in this case, since the 11-bit prescaler is not cleared, at the first overflow (INTWTN0) after the watch timer's zero-second start, an error up to $2^{11} \times 1/f_W$ seconds is generated.

9.5.2 Interval timer operation

The watch timer operates as an interval timer which generates interrupt requests (INTWTNI0) repeatedly at an interval of the preset count value.

The interval time can be selected with bits 4 to 6 and 7 (WTNM04 to WTNM06 and WTNM07) of watch timer operation mode register 0 (WTNM0).

Figure 9-3. Operation Timing of Watch Timer/Interval Timer



Caution If the watch timer and 5-bit counter are enabled by watch timer operation mode register 0 (WTNM0) (by setting bits 0 (WTNM00) and 1 (WTNM01) of WTNM0 to 1), the time from this setting to the occurrence of the first interrupt request (INTWTN0) is not exactly the value set by bits 2 and 3 (WTNM02 and WTNM03) of WTNM0. This is because the 5-bit counter is late by one output cycle of the 11-bit prescaler in starting to count. The second INTWTN0 signal and those that follow are generated exactly at the set time.

Remark n: The number of times of interval timer operations

CHAPTER 10 WATCHDOG TIMER

10.1 Outline of Watchdog Timer

The watchdog timer can also be used to generate a non-maskable interrupt request, maskable interrupt request, or $\overline{\text{RESET}}$ signal at the preset time intervals.

10.2 Watchdog Timer Functions

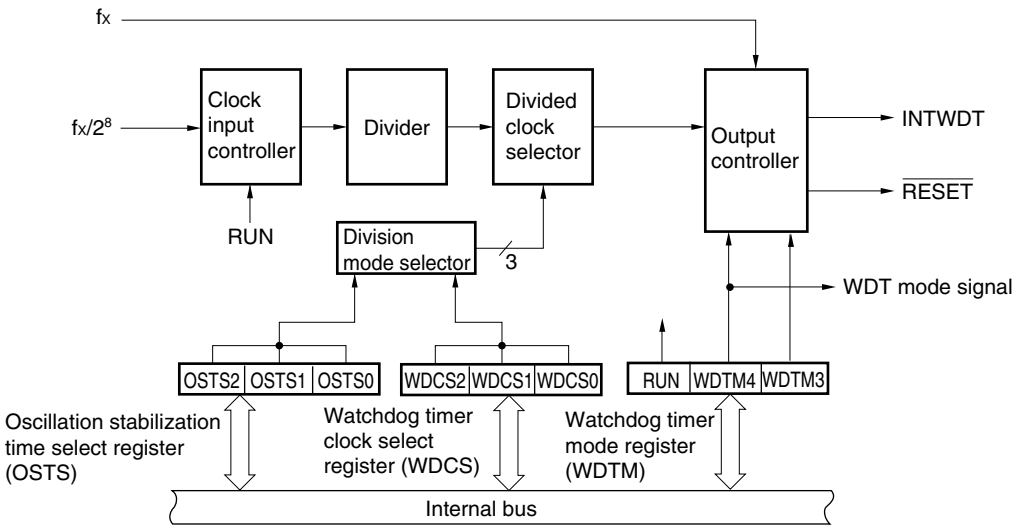
The watchdog timer has the following functions.

- Watchdog timer
- Interval timer
- Oscillation stabilization time selection

Caution Select the watchdog timer mode or the interval timer mode with the watchdog timer mode register (WDTM). (The watchdog timer and the interval timer cannot be used simultaneously.)

Figure 10-1 shows a block diagram of the watchdog timer.

Figure 10-1. Watchdog Timer Block Diagram



(1) Watchdog timer mode

A runaway is detected. Upon detection of the runaway, a non-maskable interrupt request or $\overline{\text{RESET}}$ can be generated.

Table 10-1. Watchdog Timer Runaway Detection Time

Runaway Detection Time
$2^{12} \times 1/f_X$ (410 μs)
$2^{13} \times 1/f_X$ (819 μs)
$2^{14} \times 1/f_X$ (1.64 ms)
$2^{15} \times 1/f_X$ (3.28 ms)
$2^{16} \times 1/f_X$ (6.55 ms)
$2^{17} \times 1/f_X$ (13.1 ms)
$2^{18} \times 1/f_X$ (26.2 ms)
$2^{20} \times 1/f_X$ (105 ms)

Remarks 1. fx: Main system clock oscillation frequency

2. Figures in parentheses are for operation with fx = 10 MHz

(2) Interval timer mode

Interrupt requests are generated at the preset time intervals.

Table 10-2. Interval Time

Interval Time
$2^{12} \times 1/f_X$ (410 μs)
$2^{13} \times 1/f_X$ (819 μs)
$2^{14} \times 1/f_X$ (1.64 ms)
$2^{15} \times 1/f_X$ (3.28 ms)
$2^{16} \times 1/f_X$ (6.55 ms)
$2^{17} \times 1/f_X$ (13.1 ms)
$2^{18} \times 1/f_X$ (26.2 ms)
$2^{20} \times 1/f_X$ (105 ms)

Remarks 1. fx: Main system clock oscillation frequency

2. Figures in parentheses are for operation with fx = 10 MHz

10.3 Watchdog Timer Configuration

The watchdog timer consists of the following hardware.

Table 10-3. Watchdog Timer Configuration

Item	Configuration
Control registers	Watchdog timer clock select register (WDCS) Watchdog timer mode register (WDTM) Oscillation stabilization time select register (OSTS)

10.4 Registers to Control Watchdog Timer

The following three types of registers are used to control the watchdog timer.

- Watchdog timer clock select register (WDCS)
- Watchdog timer mode register (WDTM)
- Oscillation stabilization time select register (OSTS)

(1) Watchdog timer clock select register (WDCS)

This register sets overflow time of the watchdog timer and the interval timer.

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

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

Figure 10-2. Watchdog Timer Clock Select Register (WDCS) Format

Address: FF42H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
WDCS	0	0	0	0	0	WDCS2	WDCS1	WDCS0

WDCS2	WDCS1	WDCS0	Overflow time of watchdog timer/interval timer
0	0	0	$2^{12}/f_X$ (410 μs)
0	0	1	$2^{13}/f_X$ (819 μs)
0	1	0	$2^{14}/f_X$ (1.64 ms)
0	1	1	$2^{15}/f_X$ (3.28 ms)
1	0	0	$2^{16}/f_X$ (6.55 ms)
1	0	1	$2^{17}/f_X$ (13.1 ms)
1	1	0	$2^{18}/f_X$ (26.2 ms)
1	1	1	$2^{20}/f_X$ (105 ms)

Remarks 1. fx: Main system clock oscillation frequency

2. Figures in parentheses are for operation with fx = 10 MHz

(2) Watchdog timer mode register (WDTM)

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

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

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

Figure 10-3. Watchdog Timer Mode Register (WDTM) Format

Address: FFF9H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
WDTM	RUN	0	0	WDTM4	WDTM3	0	0	0

RUN	Watchdog timer operation mode selection[Note 1]
0	Count stop
1	Counter is cleared and counting starts

WDTM4	WDTM3	Watchdog timer operation mode selection[Note 2]
0	×	Interval timer mode[Note 3] (Maskable interrupt request occurs upon generation of an overflow)
1	0	Watchdog timer mode 1 (Non-maskable interrupt request occurs upon generation of an overflow)
1	1	Watchdog timer mode 2 (Reset operation is activated upon generation of an overflow)

Notes 1. Once set to 1, RUN cannot be cleared to 0 by software.
Thus, once counting starts, it can only be stopped by $\overline{\text{RESET}}$ input.

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

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

Caution When 1 is set to RUN so that the watchdog timer is cleared, the actual overflow time is up to $2^8/f_X$ seconds shorter than the time set by watchdog timer clock select register (WDCS).

Remark ×: don't care

(3) Oscillation stabilization time select register (OSTS)

A register to select oscillation stabilization time from reset time or STOP mode released time to the time when oscillation is stabilized.

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

$\overline{\text{RESET}}$ input sets the value of this register to 04H. Thus, when releasing the STOP mode by $\overline{\text{RESET}}$ input, the time required to release is 2^{17}/fx.

Figure 10-4. Oscillation Stabilization Time Select Register (OSTS) Format

Address: FFFAH After reset: 04H R/W

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

OSTS2	OSTS1	OSTS0	Selection of oscillation stabilization time
0	0	0	2^{12}/fx (410 μs)
0	0	1	2^{14}/fx (1.64 ms)
0	1	0	2^{15}/fx (3.28 ms)
0	1	1	2^{16}/fx (6.55 ms)
1	0	0	2^{17}/fx (13.1 ms)
Other than the above			Setting prohibited

Remarks 1. fx: Main system clock oscillation frequency

2. Figures in parentheses are for operation with fx = 10 MHz

10.5 Watchdog Timer Operations

10.5.1 Watchdog timer operation

When bit 4 (WDTM4) of the watchdog timer mode register (WDTM) is set to 1, the watchdog timer is operated to detect any runaway.

The runaway detection time interval is selected with bits 0 to 2 (WDCS0 to WDCS2) of the watchdog timer clock select register (WDCS).

Watchdog timer starts by setting bit 7 (RUN) of WDTM to 1. After the watchdog timer is started, set RUN to 1 within the set runaway time interval. The watchdog timer can be cleared and counting is started by setting RUN to 1. If RUN is not set to 1 and the runaway detection time is exceeded, system reset or a non-maskable interrupt request is generated according to WDTM bit 3 (WDTM3) value.

The watchdog timer continues operating in the HALT mode but it stops in the STOP mode. Thus, set RUN to 1 before the STOP mode is set, clear the watchdog timer and then execute the STOP instruction.

Cautions 1. The actual runaway detection time may be shorter than the set time by a maximum of $2^8/f_X$ seconds.

2. When the subsystem clock is selected for CPU clock, watchdog timer count operation is stopped.

Table 10-4. Watchdog Timer Runaway Detection Time

Runaway Detection Time
$2^{12} \times 1/f_X$ (410 μs)
$2^{13} \times 1/f_X$ (819 μs)
$2^{14} \times 1/f_X$ (1.64 ms)
$2^{15} \times 1/f_X$ (3.28 ms)
$2^{16} \times 1/f_X$ (6.55 ms)
$2^{17} \times 1/f_X$ (13.1 ms)
$2^{18} \times 1/f_X$ (26.2 ms)
$2^{20} \times 1/f_X$ (105 ms)

Remarks 1. f_X: Main system clock oscillation frequency

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

10.5.2 Interval timer operation

The watchdog timer operates as an interval timer which generates interrupt requests repeatedly at an interval of the preset count value when bit 4 (WDTM4) of the watchdog timer mode register (WDTM) is set to 0.

The interval time of interval timer is selected with bits 0 to 2 (WDCS0 to WDCS2) of the watchdog timer clock select register (WDCS). When 1 is set to bit 7 (RUN) of WDTM, the watchdog timer operates as the interval timer.

When the watchdog timer operated as the interval timer, the interrupt mask flag (WDTMK) and priority specify flag (WDTPR) are validated and the maskable interrupt request (INTWDT) can be generated. Among maskable interrupts, INTWDT has the highest priority at default.

The interval timer continues operating in the HALT mode but it stops in STOP mode. Thus, set RUN to 1 before the STOP mode is set, clear the interval timer and then execute the STOP instruction.

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

2. The interval time just after setting by WDTM may be shorter than the set time by a maximum of $2^8/f_X$ seconds.

3. When the subsystem clock is selected for CPU clock, watchdog timer count operation is stopped.

Table 10-5. Interval Timer Interval Time

Interval Time
$2^{12} \times 1/f_X$ (410 μs)
$2^{13} \times 1/f_X$ (819 μs)
$2^{14} \times 1/f_X$ (1.64 ms)
$2^{15} \times 1/f_X$ (3.28 ms)
$2^{16} \times 1/f_X$ (6.55 ms)
$2^{17} \times 1/f_X$ (13.1 ms)
$2^{18} \times 1/f_X$ (26.2 ms)
$2^{20} \times 1/f_X$ (105 ms)

Remarks 1. f_X: Main system clock oscillation frequency

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

CHAPTER 11 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLERS

11.1 Outline of Clock Output/Buzzer Output Controllers

The clock output circuit supplies other devices with the divided main system clock and the subsystem clock, and buzzer output supplies the buzzer frequency with the divided main system clock.

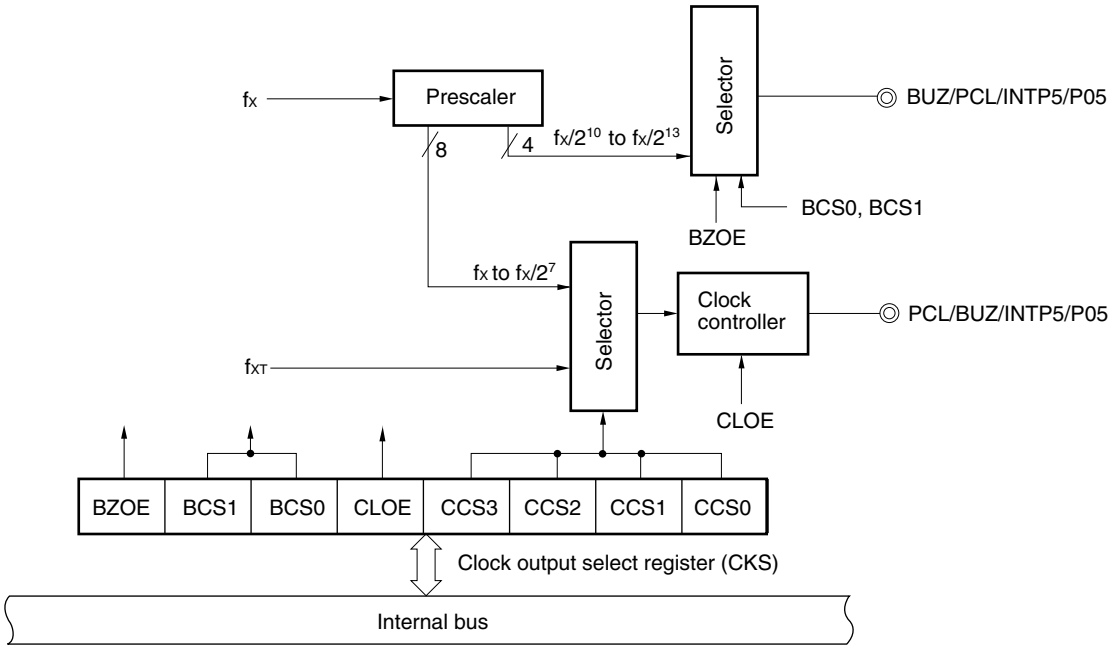
11.2 Clock Output/Buzzer Output Controller Functions

The clock output controller is intended for carrier output during remote controlled transmission and clock output for supply to peripheral LSIs. The clock selected with the clock output select register (CKS) is output.

In addition, the buzzer output is intended for square wave output of the buzzer frequency selected with CKS.

Figure 11-1 shows the block diagram of clock output/buzzer output controllers.

Figure 11-1. Clock Output/Buzzer Output Controller Block Diagram



11.3 Clock Output/Buzzer Output Controller Configuration

The clock output/buzzer output controllers consist of the following hardware.

Table 11-1. Clock Output/Buzzer Output Controllers Configuration

Item	Configuration
Control registers	Clock output select register (CKS) Port mode register 0 (PM0)[Note]

Note See **Figure 4-3 P05 Block Diagram**.

11.4 Registers to Control Clock Output/Buzzer Output Controllers

The following two types of registers are used to control the clock output/buzzer output controllers.

- Clock output select register (CKS)
- Port mode register 0 (PM0)

(1) Clock output select register (CKS)

This register sets output enable/disable for clock output (PCL) and for the buzzer frequency output (BUZ), and sets the output clock.

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

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

Figure 11-2. Clock Output Select Register (CKS) Format

Address: FF40H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CKS	BZOE	BCS1	BCS0	CLOE	CCS3	CCS2	CCS1	CCS0

BZOE	BUZ output enable/disable specification
0	Stop clock divider operation. BUZ fixed to low level.
1	Enable clock divider operation. BUZ output enabled.

BCS1	BCS0	BUZ output clock selection
0	0	$f_X/2^{10}$ (9.77 kHz)
0	1	$f_X/2^{11}$ (4.88 kHz)
1	0	$f_X/2^{12}$ (2.44 kHz)
1	1	$f_X/2^{13}$ (1.22 kHz)

CLOE	PCL output enable/disable setting
0	Stop clock divider operation. PCL fixed to low level
1	Enable clock divider operation. PCL output enabled.

CCS3	CCS2	CCS1	CCS0	PCL output clock selection
0	0	0	0	f_X (10 MHz)
0	0	0	1	$f_X/2$ (5 MHz)
0	0	1	0	$f_X/2^2$ (2.5 MHz)
0	0	1	1	$f_X/2^3$ (1.25 MHz)
0	1	0	0	$f_X/2^4$ (625 kHz)
0	1	0	1	$f_X/2^5$ (312.5 kHz)
0	1	1	0	$f_X/2^6$ (156.3 kHz)
0	1	1	1	$f_X/2^7$ (78.1 kHz)
1	0	0	0	f_{XT} (32.768 kHz)
Other than above				Setting prohibited

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

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

(2) Port mode register 0 (PM0)

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

When using the PCL/BUZ/INTP5/P05 pin for clock output or for buzzer output, set PM05 and the output latch of P05 to 0.

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

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

Figure 11-3. Port Mode Register 0 (PM0) Format

Address: FF20H After reset: FFH R/W

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

PM0n	P0n pin I/O mode selection (n = 0 to 5)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

11.5 Clock Output/Buzzer Output Controller Operations

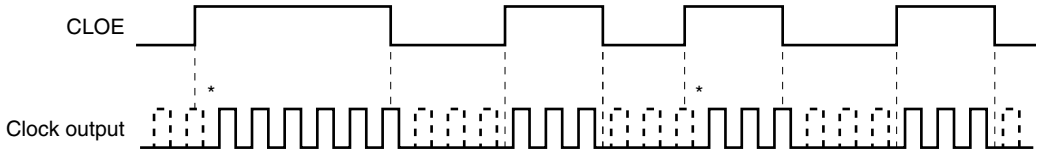
11.5.1 Operation as clock output

The clock pulse is output as the following procedure.

<1> Select the clock pulse output frequency with bits 0 to 3 (CCS0 to CCS3) of the clock output select register (CKS) (clock pulse output in disabled status).
<2> Set bit 4 (CLOE) of CKS to 1, and enable clock output.

Remark The clock output controller is designed not to output pulses with a small width during output enable/disable switching of the clock output. As shown in Figure 11-4, be sure to start output from the low period of the clock (marked with * in the figure). When stopping output, do so after securing high level of the clock.

Figure 11-4. Remote Control Output Application Example



11.5.2 Operation as buzzer output

The buzzer frequency is output as the following procedure.

<1> Select the buzzer output frequency with bits 5 and 6 (BCS0, BCS1) of the clock output select register (CKS) (buzzer output in disabled status).
<2> Set bit 7 (BZOE) of CKS to 1 to enable buzzer output.

CHAPTER 12 A/D CONVERTER

12.1 A/D Converter Functions

The A/D converter is a 10-bit resolution converter that converts analog inputs into digital signals. It can control up to 10 analog input channels (ANI0 to ANI9).

(1) Hardware start

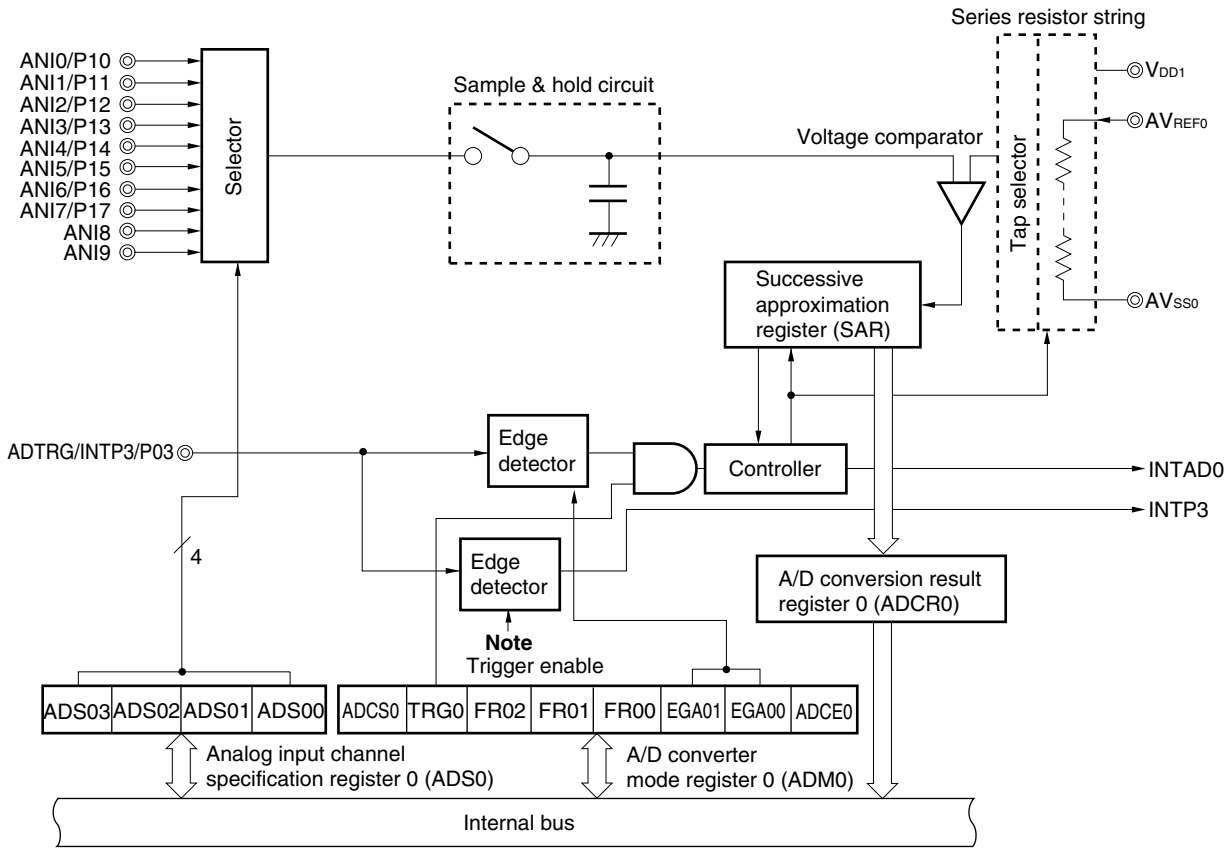
Conversion is started by trigger input (ADTRG: rising edge, falling edge, or both rising and falling edges can be specified).

(2) Software start

Conversion is started by setting A/D converter mode register 0 (ADM0).

Select one channel for analog input from ANI0 to ANI9 to start A/D conversion. In the case of hardware start, the A/D converter stops when A/D conversion is completed, and an interrupt request (INTAD0) is generated. In the case of software start, A/D conversion is repeated. Each time an A/D conversion operation ends, an interrupt request (INTAD0) is generated.

Figure 12-1. A/D Converter Block Diagram



Note The valid edge is specified by bit 3 of the EGP and EGN registers (see **Figure 18-5 External Interrupt Rising Edge Enable Register (EGP), External Interrupt Falling Edge Enable Register (EGN) Format**).

12.2 A/D Converter Configuration

The A/D converter consists of the following hardware.

Table 12-1. A/D Converter Configuration

Item	Configuration
Analog input	10 channels (ANI0 to ANI9)
Registers	Successive approximation register (SAR) A/D conversion result register 0 (ADCR0)
Control registers	A/D converter mode register 0 (ADM0) Analog input channel specification register 0 (ADS0) External interrupt rising edge enable register (EGP) External interrupt falling edge enable register (EGN)

(1) Successive approximation register (SAR)

This register compares the analog input voltage value to the voltage tap (compare voltage) value applied from the series resistor string, and holds the result from the most significant bit (MSB).

When up to the least significant bit (LSB) is held (end of A/D conversion), the SAR contents are transferred to A/D conversion result register 0 (ADCR0).

(2) A/D conversion result register 0 (ADCR0)

This is a 16-bit register which stores the A/D conversion results. The lower 6 bits are fixed to 0. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register.

ADCR0 is read by a 16-bit memory manipulation instruction.

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

Symbol	FF0FH								FF0EH								Address	After reset	R/W
ADCR0											0	0	0	0	0	0	FF0EH, FF0FH	0000H	R

Caution When writing is performed to A/D converter mode register 0 (ADM0) and analog input channel specification register 0 (ADS0), the contents of ADCR0 may become undefined. Read the conversion result following conversion completion before writing to ADM0, ADS0. Using a timing other than the above may cause an incorrect conversion result to be read.

(3) Sample & hold circuit

The sample & hold circuit samples each analog input signal sequentially applied from the input circuit, and sends it to the voltage comparator. This circuit holds the sampled analog input voltage value during A/D conversion.

(4) Voltage comparator

The voltage comparator compares the analog input to the series resistor string output voltage.

(5) Series resistor string

The series resistor string is connected between AV_{REF0} and AV_{SS0}, and generates a voltage to be compared to the analog input.

(6) ANI0 to ANI9 pins

These ten analog input pins are used to input analog signals to undergo A/D conversion to the A/D converter. ANI0 to ANI7 can be used as input ports except for the pins specified as analog input by analog input channel specification register 0 (ADS0).

Cautions 1. Use the ANI0 to ANI9 input voltages within the specification range. If a voltage higher than AV_{REF0} or lower than AV_{SS0} is applied (even if within the absolute maximum rating range), the conversion value of that channel will be undefined and the conversion values of other channels may also be affected.

2. Analog input (ANI0 to ANI7) pins are alternate function pins that can also be used as input port (P10 to P17) pins. When A/D conversion is performed by selecting any one of ANI0 through ANI7, do not execute any input instruction to port 1 during conversion. It may cause a lower conversion resolution.

When a digital pulse is applied to a pin adjacent to the pin in the process of A/D conversion, A/D conversion values may not be obtained as expected due to coupling noise. Thus, do not apply any pulse to a pin adjacent to the pin in the process of A/D conversion.

(7) AV_{REF0} pin

This pin inputs the A/D converter reference voltage.

It converts signals input to ANI0 to ANI9 into digital signals according to the voltage applied between AV_{REF0} and AV_{SS0}.

Caution A series resistor string is connected between the AV_{REF0} and AV_{SS0} pins. Therefore, when output impedance of the reference voltage is too high, it seems as if the AV_{REF0} pin and the series resistor string are connected in series. This may cause a greater reference voltage error.

(8) AV_{SS0} pin

This is the GND potential pin of the A/D converter. Always keep it at the same potential as the V_{SS0} pin when not using the A/D converter.

(9) V_{DD1} pin

This is the positive power supply pin, except for the port block.

12.3 Registers to Control A/D Converter

The following four types of registers are used to control the A/D converter.

- A/D converter mode register 0 (ADM0)
- Analog input channel specification register 0 (ADS0)
- External interrupt rising edge enable register (EGP)
- External interrupt falling edge enable register (EGN)

(1) A/D converter mode register 0 (ADM0)

This register sets the conversion time for analog input to be A/D converted, conversion start/stop, and external trigger.

ADM0 is set by a 1-bit or 8-bit memory manipulation instruction.

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

Figure 12-2. A/D Converter Mode Register 0 (ADM0) Format

Address: FF80H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADM0	ADCS0	TRG0	FR02	FR01	FR00	EGA01	EGA00	ADCE0

ADCS0	A/D conversion operation control
0	Stop conversion operation.
1	Enable conversion operation.

TRG0	Software start/hardware start selection
0	Software start
1	Hardware start

FR02	FR01	FR00	Conversion time selection[Note 1]
0	0	0	144/fx (14.4 μs)
0	0	1	120/fx (Setting prohibited[Note 2])
0	1	0	96/fx (Setting prohibited[Note 2])
1	0	0	576/fx (57.6 μs)
1	0	1	480/fx (48.0 μs)
1	1	0	384/fx (38.4 μs)
Other than above			Setting prohibited

EGA01	EGA00	External trigger signal, edge specification
0	0	No edge detection
0	1	Falling edge detection
1	0	Rising edge detection
1	1	Both falling and rising edge detection

ADCE0	Control of voltage booster for A/D converter circuit[Note 3]
0	Stops operation.
1	Enables operation.

Notes 1. Set so that the A/D conversion time is 14 μs or more.

2. Setting prohibited because A/D conversion time is less than 14 μs.

3. Before executing A/D conversion (ADCS0 = 1), be sure to start the voltage booster (ADCE0 = 1).

Cautions 1. When rewriting FR00 to FR02 to other than the same data, stop A/D conversion operations once before performing it.

2. Make sure by using software that a wait time of 14 μs (MIN.) elapses between when ADCE0 is set and when ADCS0 is set.

3. Before clearing ADCE0, clear ADCS0.

Remarks 1. fx: Main system clock oscillation frequency

2. Figures in parentheses are for operation with fx = 10 MHz.

(2) Analog input channel specification register 0 (ADS0)

This register specifies the analog voltage input port for A/D conversion.

ADS0 is set by an 8-bit memory manipulation instruction.

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

Figure 12-3. Analog Input Channel Specification Register 0 (ADS0) Format

Address: FF81H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS0	0	0	0	0	ADS03	ADS02	ADS01	ADS00

ADS03	ADS02	ADS01	ADS00	Analog input channel specification
0	0	0	0	ANI0
0	0	0	1	ANI1
0	0	1	0	ANI2
0	0	1	1	ANI3
0	1	0	0	ANI4
0	1	0	1	ANI5
0	1	1	0	ANI6
0	1	1	1	ANI7
1	0	0	0	ANI8
1	0	0	1	ANI9
Other than above				Setting prohibited

(3) External interrupt rising edge enable register (EGP), external interrupt falling edge enable register (EGN)

These registers specify the valid edge for INTP0 to INTP5.

EGP and EGN are set by a 1-bit or 8-bit memory manipulation instruction.

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

Figure 12-4. External Interrupt Rising Edge Enable Register (EGP), External Interrupt Falling Edge Enable Register (EGN) Format

Address: FF48H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP	0	0	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

Address: FF49H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN	0	0	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 5)
0	0	Interrupt disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

12.4 A/D Converter Operation

12.4.1 Basic operations of A/D converter

<1> Select one channel for A/D conversion using analog input channel specification register 0 (ADS0).

<2> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<3> When sampling has been done for a certain time, the sample & hold circuit is placed in the hold state and the input analog voltage is held until the A/D conversion operation is ended.

<4> Bit 9 of the successive approximation register (SAR) is set. The series resistor string voltage tap is set to (1/2) AV_{REF0} by the tap selector.

<5> The voltage difference between the series resistor string voltage tap and analog input is compared by the voltage comparator. If the analog input is greater than (1/2) AV_{REF0}, the MSB of SAR remains set. If the analog input is smaller than (1/2) AV_{REF0}, the MSB is reset.

<6> Next, bit 8 of SAR is automatically set, and the operation proceeds to the next comparison. The series resistor string voltage tap is selected according to the preset value of bit 9, as described below.

- Bit 9 = 1: (3/4) AV_{REF0}
- Bit 9 = 0: (1/4) AV_{REF0}

The voltage tap and analog input voltage are compared and bit 8 of SAR is manipulated as follows.

- Analog input voltage ≥ Voltage tap: Bit 8 = 1
- Analog input voltage < Voltage tap: Bit 8 = 0

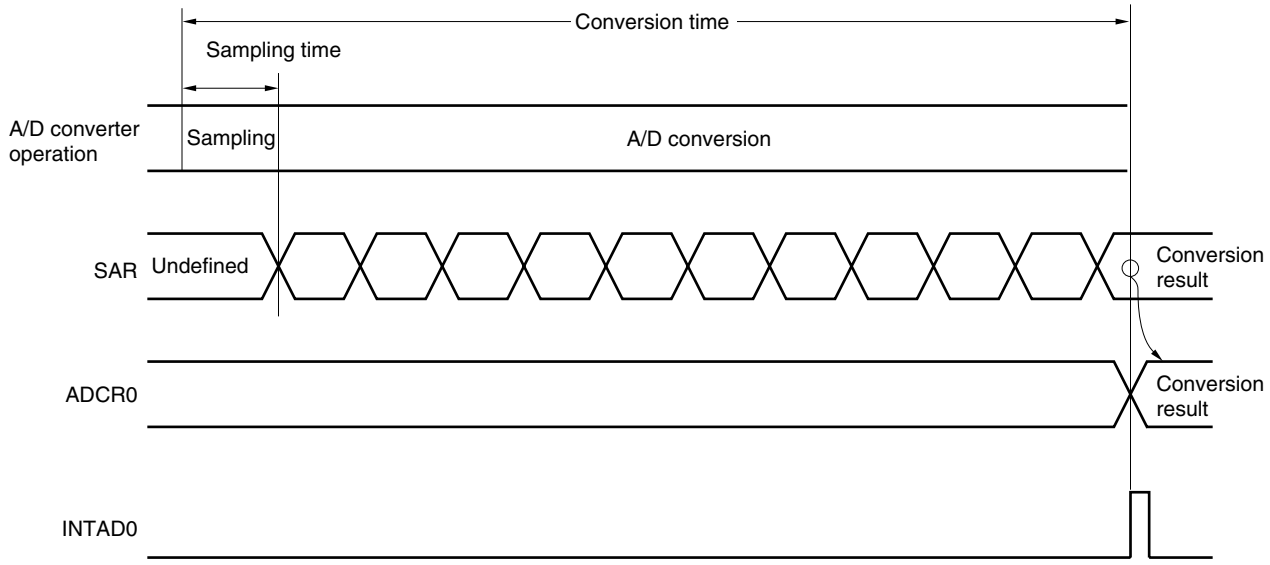
<7> Comparison is continued in this way up to bit 0 of SAR.

<8> Upon completion of the comparison of 10 bits, an effective digital result value remains in SAR, and the result value is transferred to and latched in A/D conversion result register 0 (ADCR0).

At the same time, the A/D conversion end interrupt request (INTAD0) can also be generated.

Caution The first A/D conversion value just after A/D conversion operations start may not fall within the rating.

Figure 12-5. Basic Operation of A/D Converter



A/D conversion operations are performed continuously until bit 7 (ADCS0) of A/D converter mode register 0 (ADM0) is reset (0) by software.

If a write operation is performed to the ADM0 or the analog input channel specification register 0 (ADS0) during an A/D conversion operation, the conversion operation is initialized, and if the ADCS0 bit is set (1), conversion starts again from the beginning.

$\overline{\text{RESET}}$ input sets A/D conversion result register 0 (ADCR0) to 00H.

12.4.2 Input voltage and conversion results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI9) and the A/D conversion result (stored in A/D conversion result register 0 (ADCR0)) is shown by the following expression.

$$ADCR0 = INT\left(\frac{V_{IN}}{AV_{REF0}} \times 1{,}024 + 0.5\right)$$

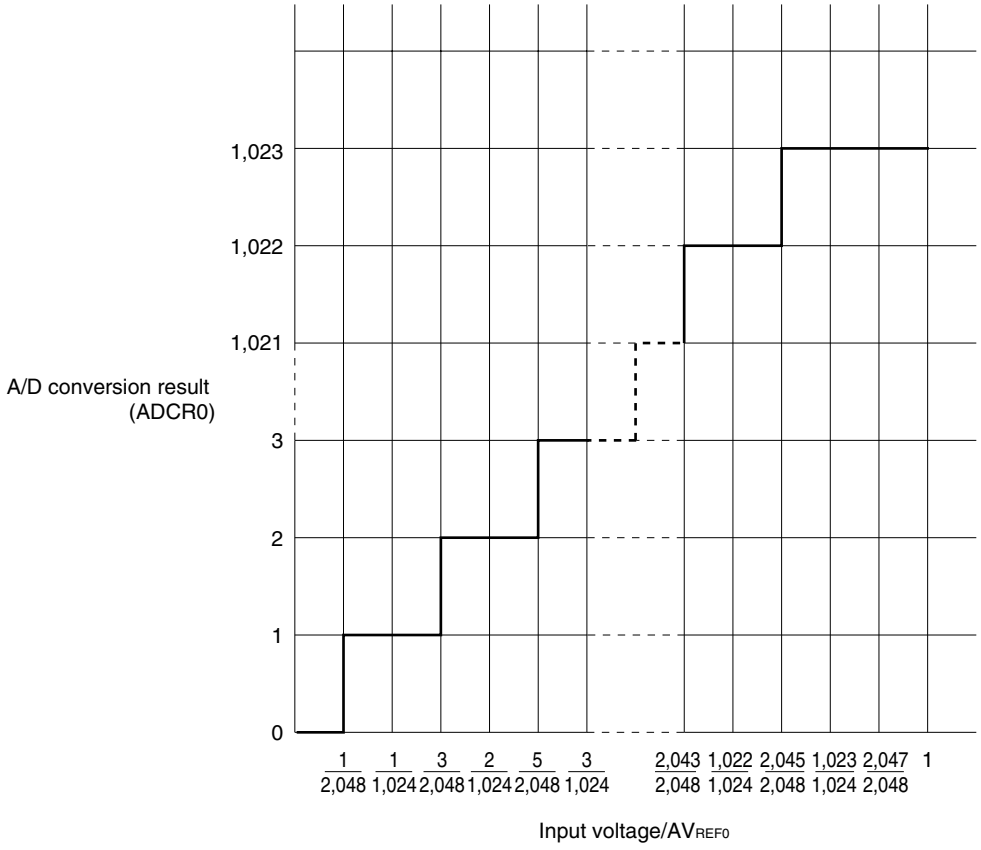
or

$$(ADCR0 - 0.5) \times \frac{AV_{REF0}}{1{,}024} \leq V_{IN} < (ADCR0 + 0.5) \times \frac{AV_{REF0}}{1{,}024}$$

where, INT( ): Function which returns integer part of value in parentheses
V_{IN}: Analog input voltage
AV_{REF0}: AV_{REF0} pin voltage
ADCR0: A/D conversion result register 0 (ADCR0) value

Figure 12-6 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 12-6. Relationship Between Analog Input Voltage and A/D Conversion Result



12.4.3 A/D converter operation mode

Select one analog input channel from among ANI0 to ANI9 using analog input channel specification register 0 (ADS0) to start A/D conversion.

A/D conversion can be started in either of the following two ways.

- Hardware start: Conversion is started by trigger input (rising edge, falling edge, or both rising and falling edges specified).
- Software start: Conversion is started by specifying A/D converter mode register 0 (ADM0).

The A/D conversion result is stored in A/D conversion result register 0 (ADCR0), and the interrupt request signal (INTAD0) is simultaneously generated.

(1) A/D conversion by hardware start

When bit 6 (TRG0) and bit 7 (ADCS0) of A/D converter mode register 0 (ADM0) are set to 1 after bit 0 (ADCE0) is set to 1, the A/D conversion standby state is set. When the external trigger signal (ADTRG) is input, A/D conversion of the voltage applied to the analog input pin specified by analog input channel specification register 0 (ADS0) starts.

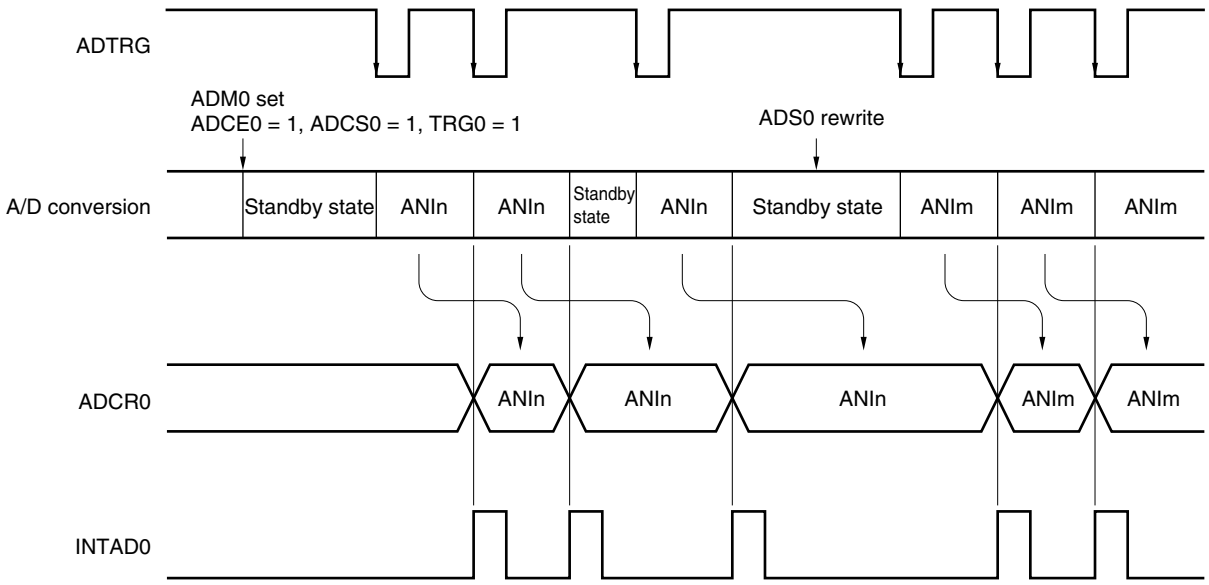
Upon the end of the A/D conversion, the conversion result is stored in A/D conversion result register 0 (ADCR0), and the interrupt request signal (INTAD0) is generated. After one A/D conversion operation is started and ended, the next conversion operation is not started until a new external trigger signal is input.

If ADS0 is rewritten during A/D conversion operation, the converter suspends A/D conversion and waits for a new external trigger signal to be input. When the external trigger input signal is reinput, A/D conversion is carried out from the beginning. If ADS0 is rewritten during A/D conversion waiting, A/D conversion starts when the following external trigger input signal is input.

If data with ADCS0 set to 0 is written to ADM0 during A/D conversion, the A/D conversion operation stops immediately.

Caution When P03/INTP3/ADTRG is used as the external trigger input (ADTRG), specify the valid edge by bits 1, 2 (EGA00, EGA01) of A/D converter mode register 0 (ADM0) and set the interrupt mask flag (PMK3) to 1.

Figure 12-7. A/D Conversion by Hardware Start (When Falling Edge Is Specified)



Remarks 1. n = 0, 1, ..., 9

2. m = 0, 1, ..., 9

(2) A/D conversion by software start

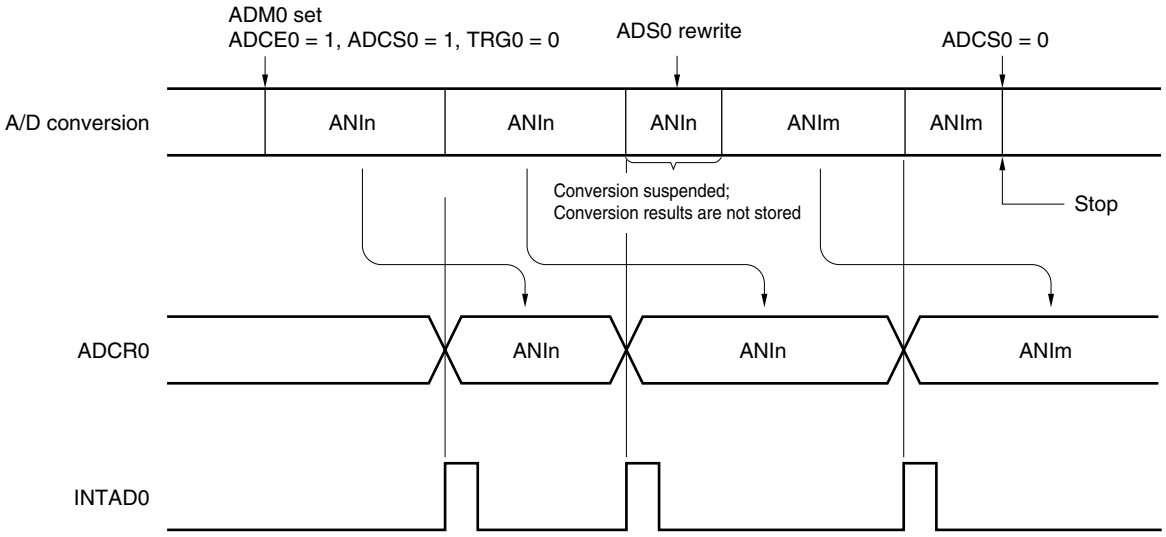
When bit 6 (TRG0) and bit 7 (ADCS0) of A/D converter mode register 0 (ADM0) are set to 0 and 1 after bit 0 (ADCE0) is set to 1, respectively, A/D conversion of the voltage applied to the analog input pin specified by analog input channel specification register 0 (ADS0) starts.

Upon the end of the A/D conversion, the conversion result is stored in A/D conversion result register 0 (ADCR0), and the interrupt request signal (INTAD0) is generated. After one A/D conversion operation is started and ended, the next conversion operation is immediately started. A/D conversion operations are repeated until new data is written to ADS0.

If ADS0 is rewritten during A/D conversion, the converter suspends A/D conversion operation and A/D conversion of the new selected analog input channel starts.

If data with ADCS0 set to 0 is written to ADM0 during A/D conversion, the A/D conversion operation stops immediately.

Figure 12-8. A/D Conversion by Software Start



Remarks 1. n = 0, 1, ..., 9

2. m = 0, 1, ..., 9

12.5 How to Read the A/D Converter Characteristics Table

Here we will explain the special terms unique to A/D converters.

(1) Resolution

This is the minimum analog input voltage that can be identified. That is, the percentage of the analog input voltage per 1 bit of digital output is called 1LSB (Least Significant Bit). The percentage of 1LSB with respect to the full scale is expressed by %FSR (Full Scale Range).

When the resolution is 10 bits,

$1LSB = 1/2^{10} = 1/1,024$
$= 0.098\%FSR$

Accuracy has no relation to resolution, but is determined by overall error.

(2) Overall error

This shows the maximum error value between the actual measured value and the theoretical value.
Zero scale error, full scale error, integral linearity error, differential linearity error and errors which are combinations of these express overall error.
Note that, quantization error is not included in overall error in the characteristics table.

(3) Quantization error

When analog values are converted to digital values, a ±1/2LSB error naturally occurs. In an A/D converter, an analog input voltage in a range of ±1/2LSB is converted to the same digital code, so a quantization error cannot be avoided.
Note that the quantization error is not included in the overall error, zero scale error, full scale error, integral linearity error, and differential linearity error in the characteristics table.

Figure 12-9. Overall Error

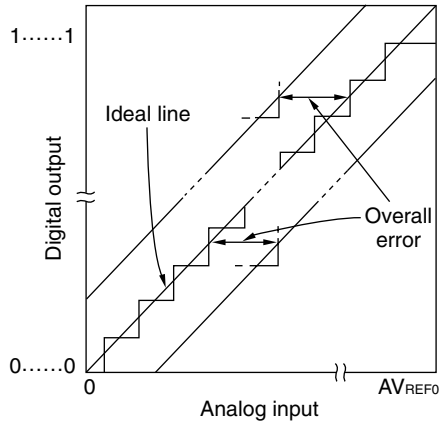
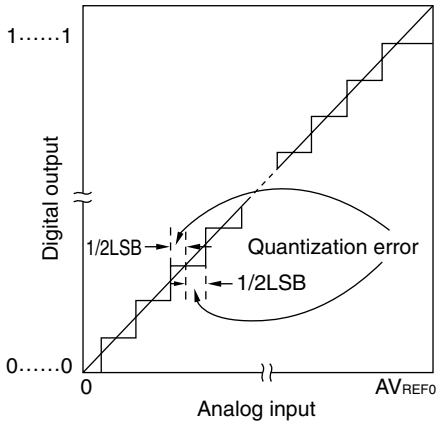


Figure 12-10. Quantization Error



(4) Zero scale error

This shows the difference between the actual measured value of the analog input voltage and the theoretical value (1/2 LSB) when the digital output changes from 0……000 to 0……001. If the actual measured value is greater than the theoretical value, it shows the difference between the actual measured value of the analog input voltage and the theoretical value (3/2LSB) when the digital output changes from 0……001 to 0……010.

(5) Full scale error

This shows the difference between the actual measured value of the analog input voltage and the theoretical value (full scale –3/2 LSB) when the digital output changes from 1……110 to 1……111.

(6) Integral linearity error

This shows the degree to which the conversion characteristics deviate from the ideal linear relationship. It expresses the maximum value of the difference between the actual measured value and the ideal straight line when the zero scale error and full scale error are 0.

(7) Differential linearity error

The ideal width to output a certain code is 1LSB. The following shows the difference between the actual measurement values and ideal values of the width when outputting a certain code.

Figure 12-11. Zero Scale Error

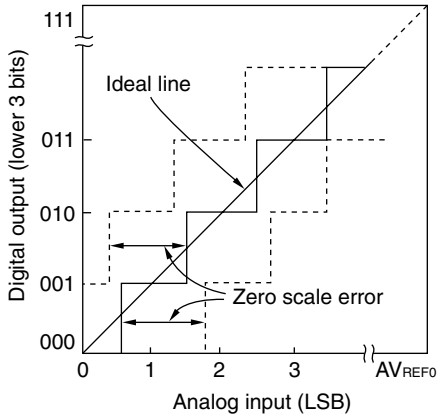


Figure 12-12. Full Scale Error

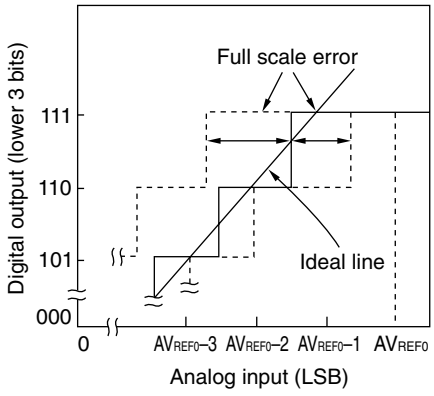


Figure 12-13. Integral Linearity Error

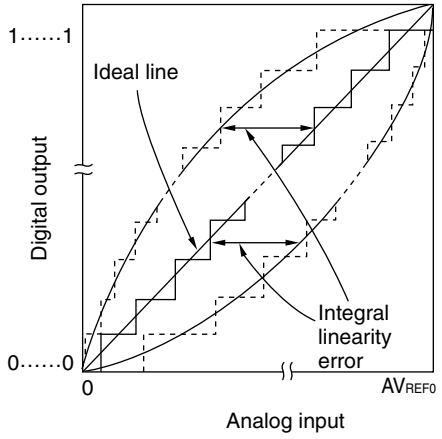
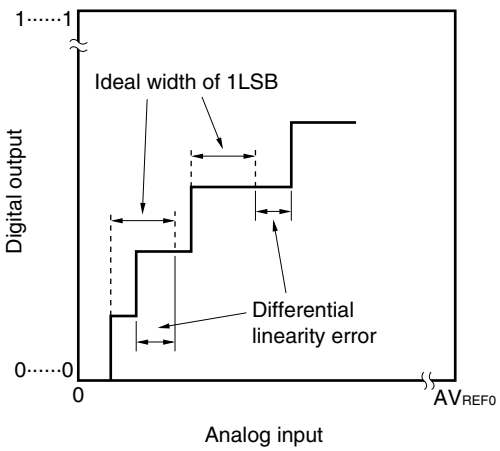


Figure 12-14. Differential Linearity Error



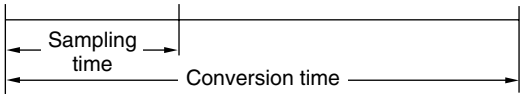
(8) Conversion time

This expresses the time from when the analog input voltage was applied to the time when the digital output was obtained.

Sampling time is included in the conversion time in the characteristics table.

(9) Sampling time

This is the time the analog switch is turned on for the analog voltage to be sampled by the sample and hold circuit.

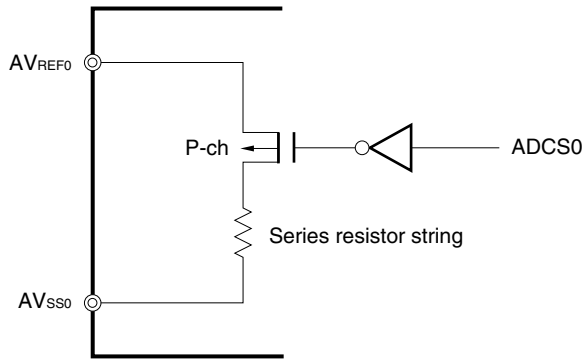


12.6 A/D Converter Cautions

(1) Current consumption in standby mode

The A/D converter stops operating in the standby mode. At this time, the current consumption can be reduced by stopping the conversion operation (by setting bit 7 (ADCS0) of A/D converter mode register 0 (ADM0) to 0). Figure 12-15 shows how to reduce current consumption in the standby mode.

Figure 12-15. Example of Method of Reducing Current Consumption in Standby Mode



(2) Input range of ANI0 to ANI9

The input voltages of ANI0 to ANI9 should be within the specification range. In particular, if a voltage higher than AV_{REF0} or lower than AV_{SS0} is input (even if within the absolute maximum rating range), the conversion value of that channel will be undefined and the conversion values of other channels may also be affected.

(3) Contending operations

<1> Contention between A/D conversion result register 0 (ADCR0) write and ADCR0 read by instruction upon the end of conversion

ADCR0 read is given priority. After the read operation, the new conversion result is written to ADCR0.

<2> Contention between ADCR0 write and external trigger signal input upon the end of conversion

The external trigger signal is not accepted during A/D conversion. Therefore, the external trigger signal is not accepted during ADCR0 write.

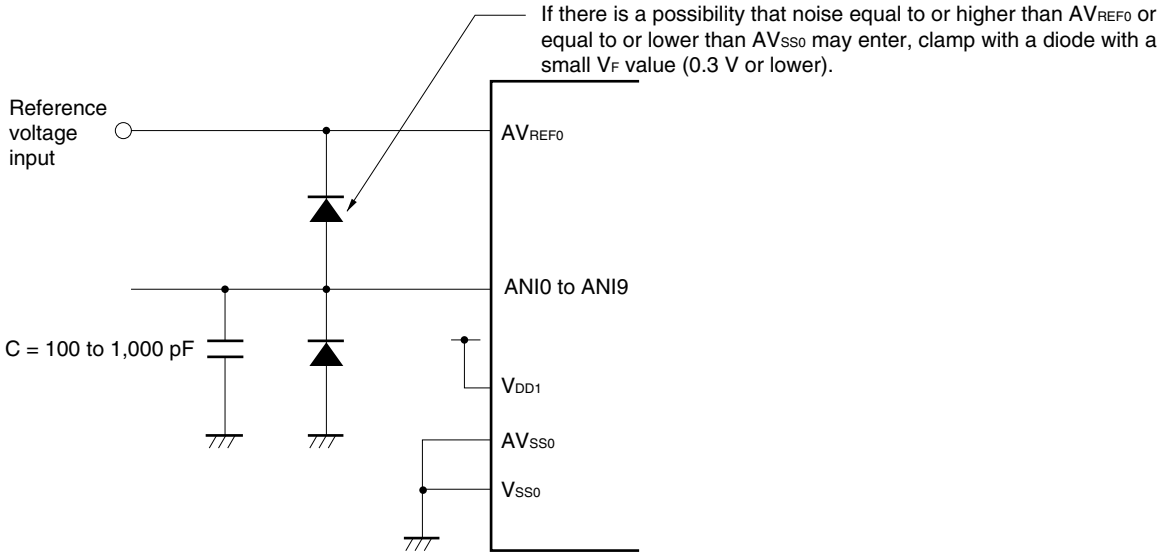
<3> Contention between ADCR0 write and A/D converter mode register 0 (ADM0) write or analog input channel specification register 0 (ADS0) write upon the end of conversion

ADM0 or ADS0 write is given priority. ADCR0 write is not performed, nor is the conversion end interrupt request signal (INTAD0) generated.

(4) Noise countermeasures

To maintain the 10-bit resolution, attention must be paid to noise input to the AV_{REF0} and ANI0 to ANI9 pins. Because the effect increases in proportion to the output impedance of the analog input source, it is recommended that a capacitor be connected externally as shown in Figure 12-16 to reduce noise.

Figure 12-16. Analog Input Pin Connection



(5) ANI0 to ANI9

The analog input pins (ANI0 to ANI9) also function as port pins.

When A/D conversion is performed with any of pins ANI0 to ANI9 selected, do not execute an input instruction to port 1 while conversion is in progress, as this may reduce the conversion resolution.

Also, if digital pulses are applied to other analog input pins during A/D conversion, the expected A/D conversion value may not be obtainable due to coupling noise. Therefore, avoid applying pulses to other analog input pins during A/D conversion.

(6) AV_{REF0} pin input impedance

A series resistor string is connected between the AV_{REF0} pin and the AV_{SS0} pin.

Therefore, when the output impedance of the reference voltage is too high, it seems as if the AV_{REF0} pin and the series resistor string are connected in series. This may cause a greater reference voltage error.

