User's Manual

μPD789467 Subseries

8-Bit Single-Chip Microcontrollers

μPD789462
μPD789464
μPD789466
μPD789467
μPD78F9468

Document No. U15552EJ2V1UD00 (2nd edition)
Date Published January 2004 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 information in this document is current as of January, 2004. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC Electronics sales representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC Electronics does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.
- NEC Electronics products are classified into the following three quality grades: "Standard", "Special" and "Specific".

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

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

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

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

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

(Note)

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

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

M8E 02. 11-1

Regional Information

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

J03.4

Major Revisions in This Edition

Page	Description
Throughout	Deletion of development status indication “under development” for μPD789466 and 789467
p. 35	Addition of description on pin connections in **2.2.15 V_{PP} (μPD78F9468 only)**
p. 55	**Table 3-3 Special-Function Registers** • Change of attribute of port 8 (P8) to read-only and change of value after reset to undefined • Modification of oscillation stabilization time selection register (OSTS) to allow use of 1-bit manipulation instruction.
p. 74	Modification of **Caution 2** in **Figure 4-10 Format of Port Function Register 8**
p. 79	Addition of **Note** on feedback resistor in **Figure 5-3 Format of Suboscillation Mode Register**
p. 91	**6.2 Configuration of 8-Bit Timers 30 and 40** • Modification of **Figure 6-3 Block Diagram of Output Controller (Timer 40)** • Addition of description to **(2) 8-bit compare register 40 (CR40)** • Addition of description to **(3) 8-bit H width compare register 40 (CRH40)**
p. 96	Addition to **Cautions** in **Figure 6-6 Format of Carrier Generator Output Control Register 40**
p. 109	Addition to **Cautions** in **6.4.3 Operation as carrier generator**
p. 115	Modification of description in **6.5 Notes on Using 8-Bit Timers 30 and 40**
p. 137	Addition of **(8) Input impedance of ANI0 pin** to **9.5 Cautions Related to 8-Bit A/D Converter**
pp. 141, 142	Addition to **Notes** and **Cautions** in **Figure 10-2 Format of LCD Display Mode Register 0**
p. 159	Addition to **Cautions** in **Figure 12-2 Format of Interrupt Request Flag Register 0**
p. 161	Addition to **Cautions** in **Figure 12-6 Format of Key Return Mode Register 00**
p. 169	Modification of oscillation stabilization time selection register (OSTS) to allow use of 1-bit manipulation instruction in **13.1.2 Register controlling standby function**
p. 183	Modification of **Table 15-3 Communication Mode List**
pp. 200, 202, 204, 205, 210	**CHAPTER 18 ELECTRICAL SPECIFICATIONS** • Change from target values to formal specifications • Addition of **Note 1** to **Absolute Maximum Ratings** • Addition of recommended oscillator constants for mask ROM versions • Modification of **Write and Erase Characteristics**
p. 212	Addition of recommended conditions for μPD789466 and 789467 in **CHAPTER 20 RECOMMENDED SOLDERING CONDITIONS**
p. 217	Change of name “conversion connector” and “conversion socket” to “conversion adapter (TGB-052SBP)” in **A.5 Debugging Tools (Hardware)**
p. 224	Addition of **APPENDIX C REVISION HISTORY**

The mark ★ shows major revised points.

INTRODUCTION

Target Readers

This manual is intended for users who wish to understand the functions of the μPD789467 Subseries and to design and develop application systems and programs using these microcontrollers.

Target products:

- μPD789467 Subseries: μPD789462, 789464, 789466, 789467, 78F9468

Purpose

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

Organization

The μPD789467 Subseries User's Manual is divided into two parts: this manual and instructions (common to the 78K/0S Series).

μPD789467 Subseries User's Manual	78K/0S Series Instructions User's Manual
• Pin functions	• CPU function
• Internal block functions	• Instruction set
• Interrupt functions	• Explanation of each instruction
• Other on-chip peripheral functions	
• Electrical specifications	

How to Use 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 understand the functions in general:
  → Read this manual in the order of the contents.
- How to interpret the register format:
  → For a bit whose number is enclosed in brackets, the bit name is defined as a reserved word in the assembler, and already defined in the header file named sfrbit.h in the C compiler.
- When you know a register name and want to confirm its details:
  → Read **APPENDIX B REGISTER INDEX**.
- To know the 78K/0S Series instruction functions in detail:
  → Read **78K/0S Series Instructions User's Manual (U11047E)**.
- To know the electrical specifications of the μPD789467 Subseries:
  → Read **CHAPTER 18 ELECTRICAL SPECIFICATIONS**.

Conventions

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

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

Documents Related to Devices

Document Name	Document No.
μPD789467 Subseries User's Manual	This manual
78K/0S Series Instructions User's Manual	U11047E

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

Document Name		Document No.
RA78K0S Assembler Package	Operation	U14876E
	Language	U14877E
	Structured Assembly Language	U11623E
CC78K0S C Compiler	Operation	U14871E
	Language	U14872E
SM78K Series System Simulator Ver. 2.30 or Later	Operation (Windows™ Based)	U15373E
	External Part User Open Interface Specifications	U15802E
ID78K Series Integrated Debugger Ver. 2.30 or Later	Operation (Windows Based)	U15185E
Project Manager Ver. 3.12 or Later (Windows Based)		U14610E

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

Document Name	Document No.
IE-78K0S-NS In-Circuit Emulator	U13549E
IE-78K0S-NS-A In-Circuit Emulator	U15207E
IE-789468-NS-EM1 Emulation Board	To be prepared

Documents Related to Flash Memory Writing

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

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

Other Related Documents

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

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

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

CONTENTS

LIST OF FIGURES (1/4)

LIST OF FIGURES (2/4)

LIST OF FIGURES (3/4)

LIST OF FIGURES (4/4)

LIST OF TABLES (1/2)

LIST OF TABLES (2/2)

CHAPTER 1 GENERAL

1.1 Features

- ROM and RAM capacities

Item Part Number	Program Memory (ROM)		Data Memory	
			Internal High-Speed RAM	LCD Display RAM
μPD789462	Mask ROM	4 KB	256 bytes	23 × 4 bits
μPD789464		8 KB		
μPD789466		16 KB	512 bytes	
μPD789467		24 KB		
μPD78F9468	Flash memory	32 KB		