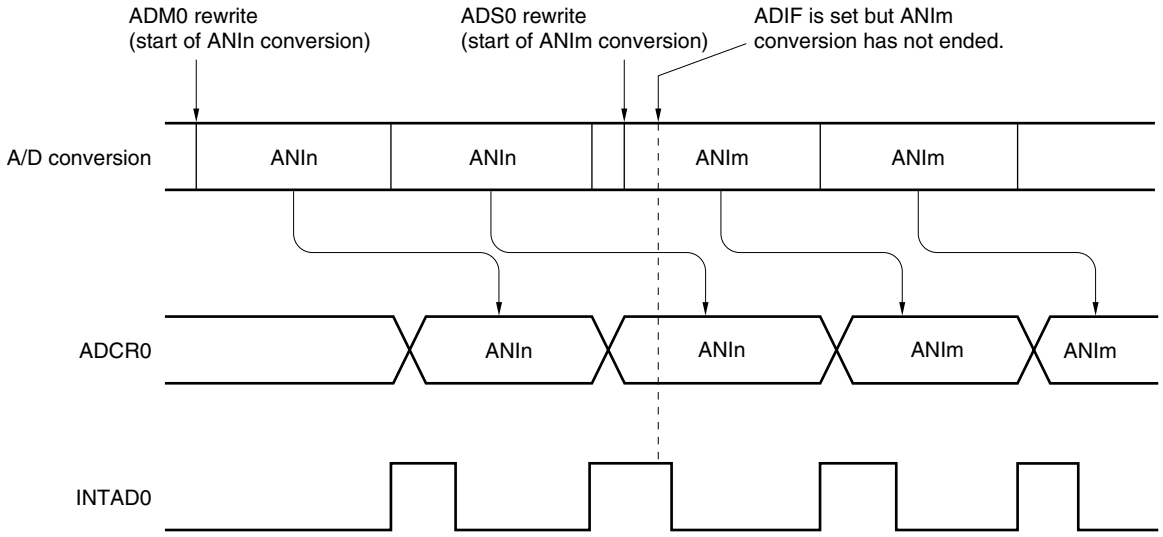
(7) Interrupt request flag (ADIF0)

The interrupt request flag (ADIF0) is not cleared even if analog input channel specification register 0 (ADS0) is changed.

Therefore, if an analog input pin is changed during A/D conversion, the A/D conversion result and conversion end interrupt request flag for the pre-change analog input may be set just before the ADS0 rewrite. Caution is therefore required since, at this time, when ADIF0 is read immediately just after the ADS0 rewrite, ADIF0 is set despite the fact that the A/D conversion for the post-change analog input has not ended.

When A/D conversion is restarted after it is stopped, clear ADIF0 before restarting.

Figure 12-17. A/D Conversion End Interrupt Request Generation Timing



Remarks 1. n = 0, 1, ..., 9

2. m = 0, 1, ..., 9

(8) Conversion results just after A/D conversion start

If bit 7 (ADCS0) of A/D converter mode register 0 (ADM0) is set to 1 without setting bit 0 (ADCE0) to 1, the first A/D conversion value immediately after A/D conversion has been started may not satisfy the rated value. Polling A/D conversion end interrupt request (INTAD0) and take measures such as removing the first conversion results. The same may apply if ADCS0 is set to 1 without the lapse of the wait time of 14 μs (MIN.) after ADCE0 has been set to 1. Make sure that the specified wait time elapses.

(9) A/D conversion result register 0 (ADCR0) read operation

When writing is performed to A/D converter mode register 0 (ADM0) and analog input channel specification register 0 (ADS0), the contents of ADCR0 may become undefined. Read the conversion result following conversion completion before writing to ADM0, ADS0. Using a timing other than the above may cause an incorrect conversion result to be read.

(10) Timing at which A/D conversion result is undefined

The A/D conversion value may be undefined if the timing of completion of A/D conversion and the timing of stopping the A/D conversion conflict with each other. Therefore, read the A/D conversion result during the A/D conversion operation. To read the conversion result after stopping the A/D conversion operation, be sure to stop the A/D conversion before the next conversion ends.

Figures 12-18 and 12-19 show the timing of reading the conversion result.

Figure 12-18. Timing of Reading Conversion Result (When Conversion Result Is Undefined)

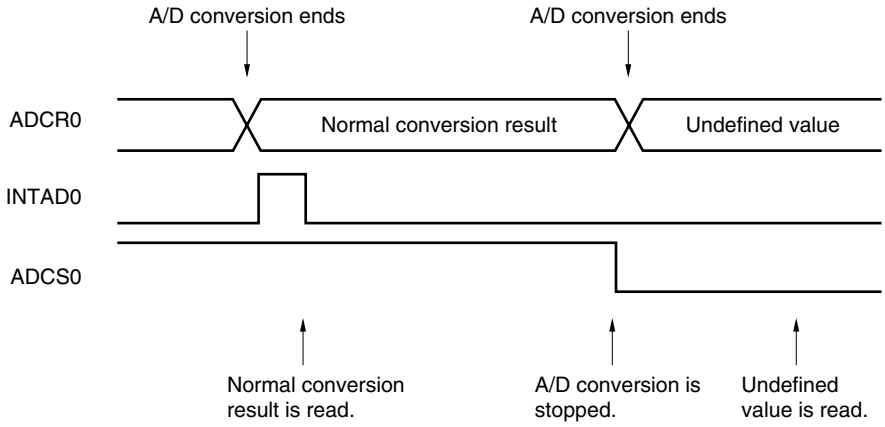
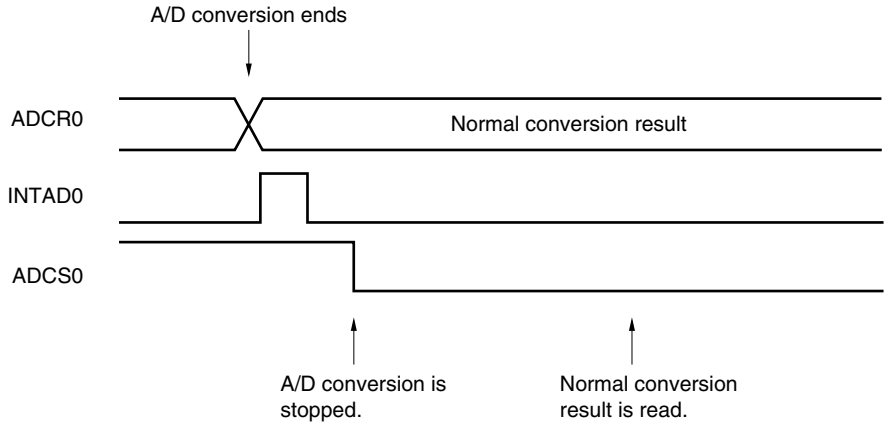


Figure 12-19. Timing of Reading Conversion Result (When Conversion Result Is Normal)



(11) Notes on board design

Locate analog circuits as far away from digital circuits as possible on the board because the analog circuits may be affected by the noise of the digital circuits. In particular, do not cross an analog signal line with a digital signal line, or wire an analog signal line in the vicinity of a digital signal line. Otherwise, the A/D conversion characteristics may be affected by the noise of the digital line.

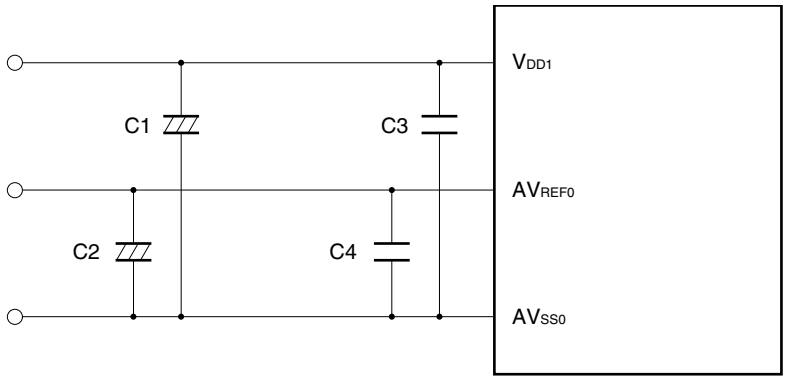
Connect AV_{SS0} and V_{SS0} at one location on the board where the voltages are stable.

(12) V_{DD1} pin and AV_{REF0} pin

Connect a capacitor to the V_{DD1} and AV_{REF0} pins to minimize conversion errors due to noise. If an A/D conversion operation has been stopped and then started, the voltage applied to the V_{DD1} and AV_{REF0} pins becomes unstable, causing the accuracy of the A/D conversion to drop. To prevent this, also connect a capacitor to the V_{DD1} and AV_{REF0} pins.

Figure 12-20 shows an example of connecting capacitors.

Figure 12-20. Example of Connecting Capacitor to V_{DD1} and AV_{REF0} Pins



Remark C1, C2: 4.7 μF to 10 μF (reference value)
C3, C4: 0.01 μF to 0.1 μF (reference value)
Connect C3 and C4 as close to the pin as possible.

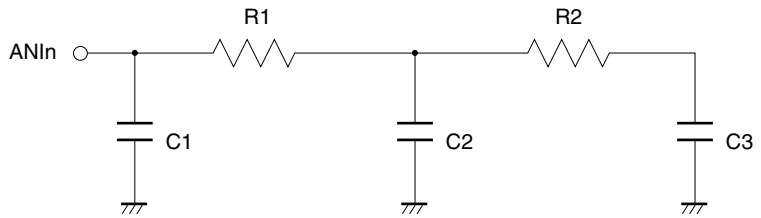
(13) Internal equivalent circuit of ANI0 to ANI9 pins and permissible signal source impedance

To complete sampling within the sampling time with sufficient A/D conversion accuracy, the impedance of the signal source such as a sensor must be sufficiently low. Figure 12-21 shows the internal equivalent circuit of the ANI0 to ANI9 pins.

If the impedance of the signal source is high, connect capacitors with a high capacitance to the ANI0 to ANI9 pins. An example of this is shown in Figure 12-22. In this case, however, the microcontroller cannot follow an analog signal with a high differential coefficient because a lowpass filter is created.

To convert a high-speed analog signal or to convert an analog signal in scan mode, insert a low-impedance buffer.

Figure 12-21. Internal Equivalent Circuit of ANI0 to ANI9 Pins



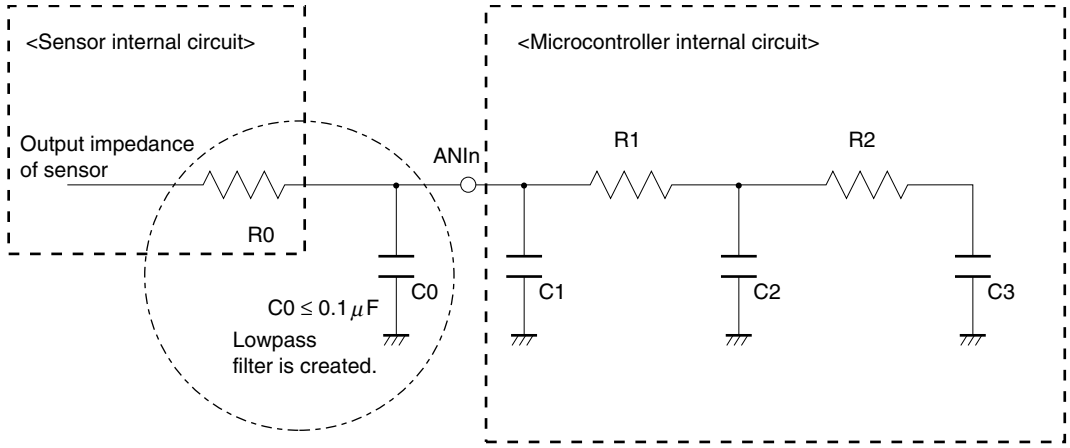
Remark n = 0 to 9

Table 12-2. Resistances and Capacitances of Equivalent Circuit (Reference Values)

V_{DD1}	R1	R2	C1	C2	C3
1.8 V	75 kΩ	30 kΩ	8 pF	4 pF	2 pF
2.7 V	12 kΩ	8 kΩ	8 pF	3 pF	2 pF
4.5 V	4 kΩ	2.7 kΩ	8 pF	1.4 pF	2 pF

Caution The resistances and capacitances in Table 12-2 are not guaranteed values.

Figure 12-22. Example of Connection If Signal Source Impedance Is High



Remark n = 0 to 9

CHAPTER 13 D/A CONVERTER

13.1 D/A Converter Functions

The D/A converter converts the digital input into analog values and consists of one channel of voltage output D/A converters with 8-bit resolution.

The conversion method is a R-2R resistor ladder.

Set DACE of D/A converter mode register 0 (DAM0) to start the D/A conversion. After D/A conversion, the analog voltage is immediately output.

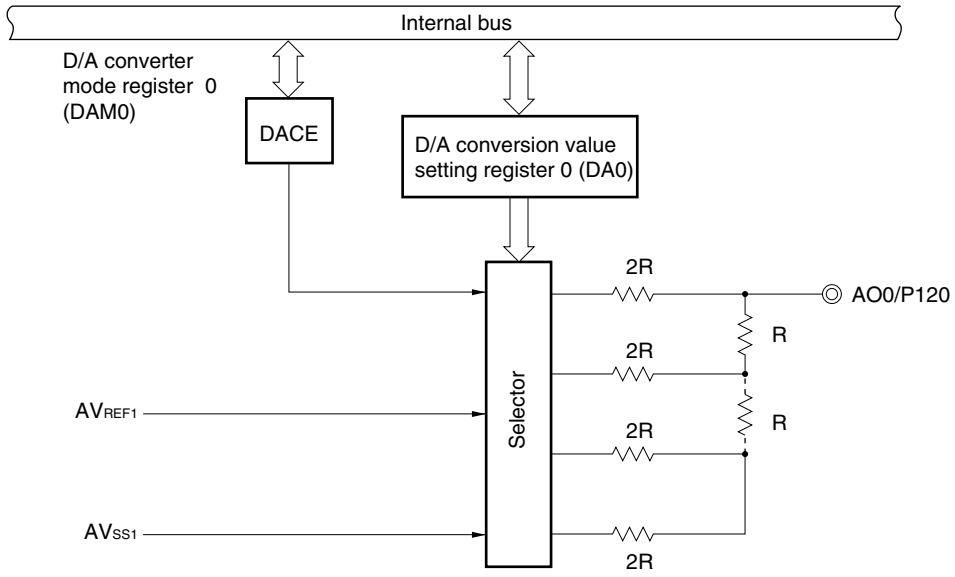
13.2 D/A Converter Configuration

The D/A converter consists of the following hardware.

Table 13-1. D/A Converter Configuration

Item	Configuration
Register	D/A conversion value setting register 0 (DA0)
Control register	D/A converter mode register 0 (DAM0)

Figure 13-1. D/A Converter Block Diagram



(1) D/A conversion value setting register 0 (DA0)

The DA0 register sets the analog voltage that is output to the AO0 pin. The analog voltage is held until new data are set in DA0.

DA0 is set by an 8-bit memory manipulation instruction.

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

The analog voltage output by the AO0 pin is determined by the following equation.

$$\text{AO0 output voltage} = AV_{REF1} \times \frac{DA0}{256}$$

13.3 Register to Control D/A Converter

(1) D/A converter mode register 0 (DAM0)

The D/A converter is controlled by D/A converter mode register 0 (DAM0). This register enables or stops the operation of the D/A converter.

DAM0 is set by a 1-bit or 8-bit memory manipulation instruction.

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

Figure 13-2. D/A Converter Mode Register 0 (DAM0) Format

Address: FF82H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
DAM0	0	0	0	0	0	0	0	DACE

DACE	D/A converter control
0	Stop conversion
1	Enable conversion

Cautions 1. When the D/A converter is used, set the alternate-function port pins to the input mode and disconnect the pull-up resistors.

2. Be sure to set bits 1 to 7 to 0.

3. The output when the D/A converter operation has stopped enters high impedance state.

13.4 D/A Converter Operation

13.4.1 Basic operations of D/A converter

<1> Set the data that corresponds to the analog voltage that is output to AO0/P120 pin of D/A conversion value setting register 0 (DA0).
<2> Set bit 0 (DACE) of D/A converter mode register 0 (DAM0) to start D/A conversion.
<3> After D/A conversion, the analog voltage is immediately output to AO0/P120 pin.
<4> The output analog voltages are held until new data are set in DA0.

Caution Set DACE after data are set in DA0.

13.4.2 Operation during standby mode

D/A converter operation is retained during standby mode.

The values in D/A converter mode register 0 (DAM0) and D/A conversion value setting register 0 (DA0) are retained.

Caution Set bit 0 (DACE) of DAM0 to 0 and stop DA0 before entering standby mode in order to reduce current consumption during standby mode.

13.4.3 Operation at reset

Reset input initializes DA0, stops D/A conversion operation, and put analog output to high-impedance state. In addition, D/A converter mode register 0 (DAM0) and D/A conversion value setting register 0 (DA0) are cleared to 00H.

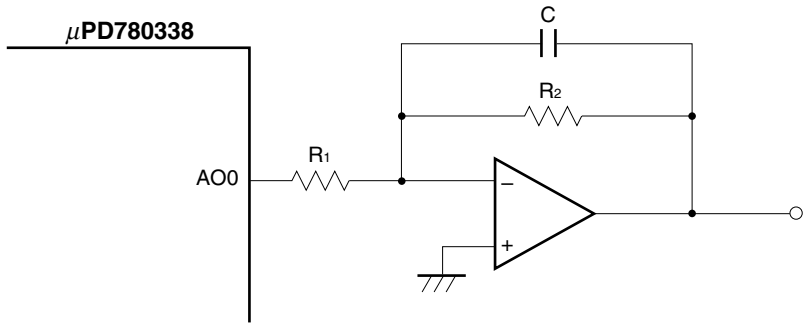
13.5 D/A Converter Cautions

(1) Output impedance of the D/A converter

Since the output impedance of the D/A converter is high, the current cannot be taken from the AO0 pin. If the input impedance of the load is low, insert a buffer amp between the load and the AO0 pin. In addition, use the shortest possible wire from the buffer amp or load (to increase the output impedance). If the wire is long, surround it with a ground pattern.

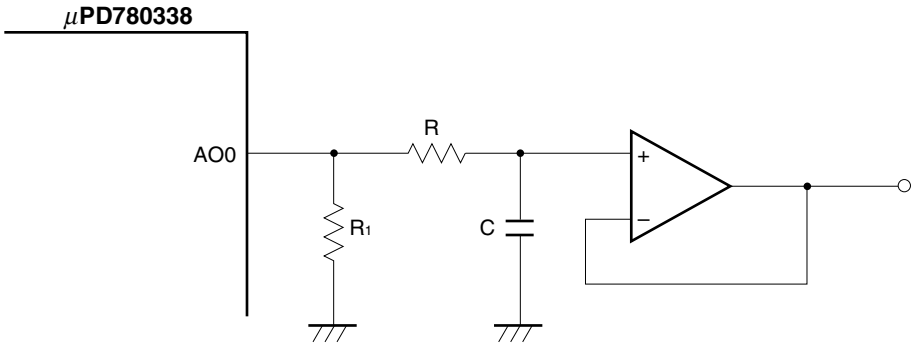
Figure 13-3. Buffer Amp Insertion Example

(a) Inverting Amp



- The input impedance of the buffer amp is R_1.

(b) Voltage follower



- The input impedance of the buffer amp is R_1.
- If there is no R_1 and $\overline{\text{RESET}}$ is low, the output is undefined.

(2) Output voltages of the D/A converters

Since the output voltages of the D/A converter change in stages, use the signals output from the D/A converter after passing them through a low-pass filter.

CHAPTER 14 SERIAL INTERFACE UART0

Serial interface UART0/SIO3 can be used in the asynchronous serial interface (UART) mode or 3-wire serial I/O mode.

Caution Do not enable UART0 and SIO3 at the same time.

14.1 Serial Interface UART0 Functions

Serial interface UART0 has the following two modes.

(1) Operation stop mode

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

For details, see **14.4.1 Operation stop mode**.

(2) Asynchronous serial interface (UART) mode

This mode enables full-duplex operation wherein one byte of data after the start bit is transmitted and received.

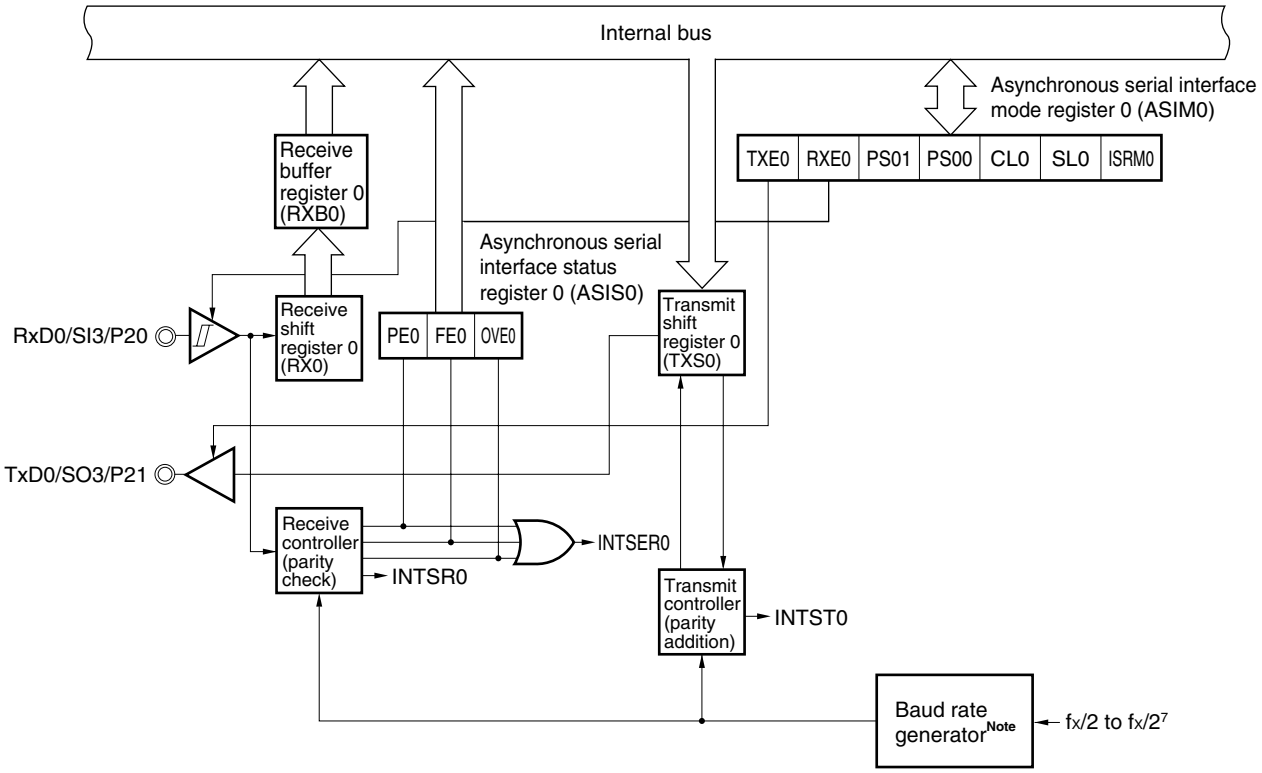
The on-chip baud rate generator dedicated to UART enables communications using a wide range of selectable baud rates.

The UART baud rate generator can also be used to generate a MIDI-standard baud rate (31.25 kbps).

For details, see **14.4.2 Asynchronous serial interface (UART) mode**.

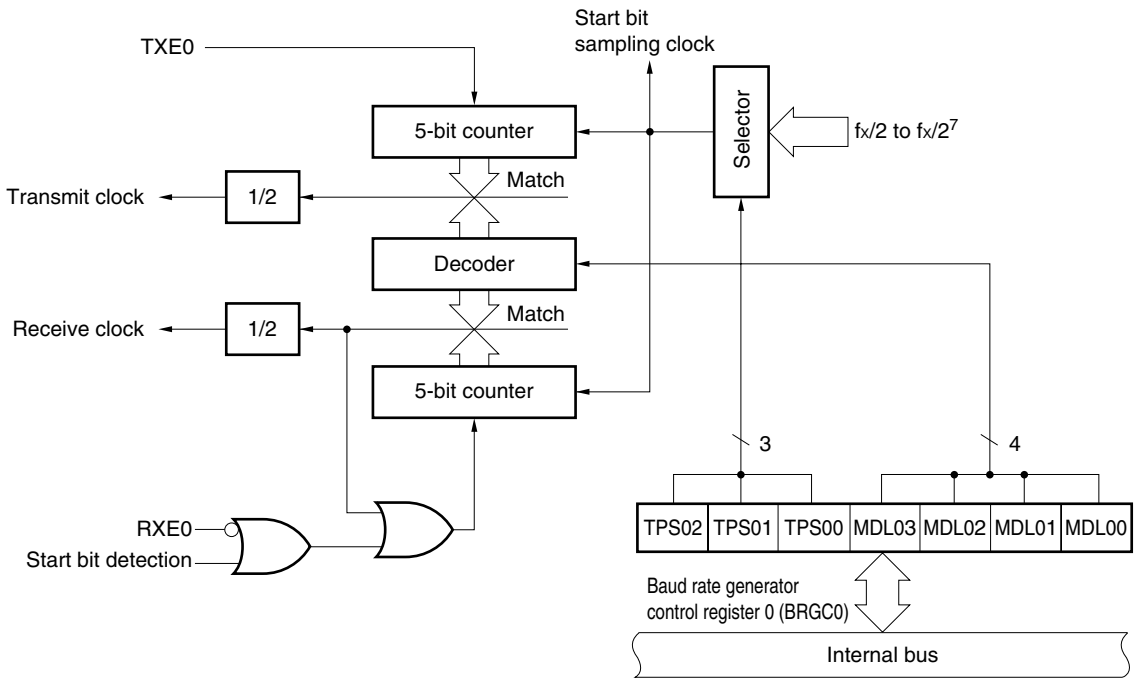
Figure 14-1 shows a block diagram of the serial interface UART0.

Figure 14-1. Serial Interface UART0 Block Diagram



Note For the configuration of the baud rate generator, refer to **Figure 14-2**.

Figure 14-2. Baud Rate Generator Block Diagram



Remark TXE0: Bit 7 of asynchronous serial interface mode register 0 (ASIM0)
RXE0: Bit 6 of asynchronous serial interface mode register 0 (ASIM0)

14.2 Serial Interface UART0 Configuration

Serial interface UART0 consists of the following hardware.

Table 14-1. Serial Interface (UART0) Configuration

Item	Configuration
Registers	Transmit shift register 0 (TXS0) Receive shift register 0 (RX0) Receive buffer register 0 (RXB0)
Control registers	Asynchronous serial interface mode register 0 (ASIM0) Asynchronous serial interface status register 0 (ASIS0) Baud rate generator control register 0 (BRGC0)

(1) Transmit shift register 0 (TXS0)

This is the register for setting transmit data. Data written to TXS0 is transmitted as serial data.

When the data length is set as 7 bits, bits 0 to 6 of the data written to TXS0 are transferred as transmit data.

Writing data to TXS0 starts the transmit operation.

TXS0 can be written by an 8-bit memory manipulation instruction. It cannot be read.

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

Caution Do not write to TXS0 during a transmit operation.
The same address is assigned to TXS0 and the receive buffer register 0 (RXB0). A read operation reads values from RXB0.

(2) Receive shift register 0 (RX0)

This register converts serial data input via the RxD0 pin to parallel data. When one byte of data is received at this register, the receive data is transferred to receive buffer register 0 (RXB0).

RX0 cannot be manipulated directly by a program.

(3) Receive buffer register 0 (RXB0)

This register is used to hold receive data. When one byte of data is received, one byte of new receive data is transferred from the receive shift register (RX0).

When the data length is set as 7 bits, receive data is transferred to bits 0 to 6 of RXB0. In this case, the MSB of RXB0 always becomes 0.

RXB0 can be read by an 8-bit memory manipulation instruction. It cannot be written to.

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

Caution The same address is assigned to RXB0 and the transmit shift register 0 (TXS0). During a write operation, values are written to TXS0.

(4) Transmit controller

The transmit controller controls transmit operations, such as adding a start bit, parity bit, and stop bit to data that is written to transmit shift register 0 (TXS0), based on the values set to asynchronous serial interface mode register 0 (ASIM0).

(5) Receive controller

The receive controller controls receive operations based on the values set to asynchronous serial interface mode register 0 (ASIM0). During a receive operation, it performs error checking, such as for parity errors, and sets various values to asynchronous serial interface status register 0 (ASIS0) according to the type of error that is detected.

14.3 Registers to Control Serial Interface UART0

Serial interface UART0 is controlled by the following three types of registers.

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

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

This is an 8-bit register that controls serial interface UART0's serial transfer operations.

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

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

Figure 14-3 shows the format of ASIM0.

Caution In UART mode, set the port mode register (PMXX) as follows. Set the output latch of the port set to output mode (PMXX = 0) to 0.

- **During receive operation**
  Set P20 (RxD0) to input mode (PM20 = 1)
- **During transmit operation**
  Set P21 (TxD0) to output mode (PM21 = 0)
- **During transmit/receive operation**
  Set P20 to input mode, and P21 to output mode

Figure 14-3. Asynchronous Serial Interface Mode Register 0 (ASIM0) Format

Address: FFA0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ASIM0	TXE0	RXE0	PS01	PS00	CL0	SL0	ISRM0	0

TXE0	RXE0	Operation mode	RxD0/P20 pin function	TxD0/P21 pin function
0	0	Operation stop	Port function (P20)	Port function (P21)
0	1	UART mode (receive only)	Serial function (RxD0)	
1	0	UART mode (transmit only)	Port function (P20)	Serial function (TxD0)
1	1	UART mode (transmit and receive)	Serial function (RxD0)	

PS01	PS00	Parity bit specification
0	0	No parity
0	1	Zero parity always added during transmission No parity detection during reception (parity errors do not occur)
1	0	Odd parity
1	1	Even parity

CL0	Character length specification
0	7 bits
1	8 bits

SL0	Stop bit length specification for transmit data
0	1 bit
1	2 bits

ISRM0	Receive completion interrupt control when error occurs
0	Receive completion interrupt request is issued when an error occurs
1	Receive completion interrupt request is not issued when an error occurs

Caution Do not switch the operation mode until the current serial transmit/receive operation has stopped.

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

When a receive error occurs during UART mode, this register indicates the type of error.

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

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

Figure 14-4. Asynchronous Serial Interface Status Register 0 (ASIS0) Format

Address: FFA1H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS0	0	0	0	0	0	PE0	FE0	OVE0

PE0	Parity error flag
0	No parity error
1	Parity error (Transmit data parity not matched)

FE0	Framing error flag
0	No framing error
1	Framing error[Note 1] (Stop bit not detected)

OVE0	Overrun error flag
0	No overrun error
1	Overrun error[Note 2] (Next receive operation was completed before data was read from receive buffer register 0 (RXB0))

Notes 1. Even if a stop bit length is set to 2 bits by setting bit 2 (SL0) in asynchronous serial interface mode register 0 (ASIM0), stop bit detection during a receive operation only applies to a stop bit length of 1 bit.

2. Be sure to read the contents of receive buffer register 0 (RXB0) when an overrun error has occurred. Until the contents of RXB0 are read, further overrun errors will occur when receiving data.

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

This register sets the serial clock for serial interface.

BRGC0 is set by an 8-bit memory manipulation instruction.

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

Figure 14-5 shows the format of BRGC0.

Figure 14-5. Baud Rate Generator Control Register 0 (BRGC0) Format

Address: FFA2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
BRGC0	0	TPS02	TPS01	TPS00	MDL03	MDL02	MDL01	MDL00

TPS02	TPS01	TPS00	Source clock selection for 5-bit counter	n
0	0	0	Setting prohibited	—
0	0	1	$f_X/2$	1
0	1	0	$f_X/2^2$	2
0	1	1	$f_X/2^3$	3
1	0	0	$f_X/2^4$	4
1	0	1	$f_X/2^5$	5
1	1	0	$f_X/2^6$	6
1	1	1	$f_X/2^7$	7

MDL03	MDL02	MDL01	MDL00	Input clock selection for baud rate generator	k
0	0	0	0	$f_{SCK}/16$	0
0	0	0	1	$f_{SCK}/17$	1
0	0	1	0	$f_{SCK}/18$	2
0	0	1	1	$f_{SCK}/19$	3
0	1	0	0	$f_{SCK}/20$	4
0	1	0	1	$f_{SCK}/21$	5
0	1	1	0	$f_{SCK}/22$	6
0	1	1	1	$f_{SCK}/23$	7
1	0	0	0	$f_{SCK}/24$	8
1	0	0	1	$f_{SCK}/25$	9
1	0	1	0	$f_{SCK}/26$	10
1	0	1	1	$f_{SCK}/27$	11
1	1	0	0	$f_{SCK}/28$	12
1	1	0	1	$f_{SCK}/29$	13
1	1	1	0	$f_{SCK}/30$	14
1	1	1	1	Setting prohibited	—

Caution Writing to BRGC0 during a communication operation may cause abnormal output from the baud rate generator and disable further communication operations. Therefore, do not write to BRGC0 during a communication operation.

Remarks 1. f_{SCK}: Source clock for 5-bit counter

2. n: Value set via TPS00 to TPS02 ($1 \leq n \leq 7$)

3. k: Value set via MDL00 to MDL03 ($0 \leq k \leq 14$)

14.4 Serial Interface UART0 Operations

This section explains the two modes of serial interface UART0.

14.4.1 Operation stop mode

Because serial transfer is not performed during this mode, the power consumption can be reduced.
In addition, pins can be used as normal ports.

(1) Register settings

Operation stop mode is set by asynchronous serial interface mode register 0 (ASIM0).
ASIM0 is set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets the value of this register to 00H.

Address: FFA0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ASIM0	TXE0	RXE0	PS01	PS00	CL0	SL0	ISRM0	0

TXE0	RXE0	Operation mode	RxD0/P20 pin function	TxD0/P21 pin function
0	0	Operation stop	Port function (P20)	Port function (P21)
0	1	UART mode (receive only)	Serial function (RxD0)	
1	0	UART mode (transmit only)	Port function (P20)	Serial function (TxD0)
1	1	UART mode (transmit and receive)	Serial function (RxD0)	

Caution Do not switch the operation mode until the current serial transmit/receive operation has stopped.

14.4.2 Asynchronous serial interface (UART) mode

This mode enables full-duplex operation wherein one byte of data after the start bit is transmitted or received.

The on-chip baud rate generator dedicated to UART enables communications using a wide range of selectable baud rates.

The UART baud rate generator can also be used to generate a MIDI-standard baud rate (31.25 kbps).

(1) Register settings

UART mode settings are performed by asynchronous serial interface mode register 0 (ASIM0), asynchronous serial interface status register 0 (ASIS0), and baud rate generator control register 0 (BRGC0).

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

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

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

Caution In UART mode, set the port mode register (PMXX) as follows. Set the output latch of the port set to output mode (PMXX = 0) to 0.

- **During receive operation**
  Set P20 (RxD0) to input mode (PM20 = 1)
- **During transmit operation**
  Set P21 (TxD0) to output mode (PM21 = 0)
- **During transmit/receive operation**
  Set P20 to input mode, and P21 to output mode

Address: FFA0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ASIM0	TXE0	RXE0	PS01	PS00	CL0	SL0	ISRM0	0

TXE0	RXE0	Operation mode	RxD0/P20 pin function	TxD0/P21 pin function
0	0	Operation stop	Port function (P20)	Port function (P21)
0	1	UART mode (receive only)	Serial function (RxD0)	
1	0	UART mode (transmit only)	Port function (P20)	Serial function (TxD0)
1	1	UART mode (transmit and receive)	Serial function (RxD0)	

PS01	PS00	Parity bit specification
0	0	No parity
0	1	Zero parity always added during transmission No parity detection during reception (parity errors do not occur)
1	0	Odd parity
1	1	Even parity

CL0	Character length specification
0	7 bits
1	8 bits

SL0	Stop bit length specification for transmit data
0	1 bit
1	2 bits

ISRM0	Receive completion interrupt control when error occurs
0	Receive completion interrupt request is issued when an error occurs
1	Receive completion interrupt request is not issued when an error occurs

Caution Do not switch the operation mode until the current serial transmit/receive operation has stopped.

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

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

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

Address: FFA1H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS0	0	0	0	0	0	PE0	FE0	OVE0

PE0	Parity error flag
0	No parity error
1	Parity error (Transmit data parity not matched)

FE0	Framing error flag
0	No framing error
1	Framing error**Note 1** (Stop bit not detected)

OVE0	Overrun error flag
0	No overrun error
1	Overrun error**Note 2** (Next receive operation was completed before data was read from receive buffer register 0 (RXB0))

Notes 1. Even if a stop bit length is set to 2 bits by setting bit 2 (SL0) in asynchronous serial interface mode register 0 (ASIM0), stop bit detection during a receive operation only applies to a stop bit length of 1 bit.

2. Be sure to read the contents of receive buffer register 0 (RXB0) when an overrun error has occurred.

Until the contents of RXB0 are read, further overrun errors will occur when receiving data.

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

BRGC0 is set by an 8-bit memory manipulation instruction.

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

Address: FFA2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
BRGC0	0	TPS02	TPS01	TPS00	MDL03	MDL02	MDL01	MDL00

TPS02	TPS01	TPS00	Source clock selection for 5-bit counter	n
0	0	0	Setting prohibited	—
0	0	1	$f_X/2$	1
0	1	0	$f_X/2^2$	2
0	1	1	$f_X/2^3$	3
1	0	0	$f_X/2^4$	4
1	0	1	$f_X/2^5$	5
1	1	0	$f_X/2^6$	6
1	1	1	$f_X/2^7$	7

MDL03	MDL02	MDL01	MDL00	Input clock selection for baud rate generator	k
0	0	0	0	$f_{SCK}/16$	0
0	0	0	1	$f_{SCK}/17$	1
0	0	1	0	$f_{SCK}/18$	2
0	0	1	1	$f_{SCK}/19$	3
0	1	0	0	$f_{SCK}/20$	4
0	1	0	1	$f_{SCK}/21$	5
0	1	1	0	$f_{SCK}/22$	6
0	1	1	1	$f_{SCK}/23$	7
1	0	0	0	$f_{SCK}/24$	8
1	0	0	1	$f_{SCK}/25$	9
1	0	1	0	$f_{SCK}/26$	10
1	0	1	1	$f_{SCK}/27$	11
1	1	0	0	$f_{SCK}/28$	12
1	1	0	1	$f_{SCK}/29$	13
1	1	1	0	$f_{SCK}/30$	14
1	1	1	1	Setting prohibited	—

Caution Writing to BRGC0 during a communication operation may cause abnormal output from the baud rate generator and disable further communication operations. Therefore, do not write to BRGC0 during a communication operation.

Remarks 1. f_{SCK}: Source clock for 5-bit counter

2. n: Value set via TPS00 to TPS02 ($1 \leq n \leq 7$)

3. k: Value set via MDL00 to MDL03 ($0 \leq k \leq 14$)

The transmit/receive clock that is used to generate the baud rate is obtained by dividing the main system clock.

- Transmit/receive clock generation for baud rate by using main system clock
  The main system clock is divided to generate the transmit/receive clock. The baud rate generated from the main system clock is determined according to the following formula.

$$[\text{Baud rate}] = \frac{f_X}{2^{n+1}(k + 16)} \quad [\text{Hz}]$$

  f_X: Main system clock oscillation frequency
  n: Value set via TPS00 to TPS02 ($1 \leq n \leq 7$)
  For details, see **Table 14-2**.
  k: Value set via MDL00 to MDL03 ($0 \leq k \leq 14$)

Table 14-2 shows the relationship between the 5-bit counter's source clock assigned to bits 4 to 6 (TPS00 to TPS02) of BRGC0 and the "n" value in the above formula and Table 14-3 shows the relationship between the main system clock and the baud rate.

Table 14-2. Relationship Between 5-Bit Counter's Source Clock and "n" Value

TPS02	TPS01	TPS00	5-Bit Counter's Source Clock Selected	n
0	0	0	Setting prohibited	—
0	0	1	$f_X/2$	1
0	1	0	$f_X/2^2$	2
0	1	1	$f_X/2^3$	3
1	0	0	$f_X/2^4$	4
1	0	1	$f_X/2^5$	5
1	1	0	$f_X/2^6$	6
1	1	1	$f_X/2^7$	7

Remark f_X: Main system clock oscillation frequency

Table 14-3. Relationship Between Main System Clock and Baud Rate

Baud Rate (bps)	fx = 10 MHz		fx = 9.8304 MHz		fx = 8.386 MHz		fx = 8 MHz	
	BRGC0	ERR (%)	BRGC0	ERR (%)	BRGC0	ERR (%)	BRGC0	ERR (%)
600	–	–	–	–	–	–	–	–
1,200	–	–	–	–	7BH	1.10	7AH	0.16
2,400	70H	1.73	70H	0.00	6BH	1.10	6AH	0.16
4,800	60H	1.73	60H	0.00	5BH	1.10	5AH	0.16
9,600	50H	1.73	50H	0.00	4BH	1.10	4AH	0.16
19,200	40H	1.73	40H	0.00	3BH	1.10	3AH	0.16
31,250	34H	0.00	34H	–1.70	31H	–3.14	30H	0.00
38,400	30H	1.73	30H	0.00	2BH	1.10	2AH	0.16
76,800	20H	1.73	20H	0.00	1BH	1.10	1AH	0.16
115,200	16H	–1.36	16H	–3.03	12H	1.10	11H	2.12
153,600	10H	1.73	10H	0.00	–	–	–	–

Baud Rate (bps)	fx = 7.3728 MHz		fx = 5 MHz		fx = 4.194304 MHz	
	BRGC0	ERR (%)	BRGC0	ERR (%)	BRGC0	ERR (%)
600	–	–	–	–	7BH	1.14
1,200	78H	0.00	70H	1.73	6BH	1.14
2,400	68H	0.00	60H	1.73	5BH	1.14
4,800	58H	0.00	50H	1.73	4BH	1.14
9,600	48H	0.00	40H	1.73	3BH	1.14
19,200	38H	0.00	30H	1.73	2BH	1.14
31,250	2DH	1.69	24H	0.00	21H	–1.31
38,400	28H	0.00	20H	1.73	1BH	1.14
76,800	18H	0.00	10H	1.73	–	–
115,200	10H	0.00	–	–	–	–
153,600	–	–	–	–	–	–

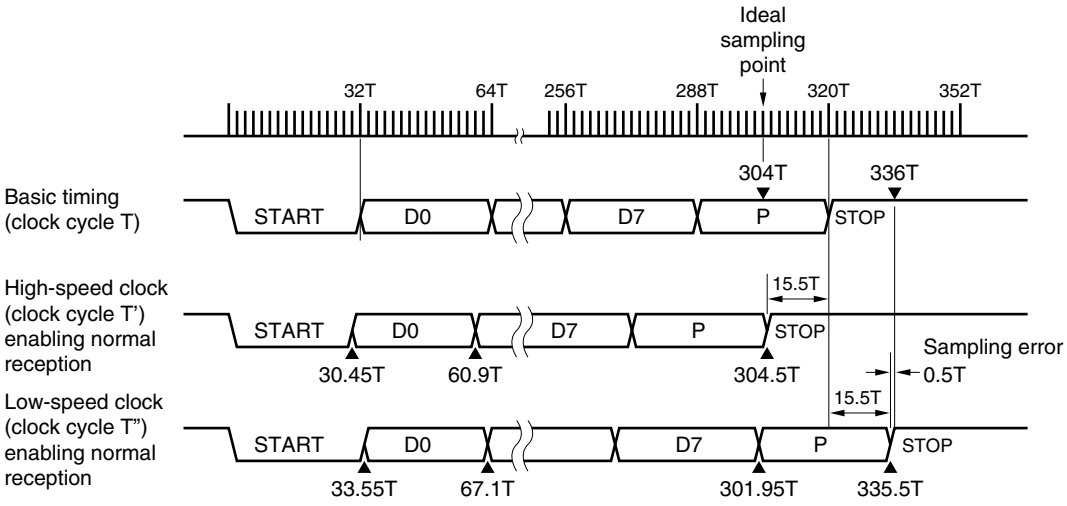
Remark fx: Main system clock oscillation frequency
n: Value set via TPS00 to TPS02 ($1 \leq n \leq 7$)
k: Value set via MDL00 to MDL03 ($0 \leq k \leq 14$)

- **Error tolerance range for baud rate**

  The tolerance range for the baud rate depends on the number of bits per frame and the counter's division rate [1/(16 + k)].

  Figure 14-6 shows an example of a baud rate error tolerance range.

Figure 14-6. Baud Rate Error Tolerance (When k = 0), Including Sampling Errors



Remark T: 5-bit counter's source clock cycle

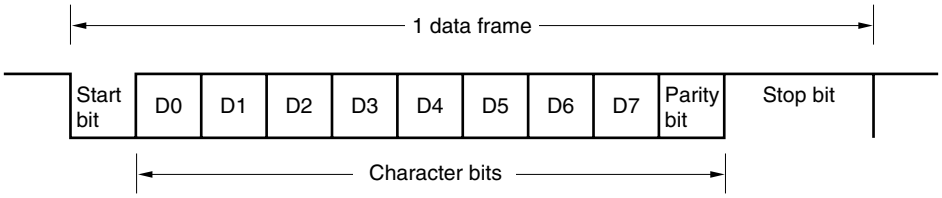
$$\text{Baud rate error tolerance (when k = 0)} = \frac{\pm 15.5}{320} \times 100 = 4.8438\ (\%)$$

(2) Communication operations

(a) Data format

Figure 14-7 shows the format of the transmit/receive data.

Figure 14-7. Format of Transmit/Receive Data in Asynchronous Serial Interface



1 data frame consists of the following bits.

- Start bit ..... 1 bit
- Character bits ... 7 bits or 8 bits
- Parity bit ..... Even parity, odd parity, zero parity, or no parity
- Stop bit(s) ..... 1 bit or 2 bits

Asynchronous serial interface mode register 0 (ASIM0) is used to set the character bit length, parity selection, and stop bit length within each data frame.

When “7 bits” is selected as the number of character bits, only the lower 7 bits (bits 0 to 6) are valid, so that during a transmission the highest bit (bit 7) is ignored and during reception the highest bit (bit 7) must be set to “0”.

ASIM0 and baud rate generator control register 0 (BRGC0) are used to set the serial transfer rate.

If a receive error occurs, information about the receive error can be recognized by reading asynchronous serial interface status register 0 (ASIS0).

(b) Parity types and operations

The parity bit is used to detect bit errors in communication data. Usually, the same type of parity bit is used by the transmitting and receiving sides. When odd parity or even parity is set, errors in the parity bit (the odd-number bit) can be detected. When zero parity or no parity is set, errors are not detected.

(i) Even parity

- During transmission
  The number of bits in transmit data that includes a parity bit is controlled so that there are an even number of bits whose value is 1. The value of the parity bit is as follows.

  If the transmit data contains an odd number of bits whose value is 1: the parity bit is “1”
  If the transmit data contains an even number of bits whose value is 1: the parity bit is “0”

- During reception
  The number of bits whose value is 1 is counted among the receive data that include a parity bit, and a parity error occurs when the counted result is an odd number.

(ii) Odd parity

- During transmission
  The number of bits in transmit data that includes a parity bit is controlled so that there is an odd number of bits whose value is 1. The value of the parity bit is as follows.

  If the transmit data contains an odd number of bits whose value is 1: the parity bit is “0”
  If the transmit data contains an even number of bits whose value is 1: the parity bit is “1”

- During reception
  The number of bits whose value is 1 is counted among the receive data that include a parity bit, and a parity error occurs when the counted result is an even number.

(iii) Zero parity

During transmission, the parity bit is set to “0” regardless of the transmit data.
During reception, the parity bit is not checked. Therefore, no parity errors will occur regardless of whether the parity bit is a “0” or a “1”.

(iv) No parity

No parity bit is added to the transmit data.
During reception, receive data is regarded as having no parity bit. Since there is no parity bit, no parity errors will occur.

(c) Transmission

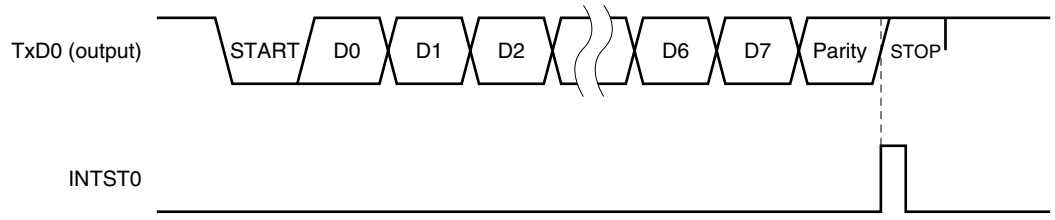
The transmit operation is enabled if bit 7 (TXE0) of asynchronous serial interface mode register 0 (ASIM0) is set to 1, the transmit operation is started when transmit data is written to transmit shift register 0 (TXS0). A start bit, parity bit, and stop bit(s) are automatically added to the data.

Starting the transmit operation shifts out the data in TXS0, thereby emptying TXS0, after which a transmit completion interrupt request (INTST0) is issued.

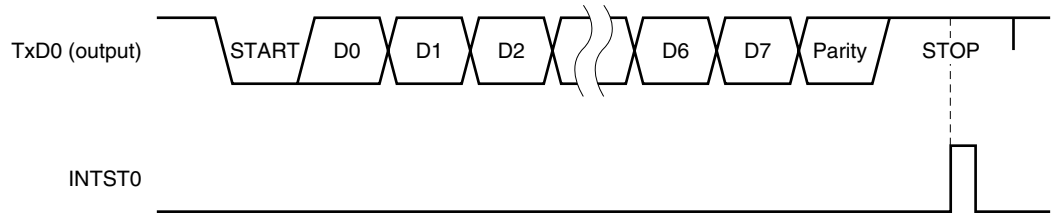
The timing of the transmit completion interrupt request is shown in Figure 14-8.

Figure 14-8. Timing of Asynchronous Serial Interface Transmit Completion Interrupt Request

(i) Stop bit length: 1 bit



(ii) Stop bit length: 2 bits



Caution Do not rewrite to asynchronous serial interface mode register 0 (ASIM0) during a transmit operation. Rewriting ASIM0 register during a transmit operation may disable further transmit operations (in such cases, enter a $\overline{\text{RESET}}$ to restore normal operation).

Whether or not a transmit operation is in progress can be determined via software using the transmit completion interrupt request (INTST0) or the interrupt request flag (STIF0) that is set by INTST0.

(d) Reception

The receive operation is enabled when “1” is set to bit 6 (RXE0) of asynchronous serial interface mode register 0 (ASIM0), and input via the RxD0 pin is sampled.

The serial clock specified by BRGC0 is used to sample the RxD0 pin.

When the RxD0 pin goes low, the 5-bit counter of the baud rate generator begins counting and the start timing signal for data sampling is output when half of the specified baud rate time has elapsed. If sampling the RxD0 pin input with this start timing signal yields a low-level result, a start bit is recognized, after which the 5-bit counter is initialized and starts counting and data sampling begins. After the start bit is recognized, the character data, parity bit, and one-bit stop bit are detected, at which point reception of one data frame is completed.

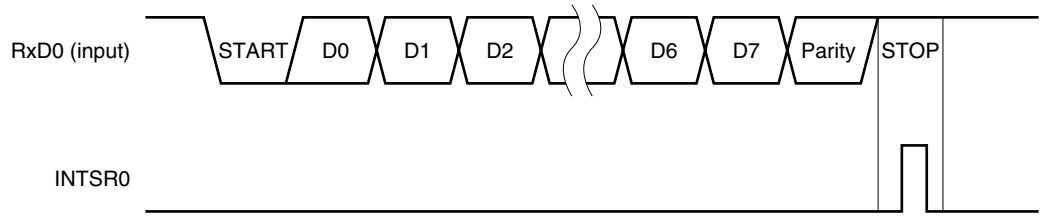
Once reception of one data frame is completed, the receive data in the shift register is transferred to receive buffer register 0 (RXB0) and a receive completion interrupt request (INTSR0) occurs.

Even if an error has occurred, the receive data in which the error occurred is still transferred to RXB0. When ASIM0 bit 1 (ISRM0) is cleared (0) upon occurrence of an error, INTSR0 occurs (see **Figure 14-10**). When ISRM0 bit is set (1), INTSR0 does not occur.

If the RXE0 bit is reset (to “0”) during a receive operation, the receive operation is stopped immediately. At this time, the contents of RXB0 and ASIS0 do not change, nor does INTSR0 or INTSER0 occur.

Figure 14-9 shows the timing of the asynchronous serial interface receive completion interrupt request.

Figure 14-9. Timing of Asynchronous Serial Interface Receive Completion Interrupt Request



Caution Be sure to read the contents of receive buffer register 0 (RXB0) even when a receive error has occurred. Overrun errors will occur during the next data receive operations and the receive error status will remain until the contents of RXB0 are read.

(e) Receive errors

Three types of errors can occur during a receive operation: parity error, framing error, or overrun error. If, as the result of data reception, an error flag is set to asynchronous serial interface status register 0 (ASIS0), a receive error interrupt request (INTSER0) will occur. Receive error interrupt requests are generated before receive completion interrupt request (INTSR0). Table 14-4 lists the causes behind receive errors.

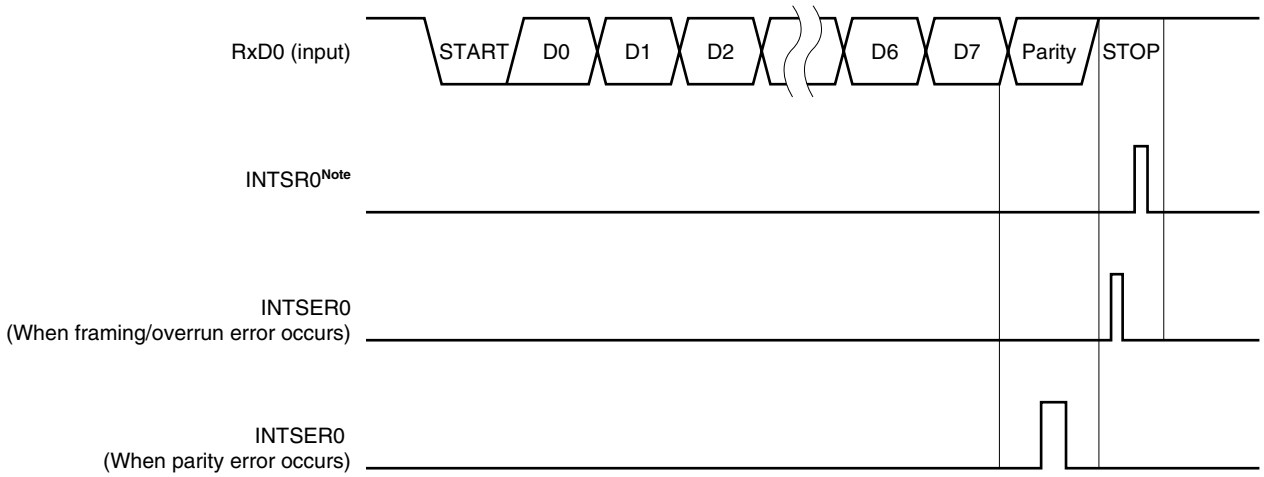
As part of receive error interrupt request (INTSER0) servicing, the contents of ASIS0 can be read to determine which type of error occurred during the receive operation (see **Table 14-4** and **Figure 14-10**).

The contents of ASIS0 are reset (to "0") when receive buffer register 0 (RXB0) is read or when the next data is received (if the next data contains an error, its error flag will be set).

Table 14-4. Causes of Receive Errors

Receive Error	Cause	ASIS0 Value
Parity error	Parity specified during transmission does not match parity of receive data	04H
Framing error	Stop bit was not detected	02H
Overrun error	Reception of the next data was completed before data was read from receive buffer register 0 (RXB0)	01H

Figure 14-10. Receive Error Timing



Note If a receive error occurs when the ISRM0 bit has been set (1), INTSR0 does not occur.

Cautions 1. The contents of asynchronous serial interface status register 0 (ASIS0) are reset (to "0") when receive buffer register 0 (RXB0) is read or when the next data is received. To obtain information about the error, be sure to read the contents of ASIS0 before reading RXB0.

2. Be sure to read the contents of receive buffer register 0 (RXB0) even when a receive error has occurred. Overrun errors will occur during the next data receive operations and the receive error status will remain until the contents of RXB0 are read.

CHAPTER 15 SERIAL INTERFACE SIO3

Serial interface UART0/SIO3 can be used in the asynchronous serial interface (UART) mode or 3-wire serial I/O mode.

Caution Do not enable UART0 and SIO3 at the same time.

15.1 Serial Interface SIO3 Functions

Serial interface SIO3 has the following two modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed. For details, see **15.4.1 Operation stop mode**.

(2) 3-wire serial I/O mode (fixed as MSB first)

This is an 8-bit data transfer mode using three lines: a serial clock line ($\overline{\text{SCK3}}$), serial output line (SO3), and serial input line (SI3).

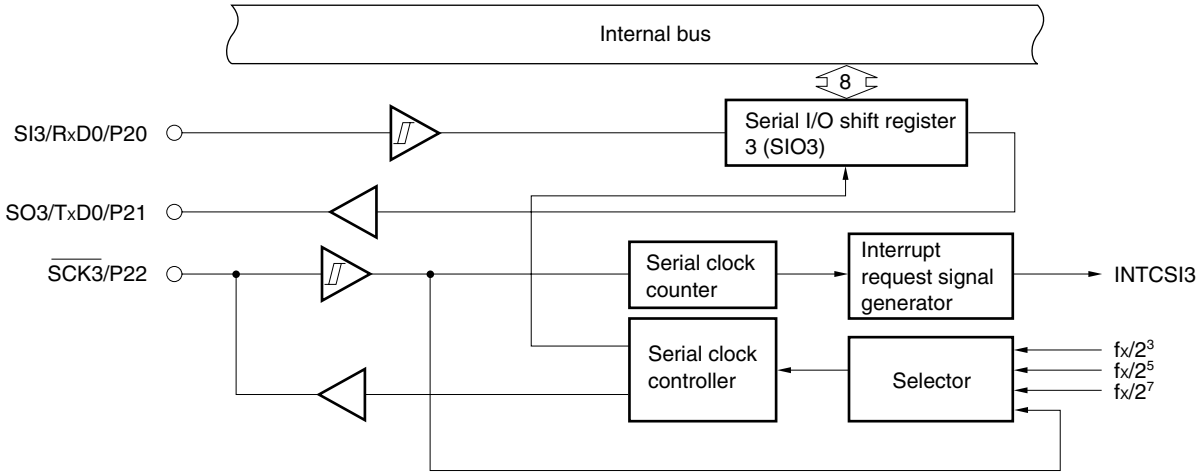
Since simultaneous transmit and receive operations are enabled in 3-wire serial I/O mode, the processing time for data transfers is reduced.

The first bit of the serial transferred 8-bit data is fixed as the MSB.

3-wire serial I/O mode is useful for connection to a peripheral I/O incorporating a clocked serial interface, or a display controller, etc. For details, see **15.4.2 3-wire serial I/O mode**.

Figure 15-1 shows a block diagram of the serial interface SIO3.

Figure 15-1. Serial Interface SIO3 Block Diagram



15.2 Serial Interface SIO3 Configuration

Serial interface SIO3 consists of the following hardware.

Table 15-1. Serial Interface SIO3 Configuration

Item	Configuration
Register	Serial I/O shift register 3 (SIO3)
Control register	Serial operation mode register 3 (CSIM3)

(1) Serial I/O shift register 3 (SIO3)

This is an 8-bit register that performs parallel-serial conversion and serial transmit/receive (shift operations) synchronized with the serial clock.

SIO3 is set by an 8-bit memory manipulation instruction.

When “1” is set to bit 7 (CSIE3) of serial operation mode register 3 (CSIM3), a serial operation can be started by writing data to or reading data from SIO3.

When transmitting, data written to SIO3 is output to the serial output (SO3).

When receiving, data is read from the serial input (SI3) and written to SIO3.

The value of this register is undefined when $\overline{\text{RESET}}$ is input.

Caution Do not access SIO3 during a transfer operation unless the access is triggered by a transfer start (read operation is disabled when MODE = 0 and write operation is disabled when MODE = 1).

15.3 Register to Control Serial Interface SIO3

Serial interface SIO3 is controlled by serial operation mode register 3 (CSIM3).

(1) Serial operation mode register 3 (CSIM3)

This register is used to enable or disable SIO3's serial clock, operation modes, and specific operations.

CSIM3 is set by a 1-bit or 8-bit memory manipulation instruction.

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

Caution In 3-wire serial I/O mode, set the port mode register (PMXX) as follows. Set the output latch of the port set to output mode (PMXX = 0) to 0.

During serial clock output (master transmission or master reception)	PM22 = 0; Sets P22 ($\overline{\text{SCK3}}$) to output mode P22 = 0; Sets output latch of P22 to 0
During serial clock input (slave transmission or slave reception)	PM22 = 1; Sets P22 ($\overline{\text{SCK3}}$) to input mode
Transmit/receive mode	PM21 = 0; Sets P21 (SO3) to output mode P21 = 0; Sets output latch of P21 to 0
Receive mode	PM20 = 1; Sets P20 (SI3) to input mode

Figure 15-2. Serial Operation Mode Register 3 (CSIM3) Format

Address: FFAFH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIM3	CSIE3	0	0	0	0	MODE	SCL31	SCL30

CSIE3	Enable/disable specification for SIO3		
	Shift register operation	Serial counter	Port
0	Operation stop	Clear	Port function**Note 1**
1	Operation enable	Count operation enable	Serial function + port function**Note 2**

MODE	Transfer operation modes and flags		
	Operation mode	Transfer start trigger	SO3 output
0	Transmit/transmit and receive mode	Write to SIO3	Normal output
1	Receive-only mode	Read from SIO3	Fixed at low level

SCL31	SCL30	Clock selection
0	0	External clock input to $\overline{\text{SCK3}}$
0	1	$f_X/2^3$ (1.25 MHz)
1	0	$f_X/2^5$ (312.5 kHz)
1	1	$f_X/2^7$ (78.125 kHz)

Notes 1. When CSIE3 = 0 (SIO3 operation stop status), the pins SI3, SO3, and $\overline{\text{SCK3}}$ can be used for port functions.

2. When CSIE3 = 1 (SIO3 operation enabled status), the SI3 pin can be used as a port pin if only the transmit function is used, and the SO3 pin can be used as a port pin if only the receive-only mode is used.

Remarks 1. f_X: Main system clock oscillation frequency

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

15.4 Serial Interface SIO3 Operations

This section explains the two modes of serial interface SIO3.

15.4.1 Operation stop mode

Because serial transfer is not performed in this mode, the power consumption can be reduced.
In addition, pins can be used as normal I/O ports.

(1) Register settings

Operation stop mode is set by serial operation mode register 3 (CSIM3).
CSIM3 is set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets the value of this register to 00H.

Address: FFAFH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIM3	CSIE3	0	0	0	0	MODE	SCL31	SCL30

CSIE3	SIO3 operation enable/disable specification		
	Shift register operation	Serial counter	Port
0	Operation stop	Clear	Port function[Note 1]
1	Operation enable	Count operation enable	Serial function + port function[Note 2]

Notes 1. When CSIE3 = 0 (SIO3 operation stop status), the pins SI3, SO3, and $\overline{\text{SCK3}}$ can be used for port functions.

2. When CSIE3 = 1 (SIO3 operation enabled status), the SI3 pin can be used as a port pin if only the transmit function is used, and the SO3 pin can be used as a port pin if only the receive-only mode is used.

15.4.2 3-wire serial I/O mode

The 3-wire serial I/O mode is useful for connection to a peripheral I/O incorporating a clocked serial interface, a display controller, etc.

This mode executes data transfers via three lines: a serial clock line ($\overline{SCK3}$), serial output line (SO3), and serial input line (SI3).

(1) Register settings

3-wire serial I/O mode is set by serial operation mode register 3 (CSIM3).

CSIM3 is set by a 1-bit or 8-bit memory manipulation instruction.

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

Caution In 3-wire serial I/O mode, set the port mode register (PMXX) as follows. Set the output latch of the port set to output mode (PMXX = 0) to 0.

During serial clock output (master transmission or master reception)	PM22 = 0; Sets P22 ($\overline{SCK3}$) to output mode P22 = 0; Sets output latch of P22 to 0
During serial clock input (slave transmission or slave reception)	PM22 = 1; Sets P22 ($\overline{SCK3}$) to input mode
Transmit/receive mode	PM21 = 0; Sets P21 (SO3) to output mode P21 = 0; Sets output latch of P21 to 0
Receive mode	PM20 = 1; Sets P20 (SI3) to input mode

Address: FFAFH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIM3	CSIE3	0	0	0	0	MODE	SCL31	SCL30

CSIE3	Enable/disable specification for SIO3		
	Shift register operation	Serial counter	Port
0	Operation stop	Clear	Port function**Note 1**
1	Operation enable	Count operation enable	Serial function + port function**Note 2**

MODE	Transfer operation modes and flags		
	Operation mode	Transfer start trigger	SO3 output
0	Transmit/transmit and receive mode	Write to SIO3	Normal output
1	Receive-only mode	Read from SIO3	Fixed at low level

SCL31	SCL30	Clock selection
0	0	External clock input to $\overline{\text{SCK3}}$
0	1	$f_X/2^3$ (1.25 MHz)
1	0	$f_X/2^5$ (312.5 kHz)
1	1	$f_X/2^7$ (78.125 kHz)

Notes 1. When CSIE3 = 0 (SIO3 operation stop status), the pins SI3, SO3, and $\overline{\text{SCK3}}$ can be used for port functions.

2. When CSIE3 = 1 (SIO3 operation enabled status), the SI3 pin can be used as a port pin if only the transmit function is used, and the SO3 pin can be used as a port pin if only the receive-only mode is used.

Remarks 1. f_X: Main system clock oscillation frequency

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

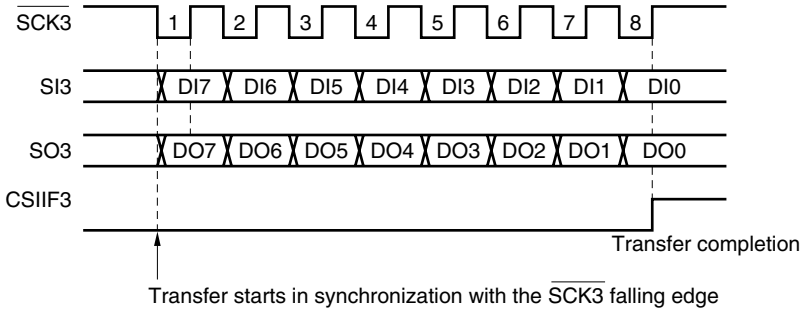
(2) Communication operations

In the 3-wire serial I/O mode, data is transmitted and received in 8-bit units. Each bit of data is transmitted or received in synchronization with the serial clock.

Serial I/O shift register 3 (SIO3) is shifted in synchronization with the falling edge of the serial clock. Transmit data is held in the SO3 latch and is output from the SO3 pin. Data that is received via the SI3 pin in synchronization with the rising edge of the serial clock is latched to SIO3.

Completion of an 8-bit transfer automatically stops operation of SIO3 and sets the interrupt request flag (CSIIF3).

Figure 15-3. Timing of 3-Wire Serial I/O Mode



(3) Transfer start

A serial transfer starts when the following two conditions have been satisfied and transfer data has been set (or read) to serial I/O shift register 3 (SIO3).

- SIO3 operation control bit (CSIE3) = 1
- After an 8-bit serial transfer, either the internal serial clock is stopped or $\overline{\text{SCK3}}$ is set to high level.
- Transmit/transmit and receive mode
  When CSIE3 = 1 and MODE = 0, transfer starts when writing to SIO3.
- Receive-only mode
  When CSIE3 = 1 and MODE = 1, transfer starts when reading from SIO3.

Caution After data has been written to SIO3, transfer will not start even if the CSIE3 bit value is set to "1".

Completion of an 8-bit transfer automatically stops the serial transfer operation and the interrupt request flag (CSIIF3) is set.

CHAPTER 16 SERIAL INTERFACE CSI1

16.1 Serial Interface CSI1 Functions

Serial interface CSI1 has the following two modes.

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

(1) Operation stop mode

This mode is used when serial transfer is not performed. In this mode, the power consumption can be reduced.

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

This mode is used to transfer 8-bit data by using three lines: a serial clock line (SCK1) and two serial data lines (SI1 and SO1).

The processing time of data transfer can be shortened in the 3-wire serial I/O mode because transmission and reception can be simultaneously executed in this mode. In addition, whether 8-bit data is transferred with the MSB or LSB first can be specified, so this interface can be connected to any device.

The 3-wire serial I/O mode is useful for connecting peripheral I/Os and display controllers having a conventional clocked serial interface, such as the 75XL Series, 78K Series, and 17K Series.

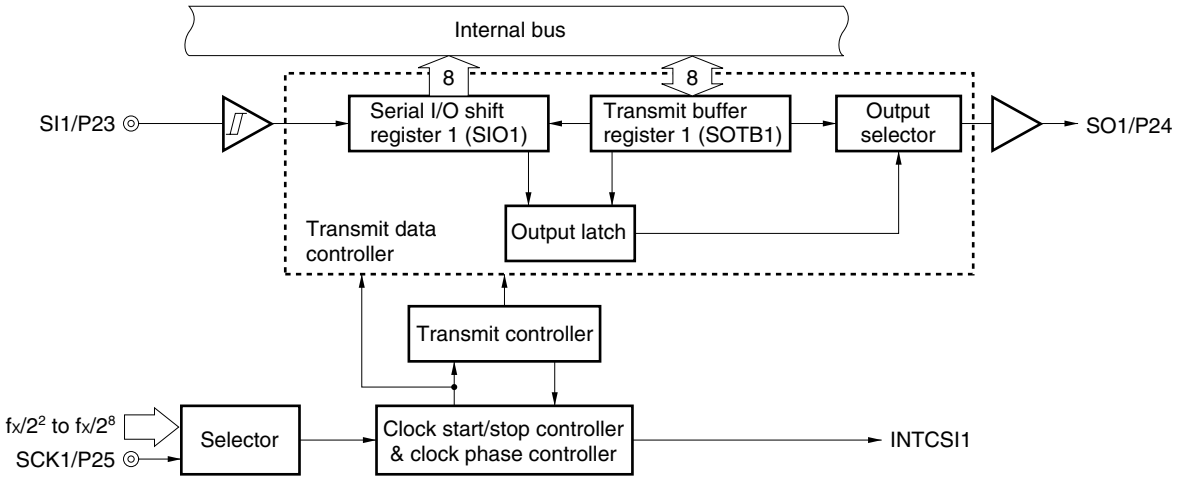
16.2 Serial Interface CSI1 Configuration

Serial interface CSI1 consists of the following hardware.

Table 16-1. Serial Interface CSI1 Configuration

Item	Configuration
Registers	Transmit buffer register 1 (SOTB1) Serial I/O shift register 1 (SIO1)
Control registers	Serial operation mode register 1 (CSIM1) Serial clock select register 1 (CSIC1)

Figure 16-1. Serial Interface CSI1 Block Diagram



(1) Transmit buffer register 1 (SOTB1)

This register sets transmit data.

Transmission/reception is started by writing data to SOTB1 when bit 6 (TRMD1) of serial operation mode register 1 (CSIM1) is 1.

The data written to SOTB1 is converted from parallel data into serial data by serial I/O shift register 1, and output to the serial output (SO1) pin.

SOTB1 can be written or read by an 8-bit memory manipulation instruction.

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

Caution Do not access SOTB1 when CSOT1 = 1 (during serial communication).

(2) Serial I/O shift register 1 (SIO1)

This is an 8-bit register that converts data from parallel into serial or vice versa.

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

Reception is started by reading data from SIO1 if bit 6 (TRMD1) of serial operation mode register 1 (CSIM1) is 0.

During reception, the data is read from the serial input pin (SI1) to SIO1.

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

Caution Do not access SIO1 when CSOT1 = 1 (during serial communication).

16.3 Registers to Control Serial Interface CSI1

Serial interface CSI1 is controlled by the following two registers.

- Serial operation mode register 1 (CSIM1)
- Serial clock select register 1 (CSIC1)

(1) Serial operation mode register 1 (CSIM1)

This register is used to select an operation mode and enable or disable the operation.

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

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

Figure 16-2. Serial Operation Mode Register 1 (CSIM1) Format

Address: FFB0H After reset: 00H R/W[Note 1]

Symbol	7	6	5	4	3	2	1	0
CSIM1	CSIE1	TRMD1	0	DIR1	0	0	0	CSOT1

CSIE1	Operation control in 3-wire serial I/O mode
0	Stops operation (SI1/P23, SO1/P24, and SCK1/P25 pins can be used as general-purpose port pins).
1	Enables operation (SI1/P23, SO1/P24, and SCK1/P25 pins are at active level).

TRMD1[Note 2]	Transmit/receive mode selection
0[Note 3]	Receive mode (transmission disabled).
1	Transmit/receive mode

DIR1[Note 4]	First bit specification
0	MSB
1	LSB

CSOT1[Note 5]	Operation mode flag
0	Communication is stopped.
1	Communication is in progress.

Notes 1. Bit 0 is a read-only bit.

2. Do not rewrite TRMD1 when CSOT1 = 1 (during serial communication).

3. The SO1 pin is fixed to the low level when TRMD1 is 0. Reception is started when data is read from SIO1.

4. Do not overwrite these bits when CSOT1 = 1 (during serial communication).

5. CSOT1 is cleared if CSIE1 is cleared to 0 (operation stops).

Caution Be sure to set bit 5 to 0.

(2) Serial clock select register 1 (CSIC1)

This register is used to select the phase of the data clock and a count clock.

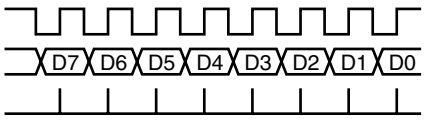
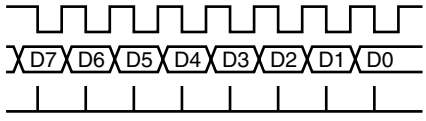
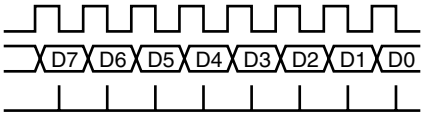
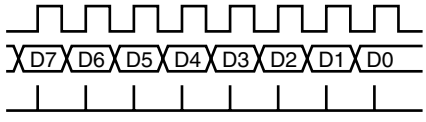
This register is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the value of this register to 10H.

Figure 16-3. Serial Clock Select Register 1 (CSIC1) Format

Address: FFB1H After reset: 10H R/W

Symbol	7	6	5	4	3	2	1	0
CSIC1	0	0	0	CKP1	DAP1	CKS12	CKS11	CKS10

CKP1	DAP1	Data clock phase selection	Type
0	0		1
0	1		2
1	0		3
1	1		4

CKS12	CKS11	CKS10	Count clock CSI1 selection
0	0	0	$f_X/2^2$ (2.5 MHz)
0	0	1	$f_X/2^3$ (1.25 MHz)
0	1	0	$f_X/2^4$ (625 kHz)
0	1	1	$f_X/2^5$ (312.5 kHz)
1	0	0	$f_X/2^6$ (156.25 kHz)
1	0	1	$f_X/2^7$ (78.125 kHz)
1	1	0	$f_X/2^8$ (39.0625 kHz)
1	1	1	External clock

Cautions 1. Do not write CSIC1 when CSIE1 = 0 (operation stops).

2. The phase type of the data clock is type 3 after reset.

Remark Figures in parentheses are for operation with f_X = 10 MHz

16.4 Serial Interface CSI1 Operations

Serial interface CSI1 can be used in the following two modes.

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

16.4.1 Operation stop mode

Serial transfer is not executed in this mode. Therefore, the power consumption can be reduced. In addition, the P23/SI1, P24/SO1, and P25/SCK1 pins can be used as normal I/O port pins in this mode.

(1) Register setting

The operation stop mode is set by serial operation mode register 1 (CSIM1).

(a) Serial operation mode register 1 (CSIM1)

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

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

Address: FFB0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIM1	CSIE1	TRMD1	0	DIR1	0	0	0	CSOT1

CSIE1	Operation control in 3-wire serial I/O mode
0	Stops operation (SI1/P23, SO1/P24, and SCK1/P25 pins can be used as general-purpose port pins).
1	Enables operation (SI1/P23, SO1/P24, and SCK1/P25 pins are at active level).

16.4.2 3-wire serial I/O mode

The 3-wire serial I/O mode is useful for connecting peripheral I/Os and display controllers having a conventional clocked serial interface, such as the 75XL Series, 78K Series, and 17K Series.

In this mode, communication is executed by using three lines: serial clock (SCK1), serial output (SO1), and serial input (SI1) lines.

(1) Register setting

The 3-wire serial I/O mode is set by using serial operation mode register 1 (CSIM1) and serial clock select register 1 (CSIC1).

(a) Serial operation mode register 1 (CSIM1)

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

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

Address: FFB0H After reset: 00H R/W[Note 1]

Symbol	7	6	5	4	3	2	1	0
CSIM1	CSIE1	TRMD1	0	DIR1	0	0	0	CSOT1

CSIE1	Operation control in 3-wire serial I/O mode
0	Stops operation (SI1/P23, SO1/P24, and SCK1/P25 pins can be used as general-purpose port pins).
1	Enables operation (SI1/P23, SO1/P24, and SCK1/P25 pins are at active level).

TRMD1[Note 2]	Transmit/receive mode selection
0[Note 3]	Receive mode (transmission disabled).
1	Transmit/receive mode

DIR1[Note 4]	First bit specification
0	MSB
1	LSB

CSOT1[Note 5]	Operation mode flag
0	Communication is stopped.
1	Communication is in progress.

Notes 1. Bit 0 is a read-only bit.

2. Do not rewrite TRMD1 when CSOT1 = 1 (during serial communication).

3. The SO1 pin is fixed to the low level when TRMD1 is 0. Reception is started when data is read from SIO1.

4. Do not overwrite these bits when CSOT1 = 1 (during serial communication).

5. CSOT1 is cleared if CSIE1 is cleared to 0 (operation stops).

Caution Be sure to set bit 5 to 0.

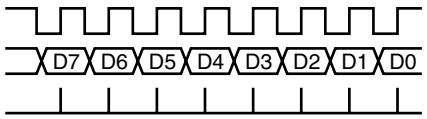
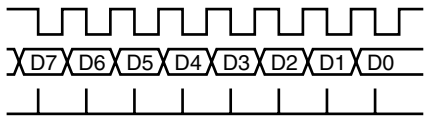
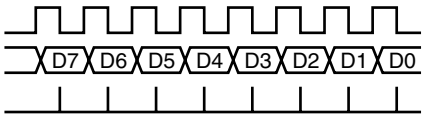
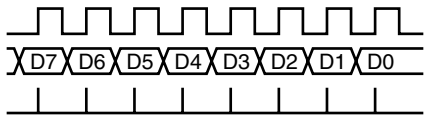
(b) Serial clock select register 1 (CSIC1)

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

$\overline{\text{RESET}}$ input sets the value of this register to 10H.

Address: FFB1H After reset: 10H R/W

Symbol	7	6	5	4	3	2	1	0
CSIC1	0	0	0	CKP1	DAP1	CKS12	CKS11	CKS10

CKP1	DAP1	Data clock phase selection	Type
0	0		1
0	1		2
1	0		3
1	1		4

CKS12	CKS11	CKS10	Count clock CSI1 selection
0	0	0	$f_X/2^2$ (2.5 MHz)
0	0	1	$f_X/2^3$ (1.25 MHz)
0	1	0	$f_X/2^4$ (625 kHz)
0	1	1	$f_X/2^5$ (312.5 kHz)
1	0	0	$f_X/2^6$ (156.25 kHz)
1	0	1	$f_X/2^7$ (78.125 kHz)
1	1	0	$f_X/2^8$ (39.0625 kHz)
1	1	1	External clock

Cautions 1. Do not write CSIC1 when CSIE1 = 0 (operation stops).

2. The phase type of the data clock is type 3 after reset.

Remark Figures in parentheses are for operation with f_X = 10 MHz

(2) Setting of port

<1> Transmit/receive mode

(a) To use externally input clock as system clock (SCK1)

- Bit 3 (PM23) of port mode register 2: Set to 1
- Bit 4 (PM24) of port mode register 2: Cleared to 0
- Bit 5 (PM25) of port mode register 2: Set to 1
- Bit 4 (P24) of port 2: Cleared to 0

(b) To use internal clock as system clock (SCK1)

- Bit 3 (PM23) of port mode register 2: Set to 1
- Bit 4 (PM24) of port mode register 2: Cleared to 0
- Bit 5 (PM25) of port mode register 2: Cleared to 0
- Bit 4 (P24) of port 2: Cleared to 0
- Bit 5 (P25) of port 2: Cleared to 0

<2> Receive mode (with transmission disabled)

(a) To use externally input clock as system clock (SCK1)

- Bit 3 (PM23) of port mode register 2: Set to 1
- Bit 5 (PM25) of port mode register 2: Set to 1

(b) To use internal clock as system clock (SCK1)

- Bit 3 (PM23) of port mode register 2: Set to 1
- Bit 5 (PM25) of port mode register 2: Cleared to 0
- Bit 5 (P25) of port 2: Cleared to 0

Remark The transmit/receive mode or receive mode is selected by using bit 6 (TRMD1) of serial operation mode register 1 (CSIM1).

(3) Communication operation

In the 3-wire serial I/O mode, data is transmitted or received in 8-bit units. Each bit of the data is transmitted or received in synchronization with the serial clock.

Data can be transmitted or received if bit 6 (TRMD1) of serial operation mode register 1 (CSIM1) is 1. Transmission/reception is started when a value is written to transmit buffer register 1 (SOTB1). Data can be received if bit 6 (TRMD1) of serial operation mode register 1 (CSIM1) is 0. Reception is started when data is read from serial I/O shift register 1 (SIO1).

After communication has been started, bit 0 (CSOT1) of CSIM1 is set to 1. When communication of 8-bit data has been completed, a communication completion interrupt flag (CSIIF1) is set, and CSOT1 is cleared to 0. Then the next communication is enabled.

Caution Do not access the control register and data register when CSOT1 = 1 (during serial communication).

Figure 16-4. Timing in 3-Wire Serial I/O Mode (1/2)

(1) Transmission/reception timing (Type 1; TRMD1 = 1, DIR1 = 0, CKP1 = 0, DAP1 = 0)

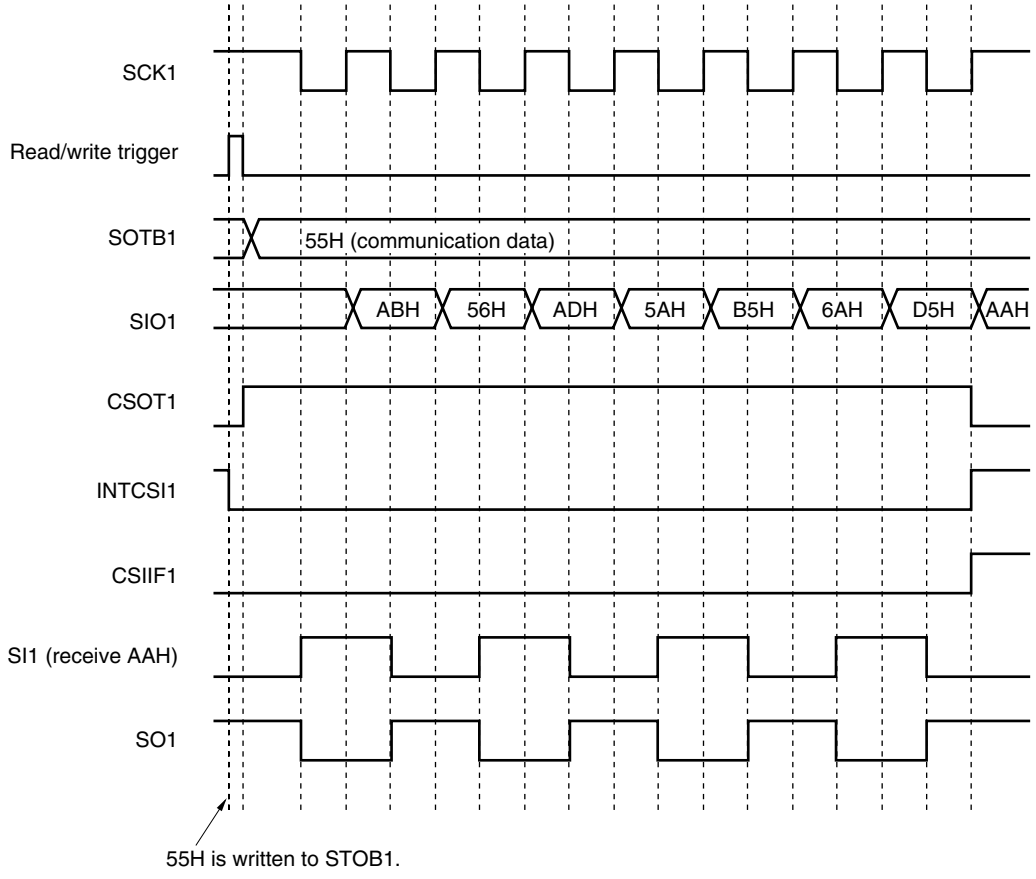


Figure 16-4. Timing in 3-Wire Serial I/O Mode (2/2)

(2) Transmission/reception timing (Type 2; TRMD1 = 1, DIR1 = 0, CKP1 = 0, DAP1 = 1)

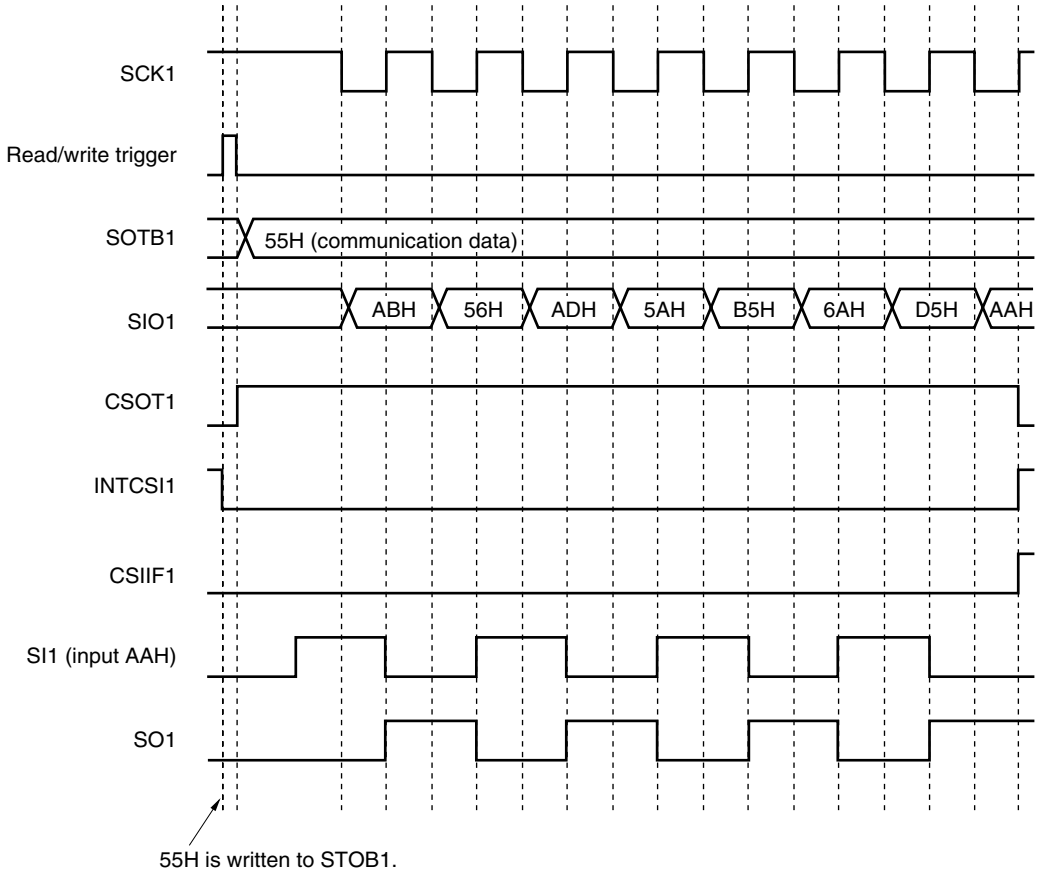
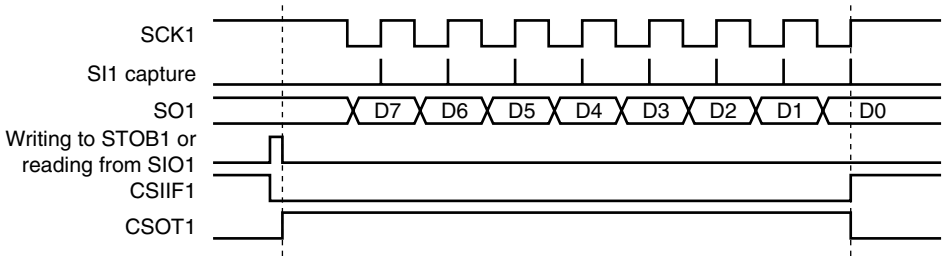
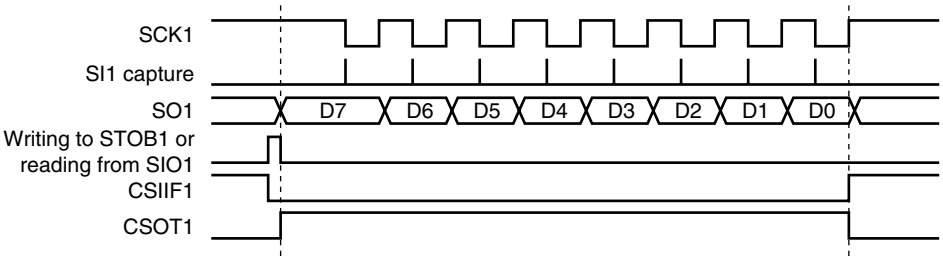


Figure 16-5. Timing of Clock/Data Phase

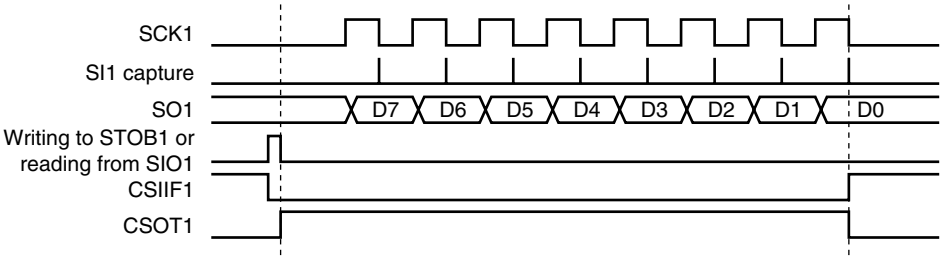
(a) Type 1; CKP1 = 0, DAP1 = 0



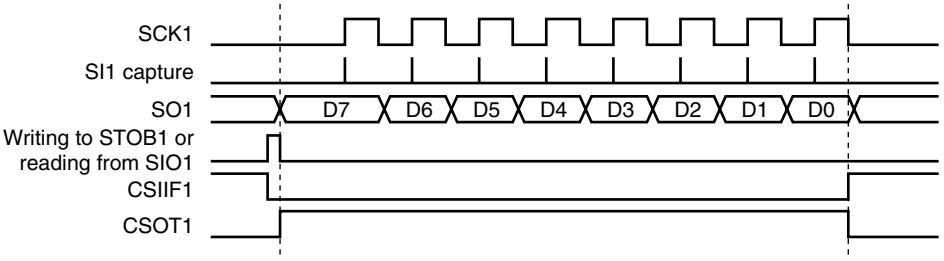
(b) Type 2; CKP1 = 0, DAP1 = 1



(c) Type 3; CKP1 = 1, DAP1 = 0



(d) Type 4; CKP = 1, DAP1 = 1

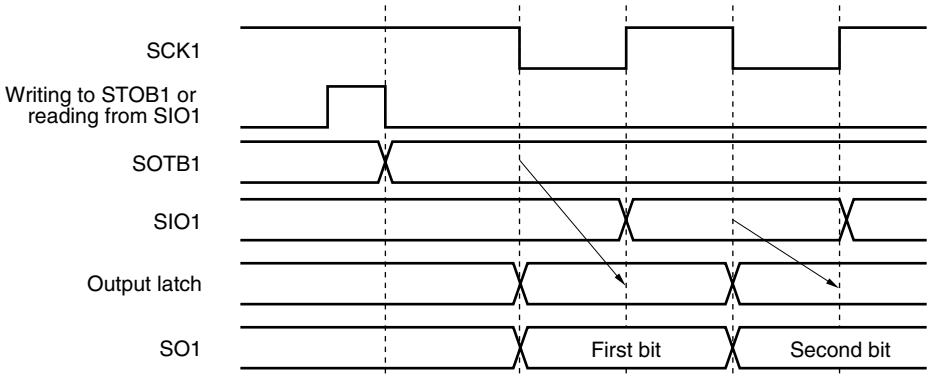


(4) Timing of output to SO1 pin (first bit)

When communication is started, the value of transmit buffer register 1 (SOTB1) is output from the SO1 pin. The output operation of the first bit at this time is explained below.

Figure 16-6. Output Operation of First Bit

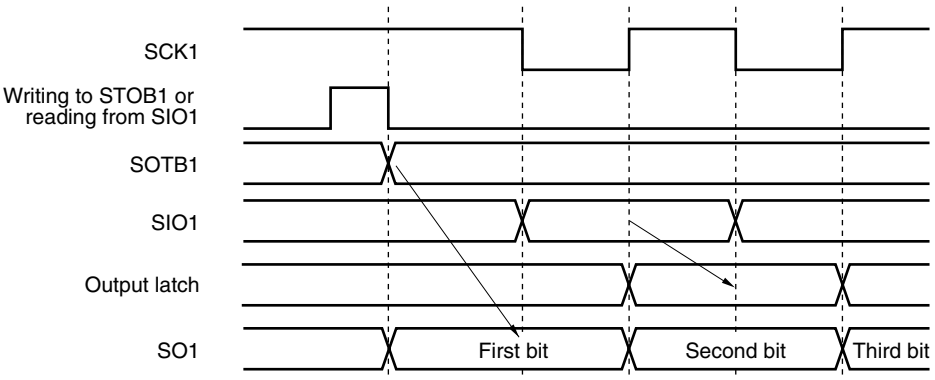
(1) CKP1 = 0, DAP1 = 0 (or CKP1 = 1, DAP1 = 0)



The first bit is directly latched to the output latch from the SOTB1 register at the falling (or rising) edge of SCK1, and is output from the SO1 pin via the output selector. At the next rising (or falling) edge of SCK1, the value of the SOTB1 register is transferred to the SIO1 register and shifted by 1 bit. At the same time, the first bit of the receive data is stored in the SIO1 register via the SI1 pin.

The second and subsequent bits are latched to the output latch from SIO1 at the next falling (or rising) edge of SCK1 and the data is output from the SO1 pin.

(2) CKP1 = 0, DAP1 = 1 (or CKP1 = 1, DAP1 = 1)



The first bit is directly output from the SOTB1 register to the SO1 pin via the output selector at the falling edge of the write signal of SOTB1 or the read signal of the SIO1 register. At the next falling (or rising) edge of SCK1, the value of the SOTB1 register is transferred to the SIO1 register and shifted by 1 bit. At the same time, the first bit of the received data is stored in the SIO1 register via the SI1 pin.

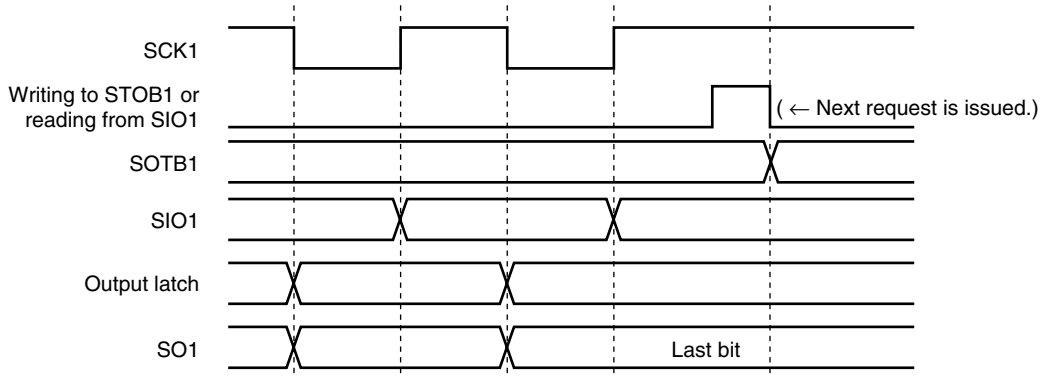
The second and subsequent bits are latched to the output latch from SIO1 at the next rising (or falling) edge of SCK1 and the data is output from the SO1 pin.

(5) Output value of SO1 pin (last bit)

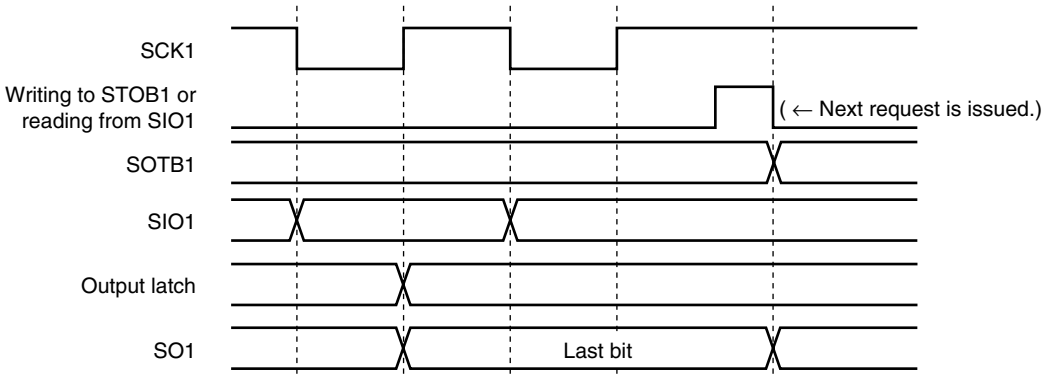
After communication has been completed, the SO1 pin holds the output value of the last bit.

Figure 16-7. Output Value of SO1 Pin (Last Bit)

(1) Type 1; CKP1 = 0 and DAP1 = 0 (or CKP1 = 1, DAP1 = 0)



(2) Type 2; CKP1 = 0 and DAP1 = 1 (or CKP1 = 1, DAP1 = 1)



(6) SCK1 pin

The status of the SCK1 pin is as follows if bit 7 (CSIE1) of serial operation mode register 1 (CSIM1) is cleared to 0.

Table 16-2. SCK1 Pin Status

CKP1	CKS12 to 10	SCK1 Pin
CKP1 = 0	CKS12, 11, 10 ≠ 1, 1, 1	Outputs high level.
	CKS12, 11, 10 = 1, 1, 1	Outputs high level.
CKP1 = 1**Note**	CKS12, 11, 10 ≠ 1, 1, 1**Note**	Outputs low level**Note**.
	CKS12, 11, 10 = 1, 1, 1	Outputs high level.

Note Status after reset

(7) SO1 pin

The status of the SO1 pin is as follows if bit 7 (CSIE1) of serial operation mode register 1 (CSIM1) is cleared to 0.

Table 16-3. SO1 Pin Status

TRMD1	DAP1	DIR1	SO1 Pin
TRMD1 = 0**Note**	–	–	Outputs low level**Note**.
TRMD1 = 1	DAP1 = 0	–	Value of SO1 latch (low-level output)
	DAP1 = 1	DIR1 = 0	Value of bit 7 of SOTB1
		DIR1 = 1	Value of bit 0 of SOTB1

Note Status after reset

Caution If a value is written to TRMD1, DAP1, and DIR1, the output value of the SO1 pin changes.

CHAPTER 17 LCD CONTROLLER/DRIVER

17.1 LCD Controller/Driver Functions

The internal LCD controller/driver of the μPD780318, 780328, and 780338 Subseries has the following functions:

(1) Automatic output of segment signals and common signals by automatically reading display data memory
(2) Internal booster circuit employed for LCD driver reference voltage generator (×3 only).
Therefore, LCD can be stably displayed even if the supply voltage drops because the battery voltage drops.
In addition, the LCD driver reference voltage can be changed by using an external resistor to adjust the brightness.
(3) Three display modes selectable
- Static (up to 12 lines)
- 1/3 duty (1/3 bias)
- 1/4 duty (1/3 bias)

(4) Four types of frame frequencies selectable in each display mode
(5) The number of segment signal output lines differs depending on the model as shown in Table 17-1.

Table 17-1. Segment Signals and Common Signals

Part Number	Maximum Number of Segment Signals	Common Signals
μPD780316, 780318	24 lines (S0 to S23), of which 12 (S0 to S11) are selectable for static display.	Dynamic display: COM0 to COM3 Static display: SCOM0
μPD780326, 780328	32 lines (S0 to S31), of which 12 (S0 to S11) are selectable for static display.	
μPD780336, 780338	40 lines (S0 to S39), of which 12 (S0 to S11) are selectable for static display.	
μPD78F0338	40 lines (S0 to S39), of which 12 (S0 to S11) are selectable for static display, and 16 (S24 to S39) are also used with output port lines (P80 to P87 and P90 to P97)**Note**.	

Note The operation mode of the alternate-function pins can be switched between the port mode and segment signal mode in 8-bit units by using pin function switching registers 8 and 9 (PF8 and PF9).

(6) Simultaneous driving of static display (up to 12 segments) and dynamic display. The operation mode of the alternate-function pins (S0 to S11) can be switched between the static display mode and dynamic display mode in 4-bit units.
(7) Blinking of LCD (only when subsystem clock is used).
Whether each segment blinks or not can be selected.
The blinking cycle can be selected from 0.5 s or 1.0 s.
(8) Operation with subsystem clock
(9) Operating voltage range: 1.8 to 5.5 V

Table 17-2 shows the maximum number of pixels that can be displayed in each display mode.

Table 17-2. Maximum Number of Pixels Displayed

Part Number	Bias Mode	Time Division	Common Signals	Maximum Number of Pixels
μPD780316, 780318, 78F0338	—	Static	SCOM0	12 (12 segment × 1 common)
	1/3	3	COM0 to COM2	72 (24 segment × 3 common)
		4	COM0 to COM3	96 (24 segment × 4 common)
μPD780326, 780328	—	Static	SCOM0	12 (12 segment × 1 common)
	1/3	3	COM0 to COM2	96 (32 segment × 3 common)
		4	COM0 to COM3	128 (32 segment × 4 common)
μPD780336, 780338	—	Static	SCOM0	12 (12 segment × 1 common)
	1/3	3	COM0 to COM2	120 (40 segment × 3 common)
		4	COM0 to COM3	160 (40 segment × 4 common)

17.2 LCD Controller/Driver Configuration

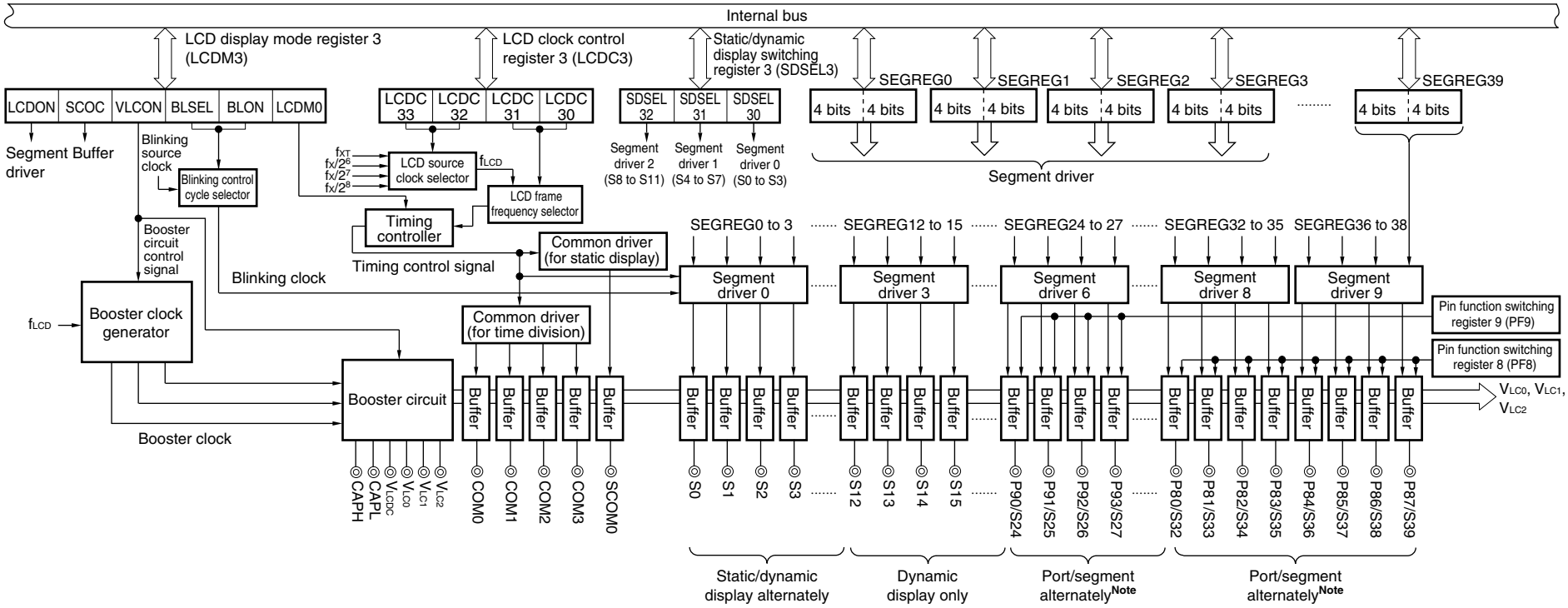
The LCD controller/driver consists of the following hardware.

Table 17-3. LCD Controller/Driver Configuration

Item		Configuration
Display output		
	μPD780316, 780318	Segment signal: 24 lines Dynamic/static alternated: 12 lines Dynamic display segment: 12 lines Common signal: 4 lines (for dynamic display) 1 line (for static display)
	μPD780326, 780328	Segment signal: 32 lines Dynamic/static alternated: 12 lines Dynamic display segment: 20 lines Common signal: 4 lines (for dynamic display) 1 line (for static display)
	μPD780336, 780338	Segment signal: 40 lines Dynamic/static alternated: 12 lines Dynamic display segment: 28 lines Common signal: 4 lines (for dynamic display) 1 line (for static display)
	μPD78F0338	Segment signal: 40 lines Dynamic/static alternated: 12 lines Dynamic display segment: 12 lines Segment/output port: 16 lines Common signal: 4 lines (for dynamic display) 1 line (for static display)
Control register		LCD display mode register 3 (LCDM3) LCD clock control register 3 (LCDC3) Static/dynamic display switching register 3 (SDSEL3) Pin function switching register 8 (PF8)**Note** Pin function switching register 9 (PF9)**Note**

Note μPD78F0338 only

Figure 17-1. LCD Controller/Driver Block Diagram



Note μPD78F0338 only

17.3 Registers to Control LCD Controller/Driver

The LCD controller/driver can be controlled by using the following three types of registers (the LCD controller/driver of the μPD78F0338 is controlled by five types of registers).

- LCD display mode register 3 (LCDM3)
- LCD clock control register 3 (LCDC3)
- Static/dynamic display switching register 3 (SDSEL3)
- Pin function switching register 8 (PF8)**Note**
- Pin function switching register 9 (PF9)**Note**

Note μPD78F0338 only

(1) LCD display mode register 3 (LCDM3)

This register enables or disables display, controls the booster circuit and blinking display, and selects a display mode.

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

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

Figure 17-2. LCD Display Mode Register 3 (LCDM3) Format

Address: FF90H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LCDM3	LCDON	SCOC	VLCON	BLSEL	BLON	0	0	LCDM0

LCDON	Display control (enables output of display data)
0	Display OFF (All segment output pins output unselect signals.)
1	Display ON

SCOC	Output control of segment/common pins
0	Outputs GND level to segment/common pins.
1	Outputs select signal to segment/common pins.

★

VLCON	Booster circuit control
0	Stops booster circuit.
1	Operates booster circuit

BLSEL[Note 1]	Blinking clock cycle selection
0	Blinking cycle of 0.5 s
1	Blinking cycle of 1.0 s

BLON[Note 2]	Blinking display control
0	Blinking display OFF
1[Note 3]	Blinking display ON

LCDM0[Note 4]	Dynamic/static display alternate pins[Notes 5, 6]		Dynamic pin	
	Time division	Bias mode	Time division	Bias mode
0	4	1/3	4	1/3
1	3	1/3	3	1/3

Notes 1. The BLSEL bit is valid only when the subsystem clock is used.

2. The corresponding segment pin can be blinked only if the blinking data memory (higher 4 bits of FA00 to FA27H) is set to 1.

3. Do not change the contents of the blinking data memory while BLON = 1.

4. Do not change LCDM0 while the LCD is in operation. Be sure to set this bit while LCDON = 0, SCOC = 0, and VLCON = 0.

5. The dynamic/static display alternate pins are in the static display mode when this mode is selected by the static/dynamic display switching register 3 (SDSEL3).

6. When static display is not used, the static display common output pin (SCOM0) outputs the GND potential.

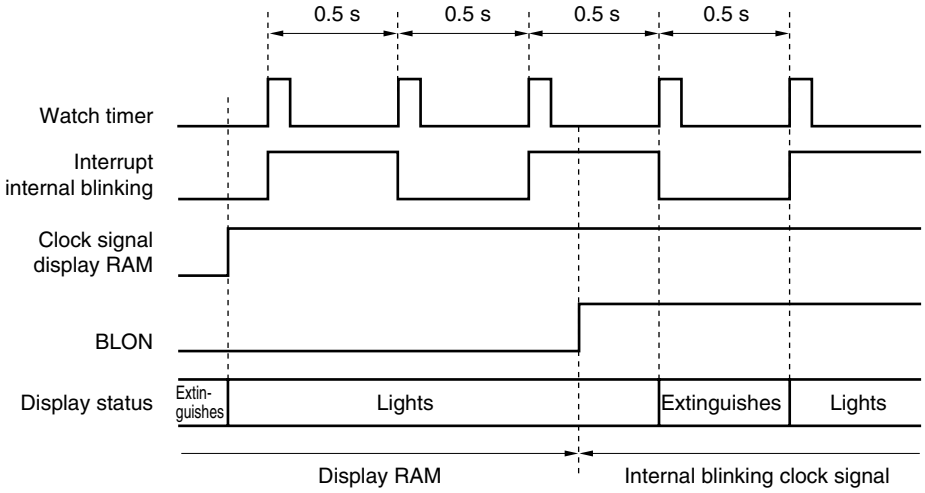
Cautions 1. Set the LCDON, SCOC, and VLCON bits in the following sequence:

- **To display LCD while LCD booster circuit stops**
  - **(1) Set VLCON to 1. All the segment and common pins are in the GND output mode (SCOC = 0).**
  - ↓
  - **(2) Set VLCON to 1 and wait 500 ms or longer with software.**
  - ↓
  - **(3) Set SCOC to 1. All the segment and common pins output an unselect waveform and are in unselect display mode.**
  - ↓
  - **(4) Set LCDON to 1. The value of the display RAM is reflected on the segment output waveform, and all segment and common pins are in select display mode.**

- **To stop LCD booster circuit while LCD displays**
  - **(1) Clear LCDON to 0. All the segment and common pins are in unselect display mode.**
  - ↓
  - **(2) Clear SCOC to 0. All the segment and common pins are in GND output mode.**
  - ↓
  - **(3) Clear VLCON to 0. The LCD booster circuit stops.**

2. The blinking cycle is generated using the interval time (0.5 s at 32.768 kHz) of the watch timer. When the blinking function is not used (BLON = 0), the LCD lights or extinguishes depending on the setting of the display RAM, as shown in Figure 17-3. To use the blinking function (BLON = 1), the LCD lights or extinguishes depending on the status of the internal blinking clock signal (set value of BLSEL), i.e., it lights if the internal blinking clock signal is "1" and extinguishes if the signal is "0".

Figure 17-3. Blinking Function



3. When using the blinking function, the LCD does not blink even if the data is rewritten while the LCD is in the extinguishing cycle (0.5 s or 1.0 s), unless the LCD is in the lighting cycle.

(2) LCD clock control register 3 (LCDC3)

This register is used to select an LCD source clock and frame frequency.

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

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

Figure 17-4. LCD Clock Control Register 3 (LCDC3) Format

Address: FF91H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LCDC3	0	0	0	0	LCDC33	LCDC32	LCDC31	LCDC30

LCDC33	LCDC32	Source clock selection (f_{LCD})
0	0	f_{XT} (32.768 kHz)
0	1	$f_X/2^6$ (156.25 kHz)
1	0	$f_X/2^7$ (78.125 kHz)
1	1	$f_X/2^8$ (39.0625 kHz)

LCDC31	LCDC30	Selection of reference clock generating frame frequency
0	0	$f_{LCD}/2^6$
0	1	$f_{LCD}/2^7$
1	0	$f_{LCD}/2^8$
1	1	$f_{LCD}/2^9$

Caution Do not rewrite LCDC3 while LCD is operating. Be sure to set this bit while LCDON = 0, SCOC = 0, and VLCON = 0.