- Minimum instruction execution time can be changed from high-speed (0.4 μs: @ 5.0 MHz operation with main system clock) to ultra-low-speed (122 μs: @ 32.768 kHz operation with subsystem clock)
- I/O ports: 18
- 8-bit resolution A/D converter: 1 channel
- Timer: 4 channels
  - 8-bit timer: 2 channels
  - Watch timer: 1 channel
  - Watchdog timer: 1 channel
- LCD controller/driver (on-chip voltage booster)
  Segment signals: 23, common signals: 4
- Vectored interrupt sources: 9
- On-chip power-on clear circuit (mask option for mask ROM versions)
- Power supply voltage: V_{DD} = 1.8 to 5.5 V[Note]
- Operating ambient temperature: T_A = –40 to +85°C

Note For mask ROM versions when the use of the POC circuit is selected or for flash memory versions, the minimum value of the operation power supply voltage is the POC detection voltage (1.9 ±0.1 V).

1.2 Applications

Remote controllers, healthcare equipment, etc.

1.3 Ordering Information

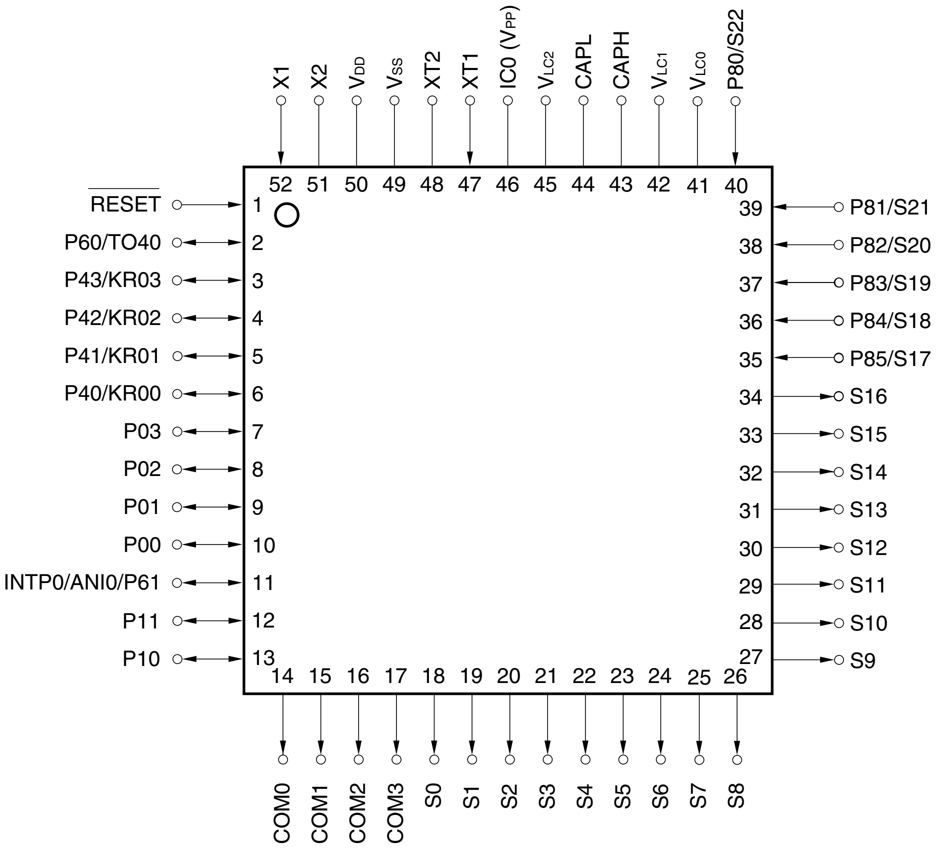
Part Number	Package	Internal ROM
μPD789462GB-×××-8ET	52-pin plastic LQFP (10 × 10)	Mask ROM
μPD789464GB-×××-8ET	52-pin plastic LQFP (10 × 10)	Mask ROM
μPD789466GB-×××-8ET	52-pin plastic LQFP (10 × 10)	Mask ROM
μPD789467GB-×××-8ET	52-pin plastic LQFP (10 × 10)	Mask ROM
μPD78F9468GB-8ET	52-pin plastic LQFP (10 × 10)	Flash memory

Remark ××× indicates ROM code suffix.

1.4 Pin Configuration (Top View)

52-pin plastic LQFP (10 × 10)