Remark Figures in parentheses are for operation with f_X = 10 MHz or f_{XT} = 32.768 kHz

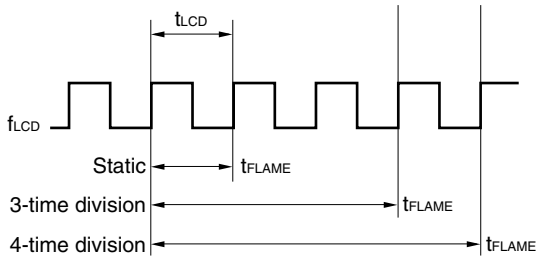
Table 17-4 shows the frame frequency if f_{XT} (32.768 kHz) is used as the source clock (f_{LCD}), and Figure 17-5 shows the relationship between the reference clock that generates the frame frequency, and the frame frequency.

Table 17-4. Frame Frequency

Frame Frequency \ Reference Clock Generating Frame Frequency		$f_{XT}/2^9$	$f_{XT}/2^8$	$f_{XT}/2^7$	$f_{XT}/2^6$
Display duty	Static	64 Hz	128 Hz	256 Hz**Note**	512 Hz**Note**
	1/3 duty	21 Hz	43 Hz	85 Hz	171 Hz**Note**
	1/4 duty	16 Hz	32 Hz	64 Hz	128 Hz

★ **Note** Set so that the frame frequency is 128 Hz or less.

Figure 17-5. Relationship Between Reference Clock Generating Frame Frequency, and Frame Frequency



Remark f_{LCD}: Reference clock that generates frame frequency
t_{LCD}: LCD clock period
t_{FLAME}: Frame period

(3) Static/dynamic display switching register 3 (SDSEL3)

This register is used to select the static or dynamic display mode of the segment pins (S0 to S11).

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

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

★ **Figure 17-6. Static/Dynamic Display Switching Register 3 (SDSEL3) Format**

Address: FF92H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SDSEL3	0	0	0	0	0	SDSEL32	SDSEL31	SDSEL30

Part Number	SDSEL32	SDSEL31	SDSEL30	Number of Segments (for Static Mode)	Number of Segments (for Dynamic Mode)
μPD780316, 780318	0	0	0	—	S0 to S23
	0	0	1	S0 to S3	S4 to S23
	0	1	1	S0 to S7	S8 to S23
	1	1	1	S0 to S11	S12 to S23
	Setting other than above is prohibited.			—	
μPD780326, 780328	0	0	0	—	S0 to S31
	0	0	1	S0 to S3	S4 to S31
	0	1	1	S0 to S7	S8 to S31
	1	1	1	S0 to S11	S12 to S31
	Setting other than above is prohibited.			—	
μPD780336, 780338, 78F0338	0	0	0	—	S0 to S39
	0	0	1	S0 to S3	S4 to S39
	0	1	1	S0 to S7	S8 to S39
	1	1	1	S0 to S11	S12 to S39
	Setting other than above is prohibited.			—	

Caution Do not rewrite SDSEL while the LCD is operating. Be sure to set this bit while LCDON = 0, SCOC = 0, and VLCON = 0. Note that SDSEL can be set only once after reset.

(4) Pin function switching registers 8 and 9 (PF8 and PF9)[Note]

These registers are used to select whether the pins of ports 8 and 9 are used as port pins or segment pins.

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

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

Note μPD78F0338 only

Figure 17-7. Pin Function Switching Registers 8 and 9 (PF8 and PF9) Format

Address: FF58H After reset: 00H W

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

Address: FF59H After reset: 00H W

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

PFn7	PFn6	PFn5	PFn4	PFn3	PFn2	PFn1	PFn0	Setting of pin
0	0	0	0	0	0	0	0	Segment output (n = 8: S32 to S39, n = 9: S24 to S31)
1	1	1	1	1	1	1	1	Output port (n = 8: P87 to P80, n = 9: P97 to P90)
Other than above								Setting prohibited

Caution PF8 and PF9 can be set to 00H or FFH only once after reset. Do not set any value other than 00H and FFH to these registers. Before changing the setting of these registers, reset the device.

17.4 LCD Display RAM

The LCD display data and the LCD blinking select bits corresponding to LCD display data are mapped to addresses FA00H to FA27H. The lower 4 bits of each of these addresses are an LCD display data area, and the higher 4 bits are an LCD blinking select bit area. The LCD blinking select bits correspond to the LCD display data (i.e., LCD blinking select bit 0 corresponds to bit 4 of the LCD display data, bit 1 to bit 5, bit 2 to bit 6, and bit 3 to bit 7). The addresses and capacity of the area that can be used for LCD display differs depending on the product, as follows:

- μPD780316, 780318: FA00H to FA17A (24 bytes)
- μPD780326, 780328: FA00H to FA1FH (32 bytes)
- μPD780336, 780338, 78F0338: FA00H to FA27H (40 bytes)

The data stored to the LCD display data area can be displayed on the LCD panel.

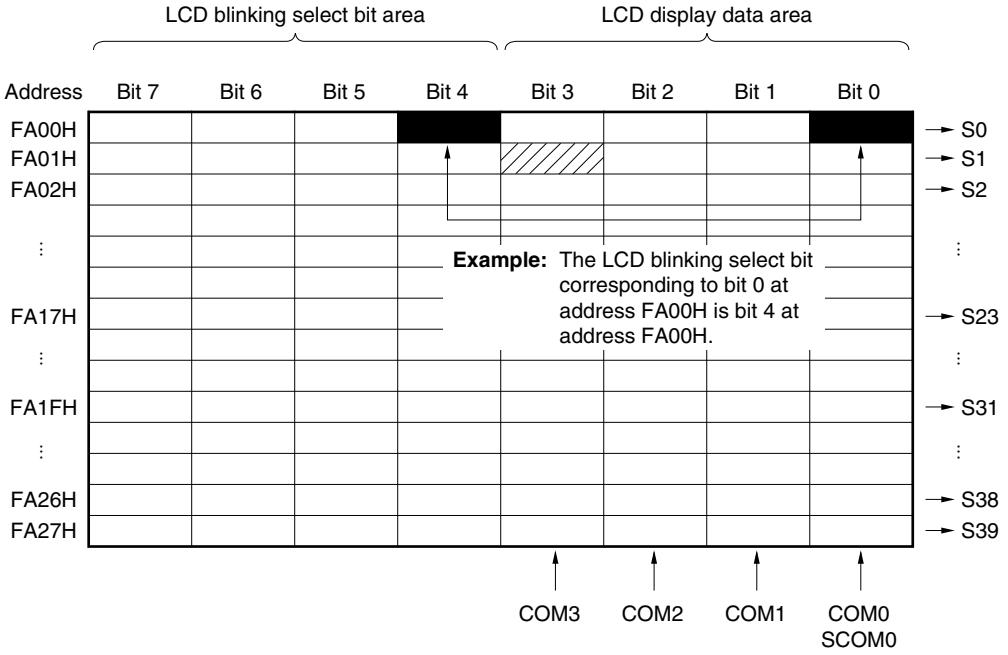
★ For example, bit 3 (shaded portion in Figure 17-8) of address FA01H is output to pin S1 at the timing of COM3.

The LCD blinking select bit is used to blink the corresponding segment by setting 1 to the bit to blink, and 1 to bit 3 (BLON) of LCD display mode register 3 (LCDM3). In this case, however, the display data of the corresponding segment must be 1.

Figure 17-8 shows the relationship between the LCD display data, contents of the blinking select bits, and segment/common output signals.

The area not used for display can be used as a normal RAM area.

★ **Figure 17-8. Relationship Between LCD Display Data, Contents of Blinking Select Bits, and Segment/Common Output Signals (4-Time Division)**



Caution The higher 4 bits (LCD blinking select bit area) of each address, FA00H to FA27H, correspond to the lower 4 bits (LCD display data area). When the LCD does not blink, therefore, be sure to clear the corresponding bit in the blinking select bit area to 0.

17.5 LCD Controller/Driver Settings

Set the LCD controller/driver as follows:

(1) When using the μPD78F0338, specify whether P80/S32 to P87/S39 and P90/S24 to P97/S31 are used as segment output pins or port output pins, by using pin function switching registers 8 and 9 (PF8 and PF9).

(2) Specify the display mode of the segment output pins (S0 to S11) by using static/dynamic display switching register 3 (SDSEL3).

★ (3) Set the displayed default value to the LCD display data area (bits 0 to 3) of the LCD display RAM. The addresses and capacity of the LCD display RAM that can be used in each device are as follows:

- μPD780316, 780318: FA00H to FA17H (24 bytes)
- μPD780326, 780328: FA00H to FA1FH (32 bytes)
- μPD780336, 780338, 78F0338: FA00H to FA27H (40 bytes)

To use the blinking function, set the corresponding bit of the blinking select bit area (bits 4 to 7) in the LCD display RAM to 1.

(4) Specify the display mode using bit 0 (LCDM0) of LCD display mode register 3 (LCDM3).

(5) Select the source clock and frame frequency of the LCD using LCD clock control register 3 (LCDC3).

(6) Set bit 5 (VLCON) of LCD display mode register 3 (LCDM3) to 1 to start the operation of the booster circuit.

(7) Make sure that a wait time of 500 ms or longer elapses with software.

(8) Set bit 6 (SCOC) of LCD display mode register 3 (LCDM3) to 1 so that unselect waveform is output to the segment pins and common pins.

(9) To use the blinking function, select a blinking cycle of 0.5 s or 1.0 s by using bit 4 (BLSEL) of LCD display mode register 3 (LCDM3).

(10) Set bit 7 (LCDON) of LCD display mode register 3 (LCDM3) to 1 to set the display to ON. To blink the LCD, set bit 3 (BLON) of LCD display mode register 3 (LCDM3) to 1 to set the display to ON.

Then, set data to the display data memory and timing of the blinking display according to the data to be displayed.

17.6 Common Signals and Segment Signals

★ An individual pixel on an LCD panel lights when the potential difference of the corresponding common signal and segment signal reaches or exceeds a given voltage (depending on the panel), and extinguishes when the potential difference drops lower than V_{LCD}.

(1) Common signals

For common signals, the selection timing order is as shown in Table 17-5 according to the number of time divisions set, and operations are repeated with these as the cycle. In the static mode, the same signal is output to SCOM0. With 3-time-division operation, the COM3 pin is left open.

Table 17-5. COM Signals

COM Signal / Time Division	COM0	COM1	COM2	COM3	SCOM0
Static	–	–	–	–	
3-time division				Open	–
4-time division					–

(2) Segment signals

Segment signals correspond to a 40-byte LCD display RAM (FA00H to FA27H[Note]). Each display data memory bit 0, bit 1, bit 2, and bit 3 is read in synchronization with the SCOM0/COM0, COM1, COM2 and COM3 timings respectively, and if the value of the bit is 1, it is converted to the selection voltage. If the value of the bit is 0, it is converted to the non-selection voltage and output to a segment pin (S0 to S39[Note]).

Consequently, it is necessary to check what combination of front surface electrodes (corresponding to the segment signals) and rear surface electrodes (corresponding to the common signals) of the LCD display to be used form the display pattern, and then write bit data corresponding on a one-to-one basis with the pattern to be displayed.

In addition, because LCD display RAM bits 1 to 3 are not used with the static method, these can be used for other than display purposes.

LCD display RAM bits 4 to 7 are bits for LCD blinking selection. To use the LCD blinking function, set the relevant bit to 1.

Note The segment signal output pins or the area that can be used as the LCD display data vary depending on the product.

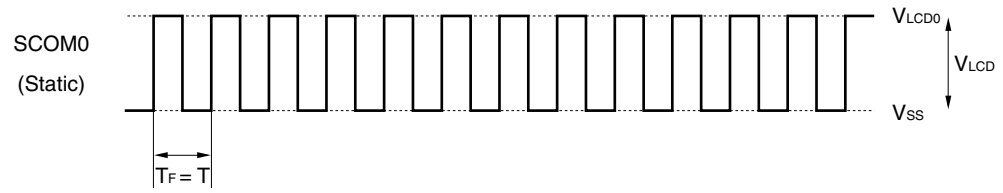
Part Number	Segment Signal Output Pins	Area That Can Be Used as LCD Display Data
μPD780316, 780318	S0 to S23	FA00H to FA17H
μPD780326, 780328	S0 to S31	FA00H to FA1FH
μPD780336, 780338	S0 to S39	FA00H to FA27H
μPD78F0338	S0 to S39 (S24 to S31, S32 to S39 are alternate with P90 to P97 and P80 to P87, respectively)	FA00H to FA27H (when ports 8 and 9 are used as the segment signal outputs)

(3) Common signal and segment signal output waveforms

The voltages shown in Figures 17-9 and 17-10 are output in the common signals and segment signals. The $\pm V_{LCD}$ ON voltage is only produced when the common signal and segment signal are both at the selection voltage; other combinations produce the OFF voltage.

Figure 17-9. Common Signal Waveform

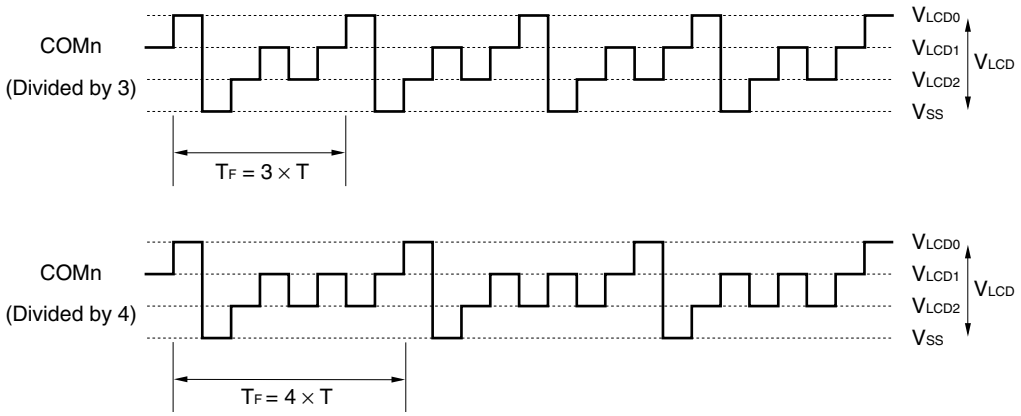
(a) Static display mode



T: One LCDCL cycle

T_F: Frame frequency

(b) Dynamic display mode
(1/3 bias method)

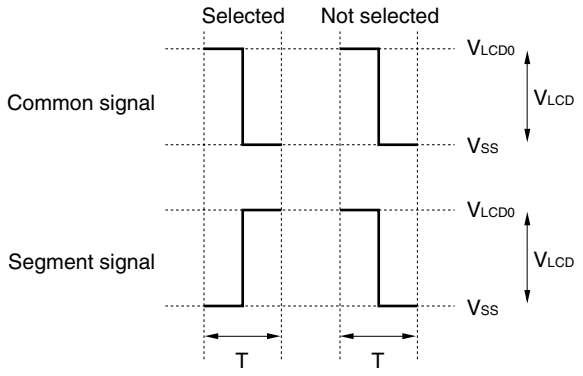


T: One LCDCL cycle

T_F: Frame frequency

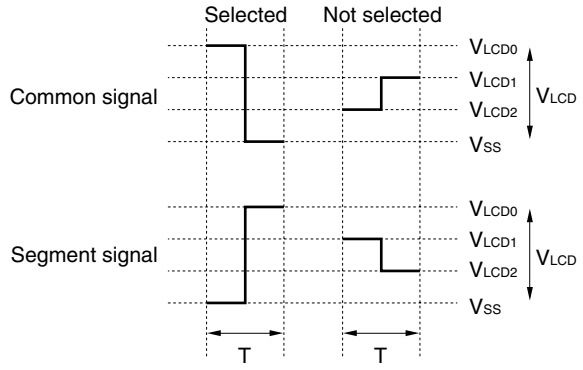
Figure 17-10. Common Signal and Segment Signal Voltages and Phases

(a) Static display mode



Remark T: One LCDCL cycle

(b) Dynamic display mode
(1/3 bias method)



Remark T: One LCDCL cycle

17.7 Supplying LCD Drive Voltages V_{LC0}, V_{LC1}, and V_{LC2}

The μPD780338 contains a booster circuit ($\times$3 only) to generate a supply voltage to drive the LCD. The internal LCD reference voltage (V_{LCD2}) is output from the V_{LC2} pin. A voltage two times higher than that on V_{LCD2} is output from the V_{LC1} pin and a voltage three times higher than that on V_{LCD2} is output from the V_{LC0} pin.

The LCD reference voltage (V_{LCD2}) can be varied by connecting external resistors as shown in Figure 17-11.

In addition, the μPD780338 requires an external capacitor (recommended value: 0.47 μF) because it employs a capacitance division method to generate a supply voltage to drive the LCD.

★ **Table 17-6. Output Voltages of V_{LC0} to V_{LC2} Pins**

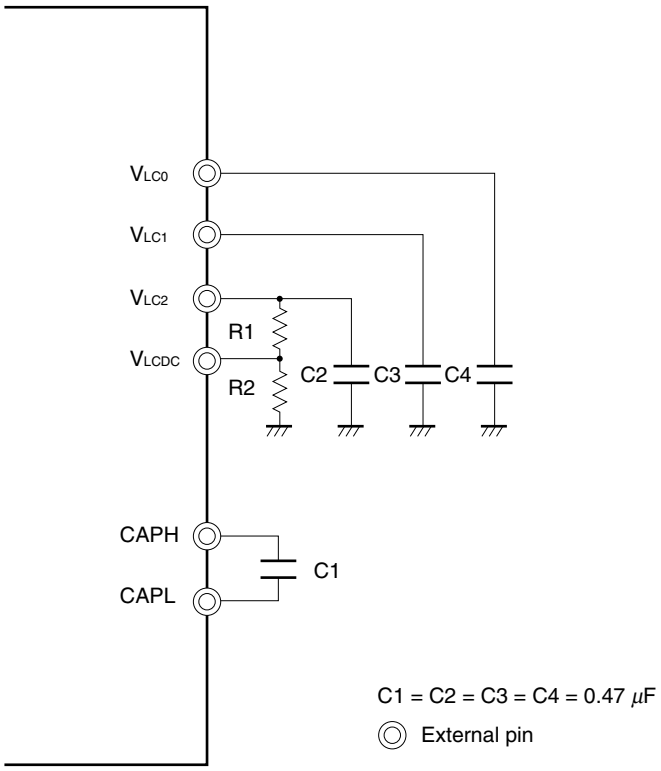
	Output Voltage
V_{LC0} pin	3 × V_{LCD2}
V_{LC1} pin	2 × V_{LCD2}
V_{LC2} pin	V_{LCD2}

Cautions 1. When using the LCD function, do not open the V_{LCDC}, V_{LC0}, V_{LC1}, and V_{LC2} pins. Refer to Figure 17-11 for connection.

★ **2. A constant LCD drive voltage can be supplied regardless of changes in V_{DD}.**

★ **Remark** For the LCD reference voltage (V_{LCD2}), refer to **LCD controller/driver characteristics** in **CHAPTER 24 ELECTRICAL SPECIFICATIONS**.

Figure 17-11. Example of Circuit to Adjust LCD Driver Reference Voltage



Remark Use a capacitor with as little leakage as possible. Use a non-polarity capacitor as C1.

External resistors (R1 and R2) must be connected as shown in Figure 17-11. The recommended resistance and capacitance are shown in the table below.

To adjust the brightness, the user must adjust the ratio of R1 to R2 depending on the LCD panel to be used.

- R1 + R2 = 3 [MΩ]
- C1 = C2 = C3 = C4 = 0.47 [μF]

V_{LCD2} can be adjusted by the division ratio of resistors R1 and R2.

- V_{LCD2} = (R1 + R2)/R2 [V]
- $V_{LCD1} = 2 \times V_{LCD2}$ [V]
- $V_{LCD0} = 3 \times V_{LCD2}$ [V]

Table 17-7. Recommended Constants of External Circuit

	V_{LCD2} [V]	V_{LCD1} [V]	V_{LCD0} [V]	R1 [MΩ]	R2 [MΩ]
V_{LCD0} = 3 [V]	1	2	3	0	3
V_{LCD0} = 4.5 [V]	1.5	3	4.5	1	2

17.8 Display Modes

17.8.1 Static display example

Figure 17-13 shows the connection of a static type 1-digit LCD panel with the display pattern shown in Figure 17-12 with the μPD780338 Subseries segment (S0 to S11) and common (SCOM0) signals. The display example is “5”, and the display data memory contents (addresses FA00H to FA07H) correspond to this.

In accordance with the display pattern in Figure 17-12, selection and non-selection voltages must be output to pins S0 to S7 as shown in Table 17-8 at the SCOM0 common signal timing. At this time, set the SDSEL3 register to 03H to set pins S0 to S7 to the static display mode. ★

Table 17-8. Selection and Non-Selection Voltages (SCOM0)

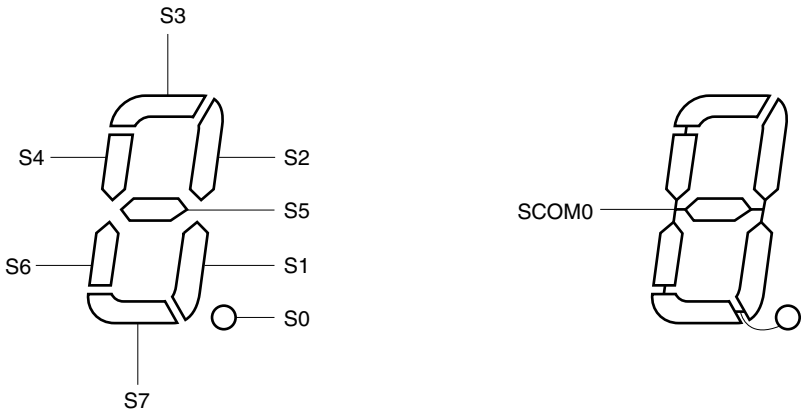
Segment / Common	S0	S1	S2	S3	S4	S5	S6	S7
SCOM0	NS	S	NS	S	S	S	NS	S

S: Selection, NS: Non-selection

From this, it can be seen that 01011101 must be prepared in the bit 0 of the display data memory (addresses FA00H to FA07H) corresponding to S0 to S7.

The LCD drive waveforms for S1, S2, and SCOM0 are shown in Figure 17-14. When S1 is at the selection voltage at the timing for selection with SCOM0, it can be seen that the $+V_{LCD}/-V_{LCD}$ AC square wave, which is the LCD illumination (ON) level, is generated.

Figure 17-12. Static LCD Panel Display Pattern and Electrode Connections



★ **Figure 17-13. Static LCD Panel Connection Example (SDSEL3n = 1: n = 0, 1)**

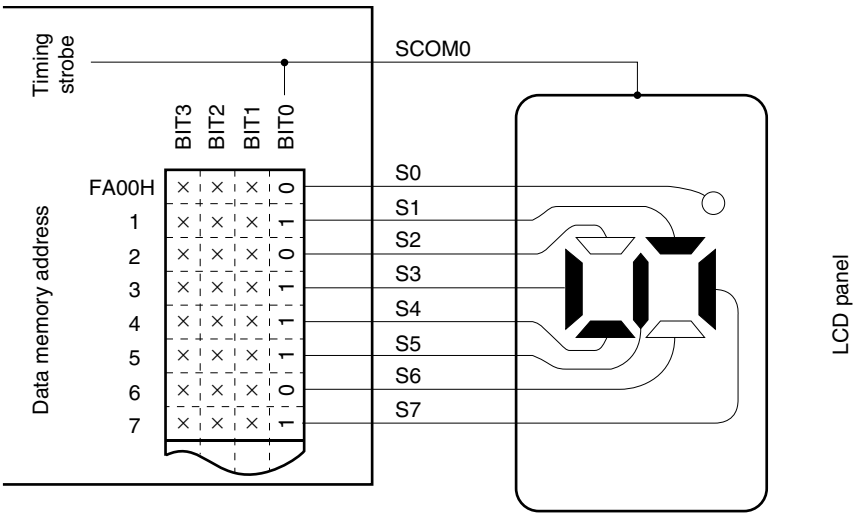
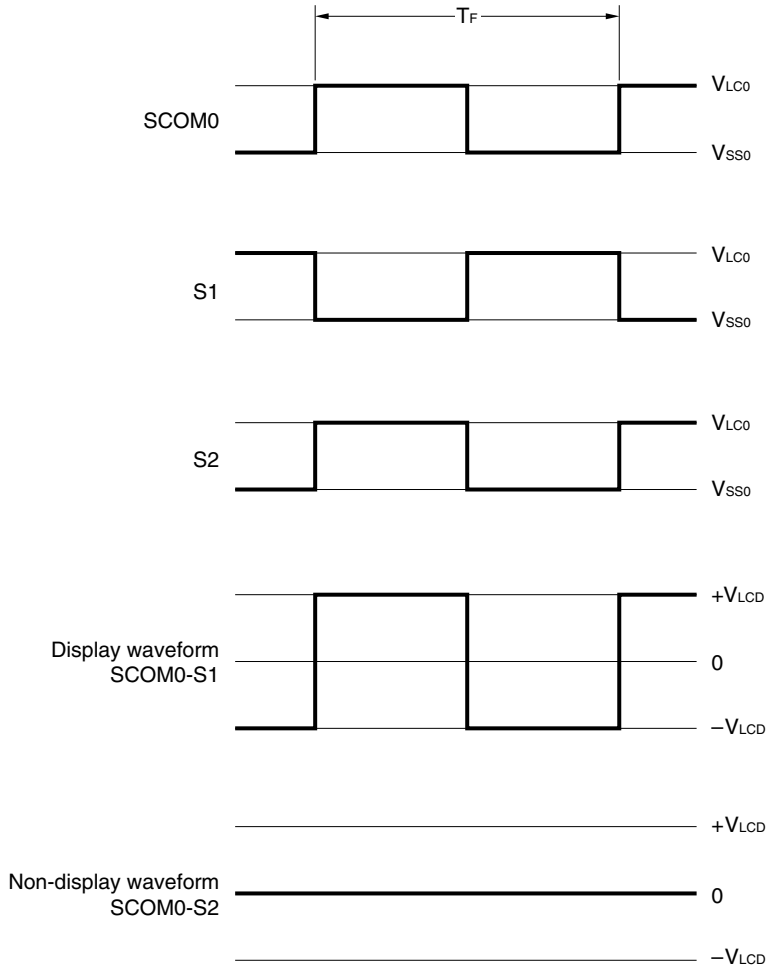


Figure 17-14. Static LCD Drive Waveform Examples



17.8.2 3-time-division display example

Figure 17-16 shows the connection of a 3-time-division type 13-digit LCD panel with the display pattern shown in Figure 17-15 with the μPD780338 Subseries segment signals (S0 to S38) and common signals (COM0 to COM2). The display example is "123456.7890123," and the display data memory contents (addresses FA00H to FA26H) correspond to this.

An explanation is given here taking the example of the eighth digit "6." (5.). In accordance with the display pattern in Figure 17-15, selection and non-selection voltages must be output to pins S21 to S23 as shown in Table 17-9 at the COM0 to COM2 common signal timings.

Table 17-9. Selection and Non-Selection Voltages (COM0 to COM2)

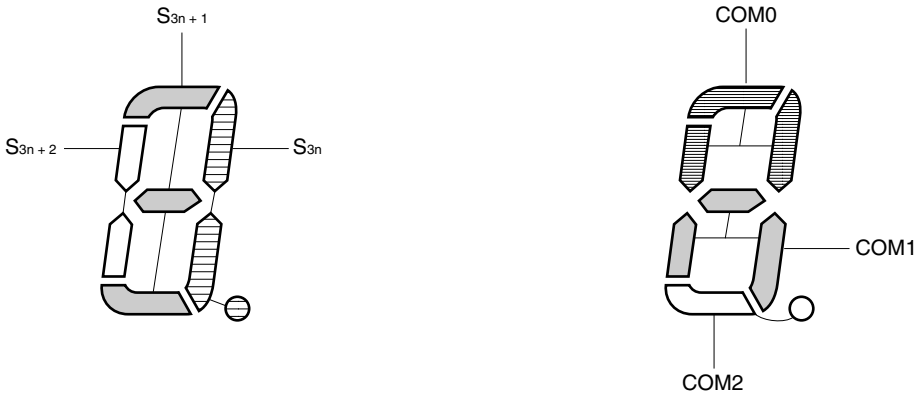
Common \ Segment	S21	S22	S23
COM0	NS	S	S
COM1	S	S	S
COM2	S	S	—

S: Selection, NS: Non-selection

From this, it can be seen that ×110 must be prepared in the display data memory (address FA15H) corresponding to S21.

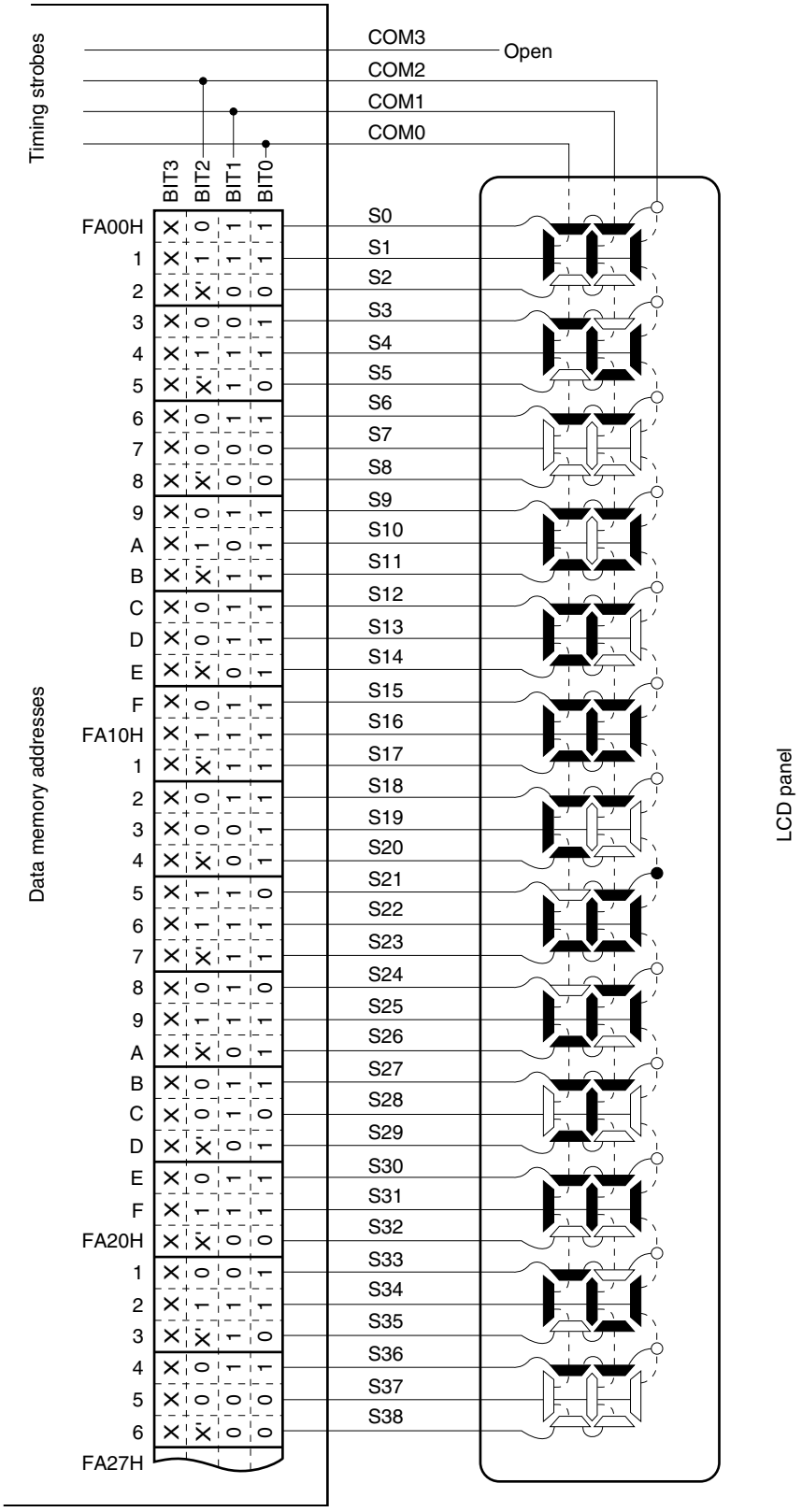
Examples of the LCD drive waveforms between S21 and the common signals are shown in Figure 17-17 (1/3 bias method). When S21 is at the selection voltage at the COM1 selection timing, and S21 is at the selection voltage at the COM2 selection timing, it can be seen that the $+V_{LCD}/-V_{LCD}$ AC square wave, which is the LCD illumination (ON) level, is generated.

Figure 17-15. 3-Time-Division LCD Display Pattern and Electrode Connections



Remark n = 0 to 12

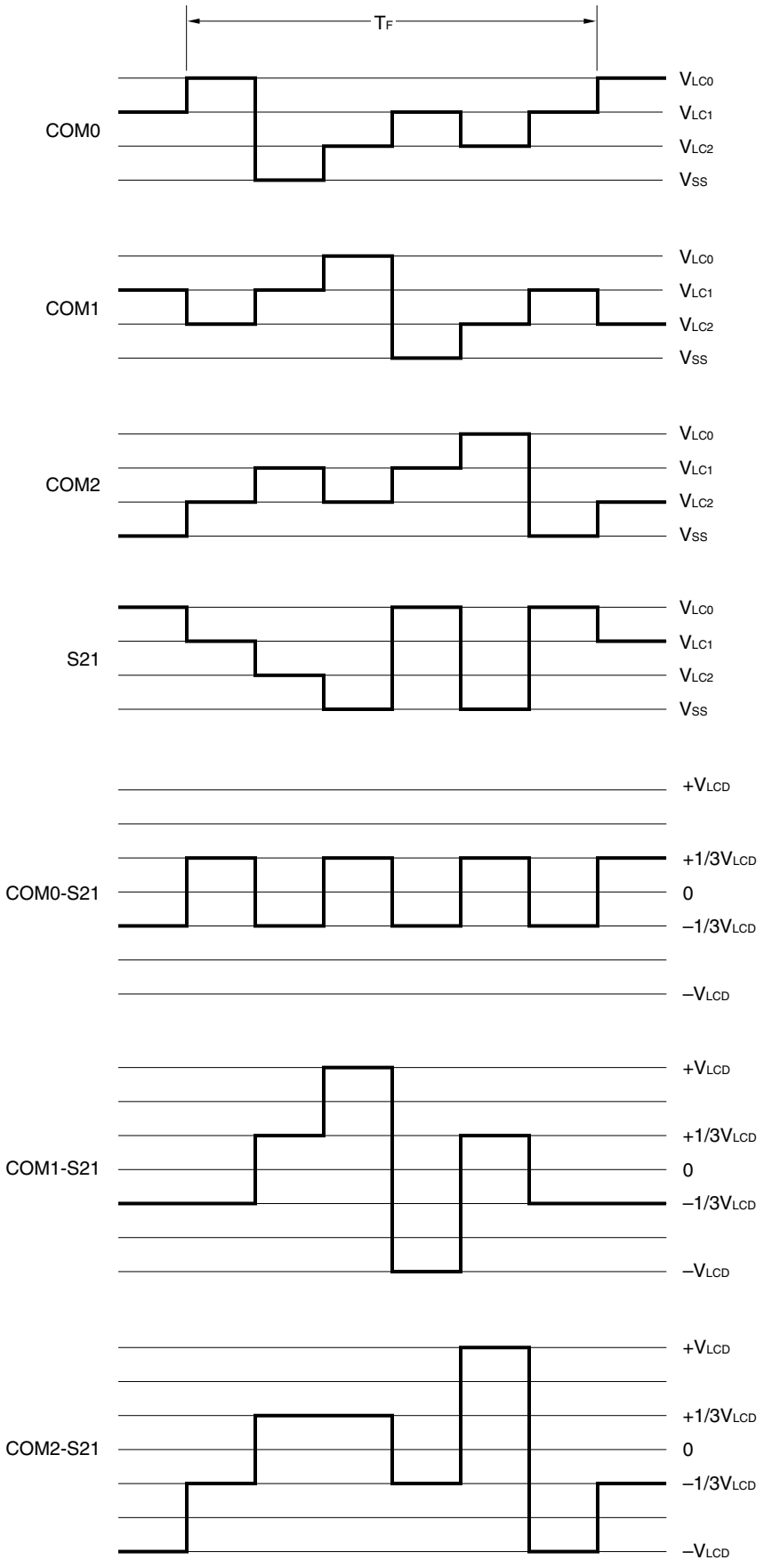
★ **Figure 17-16. 3-Time-Division LCD Panel Connection Example (SDSEL3n = 0: n = 0 to 2)**



Remarks 1. X': Irrelevant bits because they have no corresponding segment in the LCD panel

2. X: Irrelevant bits because this is a 3-time-division display

Figure 17-17. 3-Time-Division LCD Drive Waveform Examples (1/3 Bias Method)



17.8.3 4-time-division display example

Figure 17-19 shows the connection of a 4-time-division type 20-digit LCD panel with the display pattern shown in Figure 17-18 with the μPD780338 Subseries segment signals (S0 to S39) and common signals (COM0 to COM3). The display example is "123456.78901234567890," and the display data memory contents (addresses FA00H to FA27H) correspond to this.

An explanation is given here taking the example of the 15th digit "6." (E). In accordance with the display pattern in Figure 17-18, selection and non-selection voltages must be output to pins S28 and S29 as shown in Table 17-10 at the COM0 to COM3 common signal timings.

Table 17-10. Selection and Non-Selection Voltages (COM0 to COM3)

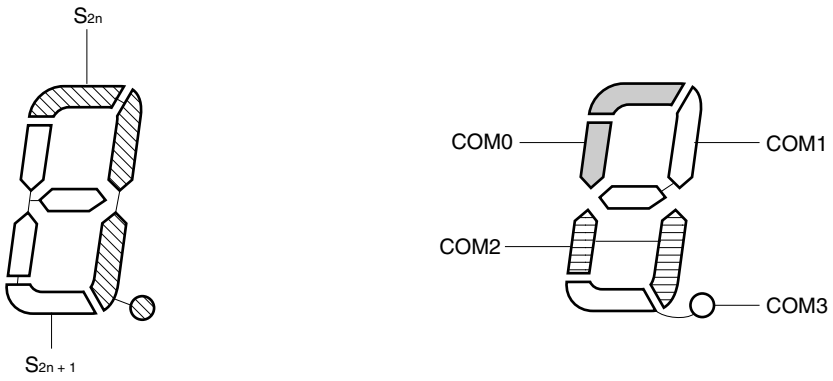
Segment / Common	S28	S29
COM0	S	S
COM1	NS	S
COM2	S	S
COM3	S	S

S: Selection, NS: Non-selection

From this, it can be seen that 1101 must be prepared in the display data memory (address FA1CH) corresponding to S28.

Examples of the LCD drive waveforms between S28 and the COM0 and COM1 signals are shown in Figure 17-20 (for the sake of simplicity, waveforms for COM2 and COM3 have been omitted). When S28 is at the selection voltage at the COM0 selection timing, it can be seen that the $+V_{LCD}/-V_{LCD}$ AC square wave, which is the LCD illumination (ON) level, is generated.

Figure 17-18. 4-Time-Division LCD Display Pattern and Electrode Connections



Remark n = 0 to 18

★ **Figure 17-19. 4-Time-Division LCD Panel Connection Example (SDSEL3n = 0, n = 0 to 2)**

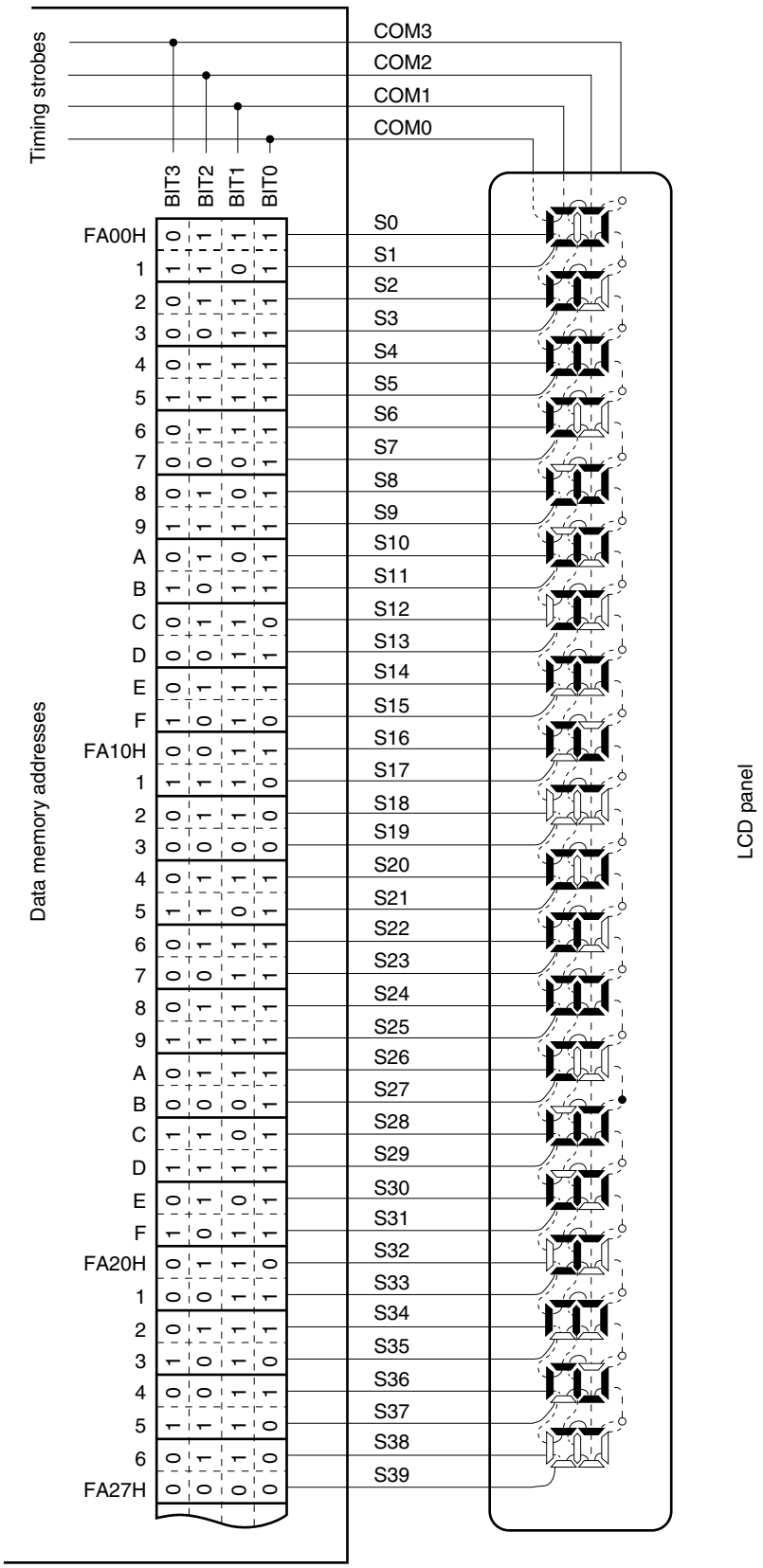
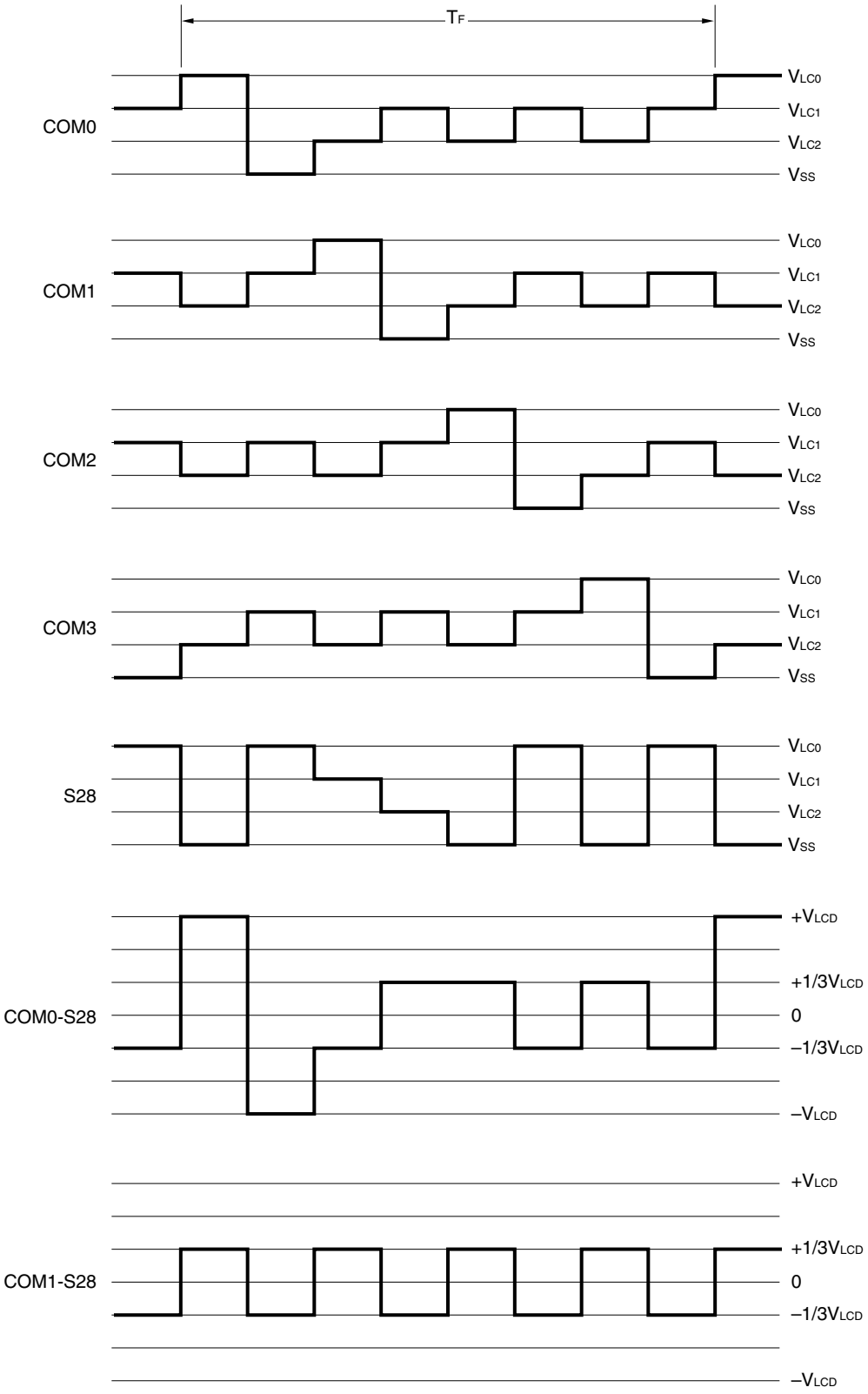


Figure 17-20. 4-Time-Division LCD Drive Waveform Examples (1/3 Bias Method)



17.8.4 Simultaneous driving of static display and dynamic display

Simultaneous driving of static display (S0 to S11) and dynamic display is possible with the μPD780338.

Refer to **Figure 17-6** for register settings.

CHAPTER 18 INTERRUPT FUNCTIONS

18.1 Interrupt Function Types

The following three types of interrupt functions are used.

(1) Non-maskable interrupt

This interrupt is acknowledged unconditionally even in an interrupt disabled state. It does not undergo priority control and is given top priority over all other interrupt requests.

A standby release signal is generated.

One interrupt request from the watchdog timer is incorporated as a non-maskable interrupt.

(2) Maskable interrupts

These interrupts undergo mask control. Maskable interrupts can be divided into a high interrupt priority group and a low interrupt priority group by setting the priority specification flag registers (PR0L, PR0H, PR1L).

Multiple high priority interrupts can be applied to low priority interrupts. If two or more interrupts with the same priority are simultaneously generated, each interrupt has a predetermined priority (see **Table 18-1**).

A standby release signal is generated.

Seven external interrupt requests and 15 internal interrupt requests are incorporated as maskable interrupts.

(3) Software interrupt

This is a vectored interrupt to be generated by executing the BRK instruction. It is acknowledged even in an interrupt disabled state. The software interrupt does not undergo interrupt priority control.

18.2 Interrupt Sources and Configuration

A total of 24 interrupt sources exist among non-maskable, maskable, and software interrupts (see **Table 18-1**).

Remark As the watchdog timer interrupt source (INTWDT), a non-maskable interrupt or maskable interrupt (internal) can be selected.

Table 18-1. Interrupt Source List

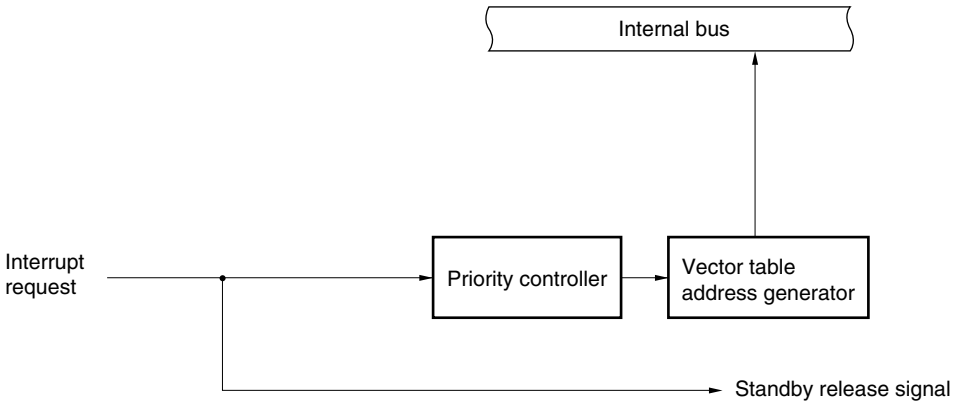
Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Non-maskable	—	INTWDT	Watchdog timer overflow (with watchdog timer mode 1 selected)	Internal	0004H	(A)
Maskable	0	INTWDT	Watchdog timer overflow (with interval timer mode selected)			(B)
	1	INTP0	Pin input edge detection	External	0006H	(C)
	2	INTP1			0008H	
	3	INTP2			000AH	
	4	INTP3			000CH	
	5	INTP4			000EH	
	6	INTP5			0010H	
	7	INTKR	Detection of port 4 falling edge		0012H	(D)
	8	INTSER0	Serial interface (UART0) reception error generation	Internal	0014H	(B)
	9	INTSR0	End of serial interface (UART0) reception		0016H	
	10	INTST0	End of serial interface (UART0) transmission		0018H	
	11	INTCSI1	End of serial interface (CSI1) transfer		001AH	
	12	INTCSI3	End of serial interface (SIO3) transfer		001CH	
	13	INTWTNI0	Reference time interval signal from watch timer		001EH	
	14	INTTM00	Match between TM00 and CR00 (when CR00 is specified as compare register) Detection of TI01 valid edge (when CR00 is specified as capture register)		0020H	
	15	INTTM01	Match between TM00 and CR01 (when CR01 is specified as compare register) Detection of TI00 valid edge (when CR01 is specified as capture register)		0022H	
	16	INTTM4	Match between TM4 and CR4 (when clear & start mode is selected by match between TM4 and CR4)		0024H	
	17	INTTM50	Match between TM50 and CR50		0026H	
	18	INTTM51	Match between TM51 and CR51		0028H	
	19	INTTM52	Match between TM52 and CR52		002AH	
	20	INTAD0	End of A/D converter conversion		002CH	
	21	INTWTN0	Watch timer overflow		002EH	
Software	—	BRK	BRK instruction execution	—	003EH	(E)

Notes 1. The default priority is the priority applicable when two or more maskable interrupts are generated simultaneously. 0 is the highest priority, and 21 is the lowest.

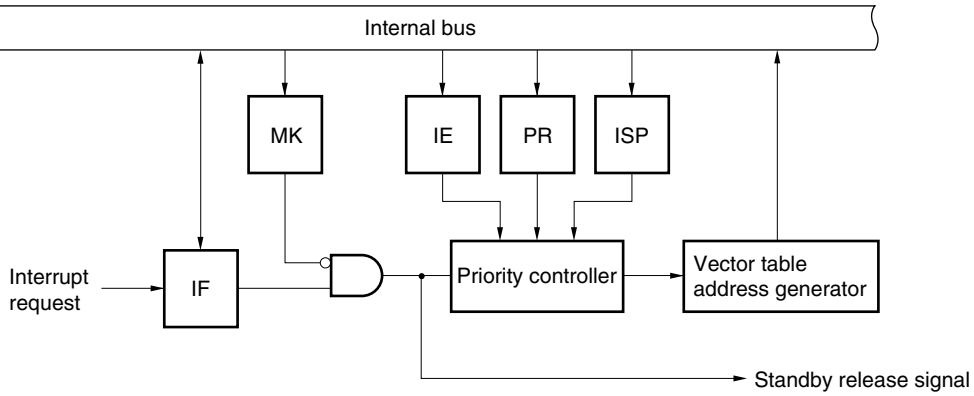
2. Basic configuration types (A) to (E) correspond to (A) to (E) in Figure 18-1.

Figure 18-1. Basic Configuration of Interrupt Function (1/2)

(A) Internal non-maskable interrupt



(B) Internal maskable interrupt



(C) External maskable interrupt (INTP0 to INTP5)

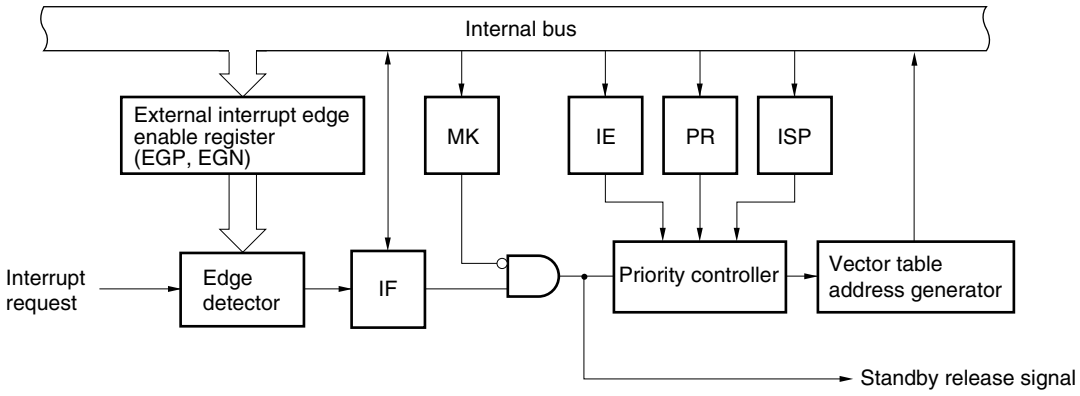
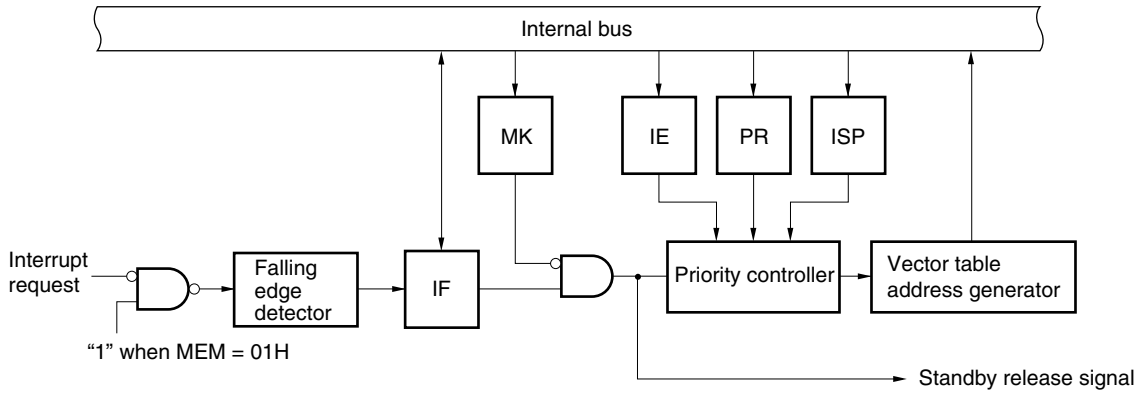
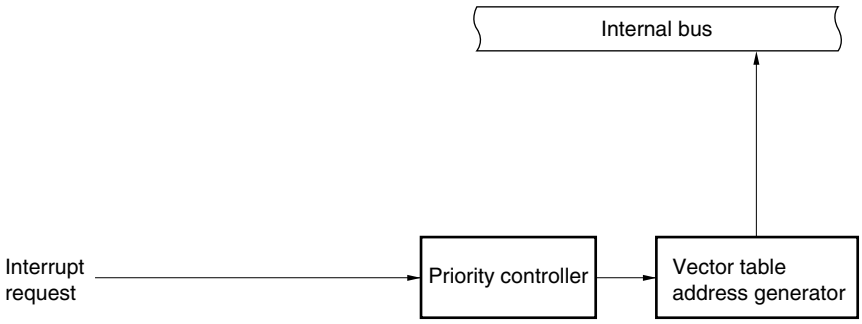


Figure 18-1. Basic Configuration of Interrupt Function (2/2)

(D) External maskable interrupt (INTKR)



(E) Software interrupt



IF: Interrupt request flag
IE: Interrupt enable flag
ISP: In-service priority flag
MK: Interrupt mask flag
PR: Priority specification flag
MEM: Memory expansion mode register

18.3 Interrupt Function Control Registers

The following six types of registers are used to control the interrupt functions.

- Interrupt request flag registers (IF0L, IF0H, IF1L)
- Interrupt mask flag registers (MK0L, MK0H, MK1L)
- Priority specification flag registers (PR0L, PR0H, PR1L)
- External interrupt rising edge enable register (EGP)
- External interrupt falling edge enable register (EGN)
- Program status word (PSW)

Table 18-2 gives a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 18-2. Flags Corresponding to Interrupt Request Sources

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
		Register		Register		Register
INTWDT	WDTIF[Note]	IF0L	WDTMK[Note]	MK0L	WDTPR[Note]	PR0L
INTP0	PIF0		PMK0		PPR0	
INTP1	PIF1		PMK1		PPR1	
INTP2	PIF2		PMK2		PPR2	
INTP3	PIF3		PMK3		PPR3	
INTP4	PIF4		PMK4		PPR4	
INTP5	PIF5		PMK5		PPR5	
INTKR	KRIF		KRMK		KRPR	
INTSER0	SERIF0	IF0H	SERMK0	MK0H	SERPR0	PR0H
INTSR0	SRIF0		SRMK0		SRPR0	
INTST0	STIF0		STMK0		STPR0	
INTCSI1	CSIIF1		CSIMK1		CSIPR1	
INTCSI3	CSIIF3		CSIMK3		CSIPR3	
INTWTNI0	WTNIIF0		WTNIMK0		WTNIPR0	
INTTM00	TMIF00		TMMK00		TMPR00	
INTTM01	TMIF01		TMMK01		TMPR01	
INTTM4	TMIF4	IF1L	TMMK4	MK1L	TMPR4	PR1L
INTTM50	TMIF50		TMMK50		TMPR50	
INTTM51	TMIF51		TMMK51		TMPR51	
INTTM52	TMIF52		TMMK52		TMPR52	
INTAD0	ADIF0		ADMK0		AD0	
INTWTN0	WTNIF0		WTNMK0		WTNPR0	

Note Interrupt control flag when the watchdog timer is used as interval timer

(1) Interrupt request flag registers (IF0L, IF0H, IF1L)

The interrupt request flags are set to 1 when the corresponding interrupt request is generated or an instruction is executed. They are cleared to 0 when an instruction is executed upon acknowledgment of an interrupt request or upon application of $\overline{\text{RESET}}$ input.

IF0L, IF0H, and IF1L are set by a 1-bit or 8-bit memory manipulation instruction. When IF0L and IF0H are combined to form 16-bit register IF0, they are set by a 16-bit memory manipulation instruction.

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

Figure 18-2. Interrupt Request Flag Registers (IF0L, IF0H, IF1L) Format

Address: FFE0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IF0L	KRIF	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	WDTIF

Address: FFE1H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IF0H	TMIF01	TMIF00	WTNIIF0	CSIIF3	CSIIF1	STIF0	SRIF0	SERIF0

Address: FFE2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IF1L	0	0	WTNIF0	ADIF0	TMIF52	TMIF51	TMIF50	TMIF4

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request signal is generated, interrupt request status

Cautions 1. The WDTIF flag is R/W enabled only when the watchdog timer is used as the interval timer. If watchdog timer mode 1 is used, set the WDTIF flag to 0.

2. Be sure to set bits 6 and 7 of IF1L to 0.

3. When operating a timer, serial interface, or A/D converter after standby release, run it once after clearing an interrupt request flag. An interrupt request flag may be set by noise.

4. When an interrupt is acknowledged, the interrupt request flag is automatically cleared and then the interrupt routine is started.

(2) Interrupt mask flag registers (MK0L, MK0H, MK1L)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt service.

MK0L, MK0H, and MK1L are set by a 1-bit or 8-bit memory manipulation instruction. When MK0L and MK0H are combined to form a 16-bit register MK0, they are set by a 16-bit memory manipulation instruction.

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

Figure 18-3. Interrupt Mask Flag Registers (MK0L, MK0H, MK1L) Format

Address: FFE4H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
MK0L	KRMK	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	WDTMK

Address: FFE5H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
MK0H	TMMK01	TMMK00	WTNIMK0	CSIMK3	CSIMK1	STMK0	SRMK0	SERMK0

Address: FFE6H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
MK1L	1	1	WTNMK0	ADMK0	TMMK52	TMMK51	TMMK50	TMMK4

XXMKX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Cautions 1. If the watchdog timer is used in watchdog timer mode 1, the contents of the WDTMK flag become undefined when read.

2. Because port 0 pins have an alternate function as external interrupt request input, when the output level is changed by specifying the output mode of the port function, an interrupt request flag is set. Therefore, 1 should be set in the interrupt mask flag before using the output mode.

3. Be sure to set bits 6 and 7 of MK1L to 1.

(3) Priority specification flag registers (PR0L, PR0H, PR1L)

The priority specification flag registers are used to set the corresponding maskable interrupt priority orders. PR0L, PR0H, and PR1L are set by a 1-bit or 8-bit memory manipulation instruction. If PR0L and PR0H are combined to form 16-bit register PR0, they are set by a 16-bit memory manipulation instruction.

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

Figure 18-4. Priority Specification Flag Registers (PR0L, PR0H, PR1L) Format

Address: FFE8H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PR0L	KRPR	PPR5	PPR4	PPR3	PPR2	PPR1	PPR0	WDTPR

Address: FFE9H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PR0H	TMPR01	TMPR00	WTNIPR0	CSIPR3	CSIPR1	STPR0	SRPR0	SERPR0

Address: FFEAH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PR1L	1	1	WTNPR0	ADPR0	TMPR52	TMPR51	TMPR50	TMPR4

XXPRX	Priority level selection
0	High priority level
1	Low priority level

Cautions 1. When the watchdog timer is used in the watchdog timer mode 1, set 1 in the WDTPR flag.

2. Be sure to set bits 6 and 7 of PR1L to 1.

(4) External interrupt rising edge enable register (EGP), external interrupt falling edge enable register (EGN)

These registers specify the valid edge for INTP0 to INTP5.

EGP and EGN are set by a 1-bit or 8-bit memory manipulation instruction.

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

Figure 18-5. External Interrupt Rising Edge Enable Register (EGP), External Interrupt Falling Edge Enable Register (EGN) Format

Address: FF48H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP	0	0	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

Address: FF49H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN	0	0	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 5)
0	0	Interrupt disable
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

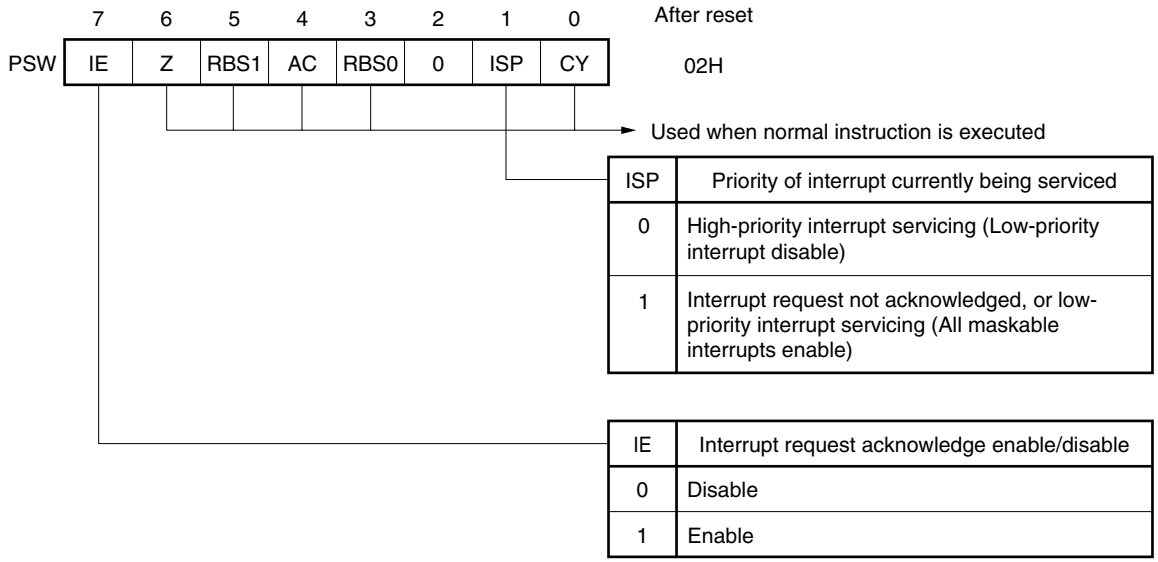
(5) Program status word (PSW)

The program status word is a register to hold the instruction execution result and the current status for an interrupt request. The IE flag to set maskable interrupt enable/disable and the ISP flag to control nesting processing are mapped.

Besides 8-bit read/write, this register can carry out operations with a bit manipulation instruction and dedicated instructions (EI and DI). When a vectored interrupt request is acknowledged, if the BRK instruction is executed, the contents of PSW are automatically saved into a stack and the IE flag is reset to 0. If a maskable interrupt request is acknowledged, the contents of the priority specification flag of the acknowledged interrupt are transferred to the ISP flag. The PSW contents are also saved into the stack with the PUSH PSW instruction. They are reset from the stack with the RETI, RETB, and POP PSW instructions.

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

Figure 18-6. Program Status Word Format



ISP	Priority of interrupt currently being serviced
0	High-priority interrupt servicing (Low-priority interrupt disable)
1	Interrupt request not acknowledged, or low-priority interrupt servicing (All maskable interrupts enable)

IE	Interrupt request acknowledge enable/disable
0	Disable
1	Enable

18.4 Interrupt Servicing Operations

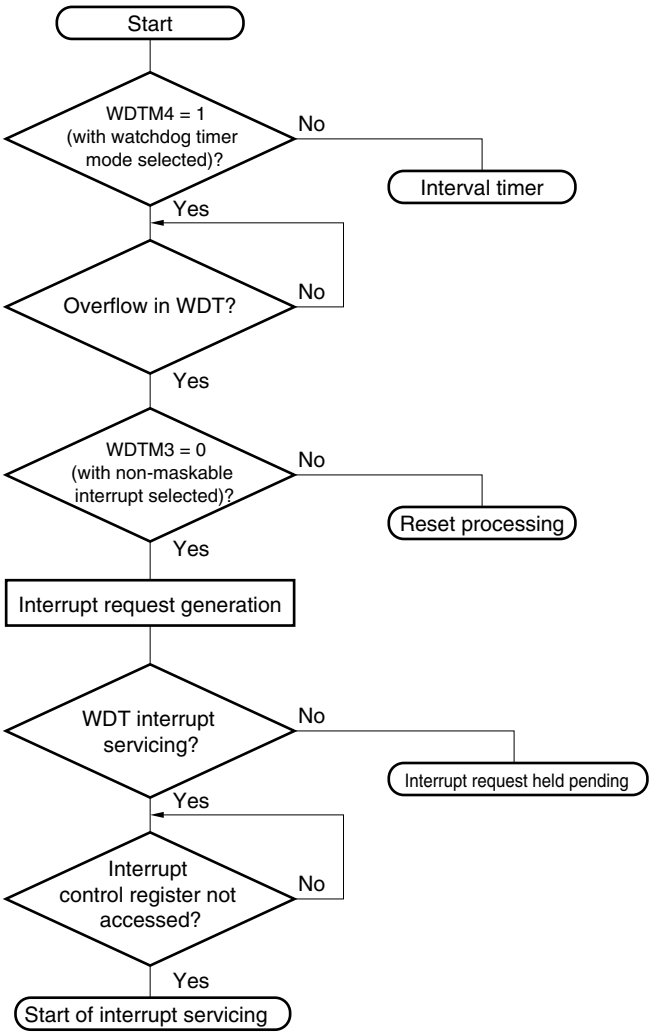
18.4.1 Non-maskable interrupt request acknowledge operation

A non-maskable interrupt request is unconditionally acknowledged even if in an interrupt acknowledge disable state. It does not undergo interrupt priority control and has highest priority over all other interrupts.

If a non-maskable interrupt request is acknowledged, the contents are saved into the stacks in the order of PSW, then PC, the IE flag and ISP flag are reset (0), and the contents of the vector table are loaded into PC and branched.

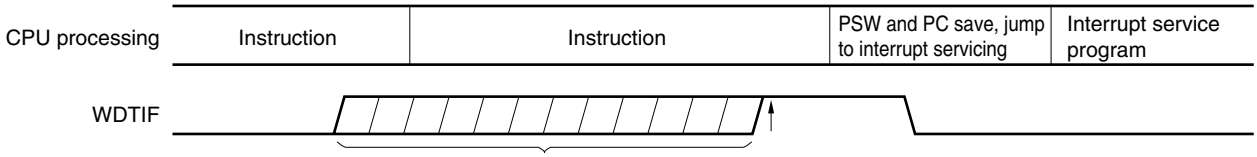
A new non-maskable interrupt request generated during execution of a non-maskable interrupt servicing program is acknowledged after the current execution of the non-maskable interrupt servicing program is terminated (following RETI instruction execution) and one main routine instruction is executed. However, if a new non-maskable interrupt request is generated twice or more during non-maskable interrupt servicing program execution, only one non-maskable interrupt request is acknowledged after termination of the non-maskable interrupt servicing program execution. Figures 18-7, 18-8, and 18-9 show the flowchart of the non-maskable interrupt request generation through acknowledge, acknowledge timing of non-maskable interrupt request, and acknowledge operation at multiple non-maskable interrupt request generation, respectively.

Figure 18-7. Non-Maskable Interrupt Request Generation to Acknowledge Flowchart



WDTM: Watchdog timer mode register
WDT: Watchdog timer

Figure 18-8. Non-Maskable Interrupt Request Acknowledge Timing

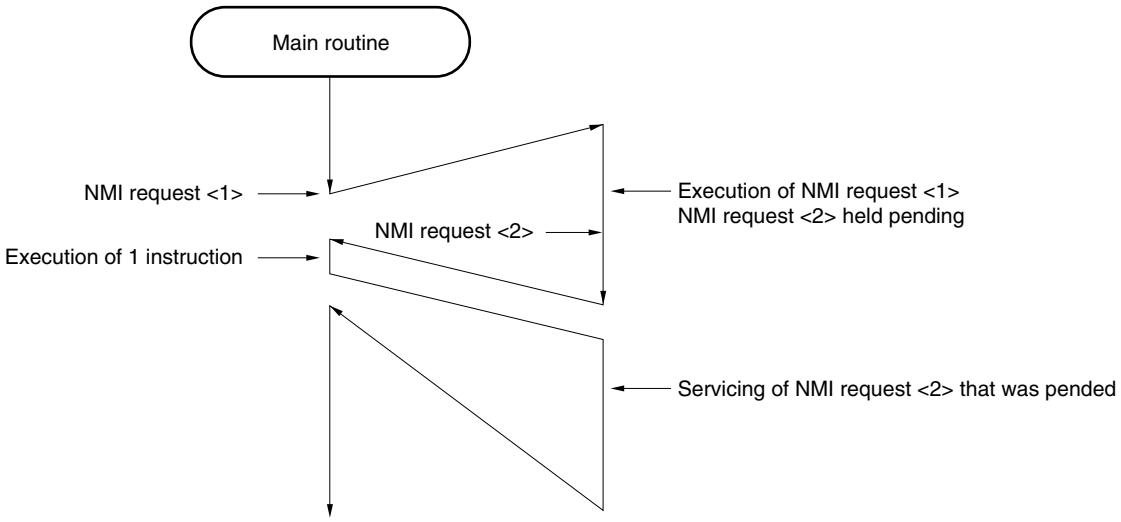


Interrupt request generated during this interval is acknowledged at ↑.

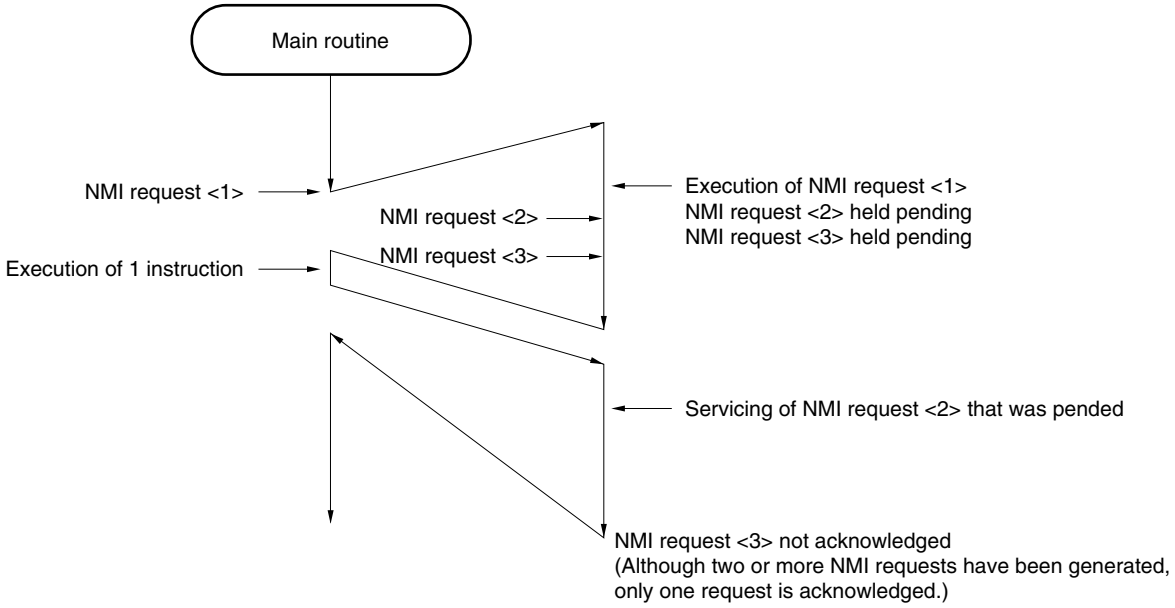
WDTIF: Watchdog timer interrupt request flag

Figure 18-9. Non-Maskable Interrupt Request Acknowledge Operation

(a) If a non-maskable interrupt request is generated during non-maskable interrupt servicing program execution



(b) If two non-maskable interrupt requests are generated during non-maskable interrupt servicing program execution



18.4.2 Maskable interrupt request acknowledge operation

A maskable interrupt request becomes acknowledgeable when an interrupt request flag is set to 1 and the mask (MK) flag corresponding to that interrupt request is cleared to 0. A vectored interrupt request is acknowledged if in the interrupt enable state (when IE flag is set to 1). However, a low-priority interrupt request is not acknowledged during servicing of a higher priority interrupt request (when the ISP flag is reset to 0).

The times from generation of a maskable interrupt request until interrupt servicing is performed are listed in Table 18-3 below.

For the interrupt request acknowledge timing, see **Figures 18-11** and **18-12**.

Table 18-3. Times from Generation of Maskable Interrupt Until Servicing

	Minimum Time	Maximum Time[Note]
When ××PR = 0	7 clocks	32 clocks
When ××PR = 1	8 clocks	33 clocks

Note If an interrupt request is generated just before a divide instruction, the wait time becomes longer.

Remark 1 clock: $1/f_{CPU}$ (f_{CPU}: CPU clock)

If two or more maskable interrupt requests are generated simultaneously, the request with a higher priority level specified in the priority specification flag is acknowledged first. If two or more maskable interrupt requests have the same priority level, the request with the highest default priority is acknowledged first.

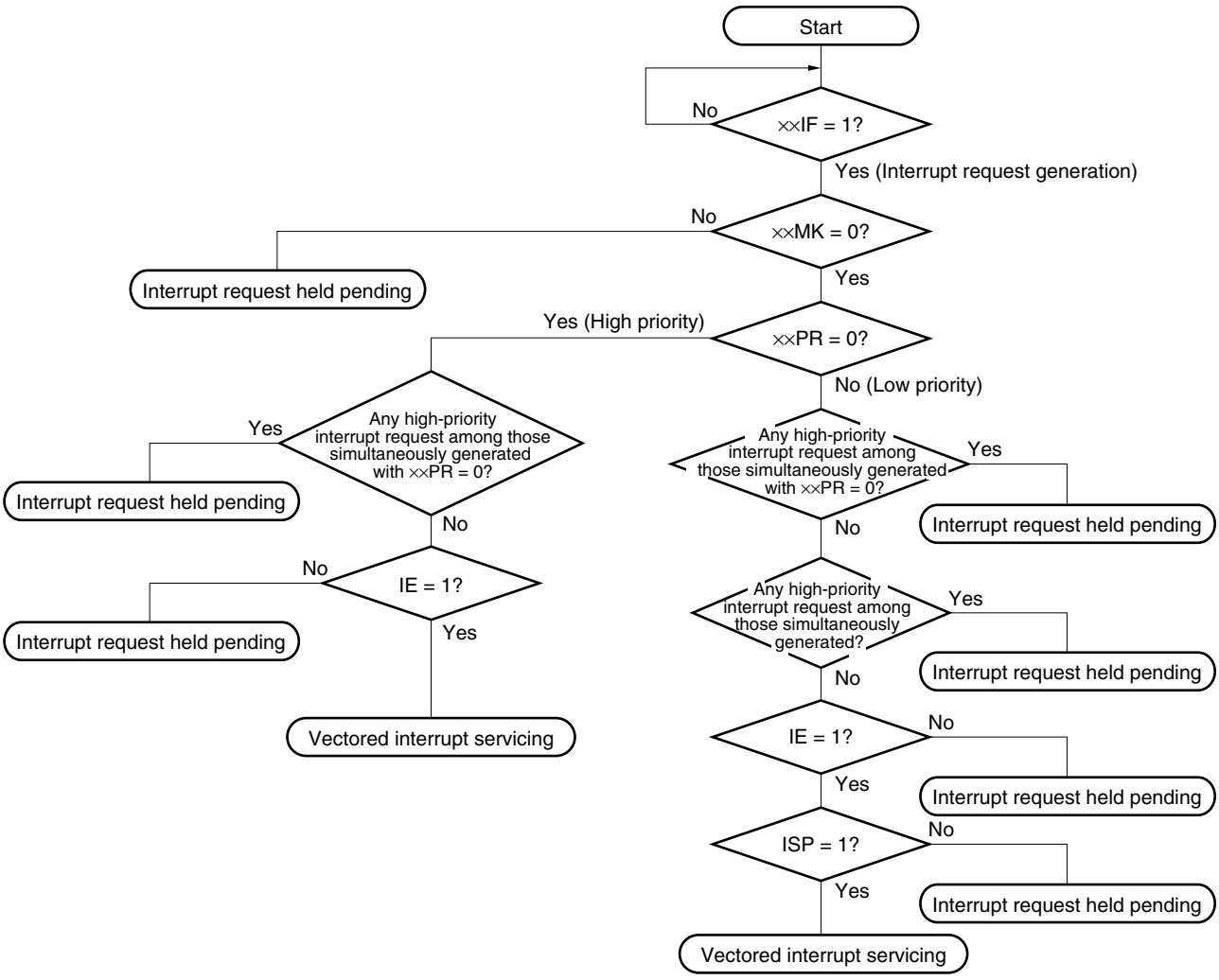
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 18-10 shows the interrupt request acknowledge algorithm.

If a maskable interrupt request is acknowledged, the contents are saved into the stacks in the order of PSW, then PC, the IE flag is reset (0), and the contents of the priority specification flag corresponding to the acknowledged interrupt are transferred to the ISP flag. Further, the vector table data determined for each interrupt request is loaded into PC and branched.

Return from an interrupt is possible with the RETI instruction.

Figure 18-10. Interrupt Request Acknowledge Processing Algorithm



××IF: Interrupt request flag

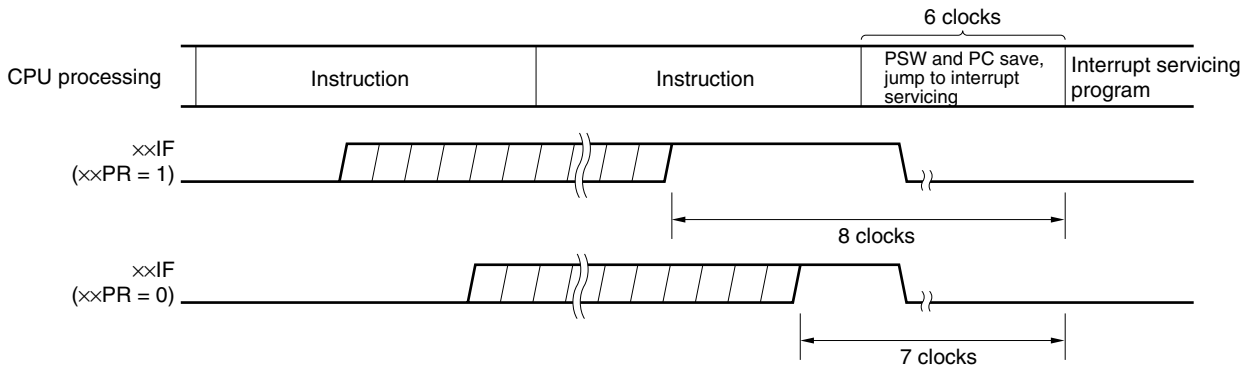
××MK: Interrupt mask flag

××PR: Priority specification flag

IE: Flag that controls acknowledge of maskable interrupt request (1 = enable, 0 = disable)

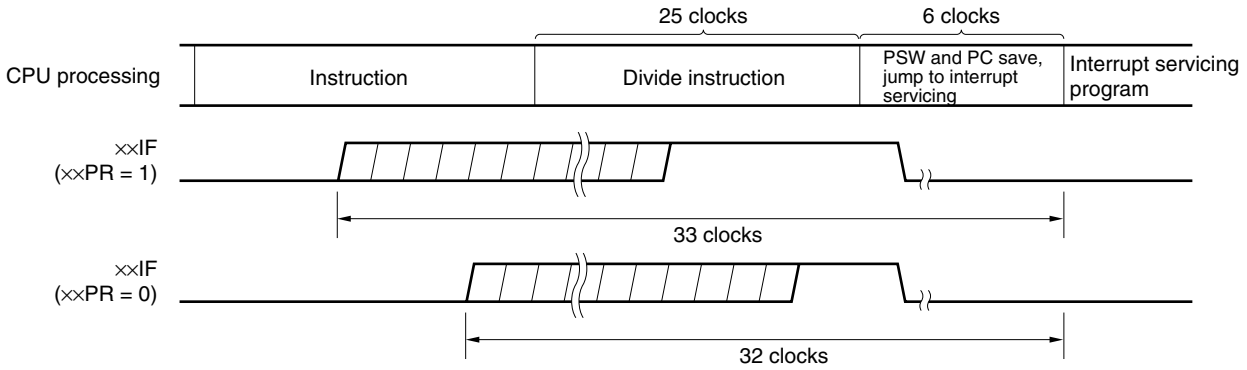
ISP: Flag that indicates the priority level of the interrupt currently being serviced (0 = high-priority interrupt servicing, 1 = no interrupt request acknowledged, or low-priority interrupt servicing)

Figure 18-11. Interrupt Request Acknowledge Timing (Minimum Time)



Remark 1 clock: 1/f_{CPU} (f_{CPU}: CPU clock)

Figure 18-12. Interrupt Request Acknowledge Timing (Maximum Time)



Remark 1 clock: 1/f_{CPU} (f_{CPU}: CPU clock)

18.4.3 Software interrupt request acknowledge operation

A software interrupt request is acknowledged by BRK instruction execution. Software interrupts cannot be disabled.

If a software interrupt request is acknowledged, the contents are saved into the stacks in the order of the program status word (PSW), then program counter (PC), the IE flag is reset (0), and the contents of the vector table (003EH, 003FH) are loaded into PC and branched.

Return from a software interrupt is possible with the RETB instruction.

Caution Do not use the RETI instruction for returning from the software interrupt.

18.4.4 Nesting processing

Nesting occurs when another interrupt request is acknowledged during execution of an interrupt.

Nesting does not occur unless the interrupt request acknowledge enable state is selected (IE = 1) (except non-maskable interrupts). Also, when an interrupt request is acknowledged, interrupt request acknowledge becomes disabled (IE = 0). Therefore, to enable nesting, it is necessary to set (1) the IE flag with the EI instruction during interrupt servicing to enable interrupt acknowledge.

Moreover, even if interrupts are enabled, nesting may not be enabled, this being subject to interrupt priority control. Two types of priority control are available: default priority control and programmable priority control. Programmable priority control is used for nesting.

In the interrupt enable state, if an interrupt request with a priority equal to or higher than that of the interrupt currently being serviced is generated, it is acknowledged for nesting. If an interrupt with a priority lower than that of the interrupt currently being serviced is generated during interrupt servicing, it is not acknowledged for nesting.

Interrupt requests that are not enabled because of the interrupt disable state or they have a lower priority are held pending. When servicing of the current interrupt ends, the pended interrupt request is acknowledged following execution of one main processing instruction execution.

Nesting is not possible during non-maskable interrupt servicing.

Table 18-4 shows interrupt requests enabled for nesting and Figure 18-13 shows nesting examples.

Table 18-4. Interrupt Request Enabled for Nesting During Interrupt Servicing

Nesting Request / Interrupt Being Serviced		Non-Maskable Interrupt Request	Maskable Interrupt Request			
			PR = 0		PR = 1	
			IE = 1	IE = 0	IE = 1	IE = 0
Non-maskable interrupt		×	×	×	×	×
Maskable interrupt	ISP = 0	○	○	×	×	×
	ISP = 1	○	○	×	○	×
Software interrupt		○	○	×	○	×

Remarks 1. ○: Nesting enabled

2. ×: Nesting disabled

3. ISP and IE are flags contained in PSW.

ISP = 0: An interrupt with higher priority is being serviced.

ISP = 1: No interrupt request has been acknowledged, or an interrupt with a lower priority is being serviced.

IE = 0: Interrupt request acknowledge is disabled.

IE = 1: Interrupt request acknowledge is enabled.

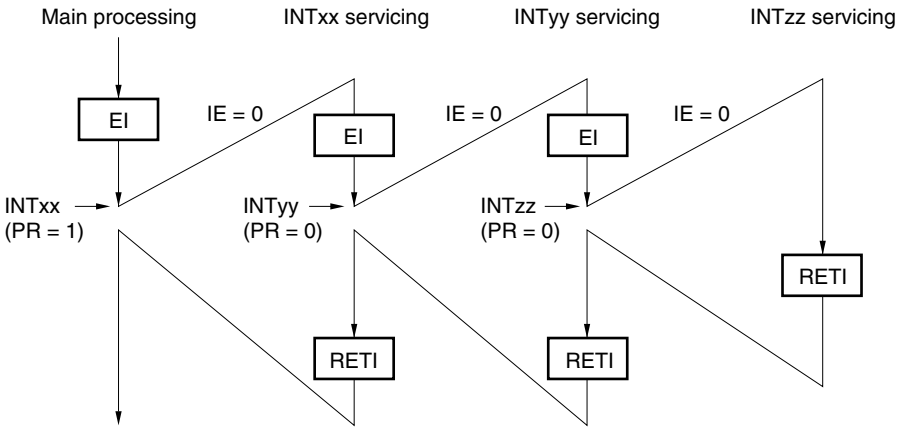
4. PR is a flag contained in PR0L, PR0H, and PR1L.

PR = 0: Higher priority level

PR = 1: Lower priority level

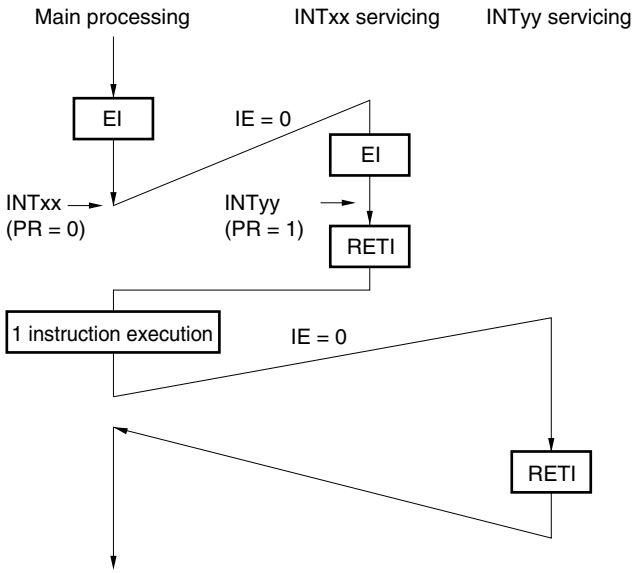
Figure 18-13. Nesting Examples (1/2)

Example 1. Nesting occurs twice



During servicing of interrupt INTxx, two interrupt requests, INTyy and INTzz, are acknowledged, and nesting takes place. Before each interrupt request is acknowledged, the EI instruction must always be issued to enable interrupt request acknowledge.

Example 2. Nesting does not occur due to priority control

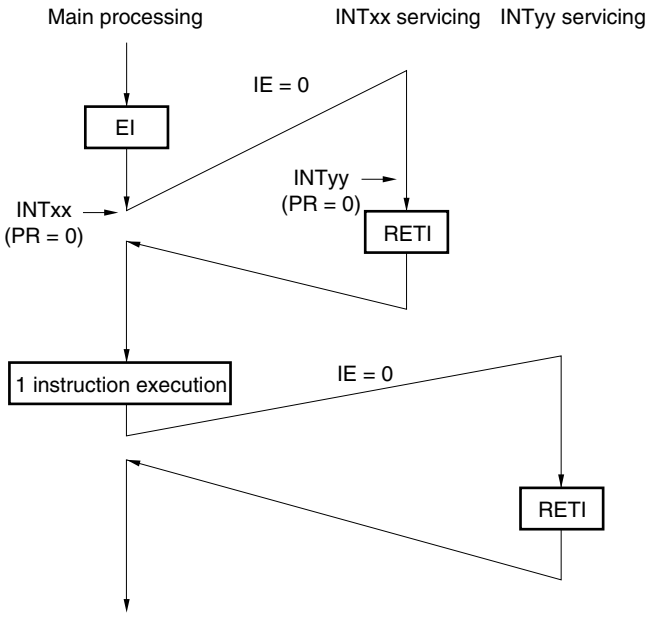


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and nesting does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 0: Higher priority level
PR = 1: Lower priority level
IE = 0: Interrupt request acknowledge disabled

Figure 18-13. Nesting Examples (2/2)

Example 3. Nesting does not occur because interrupt is not enabled



Interrupt is not enabled during servicing of interrupt INTxx (EI instruction is not issued), therefore, interrupt request INTyy is not acknowledged and nesting does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 0: Higher priority level
IE = 0: Interrupt request acknowledge disabled

18.4.5 Interrupt request hold

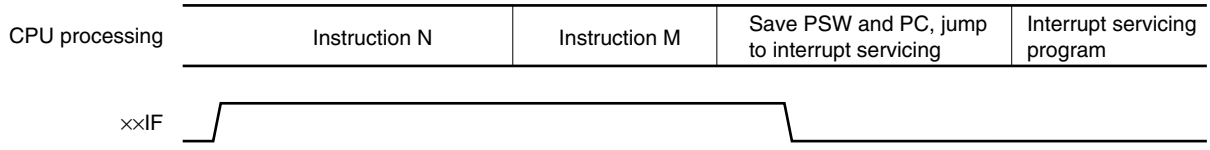
There are instructions where, even if an interrupt request is issued for them while another instruction is executed, request acknowledge is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- MOV PSW, #byte
- MOV A, PSW
- MOV PSW, A
- MOV1 PSW.bit, CY
- MOV1 CY, PSW.bit
- AND1 CY, PSW.bit
- OR1 CY, PSW.bit
- XOR1 CY, PSW.bit
- SET1 PSW.bit
- CLR1 PSW.bit
- RETB
- RETI
- PUSH PSW
- POP PSW
- BT PSW.bit, $addr16
- BF PSW.bit, $addr16
- BTCLR PSW.bit, $addr16
- EI
- DI
- Manipulate instructions for the IF0L, IF0H, IF1L, MK0L, MK0H, MK1L, PR0L, PR0H, and PR1L registers.

Caution The BRK instruction is not one of the above-listed interrupt request hold instructions. However, the software interrupt activated by executing the BRK instruction causes the IE flag to be cleared to 0. Therefore, even if a maskable interrupt request is generated during execution of the BRK instruction, the interrupt request is not acknowledged. However, a non-maskable interrupt request is acknowledged.

Figure 18-14 shows the timing with which interrupt requests are held pending.

Figure 18-14. Interrupt Request Hold



Remarks 1. Instruction N: Interrupt request hold instruction

2. Instruction M: Instruction other than interrupt request hold instruction

3. The ××PR (priority level) values do not affect the operation of ××IF (interrupt request).

CHAPTER 19 STANDBY FUNCTION

19.1 Standby Function and Configuration

19.1.1 Standby function

The standby function is designed to decrease power consumption of the system. The following two modes are available.

(1) HALT mode

HALT instruction execution sets the HALT mode. The HALT mode is intended to stop the CPU operation clock. The system clock oscillator continues oscillating. In this mode, current consumption is not decreased as much as in the STOP mode. However, the HALT mode is effective to restart operation immediately upon interrupt request and to carry out intermittent operations such as watch applications.

(2) STOP mode

STOP instruction execution sets the STOP mode. In the STOP mode, the main system clock oscillator stops, stopping the whole system, thereby considerably reducing the CPU power consumption.

Data memory low-voltage hold (down to V_{DD} = 1.6 V) is possible. Thus, the STOP mode is effective to hold data memory contents with ultra-low current consumption.

Because this mode can be cleared upon interrupt request, it enables intermittent operations to be carried out. However, because a wait time is required to secure an oscillation stabilization time after the STOP mode is cleared, select the HALT mode if it is necessary to start processing immediately upon interrupt request.

In either of these two modes, all the contents of registers, flags and data memory just before the standby mode is set are held. The I/O port output latch and output buffer statuses are also held.

Cautions 1. The STOP mode can be used only when the system operates with the main system clock (subsystem clock oscillation cannot be stopped). The HALT mode can be used with either the main system clock or the subsystem clock.

2. When operation is transferred to the STOP mode, be sure to stop the peripheral hardware operation and execute the STOP instruction.

3. The following sequence is recommended for power consumption reduction of the A/D converter when the standby function is used: First clear bit 7 (ADCS0) of the A/D converter mode register 0 (ADM0) to 0 to stop the A/D conversion operation, and then execute the HALT or STOP instruction.

19.1.2 Standby function control register

The wait time after the STOP mode is cleared upon interrupt request is controlled with the oscillation stabilization time select register (OSTS).

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

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

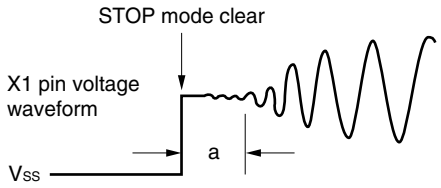
Figure 19-1. Oscillation Stabilization Time Select Register (OSTS) Format

Address: FFFAH After reset: 04H R/W

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

OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection
0	0	0	$2^{12}/f_X$ (410 μs)
0	0	1	$2^{14}/f_X$ (1.64 ms)
0	1	0	$2^{15}/f_X$ (3.28 ms)
0	1	1	$2^{16}/f_X$ (6.55 ms)
1	0	0	$2^{17}/f_X$ (13.1 ms)
Other than above			Setting prohibited

Caution The wait time after the STOP mode is cleared does not include the time (see "a" in the illustration below) from STOP mode clear to clock oscillation start. The time is not included either by $\overline{\text{RESET}}$ input or by interrupt request generation.



Remarks 1. f_X: Main system clock oscillation frequency

2. Values in parentheses are for operation with f_X = 10 MHz.

19.2 Standby Function Operations

19.2.1 HALT mode

(1) HALT mode setting and operating statuses

The HALT mode is set by executing the HALT instruction. It can be set with the main system clock or the subsystem clock.

The operating statuses in the HALT mode are described below.

Table 19-1. HALT Mode Operating Statuses

HALT Mode Setting / Item	During HALT Instruction Execution Using Main System Clock		During HALT Instruction Execution Using Subsystem Clock	
	Without Subsystem Clock[Note 1]	With Subsystem Clock[Note 2]	With Main System Clock Oscillation	With Main System Clock Oscillation Stopped
Clock generator	Both main system clock and subsystem clock can be oscillated. Clock supply to CPU stops.			
CPU	Operation stops.			
Port (output latch)	Status before HALT mode setting is held.			
16-bit timer/event counter 0	Operable			Operation stops.
16-bit timer/event counter 4	Operable			Operable when TI4 is selected as count clock.
8-bit timer/event counters 50, 51, 52	Operable			Operable when TI50, TI51, and TI52 are selected as count clock.
Watch timer	Operable when $f_X/2^8$ is selected as count clock.	Operable		Operable when f_{XT} is selected as count clock.
Watchdog timer	Operable		Operation stops.	
Clock output	Operable			Operable when f_{XT} is selected as count clock.
Buzzer output				Operation stops.
A/D converter	Operation stops.			
D/A converter	Operation stops.			
Serial interface UART0	Operable			Operation stops.
Serial interface CSI1				Operable with external SCK.
Serial interface SIO3				
LCD controller/driver	Operable when $f_X/2^6$ to $f_X/2^8$ is selected as count clock.	Operable		Operable when f_{XT} is selected as count clock.

Notes 1. Including case when external clock is not supplied.

2. Including case when external clock is supplied.

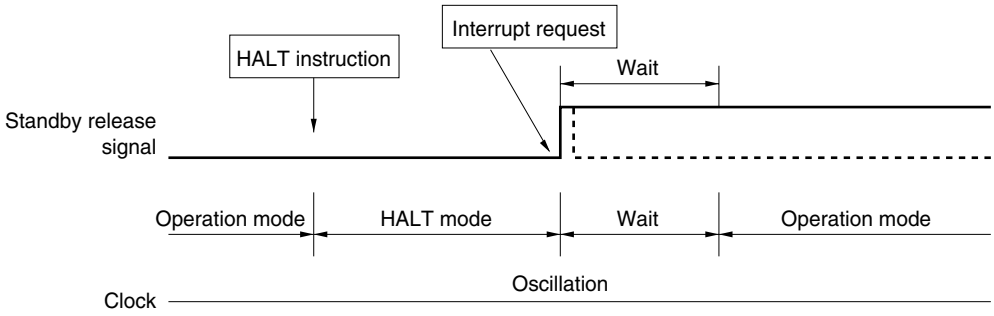
(2) HALT mode release

The HALT mode can be released with the following three types of sources.

(a) Release by unmasked interrupt request

When an unmasked interrupt request is generated, the HALT mode is released. If interrupt acknowledge is enabled, vectored interrupt service is carried out. If interrupt acknowledge is disabled, the next address instruction is executed.

Figure 19-2. HALT Mode Release by Interrupt Request Generation



Remarks 1. The broken line indicates the case when the interrupt request which has released the standby mode is acknowledged.

2. Wait times are as follows:

- When vectored interrupt service is carried out: 8 or 9 clocks
- When vectored interrupt service is not carried out: 2 or 3 clocks

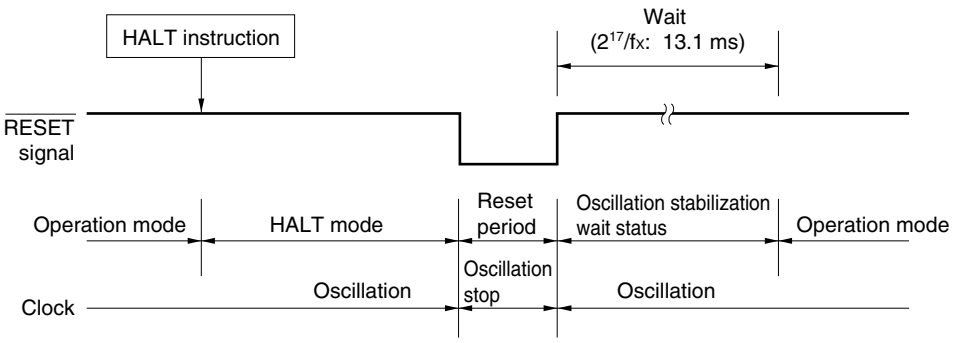
(b) Release by non-maskable interrupt request

When a non-maskable interrupt request is generated, the HALT mode is released and vectored interrupt service is carried out whether interrupt acknowledge is enabled or disabled.

(c) Release by $\overline{\text{RESET}}$ input

When $\overline{\text{RESET}}$ signal is input, HALT mode is released. And, as in the case with normal reset operation, a program is executed after branch to the reset vector address.

Figure 19-3. HALT Mode Release by $\overline{\text{RESET}}$ Input



Remarks 1. fx: Main system clock oscillation frequency

2. Values in parentheses are for operation with fx = 10 MHz.

Table 19-2. Operation After HALT Mode Release

Release Source	MK××	PR××	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Next address instruction execution
	0	0	1	×	Interrupt service execution
	0	1	0	1	Next address instruction execution
	0	1	×	0	
	0	1	1	1	Interrupt service execution
	1	×	×	×	HALT mode hold
Non-maskable interrupt request	—	—	×	×	Interrupt service execution
$\overline{\text{RESET}}$ input	—	—	×	×	Reset processing

×: Don't care

19.2.2 STOP mode

(1) STOP mode setting and operating statuses

The STOP mode is set by executing the STOP instruction. It can be set only with the main system clock.

Cautions 1. When the STOP mode is set, the X2 pin is internally connected to V_{DD1} via a pull-up resistor to minimize the leakage current at the crystal oscillator. Thus, do not use the STOP mode in a system where an external clock is used for the main system clock.

2. Because the interrupt request signal is used to clear the standby mode, if there is an interrupt source with the interrupt request flag set and the interrupt mask flag reset, the standby mode is immediately cleared if set. Thus, the STOP mode is reset to the HALT mode immediately after execution of the STOP instruction. After the wait set using the oscillation stabilization time select register (OSTS), the operation mode is set.

The operating statuses in the STOP mode are described below.

Table 19-3. STOP Mode Operating Statuses

STOP Mode Setting / Item	With Subsystem Clock	Without Subsystem Clock
Clock generator	Only main system clock oscillation is stopped.	
CPU	Operation stops.	
Port (output latch)	Status before STOP mode setting is held.	
16-bit timer/event counter 0	Operation stops.	
16-bit timer/event counter 4	Operable when TI4 is selected as count clock.	
8-bit timer/event counters 50, 51, 52	Operable when TI50, TI51, and TI52 are selected as count clock.	
Watch timer	Operable when f_{XT} is selected as count clock.	Operation stops.
Watchdog timer	Operation stops.	
Clock output	PCL is low	
Buzzer output	BUZ is low	
A/D converter	Operation stops.	
D/A converter		
Serial interface UART0	Operation stops (transmit shift register 0 (TXS0), receive shift register 0 (RX0), and receive buffer register 0 (RXB0) hold the value just before the clock stop).	
Serial interface CSI1	Operable only when externally input clock is selected as serial clock.	
Serial interface SIO3		
LCD controller/driver	Operable when f_{XT} is selected as count clock.	Operation stops.

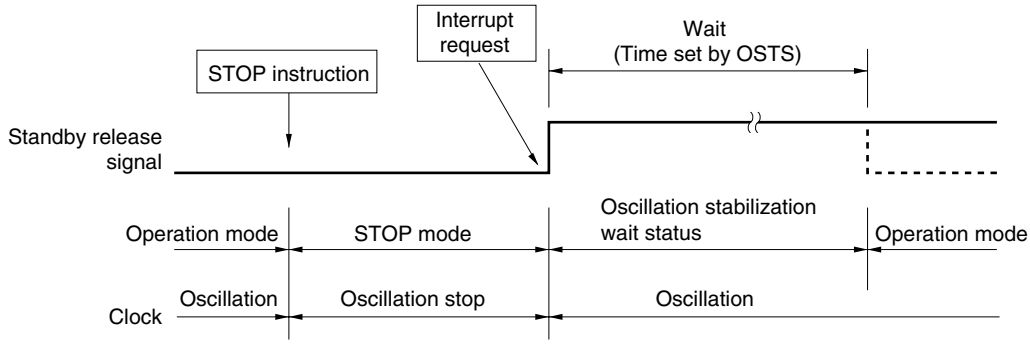
(2) STOP mode release

The STOP mode can be released by the following two types of sources.

(a) Release by unmasked interrupt request

When an unmasked interrupt request is generated, the STOP mode is released. If interrupt acknowledge is enabled after the lapse of oscillation stabilization time, vectored interrupt service is carried out. If interrupt acknowledge is disabled, the next address instruction is executed.

Figure 19-4. STOP Mode Release by Interrupt Request Generation

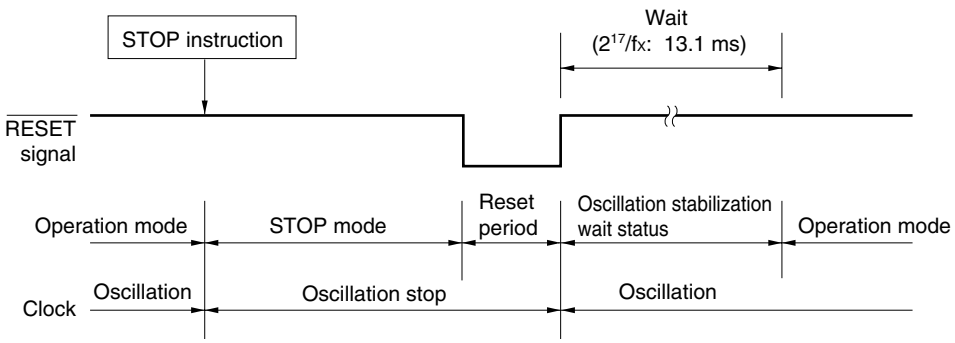


Remark The broken line indicates the case when the interrupt request which has cleared the standby status is acknowledged.

(b) Release by $\overline{\text{RESET}}$ input

The STOP mode is released when $\overline{\text{RESET}}$ signal is input, and after the lapse of oscillation stabilization time, reset operation is carried out.

Figure 19-5. STOP Mode Release by $\overline{\text{RESET}}$ Input



Remarks 1. fx: Main system clock oscillation frequency

2. Values in parentheses are for operation with fx = 10 MHz.

Table 19-4. Operation After STOP Mode Release

Release Source	MK××	PR××	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Next address instruction execution
	0	0	1	×	Interrupt service execution
	0	1	0	1	Next address instruction execution
	0	1	×	0	
	0	1	1	1	Interrupt service execution
	1	×	×	×	STOP mode hold
$\overline{\text{RESET}}$ input	—	—	×	×	Reset processing

×: Don't care

CHAPTER 20 RESET FUNCTION

20.1 Reset Function

The following two operations are available to generate the reset signal.

(1) External reset input via $\overline{\text{RESET}}$ pin
(2) Internal reset by watchdog timer runaway time detection

External reset and internal reset have no functional differences. In both cases, program execution starts at the address at 0000H and 0001H by $\overline{\text{RESET}}$ input.

When a low level is input to the $\overline{\text{RESET}}$ pin or the watchdog timer overflows, a reset is applied and each hardware is set to the status shown in Table 20-1. Each pin has high impedance during reset input or during oscillation stabilization time just after reset clear.

When a high level is input to the $\overline{\text{RESET}}$ pin, the reset is cleared and program execution starts after the lapse of oscillation stabilization time $2^{17}/f_X$. The reset applied by watchdog timer overflow is automatically cleared after a reset and program execution starts after the lapse of oscillation stabilization time $2^{17}/f_X$ (see **Figures 20-2** to **20-4**).

Cautions 1. For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin.
2. During reset input, main system clock oscillation remains stopped but subsystem clock oscillation continues.
3. When the STOP mode is cleared by reset, the STOP mode contents are held during reset input. However, the port pin becomes high-impedance.

Figure 20-1. Reset Function Block Diagram

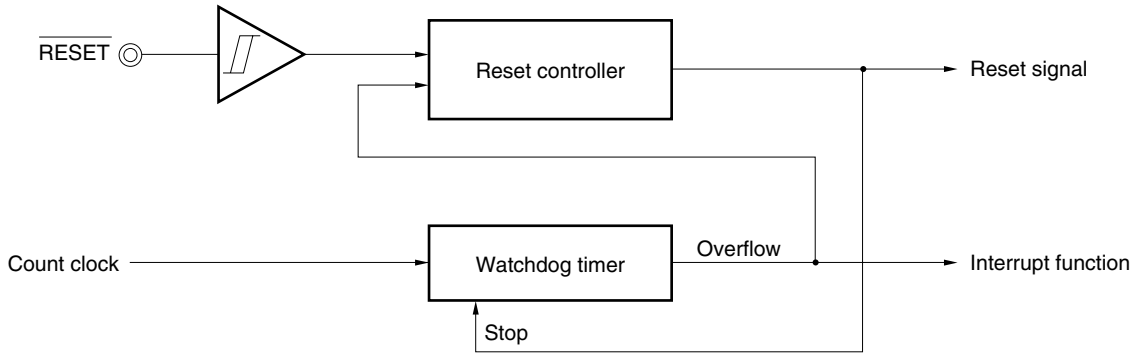


Figure 20-2. Timing of Reset by $\overline{\text{RESET}}$ Input

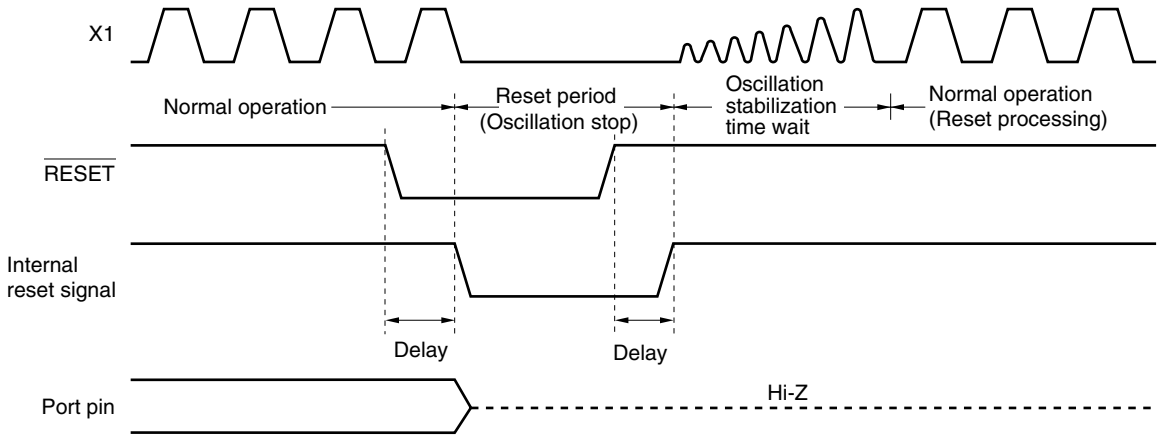


Figure 20-3. Timing of Reset Due to Watchdog Timer Overflow

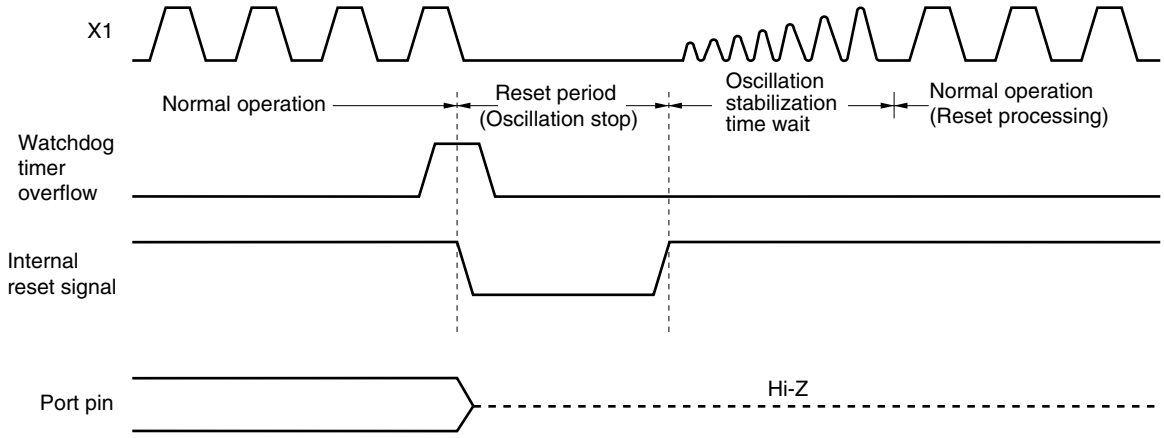


Figure 20-4. Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input

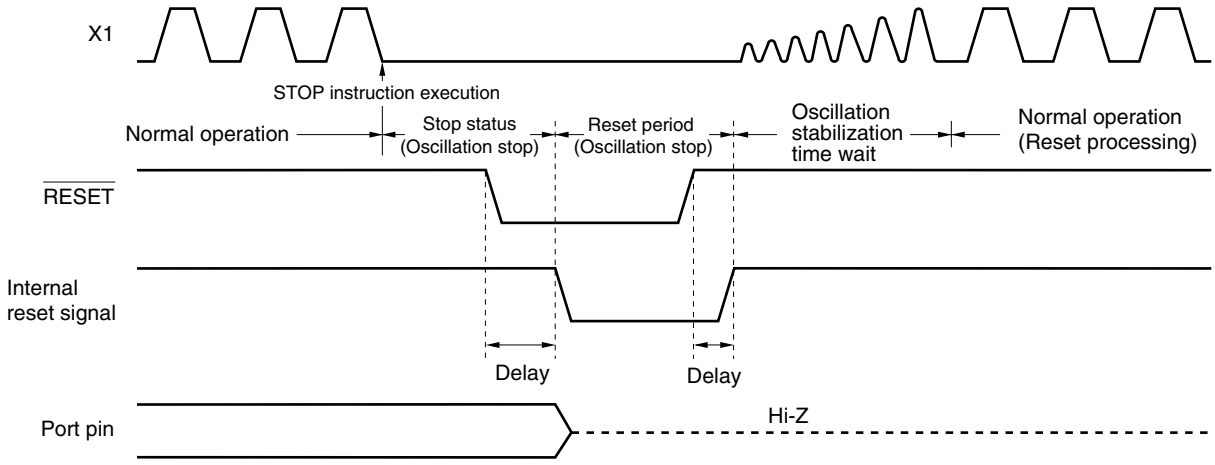


Table 20-1. Hardware Statuses After Reset (1/2)

Hardware		Status After Reset
Program counter (PC)Note 1		Contents of reset vector table (0000H, 0001H) are set.
Stack pointer (SP)		Undefined
Program status word (PSW)		02H
RAM	Data memory	UndefinedNote 2
	General-purpose register	UndefinedNote 2
Port (output latch)		00H
Port mode registers 0, 2 to 7, 8Note 5, 9Note 5, 12 (PM0, PM2 to PM7, PM8Note 5, PM9Note 5, PM12)		FFH
Pull-up resistor option registers 0, 2 to 7, 12 (PU0, PU2 to PU7, PU12)		00H
Processor clock control register (PCC)		04H
Memory size switching register (IMS)		CFHNote 3
Internal expansion RAM size switching register (IXS)		0CHNote 4
Memory expansion mode register (MEM)		00H
Key return switching register (KRSEL)		00H
Pin function switching registers 8, 9 (PF8, PF9)Note 5		00H
Oscillation stabilization time select register (OSTS)		04H
16-bit timer/event counter 0	Timer counter 0 (TM0)	0000H
	Capture/compare registers 00, 01 (CR00, CR01)	Undefined
	Prescaler mode register 0 (PRM0)	00H
	Mode control register 0 (TMC0)	00H
	Capture/compare control register 0 (CRC0)	00H
	Output control register 0 (TOC0)	00H
16-bit timer/event counter 4	Timer counter 4 (TM4)	Undefined
	Compare register 4 (CR4)	Undefined
	Mode control register 4 (TMC4)	00H
8-bit timer/event counters 50 to 52	Timer counters 50 to 52 (TM50 to TM52)	00H
	Compare registers 50 to 52 (CR50 to CR52)	Undefined
	Clock select registers 50 to 52 (TCL50 to TCL52)	00H
	Mode control registers 50 to 52 (TMC50 to TMC52)	00H

Notes 1. During reset input or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

2. When a reset is executed in the standby mode, the pre-reset status is held even after reset.

3. Although the initial value is CFH, use the following value to be set for each version.
μPD780316, 780326, 780336: CCH
μPD780318, 780328, 780338: CFH
μPD78F0338: Value for mask ROM versions

4. Although the initial value is 0CH, use this register with a setting of 09H.

5. μPD78F0338 only.