μPD789462GB-×××-8ET μPD789464GB-×××-8ET μPD789466GB-×××-8ET
μPD789467GB-×××-8ET μPD78F9468GB-8ET



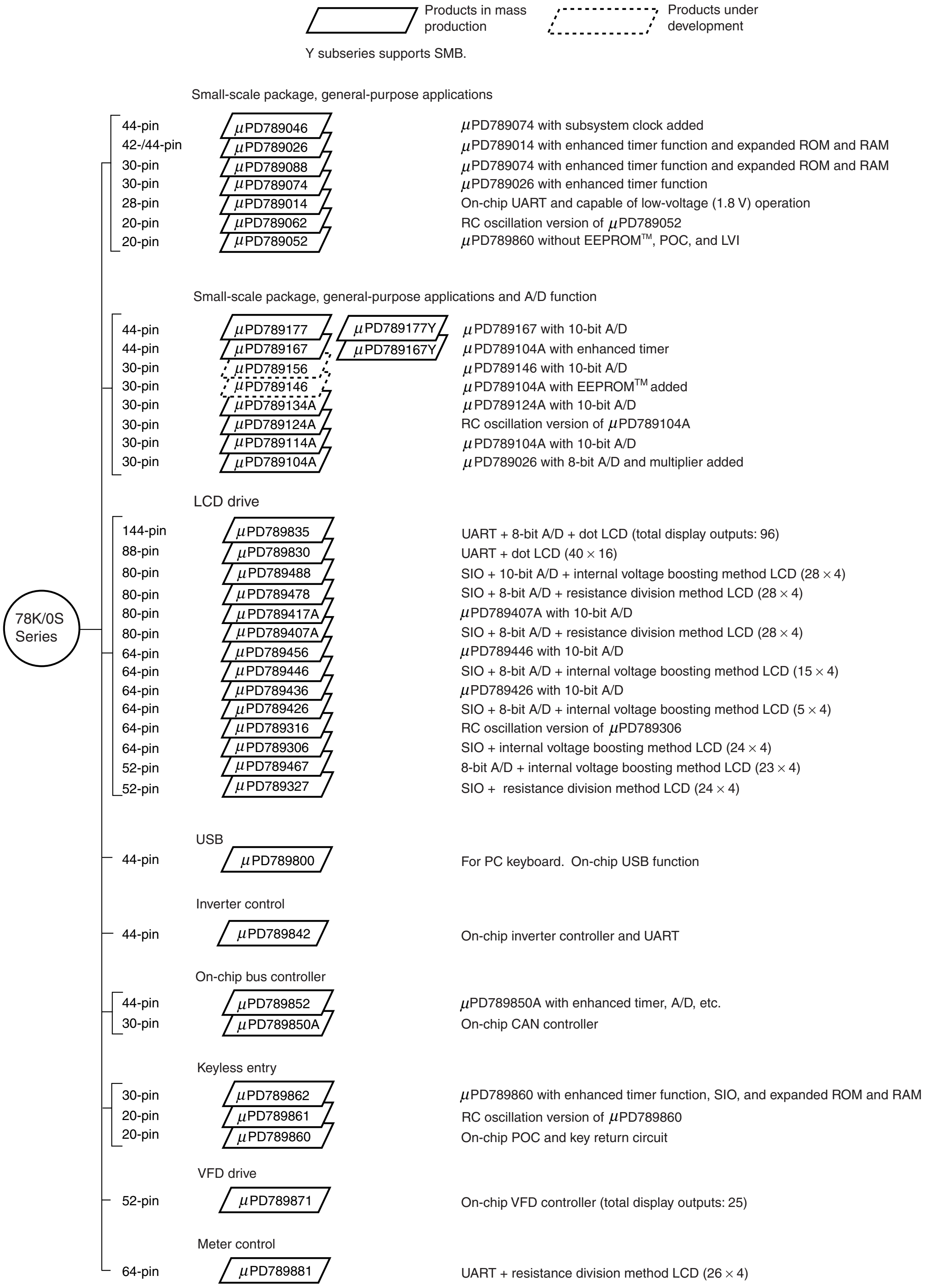
Caution Connect the IC0 (Internally Connected) pin directly to V_{SS}.

Remark The parenthesized values apply to the μPD78F9468.

ANI0:	Analog input
CAPH, CAPL:	LCD power supply capacitance control
COM0 to COM3:	Common output
IC0:	Internally connected
INTP0:	External interrupt input
KR00 to KR03:	Key return
P00 to P03:	Port 0
P10, P11:	Port 1
P40 to P43:	Port 4
P60, P61:	Port 6
P80 to P85:	Port 8
$\overline{\text{RESET}}$:	Reset
S0 to S22:	Segment output
TO40:	Timer output
V_{DD}:	Power supply
V_{LC0} to V_{LC2}:	Power supply for LCD
V_{PP}:	Programming power supply
V_{SS}:	Ground
X1, X2:	Crystal (main system clock)
XT1, XT2:	Crystal (subsystem clock)

★ 1.5 78K/0S Series Lineup

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



Products in mass production

Products under development

Y subseries supports SMB.

78K/0S Series

Small-scale package, general-purpose applications

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

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

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

LCD drive

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

USB

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

Inverter control

44-pin	µPD789842	On-chip inverter controller and UART

On-chip bus controller

44-pin	µPD789852	µPD789850A with enhanced timer, A/D, etc.
30-pin	µPD789850A	On-chip CAN controller

Keyless entry

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

VFD drive

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

Meter control

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

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

The major functional differences among the subseries are listed below.

Series for General-Purpose Applications and LCD Drive

Function		ROM Capacity	Timer				8-Bit A/D	10-Bit A/D	Serial Interface	I/O	V_{DD}	Remarks
Subseries			8-Bit	16-Bit	Watch	WDT					MIN.Value	
Small-scale package, general-purpose applications	μPD789046	16 KB	1 ch	1 ch	1 ch	1 ch	–	–	1 ch (UART: 1ch)	34	1.8 V	–
	μPD789026	4 KB to 16 KB			–							
	μPD789088	16 KB to 32 KB	3 ch							24		
	μPD789074	2 KB to 8 KB	1 ch									
	μPD789014	2 KB to 4 KB	2 ch	–						22		
	μPD789062	4 KB							–	14		RC-oscillation version
	μPD789052											–
Small-scale package, general-purpose applications + A/D converter	μPD789177	16 KB to 24 KB	3 ch	1 ch	1 ch	1ch	–	8 ch	1 ch (UART: 1ch)	31	1.8 V	–
	μPD789167						8 ch	–				
	μPD789156	8 KB to 16 KB	1 ch		–		–	4 ch		20		On-chip EEPROM
	μPD789146						4 ch	–				
	μPD789134A	2 KB to 8 KB					–	4 ch				RC-oscillation version
	μPD789124A						4 ch	–				
	μPD789114A						–	4 ch				–
	μPD789104A						4 ch	–				
LCD drive	μPD789835	24 KB to 60 KB	6 ch	–	1 ch	1 ch	3 ch	–	1 ch (UART: 1ch)	37	1.8 V[Note]	Dot LCD supported
	μPD789830	24 KB	1 ch	1 ch			–			30	2.7 V	
	μPD789488	32 KB to 48 KB	3 ch					8 ch	2 ch (UART: 1ch)	45	1.8 V	–
	μPD789478	24 KB to 48 KB					8 ch	–				
	μPD789417A	12 KB to 24 KB					–	7 ch	1 ch (UART: 1ch)	43		
	μPD789407A						7 ch	–				
	μPD789456	12 KB to 16 KB	2 ch				–	6 ch		30		
	μPD789446						6 ch	–				
	μPD789436						–	6 ch		40		
	μPD789426						6 ch	–				
	μPD789316	8 KB to 16 KB					–		2 ch (UART: 1ch)	23		RC-oscillation version
	μPD789306											–
	μPD789467	4 KB to 24 KB		–			1 ch		–	18		
	μPD789327						–		1 ch	21		