Table 20-1. Hardware Statuses After Reset (2/2)

Hardware		Status After Reset
Watch timer	Operation mode register 0 (WTNM0)	00H
Watchdog timer	Clock select register (WDCS)	00H
	Mode register (WDTM)	00H
Clock output/buzzer output controller	Clock output select register (CKS)	00H
A/D converter	Conversion result register 0 (ADCR0)	00H
	Mode register 0 (ADM0)	00H
	Analog input channel specification register 0 (ADS0)	00H
D/A converter	Conversion value setting register 0 (DA0)	00H
	Mode register 0 (DAM0)	00H
Serial interface UART0	Asynchronous serial interface mode register 0 (ASIM0)	00H
	Asynchronous serial interface status register 0 (ASIS0)	00H
	Baud rate generator control register 0 (BRGC0)	00H
	Transmit shift register 0 (TXS0)	FFH
	Receive buffer register 0 (RXB0)	
Serial interface CSI1	Shift register 1 (SIO1)	Undefined
	Transmit buffer register 1 (SOTB1)	Undefined
	Operation mode register 1 (CSIM1)	00H
	Clock select register 1 (CSIC1)	10H
Serial interface SIO3	Shift register 3 (SIO3)	Undefined
	Operation mode register 3 (CSIM3)	00H
LCD controller/driver	Operation/display mode register 3 (LCDM3)	00H
	Clock control register 3 (LCDC3)	00H
	Static/dynamic display switching register 3 (SDSEL3)	00H
Interrupt	Request flag registers 0L, 0H, 1L (IF0L, IF0H, IF1L)	00H
	Mask flag registers 0L, 0H, 1L (MK0L, MK0H, MK1L)	FFH
	Priority specification flag registers 0L, 0H, 1L (PR0L, PR0H, PR1L)	FFH
	External interrupt rising edge enable register (EGP)	00H
	External interrupt falling edge enable register (EGN)	00H
ROM correction	Correction address registers 0, 1 (CORAD0, CORAD1)	0000H
	Correction control register (CORCN)	00H

CHAPTER 21 ROM CORRECTION

21.1 ROM Correction Function

The μPD780318, 780328, 780338 Subseries can replace part of a program in the mask ROM with a program in the internal expansion RAM.

Instruction bugs found in the mask ROM can be avoided, and program flow can be changed by using ROM correction.

ROM correction can be used to correct two places (max.) of the internal ROM (program).

Caution ROM correction cannot be emulated by the in-circuit emulator (IE-78K0-NS).

21.2 ROM Correction Configuration

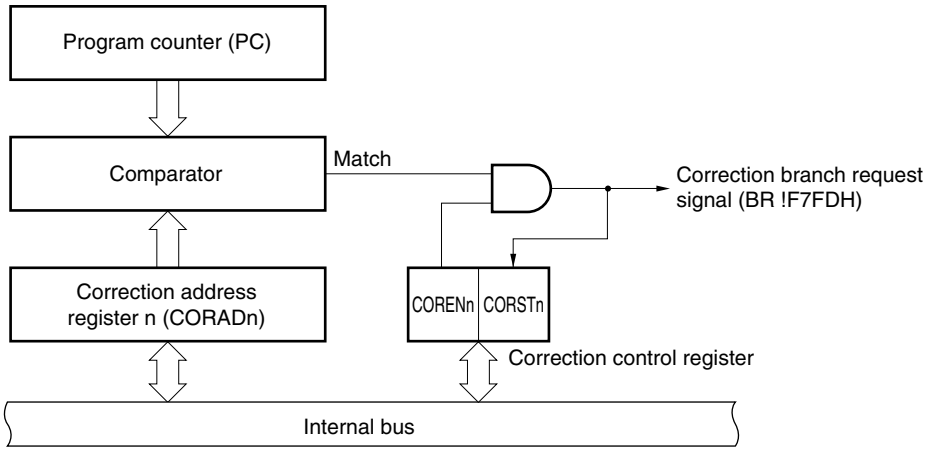
ROM correction consists of the following hardware.

Table 21-1. ROM Correction Configuration

Item	Configuration
Registers	Correction address registers 0 and 1 (CORAD0, CORAD1)
Control register	Correction control register (CORCN)

Figure 21-1 shows a block diagram of ROM correction.

Figure 21-1. ROM Correction Block Diagram



Remark n = 0, 1

(1) Correction address registers 0 and 1 (CORAD0, CORAD1)

These registers set the start address (correction address) of the instruction(s) to be corrected in the mask ROM. The ROM correction corrects two places (max.) of the program. Addresses are set to two registers, CORAD0 and CORAD1. If only one place needs to be corrected, set the address to either of the registers.

ROM correction for the start address specified in CORAD0 and CORAD1 is valid when bit 1 (COREN0) and bit 3 (COREN1) of the correction control register (CORCN) is 1.

CORAD0 and CORAD1 are set by a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CORAD0 and CORAD1 to 0000H.

Figure 21-2. Correction Address Registers 0 and 1 Format

Symbol	15	0	Address	After reset	R/W
CORAD0			FF38H/FF39H	0000H	R/W
CORAD1			FF3AH/FF3BH	0000H	R/W

Cautions 1. Set the CORAD0 and CORAD1 when bit 1 (COREN0) and bit 3 (COREN1) of the correction control register (CORCN) are 0.

2. Only start addresses where operation codes are stored can be set in CORAD0 and CORAD1.

3. Do not set the following addresses to CORAD0 and CORAD1.

- **Address value in table area of table reference instruction (CALLT instruction): 0040H to 007FH**
- **Address value in vector table area: 0000H to 003FH**

(2) Comparator

The comparator always compares the correction address value set in correction address registers 0 and 1 (CORAD0, CORAD1) with the fetch address value. When bit 1 (COREN0) or bit 3 (COREN1) of the correction control register (CORCN) is 1 and the correction address matches the fetch address value, the correction branch request signal (BR !F7FDH) is generated from the ROM correction circuit.

21.3 ROM Correction Control Register

ROM correction is controlled by the correction control register (CORCN).

(1) Correction control register (CORCN)

This register controls whether or not the correction branch request signal is generated when the fetch address matches the correction address set in correction address registers 0 and 1. The correction control register consists of correction enable flags (COREN0, COREN1) and correction status flags (CORST0, CORST1). The correction enable flags enable or disable the comparator match detection signal, and correction status flags show the values are matched.

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

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

Figure 21-3. Correction Control Register (CORCN) Format

Address: FF8AH After reset: 00H R/W[Note]

Symbol	7	6	5	4	3	2	1	0
CORCN	0	0	0	0	COREN1	CORST1	COREN0	CORST0

COREN1	Correction address register 1 and fetch address match detection control
0	Disabled
1	Enabled

CORST1	Correction address register 1 and fetch address match detection flag
0	Not detected
1	Detected

COREN0	Correction address register 0 and fetch address match detection control
0	Disabled
1	Enabled

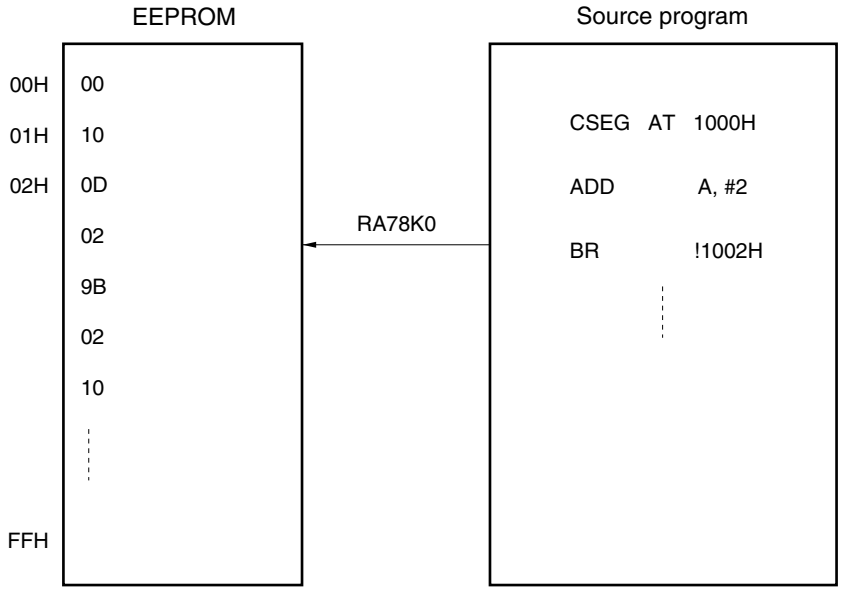
CORST0	Correction address register 0 and fetch address match detection flag
0	Not detected
1	Detected

Note Bits 0 and 2 are read-only bits. Bits 0 and 2 are set (1) only when a match is detected by comparator. Do not set these bits to 1 in software.

21.4 ROM Correction Application

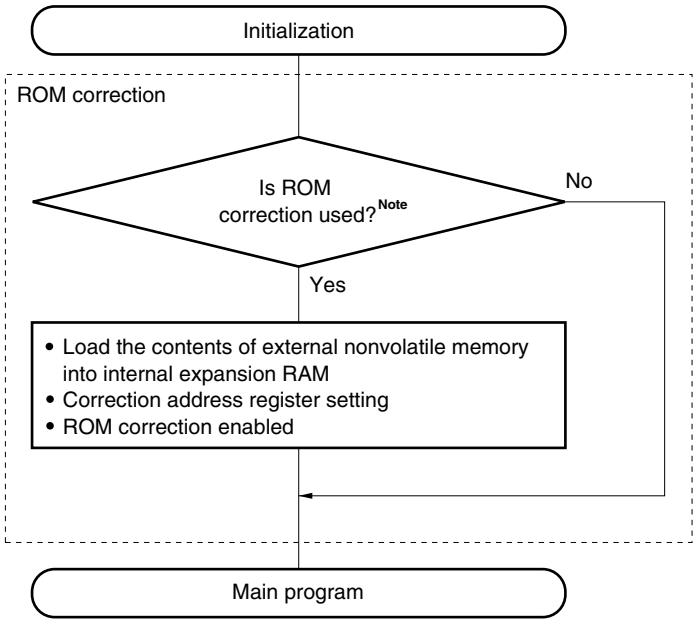
(1) Store the correction address and instruction after correction (patch program) to nonvolatile memory (such as EEPROM™) outside the microcontroller.
When two places should be corrected, store the branch destination judgment program as well. The branch destination judgment program checks which one of the addresses set to correction address register 0, 1 (CORAD0 or CORAD1) generates the correction branch.

Figure 21-4. Storing Example to EEPROM (When One Place Is Corrected)



(2) Assemble in advance the initialization routine as shown in Figure 21-5 to correct the program.

Figure 21-5. Initialization Routine



Note Whether ROM correction is used or not should be judged by the port input level. For example, when the P20 input level is high, the ROM correction is used, otherwise, it is not used.

(3) After reset, store the contents that have been previously stored in the external nonvolatile memory with initialization routine for ROM correction of the user to internal expansion RAM (see **Figure 21-5**).
Set the start address of the instruction to be corrected to CORAD0 and CORAD1, and set bits 1 and 3 (COREN0, COREN1) of the correction control register (CORCN) to 1.

(4) Set the entire-space branch instruction (BR !addr16) to the specified address (F7FDH) of the internal expansion RAM with the main program.

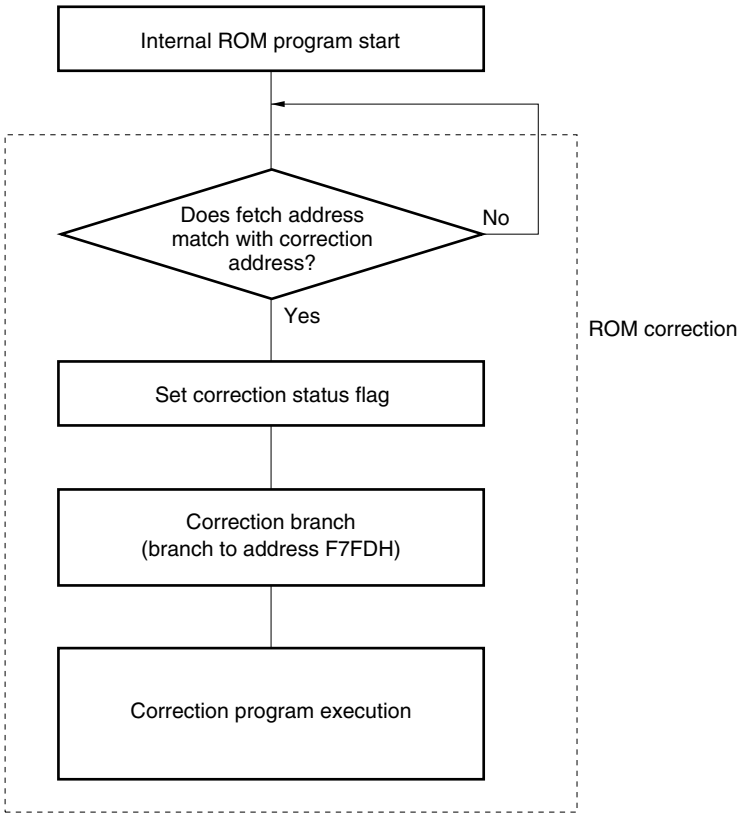
(5) After the main program is started, the fetch address value and the values set in CORAD0 and CORAD1 are always compared by the comparator in the ROM correction circuit. When these values match, the correction branch request signal is generated. Simultaneously the corresponding correction status flag (CORST0 or CORST1) is set to 1.

(6) Branch to the address F7FDH by the correction branch request signal.

(7) Branch to the internal expansion RAM address set with the main program by the entire-space branch instruction of the address F7FDH.

(8) When one place is corrected, the correction program is executed.
When two places are corrected, the correction status flag is checked with the branch destination judgment program, and branches to the correction program.

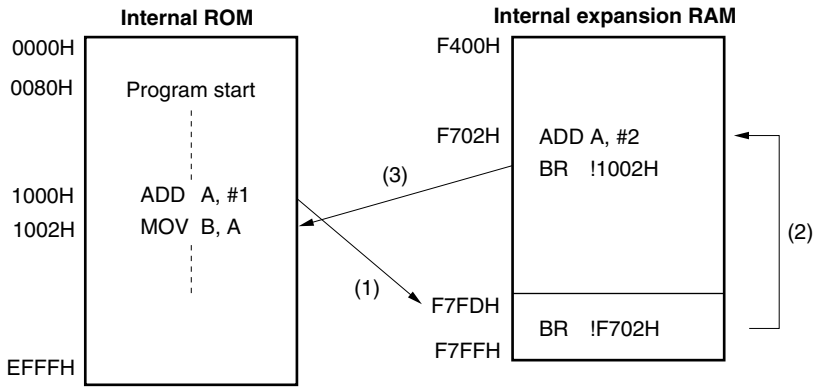
Figure 21-6. ROM Correction Operation



21.5 ROM Correction Example

An example of ROM correction when the instruction at address 1000H “ADD A, #1” is changed to “ADD A, #2” is shown below.

Figure 21-7. ROM Correction Example



(1) Branches to address F7FDH when the preset value 1000H in the correction address register 0, 1 (CORAD0, CORAD1) matches the fetch address value after the main program is started.

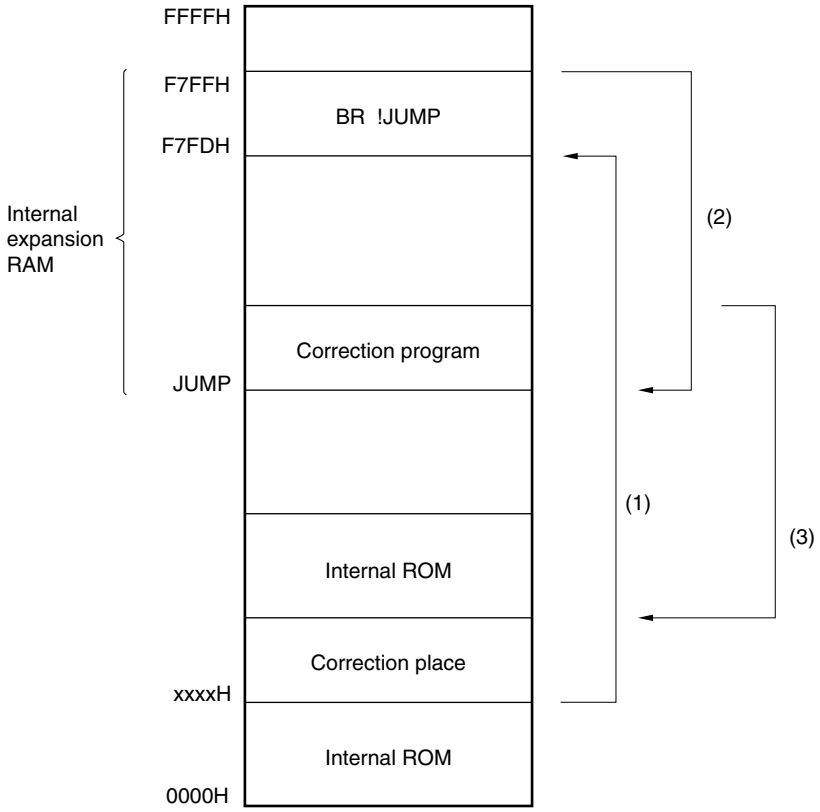
(2) Branches to any address (address F702H in this example) by setting the entire-space branch instruction (BR !addr16) to address F7FDH with the main program.

(3) Returns to the internal ROM program after executing the substitute instruction ADD A, #2.

21.6 Program Execution Flow

Figures 21-8 and 21-9 show the program transition diagrams when ROM correction is used.

Figure 21-8. Program Transition Diagram (When One Place Is Corrected)

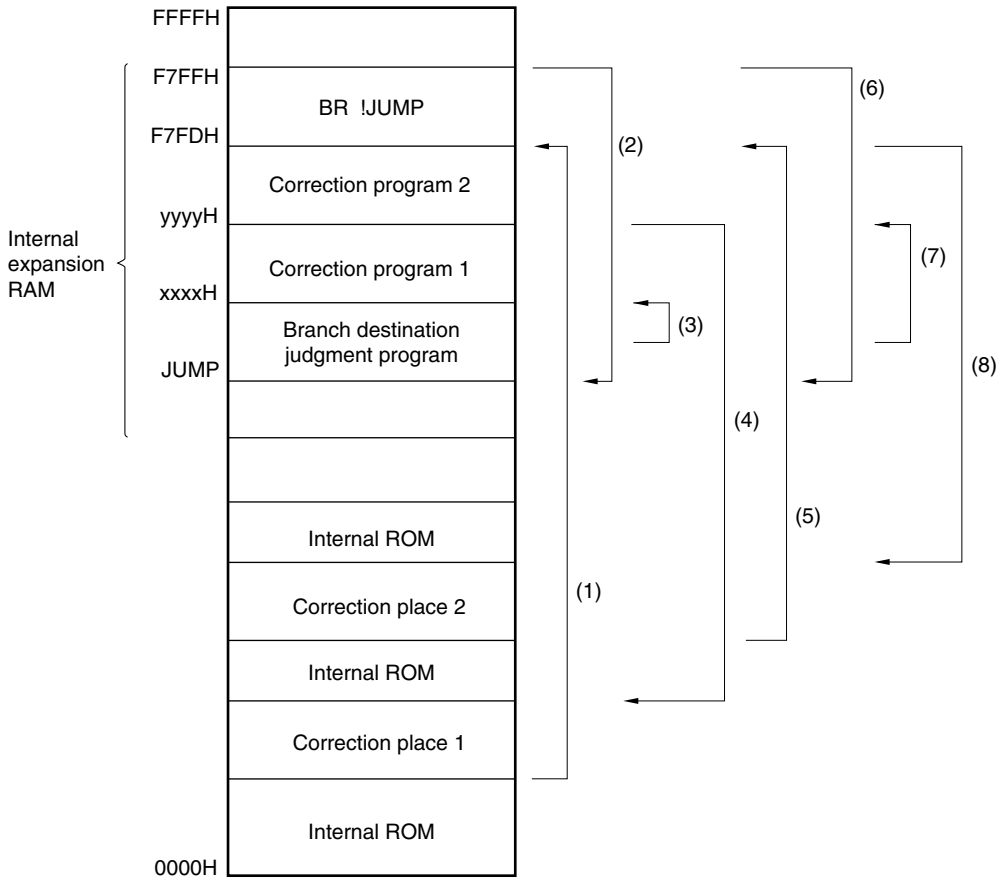


(1) Branches to address F7FDH when fetch address matches correction address
(2) Branches to correction program
(3) Returns to internal ROM program

Caution Do not use internal high-speed RAM and LCD display RAM for the ROM correction area.

Remark JUMP: Correction program start address

Figure 21-9. Program Transition Diagram (When Two Places Are Corrected)



(1) Branches to address F7FDH when fetch address matches correction address
(2) Branches to branch destination judgment program
(3) Branches to correction program 1 by branch destination judgment program (BTCLR !CORST0, $xxxxH)
(4) Returns to internal ROM program
(5) Branches to address F7FDH when fetch address matches correction address
(6) Branches to branch destination judgment program
(7) Branches to correction program 2 by branch destination judgment program (BTCLR !CORST1, $yyyyH)
(8) Returns to internal ROM program

Caution Do not use internal high-speed RAM and LCD display RAM for the ROM correction area.

Remark JUMP: Correction program start address

21.7 Cautions on ROM Correction

(1) Address values set in correction address registers 0 and 1 (CORAD0 and CORAD1) must be addresses where instruction codes are stored. In addition, address values to be set must be the start address of the instruction code.

(2) Correction address registers 0 and 1 (CORAD0 and CORAD1) should be set when the correction enable flags (COREN0, COREN1) are “0” (when correction branch processing is disabled). If address is set to CORAD0 or CORAD1 when COREN0 or COREN1 is 1 (when the correction branch is in enabled state), the correction branch may start with the different address from the set address value.

(3) Do not set the address value of instruction immediately after the instruction that sets the correction enable flag (COREN0, COREN1) to 1, to correction address register 0 or 1 (CORAD0, CORAD1); the correction branch may not start.

(4) Do not set the address value in table area of table reference instruction (CALLT instruction) (0040H to 007FH), and the address value in vector table area (0000H to 003FH) to correction address registers 0 and 1 (CORAD0, CORAD1).

(5) Do not set two addresses immediately after the instructions shown below to correction address registers 0 and 1 (CORAD0, CORAD1) (that is, when the mapped terminal address of these instructions is N, do not set the address values of N+1 and N+2).

- RET
- RETI
- RETB
- BR $addr16
- STOP
- HALT

(6) Do not set the address value set to the correction address registers 0 and 1 (CORAD0 and CORAD1) to F7FDH.

CHAPTER 22 μPD78F0338

The μPD78F0338 is provided as the flash memory version of the μPD780318, 780328, and 780338 Subseries.

The μPD78F0338 incorporates flash memory on which a program can be written, erased and overwritten while mounted on the board.

Data can be written to the flash memory with the memory mounted on the target system (on-board). To do this, connect the dedicated flash programmer to the target system.

Using flash memory in a development environment or application enables the following.

- Software can be modified after soldering the μPD78F0338 to the target system.
- Many products can be produced in small quantities by distinguishing the software of each.
- Data can be easily adjusted when mass-production is started.

Table 22-1 lists the differences between the μPD78F0338 and the mask ROM versions.

Table 22-1. Differences Between μPD78F0338 and Mask ROM Versions

Item	μPD78F0338	Mask ROM Versions		
		μPD780318 Subseries	μPD780328 Subseries	μPD780338 Subseries
Internal ROM structure	Flash memory	Mask ROM		
Internal ROM capacity	60 KB[Note 1]	μPD780316, 780326, 780336: 48 KB μPD780318, 780328, 780338: 60 KB		
I/O port	70[Note 2]	70	62	54
Segment signal output pin for LCD controller/driver	40 max.[Note 2]	24 max.	32 max.	40 max.
Mask option to specify the on-chip pull-up resistors of pins P60 to P63	Not possible	Possible		
IC pin	Not provided	Provided		
V_{PP} pin	Provided	Not provided		
Electrical specifications	Refer to data sheet of each product.			

Notes 1. The same capacity as the mask ROM versions can be specified by means of the memory size switching register (IMS).

2. The same I/O port and segment signal output pin can be specified by means of the pin function switching registers 8 and 9 (PF8 and PF9).

Caution There are differences in noise immunity and noise radiation between the flash memory and mask ROM versions. When pre-producing an application set with the flash memory version and then mass-producing it with the mask ROM version, be sure to conduct sufficient evaluations for the commercial samples (not engineering samples) of the mask ROM version.

22.1 Memory Size Switching Register

The μPD78F0338 allows users to select the internal memory capacity using the memory size switching register (IMS) so that the same memory map as that of mask ROM versions with a different size of internal memory capacity can be achieved.

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

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

Caution Be sure to set IMS to CCH or CFH as the initial setting of the program. Reset input initializes IMS to CFH. Be sure to set IMS to CCH or CFH after reset.

Figure 22-1. Memory Size Switching Register (IMS) Format

Address: FFF0H After reset: CFH R/W

Symbol	7	6	5	4	3	2	1	0
IMS	RAM2	RAM1	RAM0	0	ROM3	ROM2	ROM1	ROM0

RAM2	RAM1	RAM0	Internal high-speed RAM capacity selection
1	1	0	1,024 bytes
Other than above			Setting prohibited

ROM3	ROM2	ROM1	ROM0	Internal ROM capacity selection
1	1	0	0	48 KB
1	1	1	1	60 KB
Other than above				Setting prohibited

The IMS settings to obtain the same memory map as mask ROM versions are shown in Table 22-2.

Table 22-2. Memory Size Switching Register Settings

Target Mask ROM Versions	IMS Setting
μPD780316, 780326, 780336	CCH
μPD780318, 780328, 780338	CFH

Caution When using the mask ROM versions, be sure to set the value indicated in Table 22-2 to IMS.

22.2 Internal Expansion RAM Size Switching Register

The internal expansion RAM size switching register (IXS) is used to set the internal expansion RAM capacity.
IXS is set by an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets the value of this register to 0CH.

Caution Be sure to set IXS to 09H as the initial setting of the program. Reset input initializes IXS to 0CH. Be sure to set IXS to 09H after reset. Set the mask ROM versions in the same manner.

Figure 22-2. Internal Expansion RAM Size Switching Register (IXS) Format

Address: FFF4H After reset: 0CH R/W

Symbol	7	6	5	4	3	2	1	0
IXS	0	0	0	0	IXRAM3	IXRAM2	IXRAM1	IXRAM0

IXRAM3	IXRAM2	IXRAM1	IXRAM0	Internal expansion RAM capacity selection
1	0	0	1	1,536 bytes
Other than above				Setting prohibited

22.3 Flash Memory Characteristics

Flash memory programming is performed by connecting a dedicated flash programmer (Flashpro III (part no. FL-PR3, PG-FP3)/Flashpro IV (part no. FL-PR4, PG-FP4)) to the target system with the flash memory mounted on the target system (on-board). A flash memory writing adapter (FA adapter), which is a target board used exclusively for programming, is also provided.

Remark FL-PR3, FL-PR4, and the flash memory writing adapter are products made by Naito Densei Machida Mfg. Co., Ltd. (TEL +81-45-475-4191).

Programming using flash memory has the following advantages.

- Software can be modified after soldering the μPD78F0338 to the target system.
- Many products can be produced in small quantities by distinguishing the software of each.
- Data can be easily adjusted when mass-production is started.

22.3.1 Programming environment

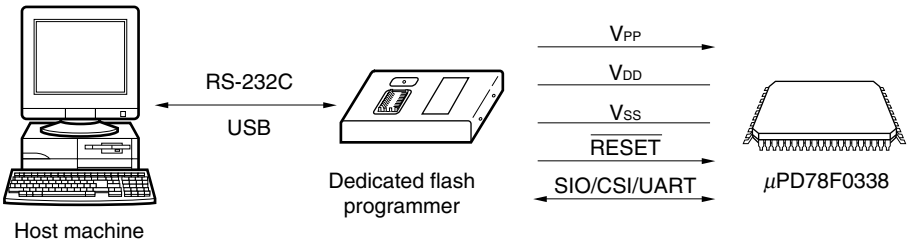
The following shows the environment required for μPD78F0338 flash memory programming.

When Flashpro III (part no. FL-PR3, PG-FP3) or Flashpro IV (part no. FL-PR4, PG-FP4) is used as a dedicated flash programmer, a host machine is required to control the dedicated flash programmer. Communication between the host machine and flash programmer is performed via RS-232C/USB (Rev. 1.1).

For details, refer to the manuals for Flashpro III/Flashpro IV.

Remark USB is supported by Flashpro IV only.

Figure 22-3. Environment for Writing Program to Flash Memory



22.3.2 Communication mode

Use the communication mode shown in Table 22-3 to perform communication between the dedicated flash programmer and μPD78F0338.

Table 22-3. 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 (SIO3)	SIO ch-0 (3-wire, sync)	100 Hz to 1.25 MHz **Note 2**	Optional	1 to 10 MHz **Note 2**	1.0	SI3/RxD0/P20 SO3/TxD0/P21 $\overline{SCK3}$/P22	0
3-wire serial I/O (CSI1)	SIO ch-1 (3-wire, sync)	100 Hz to 2 MHz[Note 2]	Optional	1 to 10 MHz **Note 2**	1.0	SI1/P23 SO1/P24 SCK1/P25	1
UART (UART0)	UART ch-0	4,800 to 76,800 bps **Notes 2, 3**	Optional	1 to 10 MHz **Note 2**	1.0	RxD0/SI3/P20 TxD0/SO3/P21	8

Notes 1. Selection items for TYPE settings on the dedicated flash programmer (Flashpro III (part no. FL-PR3, PG-FP3)/Flashpro IV (part no. FL-PR4, PG-FP4)).

2. The possible setting range differs depending on the voltage. For details, refer to **CHAPTER 24 ELECTRICAL SPECIFICATIONS**.

3. Because factors other than the baud rate error, such as the signal waveform slew, also affect UART communication, thoroughly evaluate the slew as well as the baud rate error.

Figure 22-4. 3-Wire Serial I/O (SIO3)

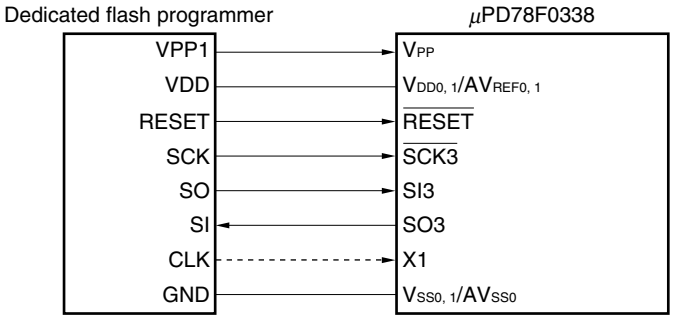


Figure 22-5. 3-Wire Serial I/O (CSI1)

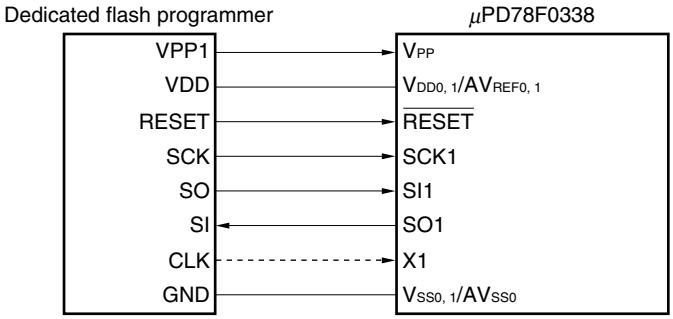
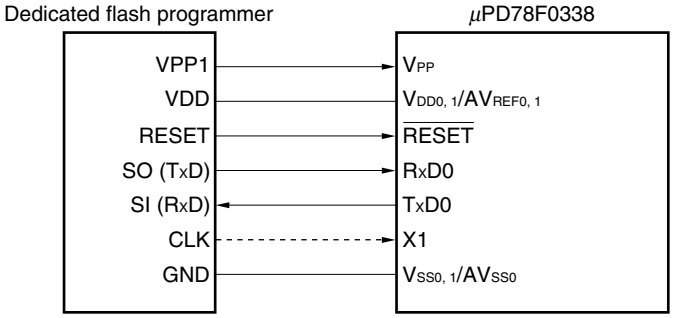


Figure 22-6. UART (UART0)



Remark CLK can be supplied on-board. It does not have to be connected to the dedicated flash programmer. V_{DD} can also be supplied on-board but it must be connected to the dedicated flash programmer. In addition, the voltage must be supplied before starting programming.

If Flashpro III (part no. FL-PR3, PG-FP3)/Flashpro IV is used as a dedicated flash programmer, the following signals are generated for the μPD78F0338. For details, refer to the manual of Flashpro III/Flashpro IV.

Table 22-4. Pin Connection List

Signal Name	I/O	Pin Function	Pin Name	SIO3	CSI1	UART0
VPP1	Output	Write voltage	V_{PP}	◎	◎	◎
VPP2	–	–	–	×	×	×
VDD	I/O	V_{DD} voltage generation/voltage monitoring	$V_{DD0}/V_{DD1}/AV_{REF}$	◎ Note	◎ Note	◎ Note
GND	–	Ground	$V_{SS0}/V_{SS1}/AV_{SS}$	◎	◎	◎
CLK	Output	Clock output	X1	○	○	○
RESET	Output	Reset signal	$\overline{RESET}$	◎	◎	◎
SI (RxD)	Input	Reception signal	SO3/SO1/TxD0	◎	◎	◎
SO (TxD)	Output	Transmit signal	SI3/SI1/RxD0	◎	◎	◎
SCK	Output	Transfer clock	$\overline{SCK3}$/SCK1	◎	◎	×
HS	–	–	–	×	×	×

Note V_{DD} voltage must be supplied before programming is started.

Remark ◎: Pin must be connected.
○: If the signal is supplied on the target board, pin need not be connected.
×: Pin need not be connected.

22.3.3 On-board pin processing

When performing programming on the target system, provide a connector on the target system to connect the dedicated flash programmer.

An on-board function that allows switching between normal operation mode and flash memory programming mode may be required in some cases.

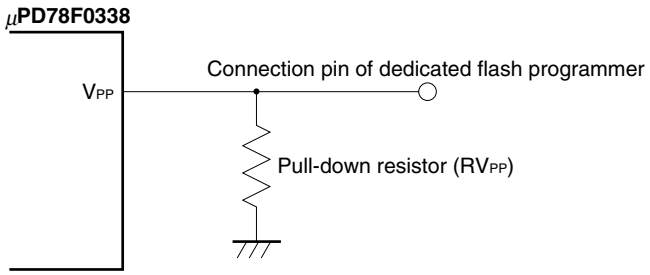
<V_{PP} pin>

In normal operation mode, input 0 V to the V_{PP} pin. In flash memory programming mode, a write voltage of 10.0 V (TYP.) is supplied to the V_{PP} pin, so perform 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 22-7. V_{PP} Pin Connection Example



<Serial interface pin>

The following shows the pins used by the serial interface.

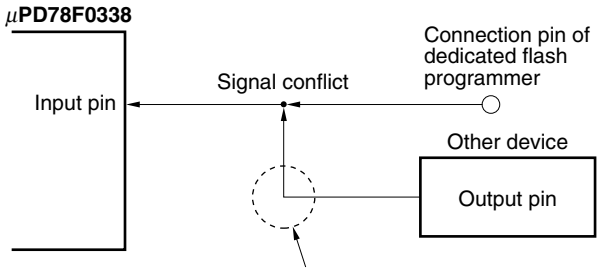
Serial Interface	Pins Used
3-wire serial I/O (SIO3)	SI3/SO3/$\overline{SCK3}$
3-wire serial I/O (CSI1)	SI1/SO1/SCK1
UART (UART0)	RxD0/TxD0

When connecting the dedicated flash programmer to a serial interface pin that is connected to another device on-board, signal conflict or abnormal operation of the other 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 22-8. Signal Conflict (Input Pin of Serial Interface)

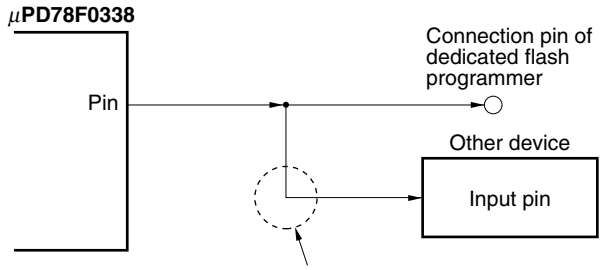


In the flash memory programming mode, the signal output by another device and the signal sent by the dedicated flash programmer conflict, therefore, isolate the signal of the other device.

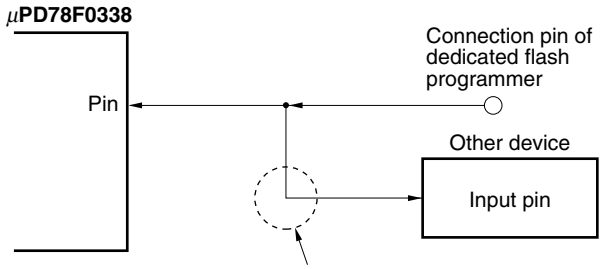
(2) Abnormal operation of other device

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 22-9. Abnormal Operation of Other Device



If the signal output by the μPD78F0338 affects another device in the flash memory programming mode, isolate the signals of the other device.

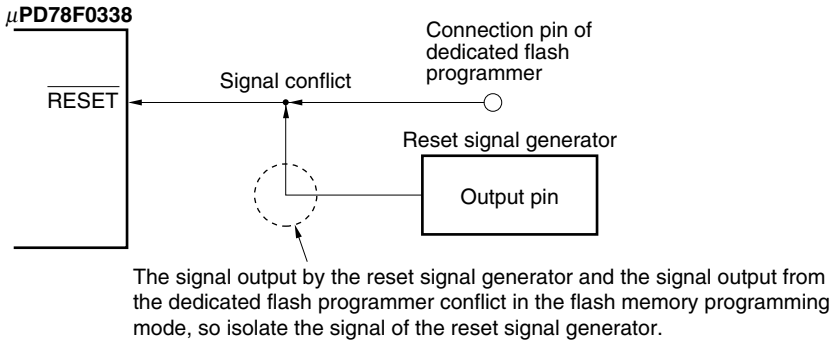


If the signal output by the dedicated flash programmer affects another device in the flash memory programming mode, isolate the signals of the other device.

<$\overline{\text{RESET}}$ 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 22-10. Signal Conflict ($\overline{\text{RESET}}$ Pin)



<Port pins (including NMI)>

When the μPD78F0338 enters the flash memory programming mode, all the pins other than those that communicate in flash memory programming are in the same status as immediately after reset.

If the external device does not recognize initial statuses such as the output high impedance status, therefore, connect the external device to V_{DD0} or V_{SS0} via a resistor.

<Oscillator>

When using the on-board clock, connect X1, X2, XT1, and XT2 as required in the normal operation mode.

When using the clock output of the flash programmer, connect it directly to X1, disconnecting the main oscillator on-board, and leave the X2 pin open. The subclock conforms to the normal operation mode.

<Power supply>

To use the power output from the flash programmer, connect the V_{DD0} and V_{DD1} pins to VDD of the flash programmer, and the V_{SS0} and V_{SS1} pins to GND of the flash programmer.

To use the on-board power supply, make connections that accord with the normal operation mode. However, because the voltage is monitored by the flash programmer, be sure to connect V_{DD0} and V_{DD1} to VDD of the flash programmer.

Supply the same power as in the normal operation mode to the other power supply pins (AV_{REF0}, AV_{REF1}, and AV_{SS0}).

<Other pins>

Process the other pins (S0 to S39, COM0 to COM3, SCOMO, V_{LC0} to V_{LC2}, V_{LCDC}, CAPH, and CAPL) in the same manner as in the normal operation mode.

CHAPTER 23 INSTRUCTION SET

This chapter lists each instruction set of the μPD780318, 780328, and 780338 Subseries in table form. For details of its operation and operation code, refer to the separate document **78K/0 Series Instructions User's Manual (U12326E)**.

23.1 Conventions

23.1.1 Operand identifiers and specification methods

Operands are written in “Operand” column of each instruction in accordance with the specification method of the instruction operand identifier (refer to the assembler specifications for details). When there are two or more methods, select one of them. Alphabetic letters in capitals and symbols, #, !, $, and [ ] are key words and must be written as they are. Each symbol has the following meaning.

- #: Immediate data specification
- !: Absolute address specification
- $: Relative address specification
- [ ]: Indirect address specification

In the case of immediate data, describe an appropriate numeric value or a label. When using a label, be sure to write the #, !, $, and [ ] symbols.

For operand register identifiers, r and rp, either function names (X, A, C, etc.) or absolute names (names in parentheses in the table below, R0, R1, R2, etc.) can be used for specification.

Table 23-1. Operand Identifiers and Specification Methods

Identifier	Specification Method
r	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7)
rp	AX (RP0), BC (RP1), DE (RP2), HL (RP3)
sfr	Special function register symbol**Note**
sfrp	Special function register symbol (16-bit manipulatable register even addresses only)**Note**
saddr	FE20H to FF1FH Immediate data or labels
saddrp	FE20H to FF1FH Immediate data or labels (even address only)
addr16	0000H to FFFFH Immediate data or labels (only even addresses for 16-bit data transfer instructions)
addr11	0800H to 0FFFH Immediate data or labels
addr5	0040H to 007FH Immediate data or labels (even address only)
word	16-bit immediate data or label
byte	8-bit immediate data or label
bit	3-bit immediate data or label
RBn	RB0 to RB3

Note Addresses from FFD0H to FFDFH cannot be accessed with these operands.

Remark For special function register symbols, refer to **Table 3-4 Special Function Register List**.

23.1.2 Description of “operation” column

A: A register; 8-bit accumulator
X: X register
B: B register
C: C register
D: D register
E: E register
H: H register
L: L register
AX: AX register pair; 16-bit accumulator
BC: BC register pair
DE: DE register pair
HL: HL register pair
PC: Program counter
SP: Stack pointer
PSW: Program status word
CY: Carry flag
AC: Auxiliary carry flag
Z: Zero flag
RBS: Register bank select flag
IE: Interrupt request enable flag
NMIS: Non-maskable interrupt servicing flag
( ): Memory contents indicated by address or register contents in parentheses
X_H, X_L: Higher 8 bits and lower 8 bits of 16-bit register
$\wedge$: Logical product (AND)
$\vee$: Logical sum (OR)
$\veebar$: Exclusive logical sum (exclusive OR)
$\overline{\quad}$: Inverted data
addr16: 16-bit immediate data or label
jdisp8: Signed 8-bit data (displacement value)

23.1.3 Description of “flag operation” column

(Blank): Not affected
0: Cleared to 0
1: Set to 1
×: Set/cleared according to the result
R: Previously saved value is restored

23.2 Operation List

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit data transfer	**MOV**	r, #byte	2	4	–	r ← byte	
		saddr, #byte	3	6	7	(saddr) ← byte	
		sfr, #byte	3	–	7	sfr ← byte	
		A, r Note 3	1	2	–	A ← r	
		r, A Note 3	1	2	–	r ← A	
		A, saddr	2	4	5	A ← (saddr)	
		saddr, A	2	4	5	(saddr) ← A	
		A, sfr	2	–	5	A ← sfr	
		sfr, A	2	–	5	sfr ← A	
		A, !addr16	3	8	9 + n	A ← (addr16)	
		!addr16, A	3	8	9 + m	(addr16) ← A	
		PSW, #byte	3	–	7	PSW ← byte	× × ×
		A, PSW	2	–	5	A ← PSW	
		PSW, A	2	–	5	PSW ← A	× × ×
		A, [DE]	1	4	5 + n	A ← (DE)	
		[DE], A	1	4	5 + m	(DE) ← A	
		A, [HL]	1	4	5 + n	A ← (HL)	
		[HL], A	1	4	5 + m	(HL) ← A	
		A, [HL + byte]	2	8	9 + n	A ← (HL + byte)	
		[HL + byte], A	2	8	9 + m	(HL + byte) ← A	
		A, [HL + B]	1	6	7 + n	A ← (HL + B)	
		[HL + B], A	1	6	7 + m	(HL + B) ← A	
		A, [HL + C]	1	6	7 + n	A ← (HL + C)	
		[HL + C], A	1	6	7 + m	(HL + C) ← A	
	XCH	A, r Note 3	1	2	–	A ↔ r	
		A, saddr	2	4	6	A ↔ (saddr)	
		A, sfr	2	–	6	A ↔ sfr	
		A, !addr16	3	8	10 + n + m	A ↔ (addr16)	
		A, [DE]	1	4	6 + n + m	A ↔ (DE)	
		A, [HL]	1	4	6 + n + m	A ↔ (HL)	
		A, [HL + byte]	2	8	10 + n + m	A ↔ (HL + byte)	
		A, [HL + B]	2	8	10 + n + m	A ↔ (HL + B)	
		A, [HL + C]	2	8	10 + n + m	A ↔ (HL + C)	

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed.

3. Except “r = A”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to internal ROM program.

3. n is the number of waits when external memory expansion area is read from.

4. m is the number of waits when external memory expansion area is written to.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
16-bit data transfer	**MOVW**	rp, #word	3	6	–	rp ← word	
		saddrp, #word	4	8	10	(saddrp) ← word	
		sfrp, #word	4	–	10	sfrp ← word	
		AX, saddrp	2	6	8	AX ← (saddrp)	
		saddrp, AX	2	6	8	(saddrp) ← AX	
		AX, sfrp	2	–	8	AX ← sfrp	
		sfrp, AX	2	–	8	sfrp ← AX	
		AX, rp Note 3	1	4	–	AX ← rp	
		rp, AX Note 3	1	4	–	rp ← AX	
		AX, !addr16	3	10	12 + 2n	AX ← (addr16)	
		!addr16, AX	3	10	12 + 2m	(addr16) ← AX	
	XCHW	AX, rp Note 3	1	4	–	AX ↔ rp	
8-bit operation	**ADD**	A, #byte	2	4	–	A, CY ← A + byte	× × ×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) + byte	× × ×
		A, r Note 4	2	4	–	A, CY ← A + r	× × ×
		r, A	2	4	–	r, CY ← r + A	× × ×
		A, saddr	2	4	5	A, CY ← A + (saddr)	× × ×
		A, !addr16	3	8	9 + n	A, CY ← A + (addr16)	× × ×
		A, [HL]	1	4	5 + n	A, CY ← A + (HL)	× × ×
		A, [HL + byte]	2	8	9 + n	A, CY ← A + (HL + byte)	× × ×
		A, [HL + B]	2	8	9 + n	A, CY ← A + (HL + B)	× × ×
		A, [HL + C]	2	8	9 + n	A, CY ← A + (HL + C)	× × ×
	ADDC	A, #byte	2	4	–	A, CY ← A + byte + CY	× × ×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) + byte + CY	× × ×
		A, r Note 4	2	4	–	A, CY ← A + r + CY	× × ×
		r, A	2	4	–	r, CY ← r + A + CY	× × ×
		A, saddr	2	4	5	A, CY ← A + (saddr) + CY	× × ×
		A, !addr16	3	8	9 + n	A, CY ← A + (addr16) + CY	× × ×
		A, [HL]	1	4	5 + n	A, CY ← A + (HL) + CY	× × ×
		A, [HL + byte]	2	8	9 + n	A, CY ← A + (HL + byte) + CY	× × ×
		A, [HL + B]	2	8	9 + n	A, CY ← A + (HL + B) + CY	× × ×
		A, [HL + C]	2	8	9 + n	A, CY ← A + (HL + C) + CY	× × ×

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Only when rp = BC, DE or HL

4. Except “r = A”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to internal ROM program.

3. n is the number of waits when external memory expansion area is read from.

4. m is the number of waits when external memory expansion area is written to.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	**SUB**	A, #byte	2	4	–	A, CY ← A – byte	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) – byte	×	×	×
		A, r Note 3	2	4	–	A, CY ← A – r	×	×	×
		r, A	2	4	–	r, CY ← r – A	×	×	×
		A, saddr	2	4	5	A, CY ← A – (saddr)	×	×	×
		A, !addr16	3	8	9 + n	A, CY ← A – (addr16)	×	×	×
		A, [HL]	1	4	5 + n	A, CY ← A – (HL)	×	×	×
		A, [HL + byte]	2	8	9 + n	A, CY ← A – (HL + byte)	×	×	×
		A, [HL + B]	2	8	9 + n	A, CY ← A – (HL + B)	×	×	×
		A, [HL + C]	2	8	9 + n	A, CY ← A – (HL + C)	×	×	×
	SUBC	A, #byte	2	4	–	A, CY ← A – byte – CY	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) – byte – CY	×	×	×
		A, r Note 3	2	4	–	A, CY ← A – r – CY	×	×	×
		r, A	2	4	–	r, CY ← r – A – CY	×	×	×
		A, saddr	2	4	5	A, CY ← A – (saddr) – CY	×	×	×
		A, !addr16	3	8	9 + n	A, CY ← A – (addr16) – CY	×	×	×
		A, [HL]	1	4	5 + n	A, CY ← A – (HL) – CY	×	×	×
		A, [HL + byte]	2	8	9 + n	A, CY ← A – (HL + byte) – CY	×	×	×
		A, [HL + B]	2	8	9 + n	A, CY ← A – (HL + B) – CY	×	×	×
		A, [HL + C]	2	8	9 + n	A, CY ← A – (HL + C) – CY	×	×	×
	AND	A, #byte	2	4	–	A ← A ∧ byte	×		
		saddr, #byte	3	6	8	(saddr) ← (saddr) ∧ byte	×		
		A, r Note 3	2	4	–	A ← A ∧ r	×		
		r, A	2	4	–	r ← r ∧ A	×		
		A, saddr	2	4	5	A ← A ∧ (saddr)	×		
		A, !addr16	3	8	9 + n	A ← A ∧ (addr16)	×		
		A, [HL]	1	4	5 + n	A ← A ∧ (HL)	×		
		A, [HL + byte]	2	8	9 + n	A ← A ∧ (HL + byte)	×		
		A, [HL + B]	2	8	9 + n	A ← A ∧ (HL + B)	×		
		A, [HL + C]	2	8	9 + n	A ← A ∧ (HL + C)	×		

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Except “r = A”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to internal ROM program.

3. n is the number of waits when external memory expansion area is read from.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit operation	**OR**	A, #byte	2	4	–	A ← A ∨ byte	×
		saddr, #byte	3	6	8	(saddr) ← (saddr) ∨ byte	×
		A, r Note 3	2	4	–	A ← A ∨ r	×
		r, A	2	4	–	r ← r ∨ A	×
		A, saddr	2	4	5	A ← A ∨ (saddr)	×
		A, !addr16	3	8	9 + n	A ← A ∨ (addr16)	×
		A, [HL]	1	4	5 + n	A ← A ∨ (HL)	×
		A, [HL + byte]	2	8	9 + n	A ← A ∨ (HL + byte)	×
		A, [HL + B]	2	8	9 + n	A ← A ∨ (HL + B)	×
		A, [HL + C]	2	8	9 + n	A ← A ∨ (HL + C)	×
	XOR	A, #byte	2	4	–	A ← A ⩛ byte	×
		saddr, #byte	3	6	8	(saddr) ← (saddr) ⩛ byte	×
		A, r Note 3	2	4	–	A ← A ⩛ r	×
		r, A	2	4	–	r ← r ⩛ A	×
		A, saddr	2	4	5	A ← A ⩛ (saddr)	×
		A, !addr16	3	8	9 + n	A ← A ⩛ (addr16)	×
		A, [HL]	1	4	5 + n	A ← A ⩛ (HL)	×
		A, [HL + byte]	2	8	9 + n	A ← A ⩛ (HL + byte)	×
		A, [HL + B]	2	8	9 + n	A ← A ⩛ (HL + B)	×
		A, [HL + C]	2	8	9 + n	A ← A ⩛ (HL + C)	×
	CMP	A, #byte	2	4	–	A – byte	× × ×
		saddr, #byte	3	6	8	(saddr) – byte	× × ×
		A, r Note 3	2	4	–	A – r	× × ×
		r, A	2	4	–	r – A	× × ×
		A, saddr	2	4	5	A – (saddr)	× × ×
		A, !addr16	3	8	9 + n	A – (addr16)	× × ×
		A, [HL]	1	4	5 + n	A – (HL)	× × ×
		A, [HL + byte]	2	8	9 + n	A – (HL + byte)	× × ×
		A, [HL + B]	2	8	9 + n	A – (HL + B)	× × ×
		A, [HL + C]	2	8	9 + n	A – (HL + C)	× × ×

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Except “r = A”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to internal ROM program.