Note Flash memory version: 3.0 V

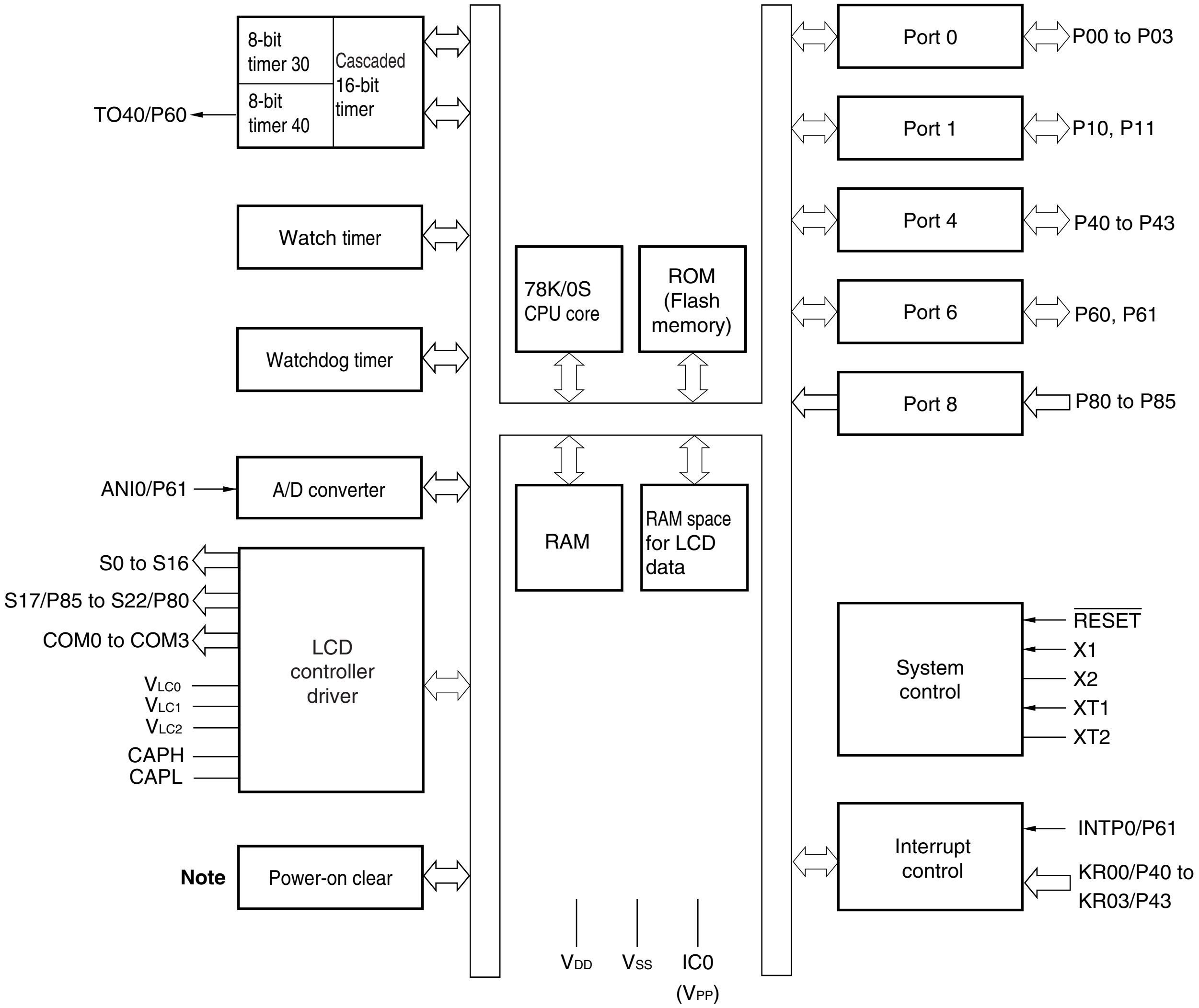
Series for ASSP

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

Notes 1. 10-bit timer: 1 channel

2. Flash memory version: 3.0 V

1.6 Block Diagram



★ **Note** Only when use of the POC circuit is selected by a mask option in the case of the mask ROM version (μPD789462, 789464, 789466, and 789467).

Remarks 1. The internal ROM and RAM capacities vary depending on the product.

2. The parenthesized values apply to the μPD78F9468.

1.7 Overview of Functions

Part Number Item		μPD789462	μPD789464	μPD789466	μPD789467	μPD78F9468
Internal memory	ROM	Mask ROM				Flash memory
		4 KB	8 KB	16 KB	24 KB	32 KB
	High-speed RAM	256 bytes		512 bytes		
	LCD display RAM	23 × 4 bits				
Main system clock (oscillation frequency)		Ceramic/crystal oscillation (1.0 to 5.0 MHz)				
Subsystem clock (oscillation frequency)		Crystal oscillation (32.768 kHz)				
Minimum instruction execution time		0.4 μs/1.6 μs (@ 5.0 MHz operation with main system clock)				
		122 μs (@ 32.768 kHz operation with subsystem clock)				
General-purpose registers		8 bits × 8 registers				
Instruction set		• 16-bit operations • Bit manipulations (such as set, reset, and test)				
I/O ports		Total: 18[Note 1] CMOS I/O: 12 CMOS input: 6[Note 1]				
Timers		• 8-bit timer: 2 channels • Watch timer: 1 channel • Watchdog timer: 1 channel				
Timer outputs		1				
A/D converter		8-bit resolution × 1 channel				
LCD controller/driver		• Segment signal outputs: 23[Note 1] • Common signal outputs: 4				
Vectored interrupt sources	Maskable	Internal: 6, external: 2				
	Non-maskable	Internal: 1				
Reset		• Reset by $\overline{\text{RESET}}$ input • Internal reset by watchdog timer • Reset by power-on-clear circuit[Note 2]				
Power supply voltage		V_{DD} = 1.8 to 5.5 V[Note 3]				
Operating ambient temperature		T_A = –40 to +85°C				
Package		52-pin plastic LQFP (10 × 10)				

Notes 1. Six of these pins are used to select either the port function or LCD segment output via the port function register.

2. For mask ROM versions (μPD789462, 789464, 789466, 789467), this is available only when use of POC circuit is selected by a mask option.

3. For mask ROM versions when use of the POC circuit is selected or for flash memory versions, the minimum value of the operating power supply voltage is the POC detection voltage (1.9 ±0.1 V).

An outline of the timers is shown below.

		8-Bit Timer 30	8-Bit Timer 40	Watch Timer	Watchdog Timer
Operation mode	Interval timer	1 channel	1 channel	1 channel[Note 1]	1 channel[Note 2]
	External event counter	–	–	–	–
Function	Timer output	–	1 output	–	–
	Square-wave output	–	1 output	–	–
	Capture	–	–	–	–
	Interrupt sources	1	1	2	2

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

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

CHAPTER 2 PIN FUNCTIONS

2.1 List of Pin Functions

(1) Port pins

Pin Name	I/O	Function	After Reset	Alternate Function
P00 to P03	I/O	Port 0. 4-bit I/O port. Input/output can be specified in 1-bit units. When used as an input port, the use of on-chip pull-up resistors can be specified in port units using pull-up resistor option register 0 (PU0).	Input	–
P10, P11	I/O	Port 1. 2-bit I/O port. Input/output can be specified in 1-bit units. When used as an input port, the use of on-chip pull-up resistors can be specified in port units using pull-up resistor option register 0 (PU0).	Input	–
P40 to P43	I/O	Port 4. 4-bit I/O port. Input/output can be specified in 1-bit units. When used as an input port, the use of on-chip pull-up resistors can be specified in port units using pull-up resistor option register 0 (PU0), or key return mode register 00 (KRM00).	Input	KR00 to KR03
P60	I/O	Port 6. 2-bit I/O port. Input/output can be specified in 1-bit units.	Input	TO40
P61				INTP0/ANI0
P80 to P85	Input	Port 8. 6-bit input port.	Input	S22 to S17

(2) Non-port pins

Pin Name	I/O	Function	After Reset	Alternate Function
INTP0	Input	External interrupt input for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.	Input	P61/ANI0
KR00 to KR03	Input	Key return signal detection	Input	P40 to P43
TO40	Output	8-bit timer 40 output	Input	P60
ANI0	Input	A/D converter analog input	Input	P61/INTP0
S0 to S16	Output	LCD controller/driver segment signal outputs	Low-level output	–
S17 to S22			Input	P85 to P80
COM0 to COM2	Output	LCD controller/driver common signal outputs	Low-level output	–
COM3	Output	Mask ROM version	Low-level output	–
		Flash memory version	High-level output	
CAPH, CAPL	–	Voltage boost capacitor for LCD drive connection pins	–	–
V_{LC0} to V_{LC2}	–	LCD drive voltage	–	–
X1	Input	Connecting crystal resonator for main system clock oscillation	–	–
X2	–		–	–
XT1	Input	Connecting crystal resonator for subsystem clock oscillation	–	–
XT2	–		–	–
$\overline{\text{RESET}}$	Input	System reset input	Input	–
V_{DD}	–	Positive power supply	–	–
V_{SS}	–	Ground potential	–	–
IC0	–	Internally connected. Connect directly to V_{SS}.	–	–
V_{PP}	–	Sets flash memory programming mode. Applies high voltage when a program is written or verified.	–	–

2.2 Description of Pin Functions

2.2.1 P00 to P03 (Port 0)

These pins constitute a 4-bit I/O port and can be set to the input or output port mode in 1-bit units using port mode register 0 (PM0). When used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 0 (PU0) in port units.

2.2.2 P10, P11 (Port 1)

These pins constitute a 2-bit I/O port and can be set to the input or output port mode in 1-bit units using port mode register 1 (PM1). When used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 0 (PU0) in port units.

2.2.3 P40 to P43 (Port 4)

These pins constitute a 4-bit I/O port. In addition, they also function as key return signal detection pins.

P40 to P43 can be specified in the following operation modes in 1-bit units.

(1) Port mode

In this mode, P40 to P43 function as a 4-bit I/O port. These pins can be set to the input or output port mode in 1-bit units using port mode register 4 (PM4). When used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 0 (PU0) or key return mode register 00 (KRM00) in port units.

(2) Control mode

In this mode, the pins function as key return signal detection pins (KR00 to KR03).

2.2.4 P60, P61 (Port 6)

These pins constitute a 2-bit I/O port. In addition, they also function as a timer output, external interrupt input, and analog input.

P60 and P61 can be specified in the following operation modes in 1-bit units.

(1) Port mode

In this mode, P60 and P61 function as a 2-bit I/O port. These pins can be set to the input or output port mode in 1-bit units using port mode register 6 (PM6).

(2) Control mode

In this mode, the pins function as a timer output, external interrupt input, and analog input.

(a) TO40

This is the timer output pin to timer 40.

(b) INTP0

This is the external interrupt input pin for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

(c) ANI0

This is the analog input pin of the A/D converter.

2.2.5 P80 to P85 (Port 8)

These pins constitute a 6-bit input port. In addition, they also function as LCD controller/driver segment signal outputs.

P80 to P85 can be specified in the following operation modes in 1-bit units by using port function register 8 (PF8).

(1) Port mode

In this mode, P80 to P85 function as a 6-bit input port.

(2) Control mode

In this mode, the pins function as LCD controller/driver segment signal outputs (S17 to S22).

2.2.6 S0 to S16

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

2.2.7 COM0 to COM3

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

2.2.8 V_{LC0} to V_{LC2}

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

2.2.9 CATH, CAPL

These pins are used to connect a voltage booster capacitor for driving the LCD.

2.2.10 $\overline{\text{RESET}}$

This pin inputs an active-low system reset signal.

2.2.11 X1, X2

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

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

2.2.12 XT1, XT2

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

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

2.2.13 V_{DD}

This is the positive power supply pin.

2.2.14 V_{SS}

This is the ground pin.

2.2.15 V_{PP} (μPD78F9468 only)

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

Perform either of the following.

- Independently connect a 10 kΩ pull-down resistor to V_{PP}.
- Use the jumper on the board to connect V_{PP} to the dedicated flash programmer in programming mode or to V_{SS} in normal operation mode.

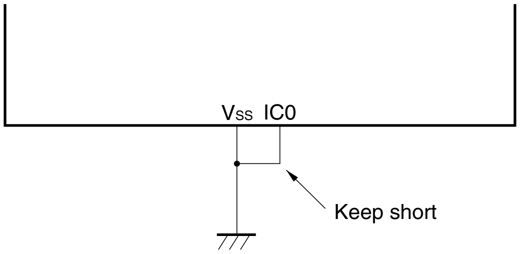
★ If the wiring between the V_{PP} and V_{SS} pins is long or external noise is superimposed on the V_{PP} pin, the user program may not run correctly.