3. n is the number of waits when external memory expansion area is read from.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit operation	**ADDW**	AX, #word	3	6	–	AX, CY ← AX + word	×	×	×
	SUBW	AX, #word	3	6	–	AX, CY ← AX – word	×	×	×
	CMPW	AX, #word	3	6	–	AX – word	×	×	×
Multiply/ divide	**MULU**	X	2	16	–	AX ← A × X			
	DIVUW	C	2	25	–	AX (Quotient), C (Remainder) ← AX ÷ C			
Increment/ decrement	**INC**	r	1	2	–	r ← r + 1	×	×	
		saddr	2	4	6	(saddr) ← (saddr) + 1	×	×	
	DEC	r	1	2	–	r ← r – 1	×	×	
		saddr	2	4	6	(saddr) ← (saddr) – 1	×	×	
	INCW	rp	1	4	–	rp ← rp + 1			
	DECW	rp	1	4	–	rp ← rp – 1			
Rotate	**ROR**	A, 1	1	2	–	$(CY, A_7 \leftarrow A_0, A_{m-1} \leftarrow A_m) \times 1$ time			×
	ROL	A, 1	1	2	–	$(CY, A_0 \leftarrow A_7, A_{m+1} \leftarrow A_m) \times 1$ time			×
	RORC	A, 1	1	2	–	$(CY \leftarrow A_0, A_7 \leftarrow CY, A_{m-1} \leftarrow A_m) \times 1$ time			×
	ROLC	A, 1	1	2	–	$(CY \leftarrow A_7, A_0 \leftarrow CY, A_{m+1} \leftarrow A_m) \times 1$ time			×
	ROR4	[HL]	2	10	12 + n + m	$A_{3-0} \leftarrow (HL)_{3-0}$, $(HL)_{7-4} \leftarrow A_{3-0}$, $(HL)_{3-0} \leftarrow (HL)_{7-4}$			
	ROL4	[HL]	2	10	12 + n + m	$A_{3-0} \leftarrow (HL)_{7-4}$, $(HL)_{3-0} \leftarrow A_{3-0}$, $(HL)_{7-4} \leftarrow (HL)_{3-0}$			
BCD adjust	**ADJBA**		2	4	–	Decimal Adjust Accumulator after Addition	×	×	×
	ADJBS		2	4	–	Decimal Adjust Accumulator after Subtract	×	×	×
Bit manipu- late	**MOV1**	CY, saddr.bit	3	6	7	CY ← (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← sfr.bit			×
		CY, A.bit	2	4	–	CY ← A.bit			×
		CY, PSW.bit	3	–	7	CY ← PSW.bit			×
		CY, [HL].bit	2	6	7 + n	CY ← (HL).bit			×
		saddr.bit, CY	3	6	8	(saddr.bit) ← CY			
		sfr.bit, CY	3	–	8	sfr.bit ← CY			
		A.bit, CY	2	4	–	A.bit ← CY			
		PSW.bit, CY	3	–	8	PSW.bit ← CY	×	×	
		[HL].bit, CY	2	6	8 + n + m	(HL).bit ← CY			

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to internal ROM program.

3. n is the number of waits when external memory expansion area is read from.

4. m is the number of waits when external memory expansion area is written to.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
Bit manipulate	**AND1**	CY, saddr.bit	3	6	7	CY ← CY ∧ (saddr.bit)	×
		CY, sfr.bit	3	–	7	CY ← CY ∧ sfr.bit	×
		CY, A.bit	2	4	–	CY ← CY ∧ A.bit	×
		CY, PSW.bit	3	–	7	CY ← CY ∧ PSW.bit	×
		CY, [HL].bit	2	6	7 + n	CY ← CY ∧ (HL).bit	×
	OR1	CY, saddr.bit	3	6	7	CY ← CY ∨ (saddr.bit)	×
		CY, sfr.bit	3	–	7	CY ← CY ∨ sfr.bit	×
		CY, A.bit	2	4	–	CY ← CY ∨ A.bit	×
		CY, PSW.bit	3	–	7	CY ← CY ∨ PSW.bit	×
		CY, [HL].bit	2	6	7 + n	CY ← CY ∨ (HL).bit	×
	XOR1	CY, saddr.bit	3	6	7	CY ← CY ∀ (saddr.bit)	×
		CY, sfr.bit	3	–	7	CY ← CY ∀ sfr.bit	×
		CY, A.bit	2	4	–	CY ← CY ∀ A.bit	×
		CY, PSW.bit	3	–	7	CY ← CY ∀ PSW.bit	×
		CY, [HL].bit	2	6	7 + n	CY ← CY ∀ (HL).bit	×
	SET1	saddr.bit	2	4	6	(saddr.bit) ← 1	
		sfr.bit	3	–	8	sfr.bit ← 1	
		A.bit	2	4	–	A.bit ← 1	
		PSW.bit	2	–	6	PSW.bit ← 1	× × ×
		[HL].bit	2	6	8 + n + m	(HL).bit ← 1	
	CLR1	saddr.bit	2	4	6	(saddr.bit) ← 0	
		sfr.bit	3	–	8	sfr.bit ← 0	
		A.bit	2	4	–	A.bit ← 0	
		PSW.bit	2	–	6	PSW.bit ← 0	× × ×
		[HL].bit	2	6	8 + n + m	(HL).bit ← 0	
	SET1	CY	1	2	–	CY ← 1	1
	CLR1	CY	1	2	–	CY ← 0	0
	NOT1	CY	1	2	–	CY ← $\overline{CY}$	×

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to internal ROM program.

3. n is the number of waits when external memory expansion area is read from.

4. m is the number of waits when external memory expansion area is written to.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
Call/return	**CALL**	!addr16	3	7	–	(SP – 1) ← (PC + 3)H, (SP – 2) ← (PC + 3)L, PC ← addr16, SP ← SP – 2	
	CALLF	!addr11	2	5	–	(SP – 1) ← (PC + 2)H, (SP – 2) ← (PC + 2)L, PC15 – 11 ← 00001, PC10 – 0 ← addr11, SP ← SP – 2	
	CALLT	[addr5]	1	6	–	(SP – 1) ← (PC + 1)H, (SP – 2) ← (PC + 1)L, PCH ← (00000000, addr5 + 1), PCL ← (00000000, addr5), SP ← SP – 2	
	BRK		1	6	–	(SP – 1) ← PSW, (SP – 2) ← (PC + 1)H, (SP – 3) ← (PC + 1)L, PCH ← (003FH), PCL ← (003EH), SP ← SP – 3, IE ← 0	
	RET		1	6	–	PCH ← (SP + 1), PCL ← (SP), SP ← SP + 2	
	RETI		1	6	–	PCH ← (SP + 1), PCL ← (SP), PSW ← (SP + 2), SP ← SP + 3, NMIS ← 0	R R R
	RETB		1	6	–	PCH ← (SP + 1), PCL ← (SP), PSW ← (SP + 2), SP ← SP + 3	R R R
Stack manipu-late	**PUSH**	PSW	1	2	–	(SP – 1) ← PSW, SP ← SP – 1	
		rp	1	4	–	(SP – 1) ← rpH, (SP – 2) ← rpL, SP ← SP – 2	
	POP	PSW	1	2	–	PSW ← (SP), SP ← SP + 1	R R R
		rp	1	4	–	rpH ← (SP + 1), rpL ← (SP), SP ← SP + 2	
	MOVW	SP, #word	4	–	10	SP ← word	
		SP, AX	2	–	8	SP ← AX	
		AX, SP	2	–	8	AX ← SP	
Uncondi-tional branch	**BR**	!addr16	3	6	–	PC ← addr16	
		$addr16	2	6	–	PC ← PC + 2 + jdisp8	
		AX	2	8	–	PCH ← A, PCL ← X	
Conditional branch	**BC**	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if CY = 1	
	BNC	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if CY = 0	
	BZ	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if Z = 1	
	BNZ	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if Z = 0	

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (fCPU) selected by the processor clock control register (PCC).

2. This clock cycle applies to internal ROM program.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
Condi-tional branch	**BT**	saddr.bit, $addr16	3	8	9	PC ← PC + 3 + jdisp8 if (saddr.bit) = 1	
		sfr.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if sfr.bit = 1	
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 1	
		PSW.bit, $addr16	3	–	9	PC ← PC + 3 + jdisp8 if PSW.bit = 1	
		[HL].bit, $addr16	3	10	11 + n	PC ← PC + 3 + jdisp8 if (HL).bit = 1	
	BF	saddr.bit, $addr16	4	10	11	PC ← PC + 4 + jdisp8 if (saddr.bit) = 0	
		sfr.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if sfr.bit = 0	
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 0	
		PSW.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if PSW.bit = 0	
		[HL].bit, $addr16	3	10	11 + n	PC ← PC + 3 + jdisp8 if (HL).bit = 0	
	BTCLR	saddr.bit, $addr16	4	10	12	PC ← PC + 4 + jdisp8 if (saddr.bit) = 1 then reset (saddr.bit)	
		sfr.bit, $addr16	4	–	12	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit	
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit	
		PSW.bit, $addr16	4	–	12	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	× × ×
		[HL].bit, $addr16	3	10	12 + n + m	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit	
	DBNZ	B, $addr16	2	6	–	B ← B – 1, then PC ← PC + 2 + jdisp8 if B ≠ 0	
		C, $addr16	2	6	–	C ← C –1, then PC ← PC + 2 + jdisp8 if C ≠ 0	
		saddr, $addr16	3	8	10	(saddr) ← (saddr) – 1, then PC ← PC + 3 + jdisp8 if (saddr) ≠ 0	
CPU control	**SEL**	RBn	2	4	–	RBS1, 0 ← n	
	NOP		1	2	–	No Operation	
	EI		2	–	6	IE ← 1 (Enable Interrupt)	
	DI		2	–	6	IE ← 0 (Disable Interrupt)	
	HALT		2	6	–	Set HALT Mode	
	STOP		2	6	–	Set STOP Mode	

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to internal ROM program.

3. n is the number of waits when external memory expansion area is read from.

4. m is the number of waits when external memory expansion area is written to.

23.3 Instructions Listed by Addressing Type

(1) 8-bit instructions

MOV, XCH, ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP, MULU, DIVUW, INC, DEC, ROR, ROL, RORC, ROLC, ROR4, ROL4, PUSH, POP, DBNZ

First Operand \ Second Operand	#byte	A	r[Note]	sfr	saddr	!addr16	PSW	[DE]	[HL]	[HL + byte] [HL + B] [HL + C]	$addr16	1	None
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP		ROR ROL RORC ROLC	
r	MOV	MOV ADD ADDC SUB SUBC AND OR XOR CMP											INC DEC
B, C											DBNZ		
sfr	MOV	MOV											
saddr	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV									DBNZ		INC DEC
!addr16		MOV											
PSW	MOV	MOV											PUSH POP
[DE]		MOV											
[HL]		MOV											ROR4 ROL4
[HL + byte] [HL + B] [HL + C]		MOV											
X													MULU
C													DIVUW

Note Except r = A

(2) 16-bit instructions

MOVW, XCHW, ADDW, SUBW, CMPW, PUSH, POP, INCW, DECW

First Operand \ Second Operand	#word	AX	rp[Note]	sfrp	saddrp	!addr16	SP	None
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	MOVW	MOVW	
rp	MOVW	MOVW[Note]						INCW DECW PUSH POP
sfrp	MOVW	MOVW						
saddrp	MOVW	MOVW						
!addr16		MOVW						
SP	MOVW	MOVW						

Note Only when rp = BC, DE, HL

(3) Bit manipulation instructions

MOV1, AND1, OR1, XOR1, SET1, CLR1, NOT1, BT, BF, BTCLR

First Operand \ Second Operand	A.bit	sfr.bit	saddr.bit	PSW.bit	[HL].bit	CY	$addr16	None
A.bit						MOV1	BT BF BTCLR	SET1 CLR1
sfr.bit						MOV1	BT BF BTCLR	SET1 CLR1
saddr.bit						MOV1	BT BF BTCLR	SET1 CLR1
PSW.bit						MOV1	BT BF BTCLR	SET1 CLR1
[HL].bit						MOV1	BT BF BTCLR	SET1 CLR1
CY	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1			SET1 CLR1 NOT1

(4) Call instructions/branch instructions

CALL, CALLF, CALLT, BR, BC, BNC, BZ, BNZ, BT, BF, BTCLR, DBNZ

Second Operand / First Operand	AX	!addr16	!addr11	[addr5]	$addr16
Basic instruction	BR	CALL BR	CALLF	CALLT	BR BC BNC BZ BNZ
Compound instruction					BT BF BTCLR DBNZ

(5) Other instructions

ADJBA, ADJBS, BRK, RET, RETI, RETB, SEL, NOP, EI, DI, HALT, STOP

Absolute maximum ratings (T_A = 25°C)

Parameter	Symbol	Conditions		Ratings	Unit
Supply voltage	V_{DD}			–0.3 to +6.5	V
	V_{PP} Note 1			–0.3 to +10.5	V
	AV_{REF0}			–0.3 to V_{DD} + 0.3 Note 2	V
	AV_{REF1}				
	AV_{SS0}			–0.3 to +0.3	V
	AV_{SS1}				
Input voltage	V_{I1}	P00 to P05, P10 to P17, P20 to P25, P30 to P34, P40 to P47, P50 to P57, P64 to P67, P70 to P73, P120, X1, X2, XT1, XT2, $\overline{\text{RESET}}$		–0.3 to V_{DD} + 0.3 Note 2	V
	V_{I2}	P60 to P63	N-ch open-drain	–0.3 to +13	V
			N-ch open-drain, mask option	–0.3 to V_{DD} + 0.3	V
Output voltage	V_O			–0.3 to V_{DD} + 0.3 Notes 2, 3	V
Analog input voltage	V_{AN}	P10 to P17, ANI8, ANI9	Analog input pin	AV_{SS} – 0.3 to AV_{REF0} + 0.3 and –0.3 to V_{DD} + 0.3	V
Output current, high	I_{OH}	Per pin for P00 to P05, P20 to P25, P30 to P34, P40 to P47, P50 to P57, P64 to P67, P70 to P73, P80 to P87, P90 to P97, P120		–10	mA
		Total for P00 to P05, P20 to P25, P30 to P34, P40 to P47, P50 to P57, P64 to P67, P70 to P73		–15	mA
		Total for P80 to P87, P90 to P97, P120		–15	mA
Output current, low	I_{OL}	Per pin for P00 to P05, P20 to P25, P30 to P34, P40 to P47, P50 to P57, P70 to P73, P80 to P87, P90 to P97, P120		20	mA
		Per pin for P60 to P63		30	mA
		Per pin for P64 to P67		30	mA
		Total for P80 to P87, P90 to P97, P120		20	mA
		Total for P00 to P05, P20 to P25, P30 to P34, P40 to P47, P50 to P57, P60 to P67, P70 to P73		170	mA
Operating ambient temperature	T_A			–40 to +85	°C
Storage temperature	T_{stg}			–65 to +150	°C

Notes 1. μPD78F0338 only

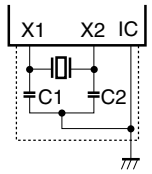
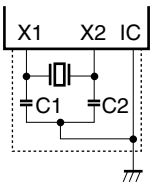
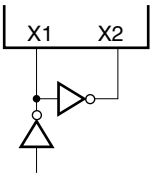
2. 6.5 V or less

3. –0.3 to V_{LC0} + 0.3 V for common and segment pins

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Main system clock oscillator characteristics (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic resonator		Oscillation frequency (f_X)**Note 1**		1.0		10	MHz
		Oscillation stabilization time**Note 2**	After V_{DD} reaches oscillation voltage range MIN.			4	ms
Crystal resonator		Oscillation frequency (f_X)**Note 1**		1.0		10	MHz
		Oscillation stabilization time**Note 2**	V_{DD} = 4.5 to 5.5 V			10	ms
			V_{DD} = 1.8 to 5.5 V			30	ms
External clock		X1 input frequency (f_X)**Note 1**	V_{DD} = 4.5 to 5.5 V	1.0		10	MHz
			V_{DD} = 1.8 to 5.5 V			5.0	MHz
		X1 input high-/low-level width (t_{XH}, t_{XL})	V_{DD} = 4.5 to 5.5 V	42.5		500	ns
			V_{DD} = 1.8 to 5.5 V	85		500	ns

Notes 1. Indicates only oscillator characteristics.

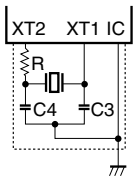
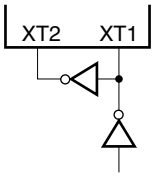
2. Time required to stabilize oscillation after reset or STOP mode release.

Cautions 1. When using the main system clock oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

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

2. When the main system clock is stopped and the system is operated by the subsystem clock, the subsystem clock should be switched again to the main system clock after the oscillation stabilization time is secured by the program.

Subsystem clock oscillator characteristics (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Crystal resonator		Oscillation frequency (f_{XT})**Note 1**		32	32.768	35	kHz
		Oscillation stabilization time**Note 2**	V_{DD} = 4.5 to 5.5 V		1.2	2	s
			V_{DD} = 1.8 to 5.5 V			10	s
External clock		XT1 input frequency (f_{XT})**Note 1**		32		38.5	kHz
		XT1 input high-/low-level width (t_{XTH}, t_{XTL})		5		15	μs

Notes 1. Indicates only oscillator characteristics.

2. Time required to stabilize oscillation after 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 figures to avoid an adverse effect from wiring capacitance.

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

2. The subsystem clock oscillator is designed as a low-amplitude circuit for reducing current consumption, and is more prone to malfunction due to noise than the main system clock oscillator. Particular care is therefore required with the wiring method when the subsystem clock is used.

Remark For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

Capacitance (T_A = 25°C, V_{DD} = V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input capacitance	C_{IN}	f = 1 MHz Unmeasured pins returned to 0 V.				15	pF
Output capacitance	C_{OUT}	f = 1 MHz Unmeasured pins returned to 0 V.				15	pF
I/O capacitance	C_{IO}	f = 1 MHz Unmeasured pins returned to 0 V.	P00 to P05, P20 to P25, P30 to P34, P40 to P47, P50 to P57, P64 to P67, P70 to P73, P120			15	pF
			P60 to P63			20	pF

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Recommended Oscillator Constants

(1) μPD780316, 780318, 780326, 780328, 780336, 780338

Main system clock: Ceramic resonator (T_A = −40 to +85°C)

Manufacturer	Part Number	Frequency (MHz)	Recommended Circuit Constant			Oscillation Voltage Range	
			C1 (pF)	C2 (pF)	Rd (kΩ)	MIN.	MAX.
Murata Mfg. Co., Ltd.	CSBFB1M00J58-R1	1.00	150	150	1	1.9	5.5
	CSBLA1M00J58-B0						
	CSTCC2M00G56-R0	2.00	Internal	Internal	0	1.8	5.5
	CSTLS2M00G56-B0						
	CSTCR4M00G53-R0	4.00	Internal	Internal	0	1.8	5.5
	CSTLS4M00G53-B0						
	CSTCC8M38G53093-R0	8.38	Internal	Internal	0	1.8	5.5
	CSTLS8M38G53093-B0						
	CSTCC8M38G53-R0		Internal	Internal	0	1.9	5.5
	CSTLS8M38G53-B0						
	CSTCC10M0G53093-R0	10	Internal	Internal	0	1.8	5.5
	CSTLS10M00G53093-B0						
	CSTCC10M0G53-R0		Internal	Internal	0	2.0	5.5
	CSTLS10M00G53-B0						

Caution The oscillator constant and oscillation voltage range indicate conditions of stable oscillation. Oscillation frequency precision is not guaranteed. For applications requiring oscillation frequency precision, the oscillation frequency must be adjusted on the implementation circuit. For details, please contact directly the manufacturer of the resonator to be used.

(2) μPD78F0338

Main system clock: Ceramic resonator (T_A = −40 to +85°C)

Manufacturer	Part Number	Frequency (MHz)	Recommended Circuit Constant			Oscillation Voltage Range	
			C1 (pF)	C2 (pF)	Rd (kΩ)	MIN.	MAX.
Murata Mfg. Co., Ltd.	CSBFB1M00J58-R1	1.00	150	150	1	2.1	5.5
	CSBLA1M00J58-B0						
	CSTCC2M00G56-R0	2.00	Internal	Internal	0	1.9	5.5
	CSTLS2M00G56-B0						
	CSTCR4M00G53093-R0	4.00	Internal	Internal	0	1.8	5.5
	CSTLS4M00G53093-B0						
	CSTCR4M00G53-R0		Internal	Internal	0	1.9	5.5
	CSTLS4M00G53-B0						
	CSTCC8M38G53U-R0	8.38	Internal	Internal	0	1.9	5.5
	CSTLS8M38G53U-B0						
	CSTCC8M38G53-R0		Internal	Internal	0	2.1	5.5
	CSTLS8M38G53-B0						
	CSTCC10M0G53U-R0	10	Internal	Internal	0	2.0	5.5
	CSTLS10M00G53U-B0						
	CSTCC10M0G53-R0		Internal	Internal	0	2.2	5.5
	CSTLS10M00G53-B0						

Caution The oscillator constant and oscillation voltage range indicate conditions of stable oscillation. Oscillation frequency precision is not guaranteed. For applications requiring oscillation frequency precision, the oscillation frequency must be adjusted on the implementation circuit. For details, please contact directly the manufacturer of the resonator to be used.

DC characteristics (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high	I_{OH}	Per pin for P00 to P05, P20 to P25, P30 to P34, P40 to P47, P50 to P57, P64 to P67, P70 to P73, P80 to P87, P90 to P97, P120				–1	mA
		All pins				–20	mA
Output current, low	I_{OL}	Per pin for P00 to P05, P20 to P25, P30 to P34, P40 to P47, P50 to P57, P70 to P73, P80 to P87, P90 to P97, P120				10	mA
		Per pin for P60 to P63				15	mA
		Per pin for P64 to P67				15	mA
		Total for P80 to P87, P90 to P97, P120				20	mA
		Total for P00 to P05, P20 to P25, P30 to P34, P40 to P47, P50 to P57, P70 to P73				10	mA
		Total for P60 to P63				60	mA
		Total for P64 to P67				60	mA
Input voltage, high	V_{IH1}	P10 to P17, P21, P24, P30, P40 to P47, P50 to P57, P64 to P67, P70, P72	2.7 V ≤ V_{DD} ≤ 5.5 V	0.7V_{DD}		V_{DD}	V
			1.8 V ≤ V_{DD} ≤ 5.5 V	0.8V_{DD}		V_{DD}	V
	V_{IH2}	P00 to P05, P20, P22, P23, P25, P31 to P34, P71, P73, $\overline{\text{RESET}}$	2.7 V ≤ V_{DD} ≤ 5.5 V	0.8V_{DD}		V_{DD}	V
			1.8 V ≤ V_{DD} ≤ 5.5 V	0.85V_{DD}		V_{DD}	V
	V_{IH3}	P60 to P63	2.7 V ≤ V_{DD} ≤ 5.5 V	0.7V_{DD}		12	V
			1.8 V ≤ V_{DD} ≤ 5.5 V	0.8V_{DD}		12	V
	V_{IH4}	X1, X2	2.7 V ≤ V_{DD} ≤ 5.5 V	V_{DD} – 0.5		V_{DD}	V
			1.8 V ≤ V_{DD} ≤ 5.5 V	V_{DD} – 0.2		V_{DD}	V
	V_{IH5}	XT1, XT2	4.5 V ≤ V_{DD} ≤ 5.5 V	0.8V_{DD}		V_{DD}	V
			1.8 V ≤ V_{DD} ≤ 5.5 V	0.9V_{DD}		V_{DD}	V
	V_{IH6}	P120	2.7 V ≤ V_{DD} ≤ 5.5 V	0.8V_{DD}		V_{DD}	V
			1.8 V ≤ V_{DD} ≤ 5.5 V	0.85V_{DD}		V_{DD}	V
Input voltage, low	V_{IL1}	P10 to P17, P21, P24, P30, P40 to P47, P50 to P57, P64 to P67, P70, P72	2.7 V ≤ V_{DD} ≤ 5.5 V	0		0.3V_{DD}	V
			1.8 V ≤ V_{DD} ≤ 5.5 V	0		0.2V_{DD}	V
	V_{IL2}	P00 to P05, P20, P22, P23, P25, P31 to P34, P71, P73, $\overline{\text{RESET}}$	2.7 V ≤ V_{DD} ≤ 5.5 V	0		0.2V_{DD}	V
			1.8 V ≤ V_{DD} ≤ 5.5 V	0		0.15V_{DD}	V
	V_{IL3}	P60 to P63	4.5 V ≤ V_{DD} ≤ 5.5 V	0		0.3V_{DD}	V
			2.7 V ≤ V_{DD} < 4.5 V	0		0.2V_{DD}	V
			1.8 V ≤ V_{DD} < 2.7 V	0		0.1V_{DD}	V
	V_{IL4}	X1, X2	2.7 V ≤ V_{DD} ≤ 5.5 V	0		0.4	V
			1.8 V ≤ V_{DD} ≤ 5.5 V	0		0.2	V
	V_{IL5}	XT1, XT2	4.5 V ≤ V_{DD} ≤ 5.5 V	0		0.2V_{DD}	V
			1.8 V ≤ V_{DD} ≤ 5.5 V	0		0.1V_{DD}	V
	V_{IL6}	P120	2.7 V ≤ V_{DD} ≤ 5.5 V	0		0.2V_{DD}	V
			1.8 V ≤ V_{DD} ≤ 5.5 V	0		0.15V_{DD}	V

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC characteristics (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output voltage, high	V_{OH}	V_{DD} = 4.0 to 5.5 V, I_{OH} = –1 mA		V_{DD} – 1.0		V_{DD}	V
		V_{DD} = 1.8 to 5.5 V, I_{OH} = –100 μA		V_{DD} – 0.5		V_{DD}	V
Output voltage, low	V_{OL1}	P60 to P63	V_{DD} = 4.5 to 5.5 V, I_{OL} = 15 mA		0.4	1.0	V
	V_{OL2}	P64 to P67	V_{DD} = 4.5 to 5.5 V, I_{OL} = 15 mA		0.4	2.0	V
	V_{OL3}	P00 to P05, P20 to P25, P30 to P34, P40 to P47, P50 to P57, P70 to P73, P80 to P87, P90 to P97, P120	V_{DD} = 4.5 to 5.5 V, I_{OL} = 1.6 mA			0.4	V
	V_{OL4}	I_{OL} = 400 μA				0.5	V
Input leakage current, high	I_{LIH1}	V_{IN} = V_{DD}	P00 to P05, P10 to P17, P20 to P25, P30 to P34, P40 to P47, P50 to P57, P60 to P67, P70 to P73, P120, $\overline{\text{RESET}}$			3	μA
	I_{LIH2}		X1, X2, XT1, XT2			20	μA
	I_{LIH3}	V_{IN} = 12 V	P60 to P63			10	μA
Input leakage current, low	I_{LIL1}	V_{IN} = 0 V	P00 to P05, P10 to P17, P20 to P25, P30 to P34, P40 to P47, P50 to P57, P64 to P67, P70 to P73, P120, $\overline{\text{RESET}}$			–3	μA
	I_{LIL2}		X1, X2, XT1, XT2			–20	μA
	I_{LIL3}		P60 to P63 (N-ch open-drain)			–3[Note]	μA
Output leakage current, high	I_{LOH}	V_{OUT} = V_{DD}				3	μA
Output leakage current, low	I_{LOL}	V_{OUT} = 0 V				–3	μA
Mask option pull-up resistor (μPD780316, 780318, 780326, 780328, 780336, 780338 only)	R_1	V_{IN} = 0 V, P60, P61, P62, P63		20	40	90	kΩ
Software pull-up resistor	R_2	V_{IN} = 0 V, P00 to P05, P20 to P25, P30 to P34, P40 to P47, P50 to P57, P64 to P67, P70 to P73, P120		15	30	90	kΩ

Note During input instruction execution, the low-level input leakage current for P60 to P63 is –200 μA (MAX.) only for 1 clock (no wait). During execution of other instructions, this value is –3 μA (MAX.).

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC characteristics (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V): μPD780316, 780318, 780326, 780328, 780336, 780338

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Power supply current[Note 1]	I_{DD1}[Note 2]	10 MHz crystal oscillation operating mode	V_{DD} = 5.0 V ±10%[Note 3]	When A/D converter stopped		6.3	12.6	mA
				When A/D converter is operating		7.3	14.6	mA
		5.0 MHz crystal oscillation operating mode	V_{DD} = 3.0 V ±10%[Note 3]	When A/D converter stopped		2.0	4.0	mA
				When A/D converter is operating		3.0	6.0	mA
			V_{DD} = 2.0 V ±10%[Note 4]	When A/D converter stopped		0.4	1.5	mA
				When A/D converter is operating		1.4	4.2	mA
	I_{DD2}	10 MHz crystal oscillation HALT mode	V_{DD} = 5.0 V ±10%[Note 3]	When peripheral function stopped		1.15	2.3	mA
				When peripheral function is operating			5.7	mA
		5.0 MHz crystal oscillation HALT mode	V_{DD} = 3.0 V ±10%[Note 3]	When peripheral function stopped		0.35	0.7	mA
				When peripheral function is operating			1.7	mA
			V_{DD} = 2.0 V ±10%[Note 4]	When peripheral function stopped		0.15	0.4	mA
				When peripheral function is operating			1.1	mA
	I_{DD3}	32.768 kHz crystal oscillation operating mode[Note 5]	V_{DD} = 5.0 V ±10%			40	80	μA
			V_{DD} = 3.0 V ±10%			20	40	μA
			V_{DD} = 2.0 V ±10%			10	20	μA
	I_{DD4}	32.768 kHz crystal oscillation HALT mode	V_{DD} = 5.0 V ±10%	When LCD stopped[Note 6]		25	45	μA
				Only when LCD boost function is operating[Note 7]		27	51	μA
				When LCD is operating[Note 8]		30	60	μA
			V_{DD} = 3.0 V ±10%	When LCD stopped[Note 6]		6	18	μA
				Only when LCD boost function is operating[Note 7]		7.5	23	μA
				When LCD is operating[Note 8]		10	30	μA
			V_{DD} = 2.0 V ±10%	When LCD stopped[Note 6]		3	10	μA
				Only when LCD boost function is operating[Note 7]		4	12	μA
				When LCD is operating[Note 8]		6	18	μA
	I_{DD5}	STOP mode	V_{DD} = 5.0 V ±10%			0.1	30	μA
			V_{DD} = 3.0 V ±10%			0.05	10	μA
			V_{DD} = 2.0 V ±10%			0.05	10	μA

Notes 1. Total current flowing in the internal power supply (V_{DD1}, AV_{REF0}).

2. Includes the peripheral operating current. However, the current flowing in the pull-up resistor on the port is not included.

3. When the processor clock control register (PCC) is set to 00H.

4. When PCC is set to 02H.

5. When the main system clock has been stopped.

6. Supply current when LCD is stopped (LCDON = 0, SCOC = 0, VLCON = 0)

7. Supply current only when the LCD boost function is operating (LCDON = 0, SCOC = 0, VLCON = 1) in the following status:

- No load without LCD display panel connected
- Capacitors C1 to C4 for boost: 0.47 μF
- When boosting is stabilized

8. Supply current when the LCD is operating (LCDON = 1, SCOC = 1, VLCON = 1) in the following status:

- No load without LCD display panel connected
- Capacitors C1 to C4 for boost: 0.47 μF
- When boosting is stabilized

DC characteristics (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V): μPD78F0338

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Power supply current[Note 1]	I_{DD1}[Note 2]	10 MHz crystal oscillation operating mode	V_{DD} = 5.0 V ±10%[Note 3]	When A/D converter stopped		15	30	mA
				When A/D converter is operating		16	32	mA
		5.0 MHz crystal oscillation operating mode	V_{DD} = 3.0 V ±10%[Note 3]	When A/D converter stopped		4.5	9	mA
				When A/D converter is operating		5.5	11	mA
			V_{DD} = 2.0 V ±10%[Note 4]	When A/D converter stopped		2.8	5.6	mA
				When A/D converter is operating		3.8	7.6	mA
	I_{DD2}	10 MHz crystal oscillation HALT mode	V_{DD} = 5.0 V ±10%[Note 3]	When peripheral function stopped		1.25	2.5	mA
				When peripheral function is operating			5.7	mA
		5.0 MHz crystal oscillation HALT mode	V_{DD} = 3.0 V ±10%[Note 3]	When peripheral function stopped		0.4	0.8	mA
				When peripheral function is operating			1.7	mA
			V_{DD} = 2.0 V ±10%[Note 4]	When peripheral function stopped		0.2	0.4	mA
				When peripheral function is operating			1.1	mA
	I_{DD3}	32.768 kHz crystal oscillation operating mode[Note 5]	V_{DD} = 5.0 V ±10%			115	230	μA
			V_{DD} = 3.0 V ±10%			95	190	μA
			V_{DD} = 2.0 V ±10%			75	150	μA
	I_{DD4}	32.768 kHz crystal oscillation HALT mode	V_{DD} = 5.0 V ±10%	When LCD stopped[Note 6]		25	45	μA
				Only when LCD boost function is operating[Note 7]		27	51	μA
				When LCD is operating[Note 8]		30	60	μA
			V_{DD} = 3.0 V ±10%	When LCD stopped[Note 6]		6	18	μA
				Only when LCD boost function is operating[Note 7]		7.5	23	μA
				When LCD is operating[Note 8]		10	30	μA
			V_{DD} = 2.0 V ±10%	When LCD stopped[Note 6]		3	10	μA
				Only when LCD boost function is operating[Note 7]		4	12	μA
				When LCD is operating[Note 8]		6	18	μA
	I_{DD5}	STOP mode	V_{DD} = 5.0 V ±10%			0.1	30	μA
			V_{DD} = 3.0 V ±10%			0.05	10	μA
			V_{DD} = 2.0 V ±10%			0.05	10	μA

Notes 1. Total current flowing in the internal power supply (V_{DD1}, AV_{REF0}).

2. Includes the peripheral operating current. However, the current flowing in the pull-up resistor on the port is not included.

3. When the processor clock control register (PCC) is set to 00H.

4. When PCC is set to 02H.

5. When the main system clock has been stopped.

6. Supply current when LCD is stopped (LCDON = 0, SCOC = 0, VLCON = 0)

7. Supply current only when the LCD boost function is operating (LCDON = 0, SCOC = 0, VLCON = 1) in the following status:

- No load without LCD display panel connected
- Capacitors C1 to C4 for boost: 0.47 μF
- When boosting is stabilized

8. Supply current when the LCD is operating (LCDON = 1, SCOC = 1, VLCON = 1) in the following status:

- No load without LCD display panel connected
- Capacitors C1 to C4 for boost: 0.47 μF
- When boosting is stabilized

AC characteristics

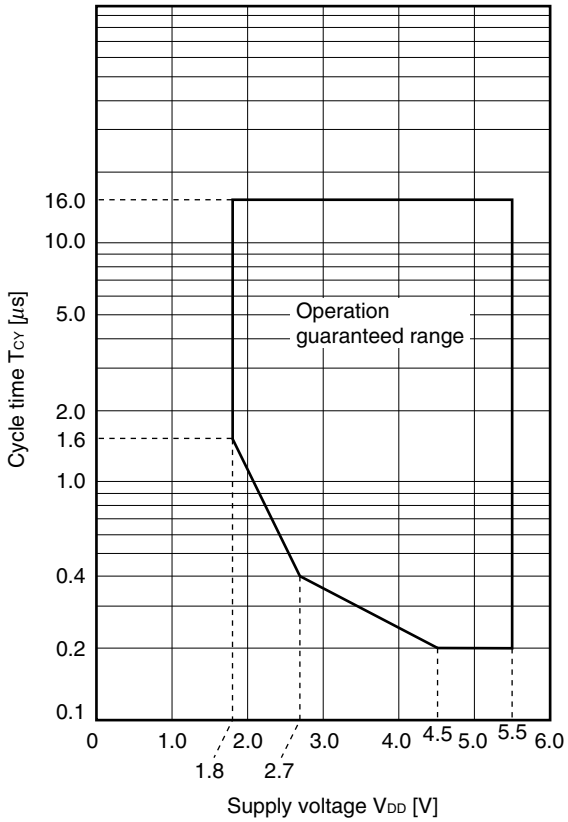
(1) Basic operation (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Cycle time (minimum instruction execution time)	T_{CY}	Operating with main system clock	4.5 V ≤ V_{DD} ≤ 5.5 V	0.2		16	μs
			2.7 V ≤ V_{DD} < 4.5 V	0.4		16	μs
			1.8 V ≤ V_{DD} < 2.7 V	1.6		16	μs
		Operating with subsystem clock		103.9 Note 1	122	125	μs
TI00, TI01 input high-/low-level width	t_{TIH0} t_{TIL0}		3.5 V ≤ V_{DD} ≤ 5.5 V	$2/f_{sam}$ + 0.1 Note 2			μs
			2.7 V ≤ V_{DD} < 3.5 V	$2/f_{sam}$ + 0.2 Note 2			μs
			1.8 V ≤ V_{DD} < 2.7 V	$2/f_{sam}$ + 0.5 Note 2			μs
TI4 input frequency	f_{TI4}	V_{DD} = 2.7 to 5.5 V		0		4	MHz
		V_{DD} = 1.8 to 5.5 V		0		275	kHz
TI4 input high-/low-level width	t_{TIH4} t_{TIL4}	V_{DD} = 2.7 to 5.5 V		100			ns
		V_{DD} = 1.8 to 5.5 V		1.8			μs
TI50, TI51, TI52 input frequency	f_{TI5}	V_{DD} = 2.7 to 5.5 V		0		4	MHz
		V_{DD} = 1.8 to 5.5 V		0		275	kHz
TI50, TI51, TI52 input high-/low-level width	t_{TIH5} t_{TIL5}	V_{DD} = 2.7 to 5.5 V		100			ns
		V_{DD} = 1.8 to 5.5 V		1.8			μs
Interrupt request input high-/low-level width	t_{INTH} t_{INTL}	INTP0 to INTP5, P40 to P47	V_{DD} = 2.7 to 5.5 V	1			μs
			V_{DD} = 1.8 to 5.5 V	2			μs
$\overline{RESET}$ low-level width	t_{RSL}			10			μs

Notes 1. Value when using the external clock. When using a crystal resonator, the value becomes 114 μs (MIN.).

2. Selection of f_{sam} = f_X, f_X/4, f_X/64 is available with bits 0 and 1 (PRM00, PRM01) of prescaler mode register 0 (PRM0). However, if the TI00 valid edge is selected as the count clock, the value becomes f_{sam} = f_X/8.

T_{CY} vs. V_{DD} (with main system clock operation)



(2) Serial interface (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

(a) SIO3 3-wire serial I/O mode ($\overline{SCK3}$ ... internal clock output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{SCK3}$ cycle time	t_{KCY1}	4.5 V ≤ V_{DD} ≤ 5.5 V	800			ns
		2.7 V ≤ V_{DD} < 4.5 V	1,600			ns
		1.8 V ≤ V_{DD} < 2.7 V	3,200			ns
$\overline{SCK3}$ high-/low-level width	t_{KH1}, t_{KL1}	4.5 V ≤ V_{DD} ≤ 5.5 V	$t_{KCY1}/2 - 50$			ns
		1.8 V ≤ V_{DD} < 4.5 V	$t_{KCY1}/2 - 100$			ns
SI3 setup time (to $\overline{SCK3}$↑)	t_{SIK1}	4.5 V ≤ V_{DD} ≤ 5.5 V	100			ns
		2.7 V ≤ V_{DD} < 4.5 V	150			ns
		1.8 V ≤ V_{DD} < 2.7 V	300			ns
SI3 hold time (from $\overline{SCK3}$↑)	t_{KSI1}		400			ns
Delay time from $\overline{SCK3}$↓ to SO3 output	t_{KSO1}	C = 100 pF**Note**			300	ns

Note C is the load capacitance of the $\overline{SCK3}$ and SO3 output lines.

(b) SIO3 3-wire serial I/O mode ($\overline{SCK3}$ ... external clock input)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{SCK3}$ cycle time	t_{KCY2}	4.5 V ≤ V_{DD} ≤ 5.5 V	800			ns
		2.7 V ≤ V_{DD} < 4.5 V	1,600			ns
		1.8 V ≤ V_{DD} < 2.7 V	3,200			ns
$\overline{SCK3}$ high-/low-level width	t_{KH2}, t_{KL2}	4.5 V ≤ V_{DD} ≤ 5.5 V	400			ns
		2.7 V ≤ V_{DD} < 4.5 V	800			ns
		1.8 V ≤ V_{DD} < 2.7 V	1,600			ns
SI3 setup time (to $\overline{SCK3}$↑)	t_{SIK2}		100			ns
SI3 hold time (from $\overline{SCK3}$↑)	t_{KSI2}		400			ns
Delay time from $\overline{SCK3}$↓ to SO3 output	t_{KSO2}	C = 100 pF**Note**			300	ns

Note C is the load capacitance of the SO3 output line.

(c) CSI1 3-wire serial I/O mode (SCK1 ... internal clock output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
SCK1 cycle time	t_{KCY3}	4.5 V ≤ V_{DD} ≤ 5.5 V	200			ns
		2.7 V ≤ V_{DD} < 4.5 V	500			ns
		1.8 V ≤ V_{DD} < 2.7 V	1			μs
SCK1 high-/low-level width	t_{KH3} t_{KL3}	4.5 V ≤ V_{DD} ≤ 5.5 V	$t_{KCY3}/2 - 5$			ns
		2.7 V ≤ V_{DD} < 4.5 V	$t_{KCY3}/2 - 20$			ns
		1.8 V ≤ V_{DD} < 2.7 V	$t_{KCY3}/2 - 30$			ns
SI1 setup time (to SCK1↑)	t_{SIK3}		20			ns
SI1 hold time (from SCK1↑)	t_{KSI3}		110			ns
Delay time from SCK1↓ to SO1 output	t_{KSO3}	C = 100 pF[Note]			150	ns

Note C is the load capacitance of the SCK1 and SO1 output lines.

(d) CSI1 3-wire serial I/O mode (SCK1 ... external clock input)

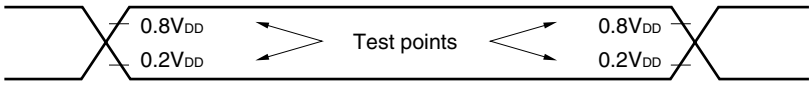
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
SCK1 cycle time	t_{KCY4}	4.5 V ≤ V_{DD} ≤ 5.5 V	200			ns
		2.7 V ≤ V_{DD} < 4.5 V	500			ns
		1.8 V ≤ V_{DD} < 2.7 V	1			μs
SCK1 high-/low-level width	t_{KH4} t_{KL4}	4.5 V ≤ V_{DD} ≤ 5.5 V	100			ns
		2.7 V ≤ V_{DD} < 4.5 V	250			ns
		1.8 V ≤ V_{DD} < 2.7 V	500			ns
SI1 setup time (to SCK1↑)	t_{SIK4}		25			ns
SI1 hold time (from SCK1↑)	t_{KSI4}		110			ns
Delay time from SCK1↓ to SO1 output	t_{KSO4}	C = 100 pF[Note]			150	ns

Note C is the load capacitance of the SO1 output line.

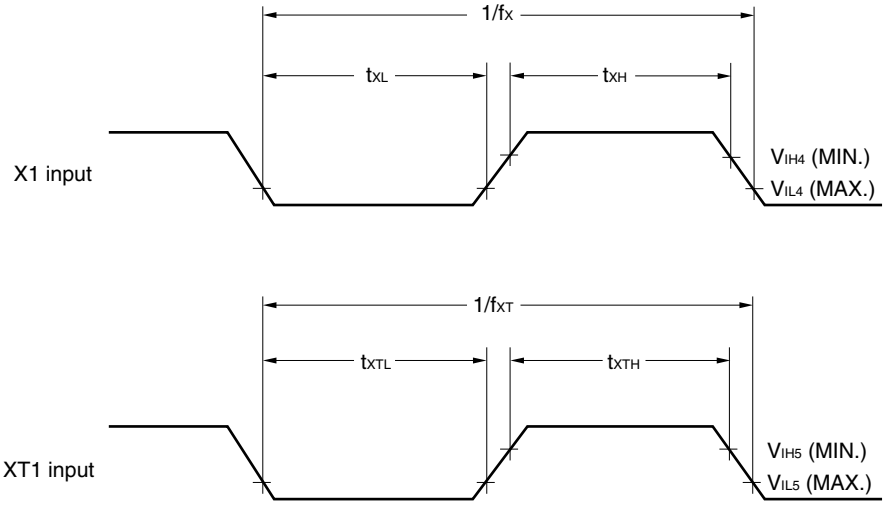
(e) UART0 (dedicated baud rate generator output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		4.5 V ≤ V_{DD} ≤ 5.5 V			156,250	bps
		2.7 V ≤ V_{DD} < 4.5 V			78,125	bps
		1.8 V ≤ V_{DD} < 2.7 V			39,063	bps

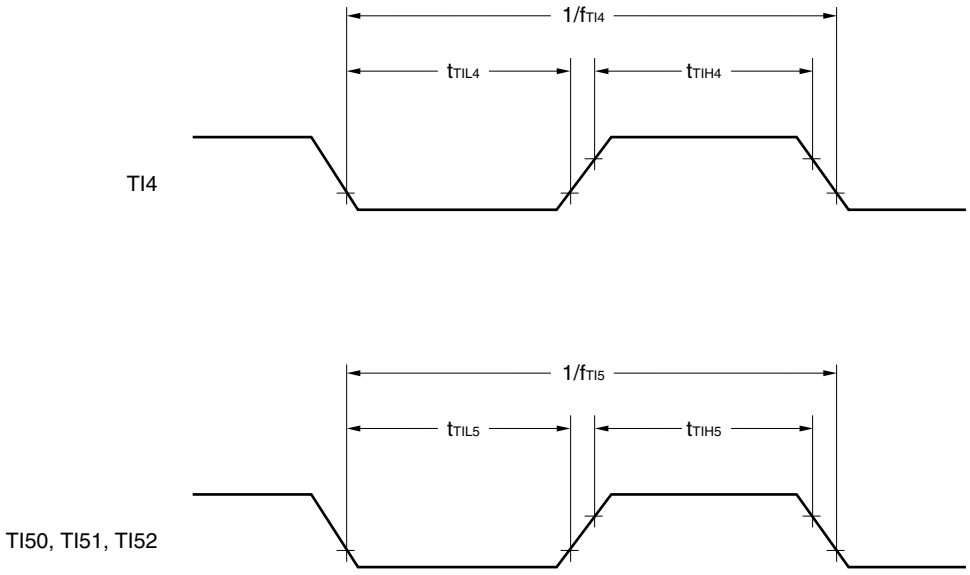
AC timing test point (excluding X1, XT1 input)



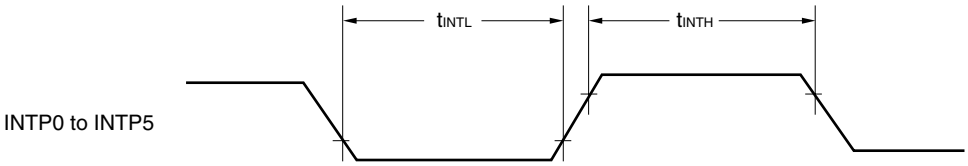
Clock timing



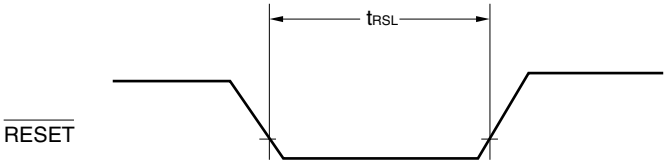
TI timing



Interrupt request input timing

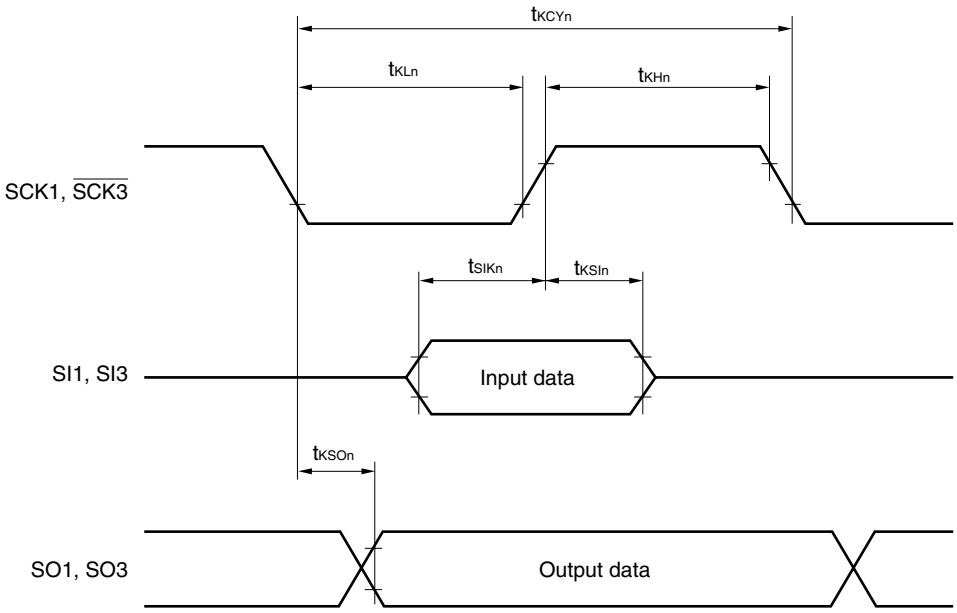


$\overline{\text{RESET}}$ input timing



Serial transfer timing

3-wire serial I/O mode (SIO3, CSI1):



n = 1 to 4

A/D converter characteristics (T_A = –40 to +85°C, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution			10	10	10	bit
Overall error[Note]		4.5 V ≤ V_{DD} ≤ 5.5 V		±0.2	±0.4	%FSR
		2.7 V ≤ V_{DD} < 4.5 V		±0.3	±0.6	%FSR
		2.05 V ≤ V_{DD} < 2.7 V		±0.6	±1.2	%FSR
Conversion time	t_{CONV}	4.5 V ≤ V_{DD} ≤ 5.5 V	14		100	μs
		2.7 V ≤ V_{DD} < 4.5 V	19		100	μs
		2.05 V ≤ V_{DD} < 2.7 V	48		100	μs
Zero-scale error[Note]		4.5 V ≤ V_{DD} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ V_{DD} < 4.5 V			±0.6	%FSR
		2.05 V ≤ V_{DD} < 2.7 V			±1.2	%FSR
Full-scale error[Note]		4.5 V ≤ V_{DD} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ V_{DD} < 4.5 V			±0.6	%FSR
		2.05 V ≤ V_{DD} < 2.7 V			±1.2	%FSR
Integral linearity error		4.5 V ≤ V_{DD} ≤ 5.5 V			±2.5	LSB
		2.7 V ≤ V_{DD} < 4.5 V			±4.5	LSB
		2.05 V ≤ V_{DD} < 2.7 V			±8.5	LSB
Differential linearity error		4.5 V ≤ V_{DD} ≤ 5.5 V			±1.5	LSB
		2.7 V ≤ V_{DD} < 4.5 V			±2.0	LSB
		2.05 V ≤ V_{DD} < 2.7 V			±3.5	LSB
Analog input voltage	V_{IAN}		0		AV_{REF}	V
Analog reference voltage	AV_{REF0}		2.05		V_{DD}	V
Resistance between AV_{REF0} and AV_{SS}	R_{REF0}	At A/D conversion operation	20	40		kΩ

Note Overall error excluding quantization error (±1/2 LSB). It is indicated as a ratio (%FSR) to the full-scale value.

D/A converter characteristics (T_A = –40 to +85°C, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution						8	bit
Overall error[Note 1]		R = 2 MΩ[Note 2]				1.2	%
		R = 4 MΩ[Note 2]				0.8	%
		R = 10 MΩ[Note 2]				0.6	%
Settling time		C = 30 pF	4.5 V ≤ V_{DD} ≤ 5.5 V			10	μs
			2.7 V ≤ V_{DD} < 4.5 V			15	μs
			1.8 V ≤ V_{DD} < 2.7 V			20	μs
Output resistance	R_O	**Note 3**			10		kΩ
Analog reference voltage	AV_{REF1}			1.8		V_{DD}	V
Resistance between AV_{REF1} and AV_{SS}	R_{REF1}	DA0 = 55H[Note 3]		4	8		kΩ

Notes 1. Overall error excluding quantization error (±1/2 LSB). It is indicated as a ratio (%FSR) to the full-scale value.

2. R and C are the D/A converter output pin load resistance and load capacitance, respectively.

3. Value for one D/A converter channel

LCD controller/driver characteristics (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
LCD reference voltage	V_{LCD2}	C1 to C4 = 0.47 μF	Gain[Note 1] = 1	0.84	1	1.165	V
			Gain[Note 1] = 1.5	1.26	1.5	1.74	V
Gain adjustment				1.0		1.5	Times
Doubler output voltage	V_{LCD1}	C1 to C4 = 0.47 μF		$2.0V_{LCD2} - 0.1$	$2.0V_{LCD2}$	$2.0V_{LCD2}$	V
Tripler output voltage	V_{LCD0}	C1 to C4 = 0.47 μF		$3.0V_{LCD2} - 0.15$	$3.0V_{LCD2}$	$3.0V_{LCD2}$	V
Boost wait time[Note 2]	t_{VAWAIT}	Gain = 1	4.5 V ≤ V_{DD} ≤ 5.5 V	4			s
			1.8 V ≤ V_{DD} < 4.5 V	0.5			s
		Gain = 1.5		0.5			s
LCD output resistance[Note 3] (common)	R_{OVC}					40	kΩ
LCD output resistance[Note 3] (segment)	R_{OVS}					200	kΩ

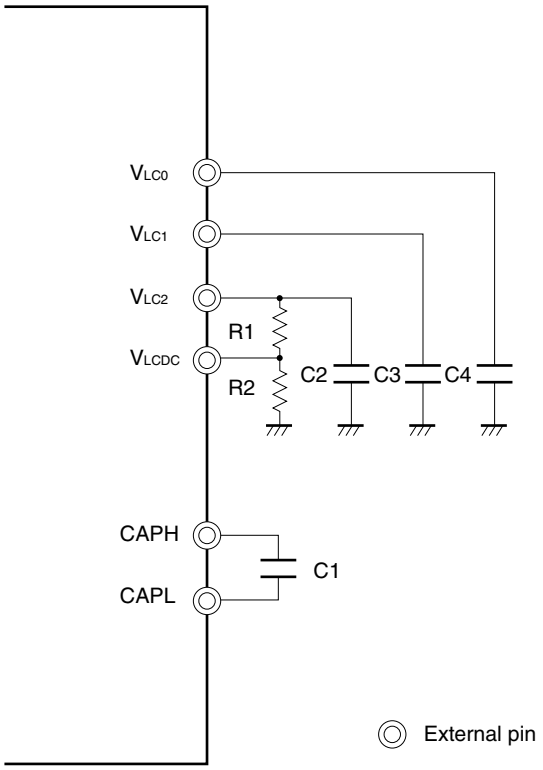
★ **Notes 1.** The gain is a determined by R1 and R2 as follows. For details, refer to **Remark**.

- Gain = (R1 + R2)/R2

2. The boost wait time is the wait time from when boosting is started to when display is enabled.

3. The output resistance is the resistance between one of the V_{LC0}, V_{LC1}, V_{LC2}, V_{SS0}, and V_{SS1} pins, and a segment signal output pin or common signal output pin.

Remark C1, C2, C3, and C4 are the capacitors connected between CAPH and CAPL, V_{LC2} and GND, V_{LC1} and GND, and V_{LC0} and GND, respectively.



- R1 + R2 = 3 [MΩ]
- C1 = C2 = C3 = C4 = 0.47 [μF]

V_{LCD2} can be adjusted according to the voltage division ratio of the resistance of R1 and R2.

- V_{LCD2} = (R1 + R2)/R2 [V]
- $V_{LCD1} = 2 \times V_{LCD2}$ [V]
- $V_{LCD0} = 3 \times V_{LCD2}$ [V]

Recommended values for external circuits are shown below.

	V_{LCD2} (V)	V_{LCD1} (V)	V_{LCD0} (V)	R1 (MΩ)	R2 (MΩ)
V_{LCD0} = 3 V (gain = 1)	1	2	3	0	3
V_{LCD0} = 4.5 V (gain = 1.5)	1.5	3	4.5	1	2

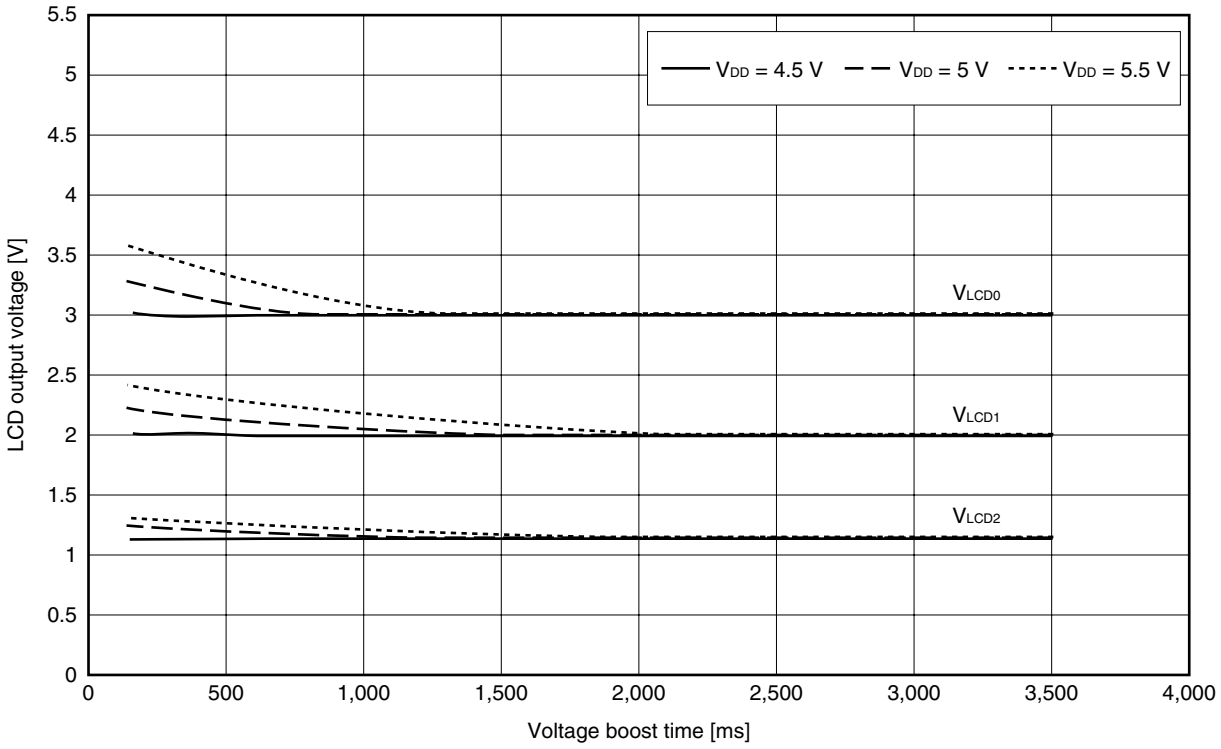
Remark The above LCD output voltage applies when the wiring resistance and capacitance between the V_{LC0}, V_{LC1}, V_{LC2}, V_{LCDC}, CAPH, and CAPL pins and the external circuit is ignored.

Characteristics curves of LCD controller/driver (reference values)

(1) Characteristics curves of voltage boost stabilization time

The following shows the characteristics curves of the time from the start of voltage boost (VLCON = 1) and the changes in the LCD output voltage (when gain = 1 (3 V boost mode), V_{DD} = 4.5 to 5.5 V).

LCD output voltage/voltage boost time (when gain = 1 (3 V boost mode), V_{DD} = 4.5 to 5.5 V)

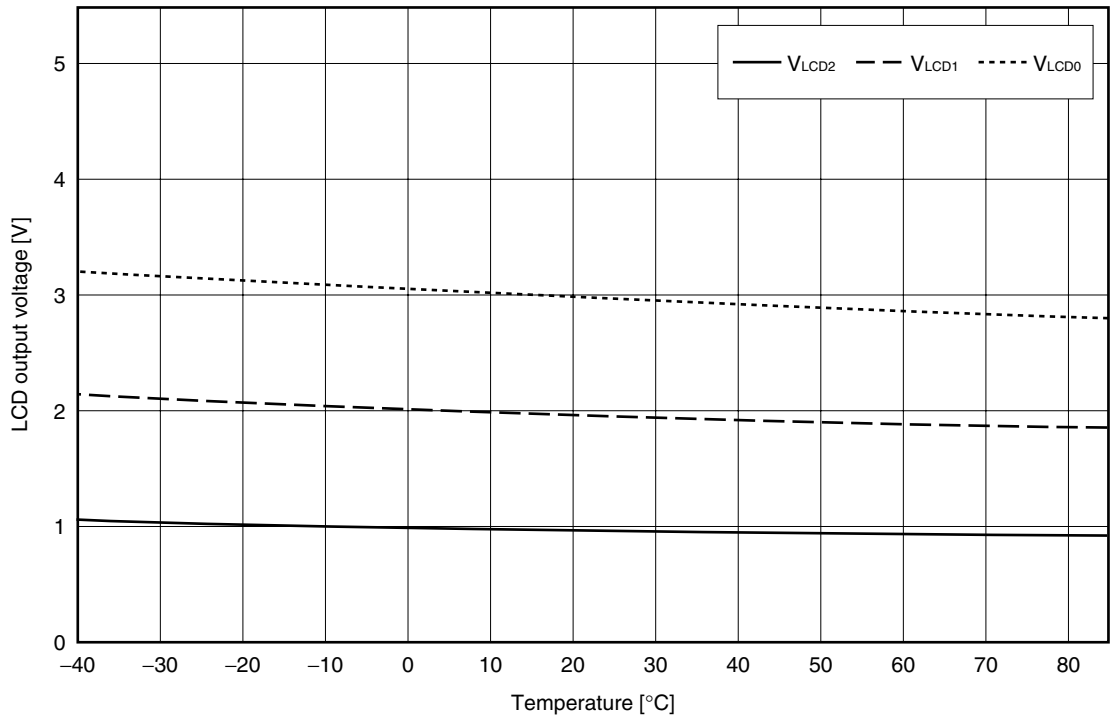


Remark The above characteristics curves are when the external resistance is R1 = 0 [MΩ] and R2 = 3 [MΩ].

(2) Temperature characteristics of LCD output voltage

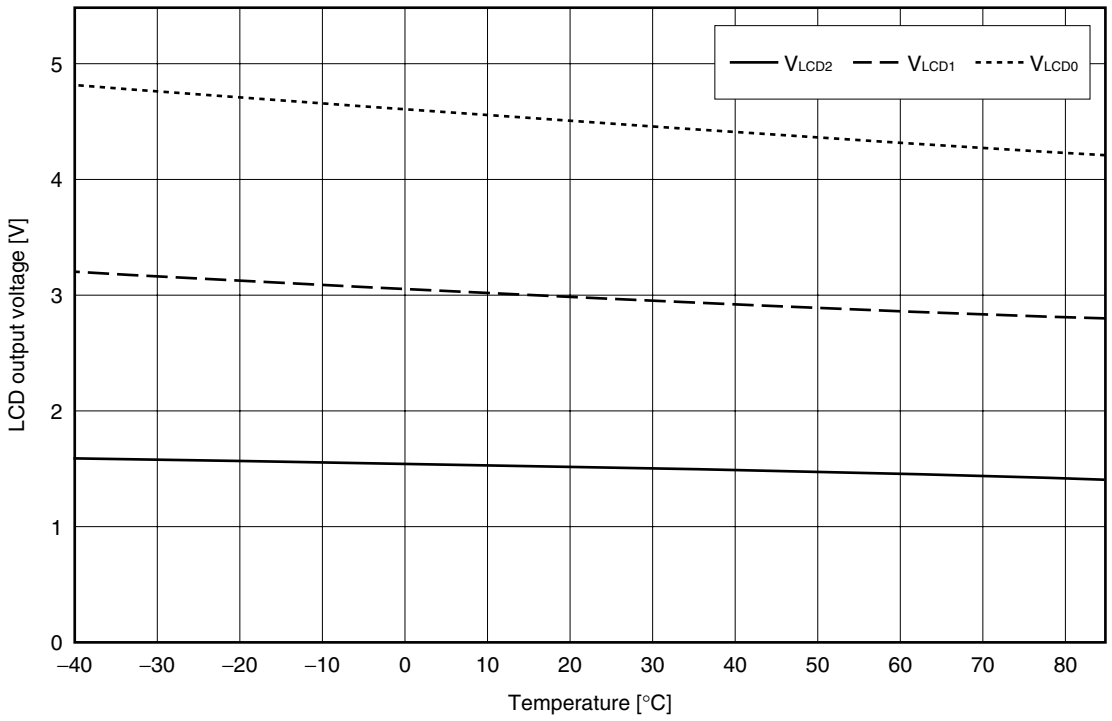
The following shows the temperature characteristics curves of LCD output voltage.

LCD output voltage/temperature (when gain = 1)



Remark The above characteristics curves are when the external resistance is R1 = 0 [MΩ] and R2 = 3 [MΩ].

LCD output voltage/temperature (when gain = 1.5)



Remark The above characteristics curves are when the external resistance is R1 = 1 [MΩ] and R2 = 2 [MΩ].

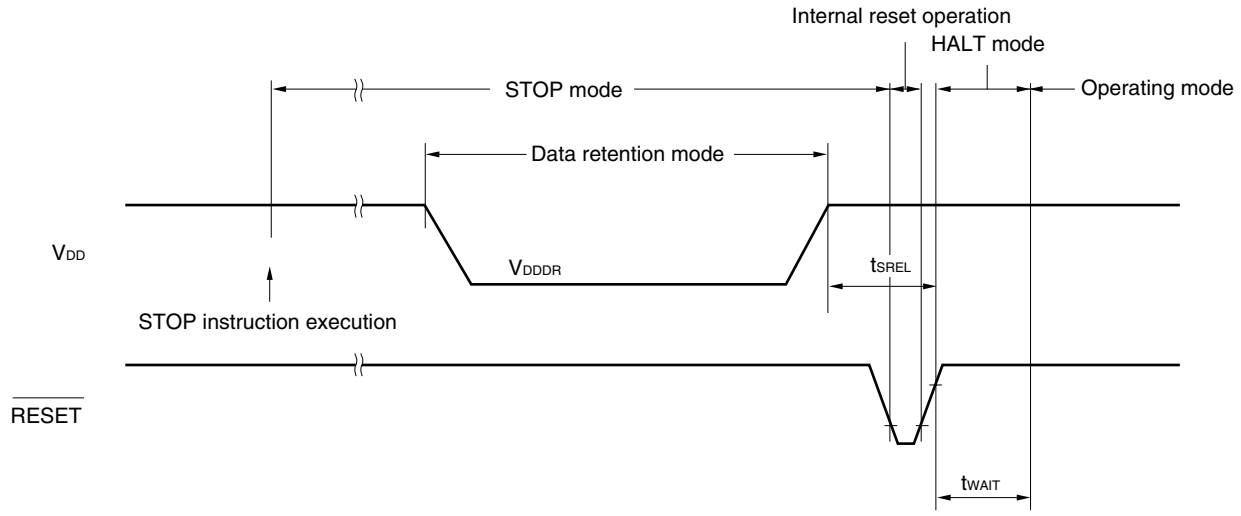
Data memory STOP mode low power supply voltage data retention characteristics (T_A = –40 to +85°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention power supply voltage	V_{DDDR}		1.6		5.5	V
Data retention power supply current	I_{DDDR}	V_{DDDR} = 1.6 V (with subsystem clock stopped and feedback resistor disconnected)		0.1	10	μA
Release signal set time	t_{SREL}		0			μs
Oscillation stabilization wait time	t_{WAIT}	Release by $\overline{\text{RESET}}$		$2^{17}/f_X$		s
		Release by interrupt request		**Note**		s

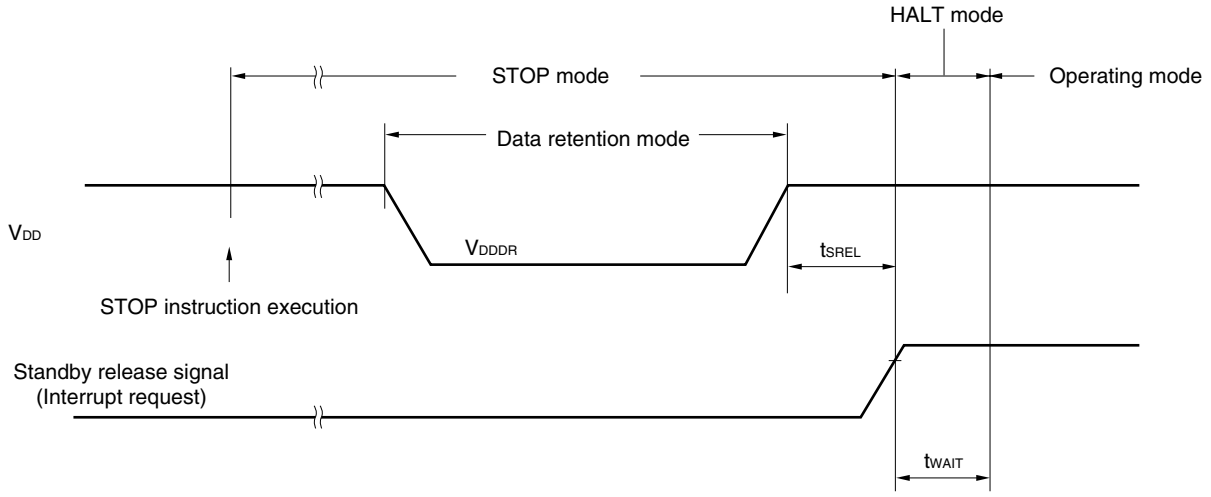
Note Selection of $2^{12}/f_X$, $2^{14}/f_X$, $2^{15}/f_X$, $2^{16}/f_X$, and $2^{17}/f_X$ is possible with bits 0 to 2 (OSTS0 to OSTS2) of the oscillation stabilization time select register (OSTS).

Remark fx: Main system clock oscillation frequency

Data retention timing (STOP mode release by $\overline{\text{RESET}}$)



Data retention timing (standby release signal: STOP mode release by interrupt request signal)



Flash memory programming characteristics (T_A = +10 to +40°C, V_{DD} = 1.8 to 5.5 V): μPD78F0338 only

(1) Write/erase characteristics

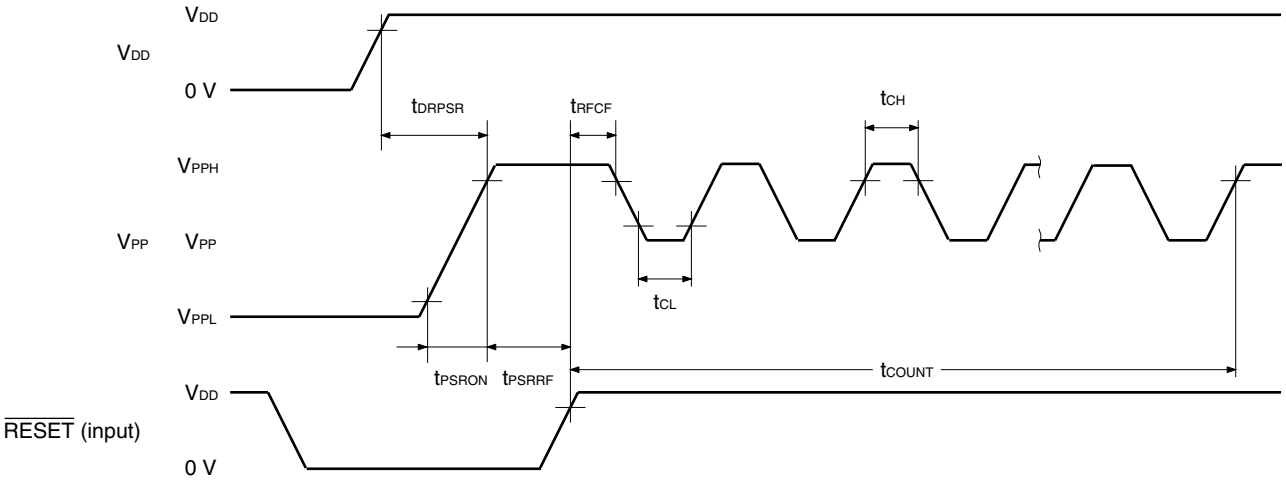
Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Operating frequency	f_X	4.5 V ≤ V_{DD} ≤ 5.5 V			1		10	MHz
		2.7 V ≤ V_{DD} < 4.5 V			1		5	
		1.8 V ≤ V_{DD} < 2.7 V			1		1.25	
V_{DD} write supply current[Note]	I_{DDW}	When V_{PP} = V_{PP1}	10 MHz crystal oscillation operating mode	V_{DD} = 4.5 to 5.5 V			35	mA
			5 MHz crystal oscillation operating mode	V_{DD} = 1.8 to 5.5 V			12	
V_{PP} write supply current[Note]	I_{PPW}	When V_{PP} = V_{PP1}	10 MHz crystal oscillation operating mode	V_{DD} = 4.5 to 5.5 V			39.5	mA
			5 MHz crystal oscillation operating mode	V_{DD} = 1.8 to 5.5 V			16.5	
V_{DD} erase supply current[Note]	I_{DDE}	When V_{PP} = V_{PP1}	10 MHz crystal oscillation operating mode	V_{DD} = 4.5 to 5.5 V			35	mA
			5 MHz crystal oscillation operating mode	V_{DD} = 1.8 to 5.5 V			12	
V_{PP} erase supply current[Note]	I_{PPE}	When V_{PP} = V_{PP1}					100	mA
Unit erase time	ter				0.5	1	1	s
Total erase time	tera						20	s
Number of rewriting times	C_{WRT}	Where erase and write make up 1 cycle					20	Times
V_{PP} supply voltage	V_{PP0}	Normal operation mode			0		0.2V_{DD}	V
	V_{PP1}	Flash memory program			9.7	10.0	10.3	V

Note Excluding port current (including current flowing through on-chip pull-up resistor)

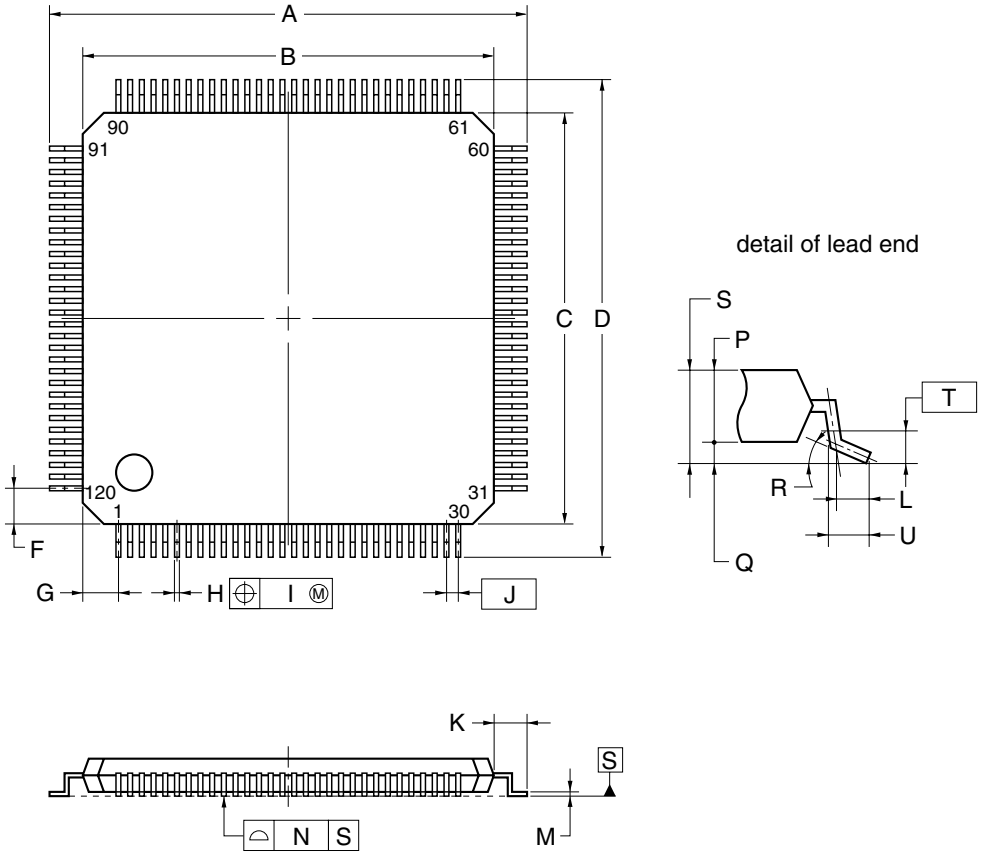
(2) Write operation characteristics

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
V_{PP} set time	t_{PSRON}	V_{PP} high voltage	1.0			μs
V_{PP}↑ set time from V_{DD}↑	t_{DRPSR}	V_{PP} high voltage	1.0			μs
$\overline{RESET}$↑ set time from V_{PP}↑	t_{PSRRF}	V_{PP} high voltage	1.0			μs
V_{PP} count start time from $\overline{RESET}$↑	t_{RFCF}		1.0			μs
Count execution time	t_{COUNT}				2.0	ms
V_{PP} counter high-/low-level width	t_{CH}, t_{CL}		8.0			μs
V_{PP} counter noise elimination width	t_{NFW}			40		ns

Flash write mode setting timing



120-PIN PLASTIC TQFP (FINE PITCH) (14x14)



NOTE

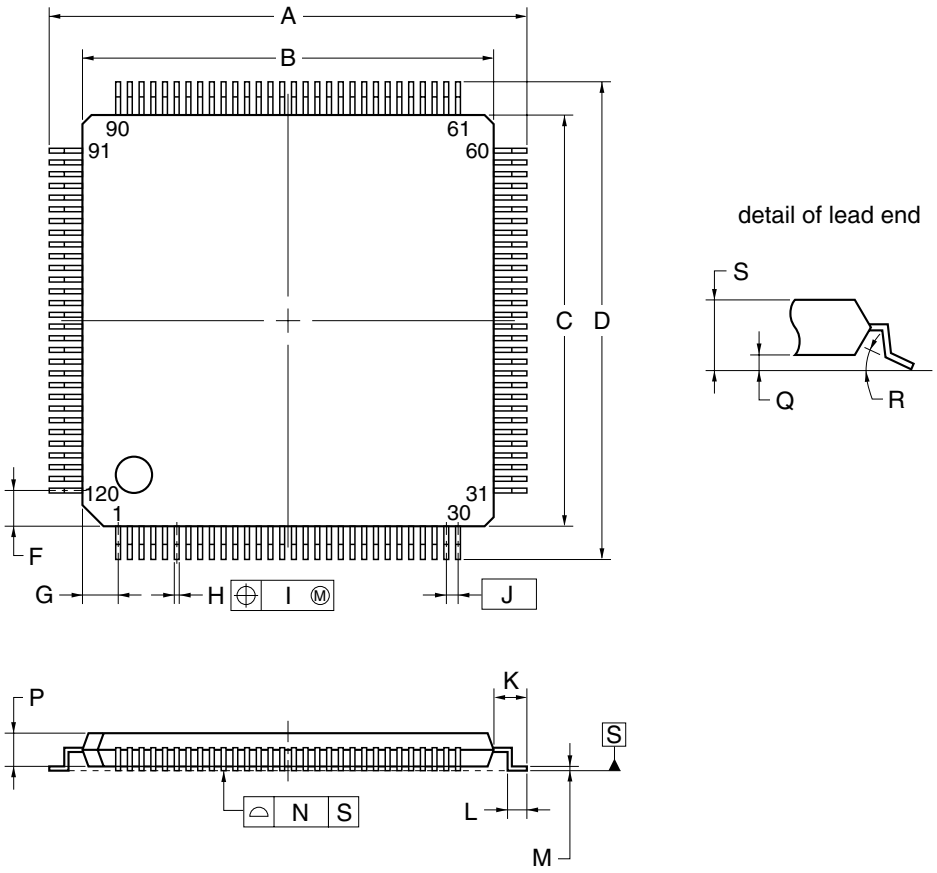
Each lead centerline is located within 0.07 mm of its true position (T.P.) at maximum material condition.-1

ITEM	MILLIMETERS
A	16.0±0.2
B	14.0±0.2
C	14.0±0.2
D	16.0±0.2
F	1.2
G	1.2
H	0.18±0.05
I	0.07
J	0.4 (T.P.)
K	1.0±0.2
L	0.5
M	$0.17^{+0.03}_{-0.07}$
N	0.08
P	1.0
Q	0.1±0.05
R	$3^{\circ}{}^{+4^{\circ}}_{-3^{\circ}}$
S	1.1±0.1
T	0.25

S120GC-40-9EB-1

Remark The dimensions and materials of the ES version are the same as those of the mass-produced version.

120-PIN PLASTIC TQFP (FINE PITCH) (14x14)



NOTE

Each lead centerline is located within 0.09 mm of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS
A	16.0±0.2
B	14.0±0.2
C	14.0±0.2
D	16.0±0.2
F	1.2
G	1.2
H	0.18±0.05
I	0.09
J	0.4 (T.P.)
K	1.0±0.2
L	0.5±0.2
M	0.145±0.05
N	0.08
P	1.0±0.1
Q	0.1±0.05
R	$3^{\circ}{}^{+7^{\circ}}_{-3^{\circ}}$
S	1.2 MAX.

S120GC-40-9EV-1

Remark The dimensions and materials of the ES version are the same as those of the mass-produced version.

CHAPTER 26 RECOMMENDED SOLDERING CONDITIONS

These products should be soldered and mounted under the following recommended conditions.

For details of the recommended soldering conditions, refer to the document **Semiconductor Device Mounting Technology Manual (C10535E)**.

For soldering methods and conditions other than those recommended below, please contact an NEC sales representative.

Table 26-1. Surface Mounting Type Soldering Conditions

μPD780316GC-×××-9EB: 120-pin plastic TQFP (fine pitch) (14 × 14)
μPD780316GC-×××-9EV: 120-pin plastic TQFP (fine pitch) (14 × 14)
μPD780318GC-×××-9EB: 120-pin plastic TQFP (fine pitch) (14 × 14)
μPD780318GC-×××-9EV: 120-pin plastic TQFP (fine pitch) (14 × 14)
μPD780326GC-×××-9EB: 120-pin plastic TQFP (fine pitch) (14 × 14)
μPD780326GC-×××-9EV: 120-pin plastic TQFP (fine pitch) (14 × 14)
μPD780328GC-×××-9EB: 120-pin plastic TQFP (fine pitch) (14 × 14)
μPD780328GC-×××-9EV: 120-pin plastic TQFP (fine pitch) (14 × 14)
μPD780336GC-×××-9EB: 120-pin plastic TQFP (fine pitch) (14 × 14)
μPD780336GC-×××-9EV: 120-pin plastic TQFP (fine pitch) (14 × 14)
μPD780338GC-×××-9EB: 120-pin plastic TQFP (fine pitch) (14 × 14)
μPD780338GC-×××-9EV: 120-pin plastic TQFP (fine pitch) (14 × 14)
μPD78F0338GC-9EB: 120-pin plastic TQFP (fine pitch) (14 × 14)
μPD78F0338GC-9EV: 120-pin plastic TQFP (fine pitch) (14 × 14)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 235°C, Time: 30 seconds max. (at 210°C or higher), Count: 2 times 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: 2 times or less, Exposure limit: 3 days[Note] (after that, prebake at 125°C for 10 hours)	VP15-103-2
Partial heating	Pin temperature: 300°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).

APPENDIX A DIFFERENCES BETWEEN μPD780308, 780318, 780328, AND 780338 SUBSERIES

Table A-1 shows the major differences between μPD780308, 780318, 780328, and 780338 Subseries.

Table A-1. Major Differences Between μPD780308, 780318, 780328, and 780338 Subseries (1/2)

Item \ Part Number		μPD780308 Subseries	μPD780318 Subseries	μPD780328 Subseries	μPD780338 Subseries
I^2C bus built-in model (Y Subseries)		Provided	Not provided		
PROM (flash memory) version		μPD78P0308	μPD78F0338		
Power supply voltage		V_{DD} = 2.0 to 5.5 V	V_{DD} = 1.8 to 5.5 V		
ROM		• μPD780306: 48 KB • μPD780308: 60 KB	• μPD780316, 780326, 780336: 48 KB • μPD780318, 780328, 780338: 60 KB		
Internal high-speed RAM		1,024 bytes			
Internal expansion RAM		1,024 bytes	1,536 bytes		
LCD display RAM		40 × 4 bits	40 × 8 bits		
Minimum instruction execution time		0.4 μs/0.8 μs/1.6 μs/3.2 μs/ 6.4 μs/12.8 μs (@f_X = 5.0 MHz)	0.2 μs/0.4 μs/0.8 μs/1.6 μs/3.2 μs (@f_X = 10.0 MHz)		
Number of I/O ports		57	70	62	54
A/D converter		8 bits × 8	10 bits × 8		
D/A converter		—	8 bits × 1		
LCD controller/driver		Bias: 1/2 and 1/3 can be selected	• Bias: 1/3 only • Internal booster circuit employed for LCD driver reference voltage generator (×3) • Blinking display possible (blinking period can be selected: 0.5 s or 1 s)		
	Segment signal output	40 max.	24 max.	32 max.	40 max.
	Common signal output	4 max. (for dynamic display only)	4 max. (for dynamic display) 1 max. (for static display)		
Serial interface	Subseries without Y	• 3-wire/2-wire/SBI: 1 • 3-wire/UART: 1 • 3-wire: 1	• 3-wire/UART: 1 • 3-wire: 1		
	Y Subseries	• 3-wire/2-wire/I^2C: 1 • 3-wire/UART: 1 • 3-wire: 1	—		
Timer		• 16-bit timer/event counter: 1 • 8-bit timer/event counter: 2 • Watch timer: 1 • Watchdog timer: 1	• 16-bit timer/event counter: 2 • 8-bit timer/event counter: 3 • Watch timer: 1 • Watchdog timer: 1		

Table A-1. Major Differences Between μPD780308, 780318, 780328, and 780338 Subseries (2/2)

Item \ Part Number	μPD780308 Subseries	μPD780318 Subseries	μPD780328 Subseries	μPD780338 Subseries
Timer output	3 (14-bit PWM output: 1)	5 (8-bit PWM output: 3)		
Clock output	19.5 kHz, 39.1 kHz, 78.1 kHz, 156 kHz, 313 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5.0 MHz (5.0 MHz with main system clock)	78.1 kHz, 156 kHz, 313 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5.0 MHz, 10 MHz (10 MHz with main system clock)		
Test input	Internal: 1, external: 1	—		
Package	• 100-pin plastic LQFP (fine pitch) (14 × 14) • 100-pin plastic QFP (14 × 20)	• 120-pin plastic TQFP (fine pitch) (14 × 14)		
Device file	DF780308	DF780338		
Emulation board	IE-780308-NS-EM1	IE-780338-NS-EM1		
Electrical specifications and recommended soldering conditions	Refer to the document of each product.			

APPENDIX B DEVELOPMENT TOOLS

The following development tools are available for the development of systems that employ the μPD780318, 780328, and 780338 Subseries.

Figure B-1 shows the development tool configuration.

- **Support for PC98-NX series**

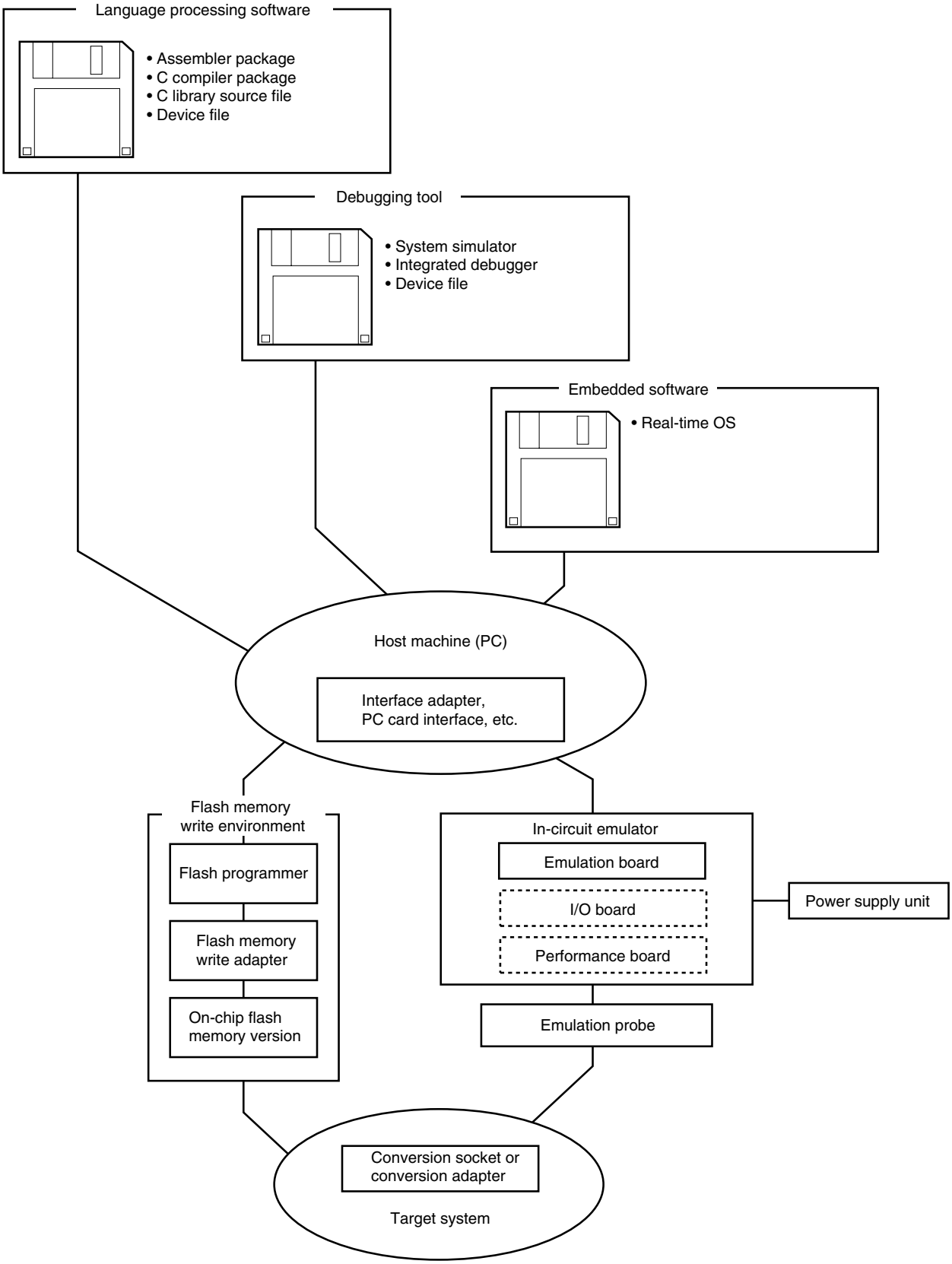
  Unless otherwise specified, products compatible with IBM PC/AT™ computers are compatible with PC98-NX series computers. When using PC98-NX series computers, refer to the explanation for IBM PC/AT computers.

- **Windows**

  Unless otherwise specified, "Windows" means the following OSs.

  - Windows 3.1
  - Windows 95, 98, 2000
  - Windows NT™ Ver. 4.0

Figure B-1. Development Tool Configuration



Remark Items in broken line boxes differ according to the development environment. See **B.3.1 Hardware**.

B.1 Language Processing Software

SP78K0 78K/0 Series Software Package	This is a software package that includes the development tools common to the 78K/0 Series.
	Part number: μS××××SP78K0
RA78K0 Assembler Package	This assembler converts programs written in mnemonics into object codes executable with a microcontroller. Further, this assembler is provided with functions capable of automatically creating symbol tables and branch instruction optimization. This assembler should be used in combination with an optional device file (DF780338). **<Precaution when using RA78K0 in PC environment>** This assembler package is a DOS-based application. It can also be used in Windows, however, by using the Project Manager (included in assembler package) on Windows.
	Part number: μS××××RA78K0
CC78K0 C Compiler Package	This compiler converts programs written in C language into object codes executable with a microcontroller. This compiler should be used in combination with an optional assembler package and device file. **<Precaution when using CC78K0 in PC environment>** This C compiler package is a DOS-based application. It can also be used in Windows, however, by using the Project Manager (included in assembler package) on Windows.
	Part number: μS××××CC78K0
DF780338[Note] Device File	This file contains information peculiar to the device. This device file should be used in combination with an optional tool (RA78K0, CC78K0, SM78K0, and ID78K0-NS). Corresponding OS and host machine differ depending on the tool to be used with.
	Part number: μS××××DF780338
CC78K0-L C Library Source File	This is a source file of functions configuring the object library included in the C compiler package. This file is required to match the object library included in C compiler package to the customer's specifications. Operating environment for the source file is not dependent on the OS.
	Part number: μS××××CC78K0-L

Note The DF780338 can be used in common with the RA78K0, CC78K0, SM78K0, and ID78K0-NS.

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××SP78K0

××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	

μS××××RA78K0
μS××××CC78K0

××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	3.5-inch 2HD FD
BB13		Windows (English version)	
AB17		Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	
3P17	HP9000 series 700™	HP-UX™ (Rel. 10.10)	
3K17	SPARCstation™	SunOS™ (Rel. 4.1.4), Solaris™ (Rel. 2.5.1)	

μS××××DF780338
μS××××CC78K0-L

××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	3.5-inch 2HD FD
BB13		Windows (English version)	
3P16	HP9000 series 700	HP-UX (Rel. 10.10)	DAT
3K13	SPARCstation	SunOS (Rel. 4.1.4), Solaris (Rel. 2.5.1)	3.5-inch 2HD FD
3K15			1/4-inch CGMT

B.2 Flash Memory Writing Tools

Flashpro III (part number: FL-PR3, PG-FP3) Flash Programmer	Flash programmer dedicated to microcontrollers with on-chip flash memory.
FA-120GC Flash Memory Writing Adapter	Flash memory writing adapter used connected to the Flashpro III. • FA-120GC: 120-pin plastic TQFP (GC-9EB, GC-9EV type)

Remark FL-PR3 and FA-120GC are products of Naito Densei Machida Mfg. Co., Ltd.
Phone: +81-45-475-4191 Naito Densei Machida Mfg. Co., Ltd.

B.3 Debugging Tools

B.3.1 Hardware

IE-78K0-NS In-Circuit Emulator	The in-circuit emulator serves to debug hardware and software when developing application systems using a 78K/0 Series product. It corresponds to integrated debugger (ID78K0-NS). This emulator should be used in combination with power supply unit, emulation probe, and interface adapter which is required to connect this emulator to the host machine.
IE-78K0-NS-PA Performance Board	This board is connected to the IE-78K0-NS to expand its functions. Adding this board adds a coverage function and enhances debugging functions such as tracer and timer functions.
IE-78K0-NS-A In-Circuit Emulator (with performance board)	This is a combination of the IE-78K0-NS and IE-78K0-NS-PA.
IE-70000-MC-PS-B Power Supply Unit	This adapter is used for supplying power from a receptacle of 100 V to 240 VAC.
IE-70000-98-IF-C Interface Adapter	This adapter is required when using the PC-9800 series computer (except notebook type) as the IE-78K0-NS host machine (C bus compatible).
IE-70000-CD-IF-A PC Card Interface	This is PC card and interface cable required when using notebook computer as the IE-78K0-NS host machine (PCMCIA socket compatible).
IE-70000-PC-IF-C Interface Adapter	This adapter is required when using the IBM PC/AT compatible computers as the IE-78K0-NS host machine (ISA bus compatible).
IE-70000-PCI-IF-A Interface Adapter	This adapter is required when using a computer with PCI bus as the IE-78K0-NS host machine.
IE-780338-NS-EM1 Emulation Board	This board emulates the operations of the peripheral hardware peculiar to a device. It should be used in combination with an in-circuit emulator.
★ SWEX-120SE-1 Emulation Probe	This probe is used to connect the in-circuit emulator to a target system and is designed for use with 120-pin plastic TQFP (GC-9EB, GC-9EV type).
NQPACK120SE/ YQPACK120SE/ YQ-QUIDE Conversion Socket	This conversion adapter connects the SWEX-120SE-1 to a target system board designed for a 120-pin plastic TQFP (GC-9EB, GC-9EV type).

Remark SWEX-120SE-1 and NQPACK120SE/YQPACK120SE/YQ-GUIDE are products of TOKYO ELETECH CORPORATION.

Inquiry: Daimaru Kogyo, Ltd. Phone: Tokyo +81-3-3820-7112 Electronics Dept.
Osaka +81-6-6244-6672 Electronics 2nd Dept.

B.3.2 Software (1/2)

SM78K0 System Simulator	This system simulator is used to perform debugging at C source level or assembler level while simulating the operation of the target system on a host machine. This simulator runs on Windows. Use of the SM78K0 allows the execution of application logical testing and performance testing on an independent basis from hardware development without having to use an in-circuit emulator, thereby providing higher development efficiency and software quality. The SM78K0 should be used in combination with an optional device file (DF780338).
	Part number: μS××××SM78K0

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××SM78K0

××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	3.5-inch 2HD FD
BB13		Windows (English version)	
AB17		Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	

B.3.2 Software (2/2)

ID78K0-NS Integrated Debugger (supporting in-circuit emulator IE-78K0-NS)	This debugger is a control program to debug 78K/0 Series microcontrollers. It adopts a graphical user interface, which is equivalent visually and operationally to Windows or OSF/Motif™. It also has an enhanced debugging function for C programs, and thus trace results can be displayed on screen in C level by using the windows integration function which links a trace result with its source program, disassembled display, and memory display. In addition, by incorporating function modules such as task debugger and system performance analyzer, the efficiency of debugging programs, which run on real-time OSs can be improved. It should be used in combination with the optional device file.
	Part number: μS××××ID78K0-NS

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××ID78K0-NS

××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	3.5-inch 2HD FD
BB13		Windows (English version)	
AB17		Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	

APPENDIX C EMBEDDED SOFTWARE

For efficient development and maintenance of the μPD780318, 780328, and 780338 Subseries, the following embedded products are available.

Real-Time OS

RX78K0 Real-Time OS	RX78K0 is a real-time OS conforming to the μITRON specifications. Tool (configurator) for generating nucleus of RX78K0 and plural information tables is supplied. Used in combination with an optional assembler package (RA78K0) and device file (DF780338). **<Precaution when using RX78K0 in PC environment>** The real-time OS is a DOS-based application. It should be used in the DOS Prompt when using in Windows.
	Part number: μS××××RX78013-ΔΔΔΔ

Caution When purchasing the RX78K0, fill in the purchase application form in advance and sign the user agreement.

Remark ×××× and ΔΔΔΔ in the part number differ depending on the host machine and OS used.

μS××××RX78013-ΔΔΔΔ

ΔΔΔΔ	Product Outline	Maximum Number for Use in Mass Production
001	Evaluation object	Do not use for mass-produced product.
100K	Mass-production object	0.1 million units
001M		1 million units
010M		10 million units
S01	Source program	Source program for mass-produced object

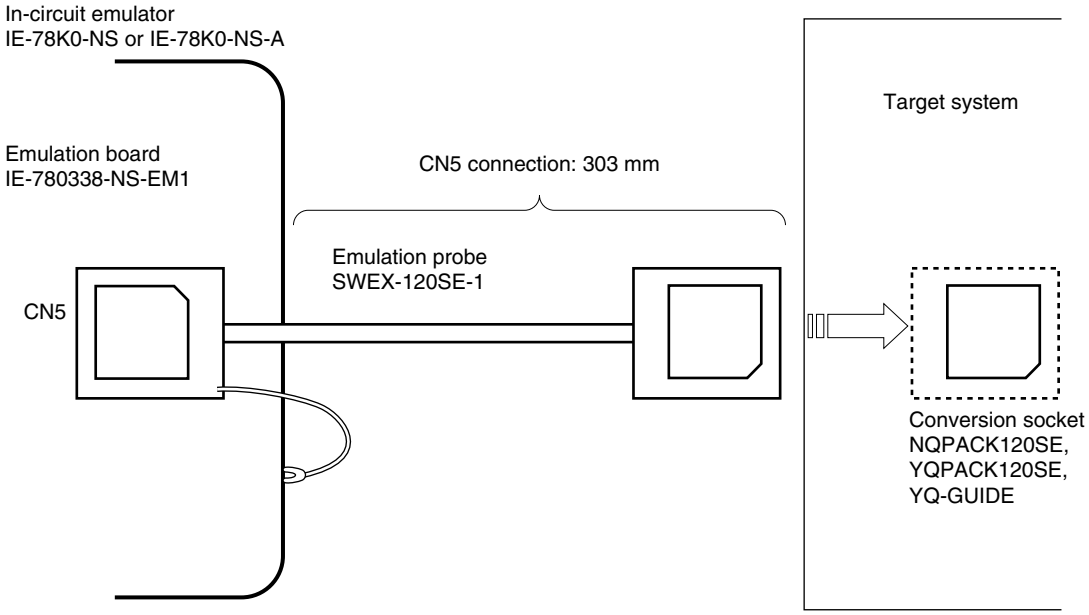
××××	Host Machine	OS	Supply Medium
AA13	PC-9800 series	Windows (Japanese version)[Note]	3.5-inch 2HD FD
AB13	IBM PC/AT compatibles	Windows (Japanese version)[Note]	3.5-inch 2HD FD
BB13		Windows (English version)[Note]	
3P16	HP9000 series 700	HP-UX (Rel. 10.10)	DAT
3K13	SPARCstation	SunOS (Rel. 4.1.4), Solaris (Rel. 2.5.1)	3.5-inch 2HD FD
3K15			1/4-inch CGMT

Note Can also be operated in DOS environment.

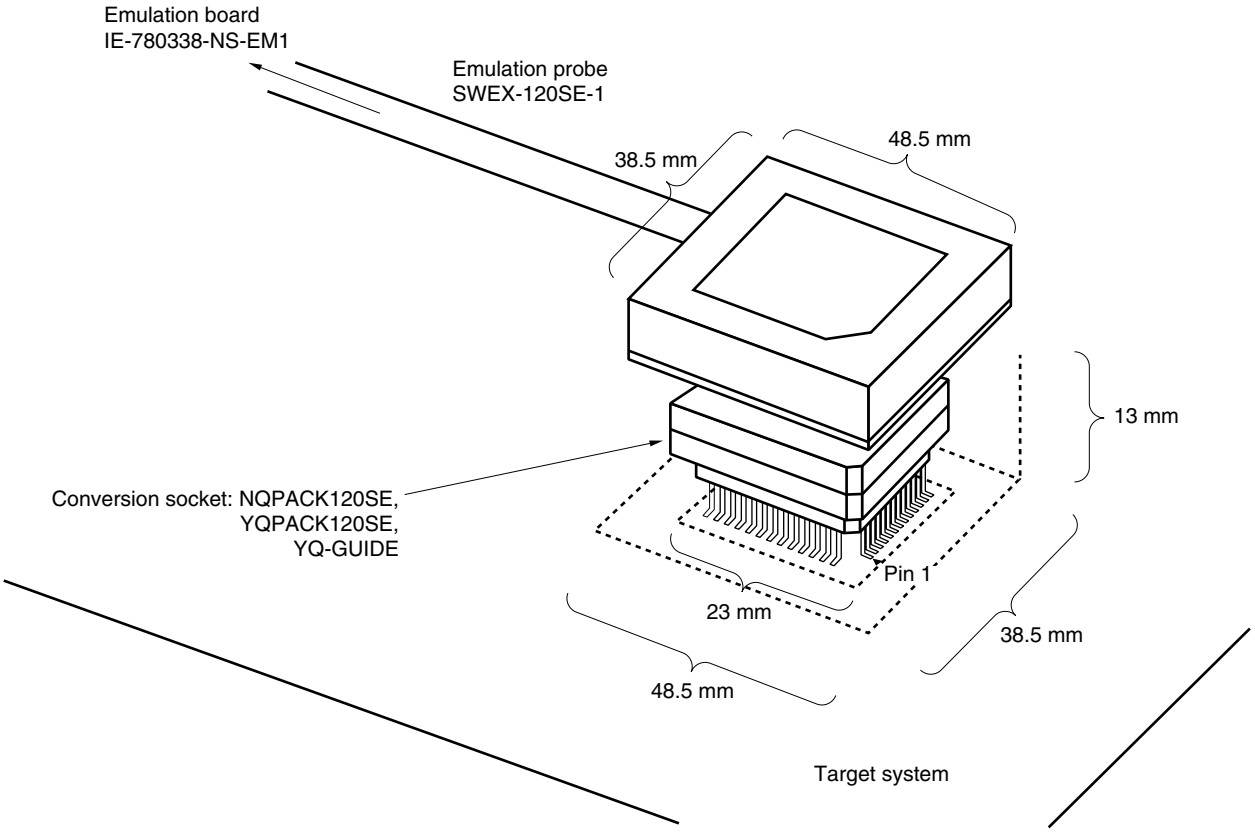
APPENDIX D NOTES ON DESIGNING TARGET SYSTEM

The following figure shows the conditions when connecting the emulation probe to the conversion socket. Design the system taking into consideration the shapes and other conditions of the components to be mounted on the target system, and be sure to follow the configuration below.

★ **Figure D-1. Distance from In-Circuit Emulator to Conversion Socket**



★ **Figure D-2. Connection Condition of Target System**



Remark SWEX-120SE-1, NQPACK120SE, YQPACK120SE, and YQ-GUIDE are products of TOKYO ELETECH CORPORATION.

APPENDIX E REGISTER INDEX

E.1 Register Name Index

[S]

[T]

[W]

E.2 Register Symbol Index

[M]

[O]

[P]

[S]

[T]

[W]

APPENDIX F REVISION HISTORY

The history of revisions up to this edition is shown below. “Applied to:” indicates the chapters to which the revision was applied.

(1/3)

Edition	Contents	Applied to:
2nd edition	Addition of packages μPD780316GC-×××-9EV, 780318GC-×××-9EV μPD780326GC-×××-9EV, 780328GC-×××-9EV μPD780336GC-×××-9EV, 780338GC-×××-9EV μPD78F0338GC-9EV	Throughout
	Change of block diagrams **Figure 4-2 P00 to P04 Block Diagram** **Figure 4-3 P05 Block Diagram** **Figure 4-5 P20, P22, P23, P25 Block Diagram** **Figure 4-8 P31, P32 Block Diagram** **Figure 4-9 P33, P34 Block Diagram** **Figure 4-11 Falling Edge Detector Block Diagram** **Figure 4-16 P71, P73 Block Diagram**	**CHAPTER 4 PORT FUNCTIONS**
	Addition of **Caution** to **Figure 4-24 Pin Function Switching Registers 8 and 9 (PF8, PF9) Format**	
	Addition of **Note 3** to **Figure 5-3 Processor Clock Control Register (PCC) Format**	**CHAPTER 5 CLOCK GENERATOR**
	Change of **Figure 6-13 Timing of Pulse Width Measurement Operation by Free-Running Counter and One Capture Register (with Both Edges Specified)**	**CHAPTER 6 16-BIT TIMER/EVENT COUNTER 0**
	Change of **Figure 6-15 Capture Operation of CR01 with Rising Edge Specified**	
	Change of **Figure 6-16 Timing of Pulse Width Measurement Operation with Free-Running Counter (with Both Edges Specified)**	
	Change of **Figure 6-18 Timing of Pulse Width Measurement Operation by Free-Running Counter and Two Capture Registers (with Rising Edge Specified)**	
	Change of **Figure 6-20 Timing of Pulse Width Measurement Operation by Means of Restart (with Rising Edge Specified)**	
	Change of following items in **6.6 16-Bit Timer/Event Counter 0 Cautions** **(2) 16-bit timer compare register setting (in the clear & start mode on match between TM0 and CR00)** **(3) Operation after compare register change during timer count operation** **(4) Capture register data retention timings** **(6) Operation of OVF0 flag** <1> **(11) Edge detection** <2>	
	Deletion of **Caution** in **Figure 8-7 8-Bit Timer Mode Control Register 5n (TMC5n) Format**	**CHAPTER 8 8-BIT TIMER/EVENT COUNTERS 50, 51, 52**
	Change of **Figure 12-2 A/D Converter Mode Register 0 (ADM0) Format**	**CHAPTER 12 A/D CONVERTER**

(2/3)

Edition	Contents	Applied to:
2nd edition	Deletion of infrared data transfer mode in **CHAPTER 14 SERIAL INTERFACE UART0**	**CHAPTER 14 SERIAL INTERFACE UART0**
	Change of description in **16.4.2 3-wire serial I/O mode (3) Communication operation**	**CHAPTER 16 SERIAL INTERFACE CSI1**
	Change of **Figure 16-4 Timing in 3-Wire Serial I/O Mode**	
	Change of **Figure 16-6 Output Operation of First Bit**	
	Change of **Figure 16-7 Output Value of SO1 Pin (Last Bit)**	
	Addition of **Figure 17-5 Relationship Between Reference Clock Generating Frame Frequency, and Frame Frequency**	**CHAPTER 17 LCD CONTROLLER/ DRIVER**
	Addition of **Caution** to **Figure 18-2 Interrupt Request Flag Registers (IF0L, IF0H, IF1L) Format**	**CHAPTER 18 INTERRUPT FUNCTIONS**
	Addition of **22.3 Flash Memory Characteristics**	**CHAPTER 22 μPD78F0338**
	Addition of **CHAPTER 24 ELECTRICAL SPECIFICATIONS**	**CHAPTER 24 ELECTRICAL SPECIFICATIONS**
	Addition of **CHAPTER 25 PACKAGE DRAWINGS**	**CHAPTER 25 PACKAGE DRAWINGS**
	Addition of **CHAPTER 26 RECOMMENDED SOLDERING CONDITIONS**	**CHAPTER 26 RECOMMENDED SOLDERING CONDITIONS**
	Change of **APPENDIX B DEVELOPMENT TOOLS**	**APPENDIX B DEVELOPMENT TOOLS**
	Change of **APPENDIX C EMBEDDED SOFTWARE**	**APPENDIX C EMBEDDED SOFTWARE**
	Addition of **APPENDIX D NOTES ON DESIGNING TARGET SYSTEM**	**APPENDIX D NOTES ON DESIGNING TARGET SYSTEM**
3rd edition	Change of Recommended Connection of Unused Pins for the following pins in **Table 2-1 Pin I/O Circuit Types** • P60 to P63 • P80/S32 to P87/S39 (for flash memory version) • P90/S24 to P97/S31 (for flash memory version)	**CHAPTER 2 PIN FUNCTIONS**
	Addition of description to **(1) Internal high-speed RAM** and **(2) Internal expansion RAM** in **3.1.2 Internal data memory space**	**CHAPTER 3 CPU ARCHITECTURE**
	Change of Manipulatable Bit Unit for ports 8 and 9 in **Table 3-4 Special Function Register List**	

(3/3)

Edition	Contents	Applied to:
3rd edition	Change of **Figure 4-18 P80 to P87 and P90 to P97 Block Diagram (Flash Memory Version)**	**CHAPTER 4 PORT FUNCTIONS**
	Modification of **Caution** in **4.2.11 Port 12**	
	Modification of clear conditions in **7.3 (1) 16-bit timer counter 4 (TM4)**	**CHAPTER 7 16-BIT TIMER/EVENT COUNTER 4**
	Modification of **Figure 7-1 16-Bit Timer/Event Counter 4 Block Diagram**	
	Modification of **Note** in **Table 17-4 Frame Frequency**	**CHAPTER 17 LCD CONTROLLER/ DRIVER**
	Modification of **Figure 17-6 Static/Dynamic Display Switching Register 3 (SDSEL3) Format**	
	Switch in order between **17.4 LCD Controller/Driver Settings** and **17.5 LCD Display RAM** of previous edition	
	Deletion of **Table 17-7 LCD Drive Voltages** of previous edition	
	Standardization of abbreviations • Output voltage of V_{LC0} pin: V_{LCD0} • Output voltage of V_{LC1} pin: V_{LCD1} • Output voltage of V_{LC2} pin: V_{LCD2}	
	Addition of description to **17.8.1 Static display example**	
	Modification of LCD panel connection example • **Figure 17-13 Static LCD Panel Connection Example (SDSEL3n = 1: n = 0, 1)** • **Figure 17-16 3-Time-Division LCD Panel Connection Example (SDSEL3n = 0: n = 0 to 2)** • **Figure 17-19 4-Time-Division LCD Panel Connection Example (SDSEL3n = 0, n = 0 to 2)**	
	Change of emulation probe name SWEX-120SE → SWEX-120SE-1	**APPENDIX B DEVELOPMENT TOOLS**
	Modification of **Figure D-1 Distance from In-Circuit Emulator to Conversion Socket**	**APPENDIX D NOTES ON DESIGNING TARGET SYSTEM**
	Modification of **Figure D-2 Connection Condition of Target System**	

NEC

Facsimile Message

From:

Name

Company

Tel. FAX

Address

Although NEC has taken all possible steps to ensure that the documentation supplied to our customers is complete, bug free and up-to-date, we readily accept that errors may occur. Despite all the care and precautions we've taken, you may encounter problems in the documentation. Please complete this form whenever you'd like to report errors or suggest improvements to us.

Thank you for your kind support.

North America
NEC Electronics Inc.
Corporate Communications Dept.
Fax: +1-800-729-9288
+1-408-588-6130

Europe
NEC Electronics (Europe) GmbH
Market Communication Dept.
Fax: +49-211-6503-274

South America
NEC do Brasil S.A.
Fax: +55-11-6462-6829

Hong Kong, Philippines, Oceania
NEC Electronics Hong Kong Ltd.
Fax: +852-2886-9022/9044

Korea
NEC Electronics Hong Kong Ltd.
Seoul Branch
Fax: +82-2-528-4411

P.R. China
NEC Electronics Shanghai, Ltd.
Fax: +86-21-6841-1137

Taiwan
NEC Electronics Taiwan Ltd.
Fax: +886-2-2719-5951

Asian Nations except Philippines
NEC Electronics Singapore Pte. Ltd.
Fax: +65-250-3583

Japan
NEC Semiconductor Technical Hotline
Fax: +81- 44-435-9608

I would like to report the following error/make the following suggestion:

Document title: _____

Document number: _____ Page number: _____

If possible, please fax the referenced page or drawing.

Document Rating	Excellent	Good	Acceptable	Poor
Clarity	❑	❑	❑	❑
Technical Accuracy	❑	❑	❑	❑
Organization	❑	❑	❑	❑