2.2.16 IC0 (mask ROM version only)

The IC0 (Internally Connected) pin is used to set the μPD789462, 789464, 789466, and 789467 in the test mode before shipment. In the normal operation mode, directly connect this pin to the V_{SS} pin with as short a wiring length as possible.

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

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



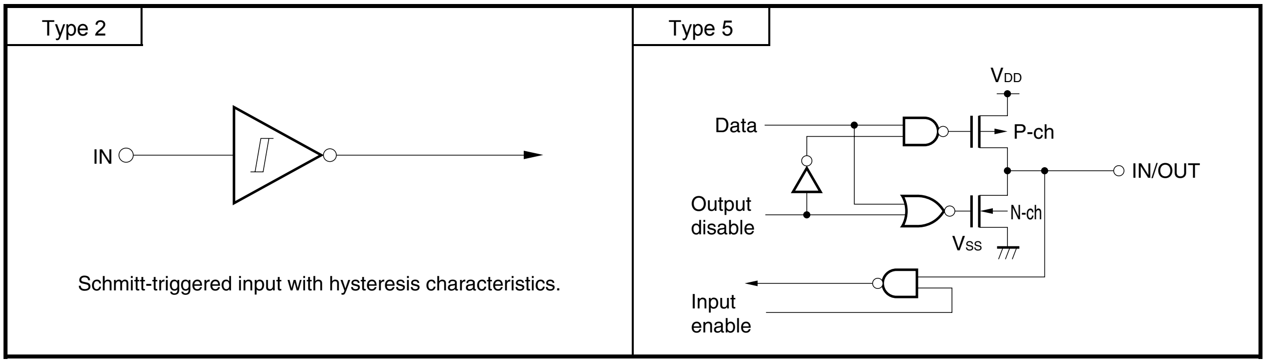
2.3 Pin I/O Circuits and Recommended Connection of Unused Pins

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

Table 2-1. Types of Pin I/O Circuits and Recommended Connection of Unused Pins

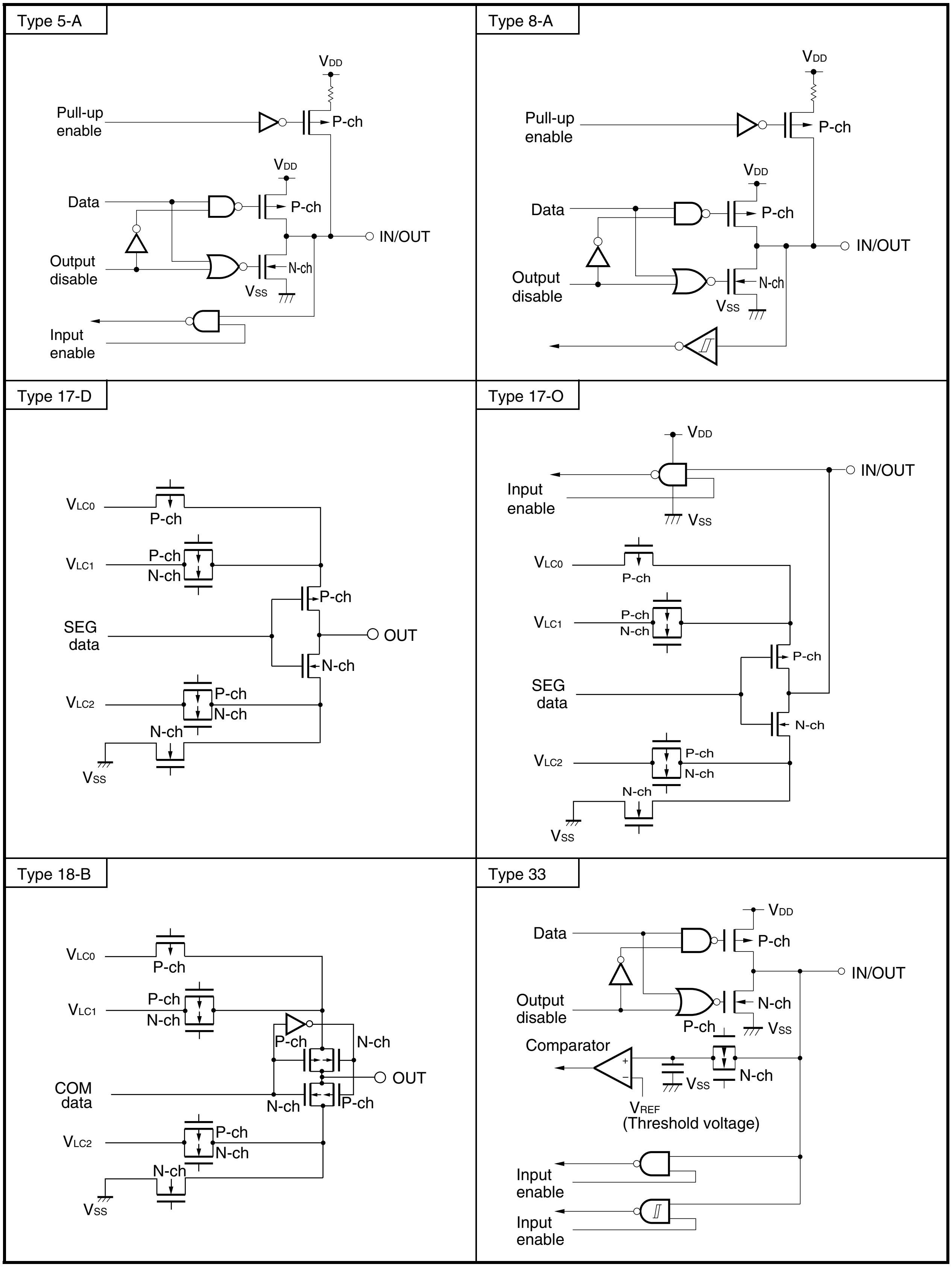
Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00 to P03	5-A	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P10, P11			
P40/KR00 to P43/KR03	8-A		
P60/TO40	5		
P61/INTP0/ANI0	33		Input: Independently connect to V_{SS} via a resistor. Output: Leave open.
P80/S22 to P85/S17	17-O		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
S0 to S16	17-D	Output	Leave open.
COM0 to COM3	18-B		
V_{LC0} to V_{LC2}	–	–	
CAPH, CAPL			
XT1		Input	Connect to V_{SS}.
XT2		–	Leave open.
$\overline{\text{RESET}}$	2	Input	–
IC0 (mask ROM version)	–	–	Connect directly to V_{SS}.
V_{PP} (μPD78F9468)			Independently connect V_{PP} to a 10 kΩ pull-down resistor or directly connect to V_{SS}.

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



Schmitt-triggered input with hysteresis characteristics.

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



3.1 Memory Space

The μPD78467 Subseries can access 64 KB of memory space. Figures 3-1 to 3-5 show the memory maps.

Figure 3-1. Memory Map (μPD789462)

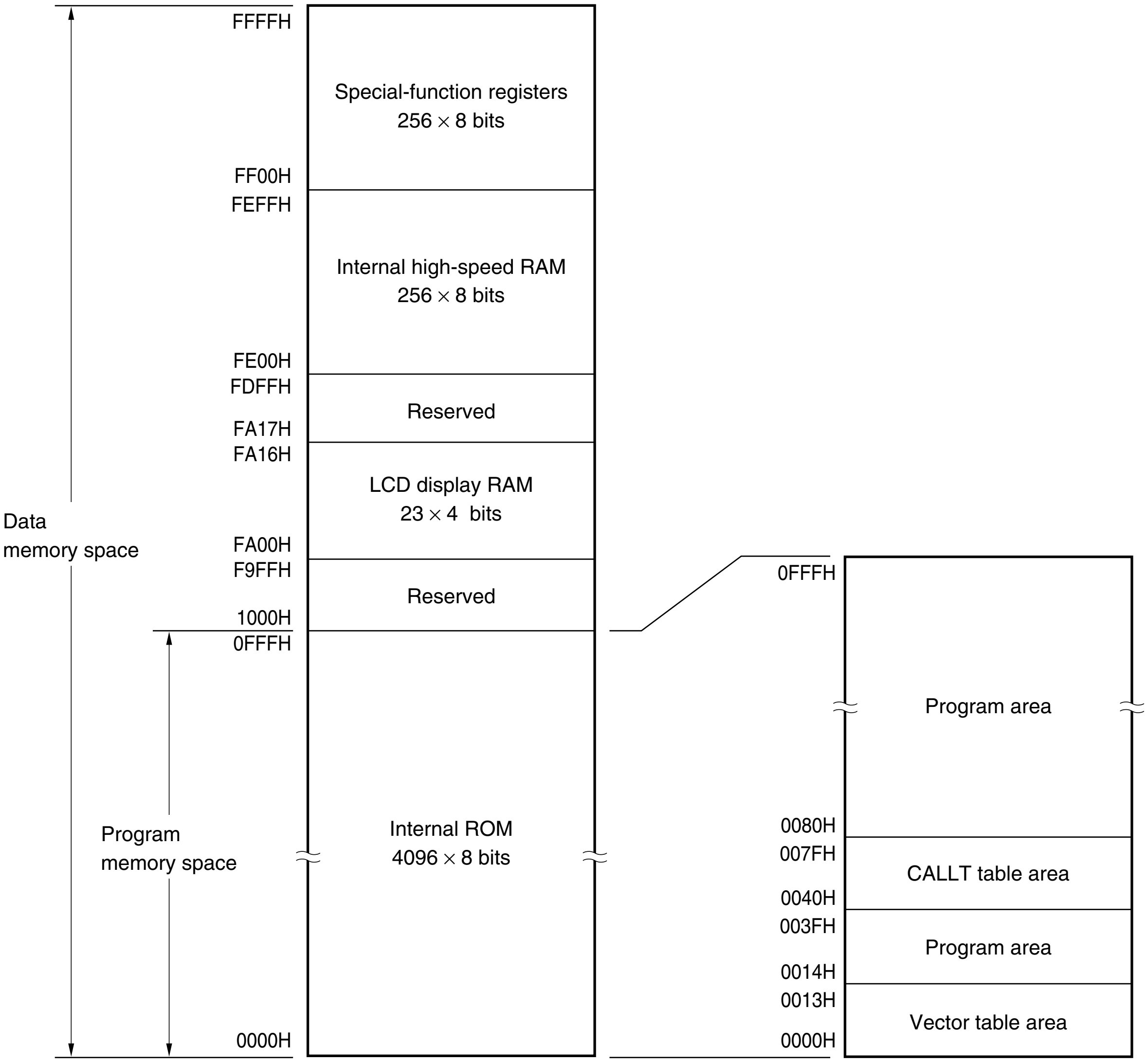


Figure 3-2. Memory Map (μPD789464)

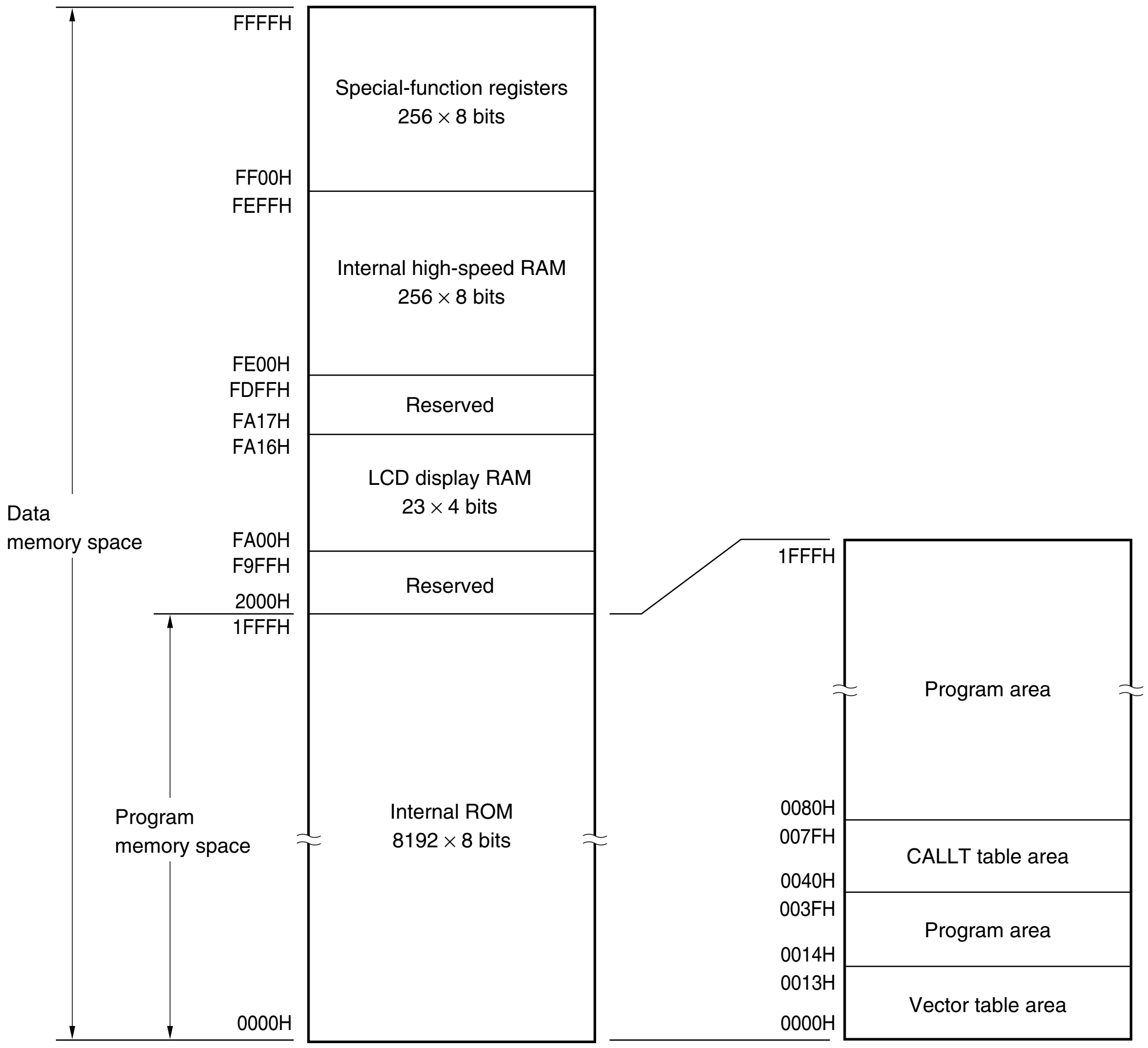


Figure 3-3. Memory Map (μPD789466)

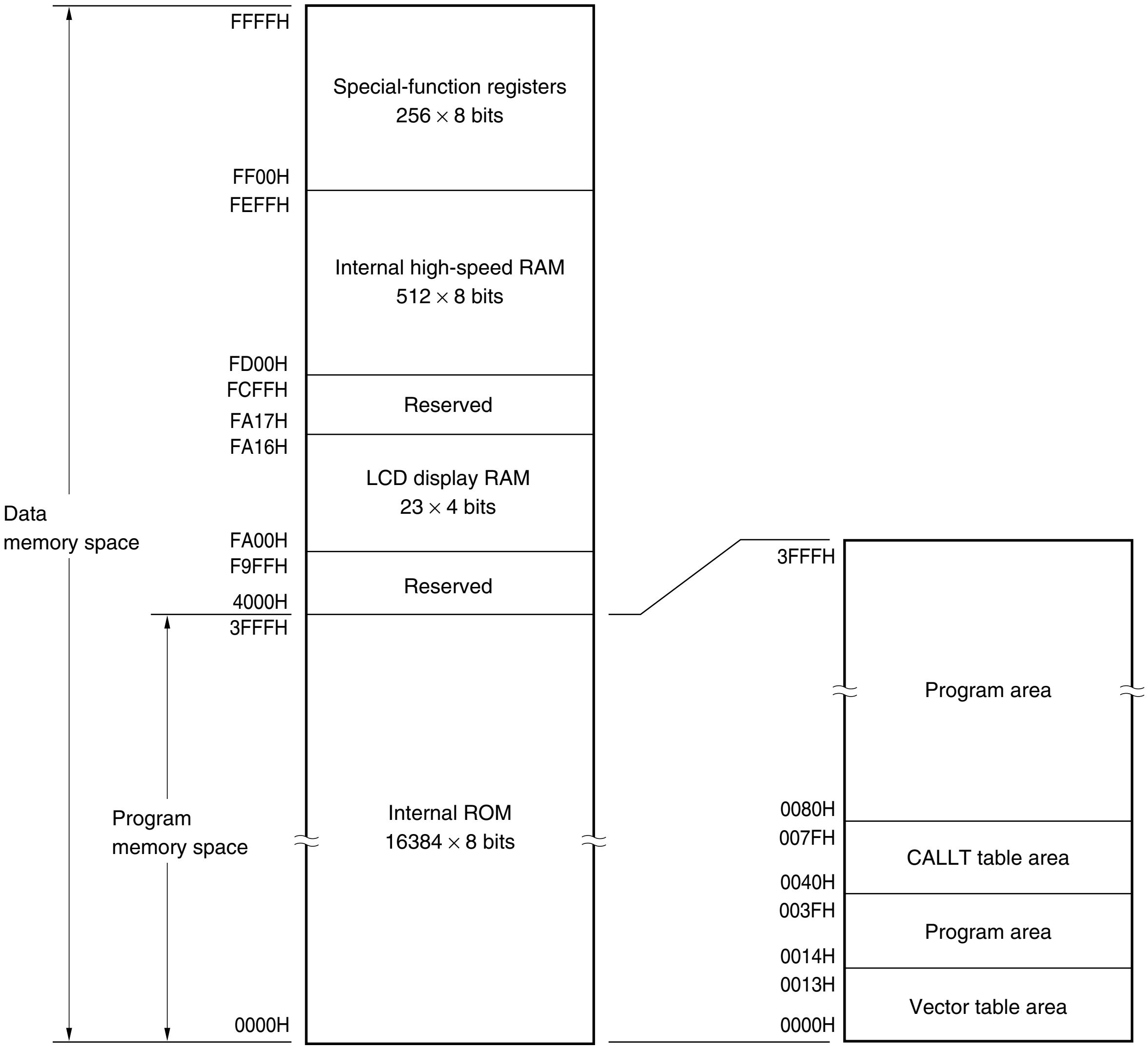


Figure 3-4. Memory Map (μPD789467)

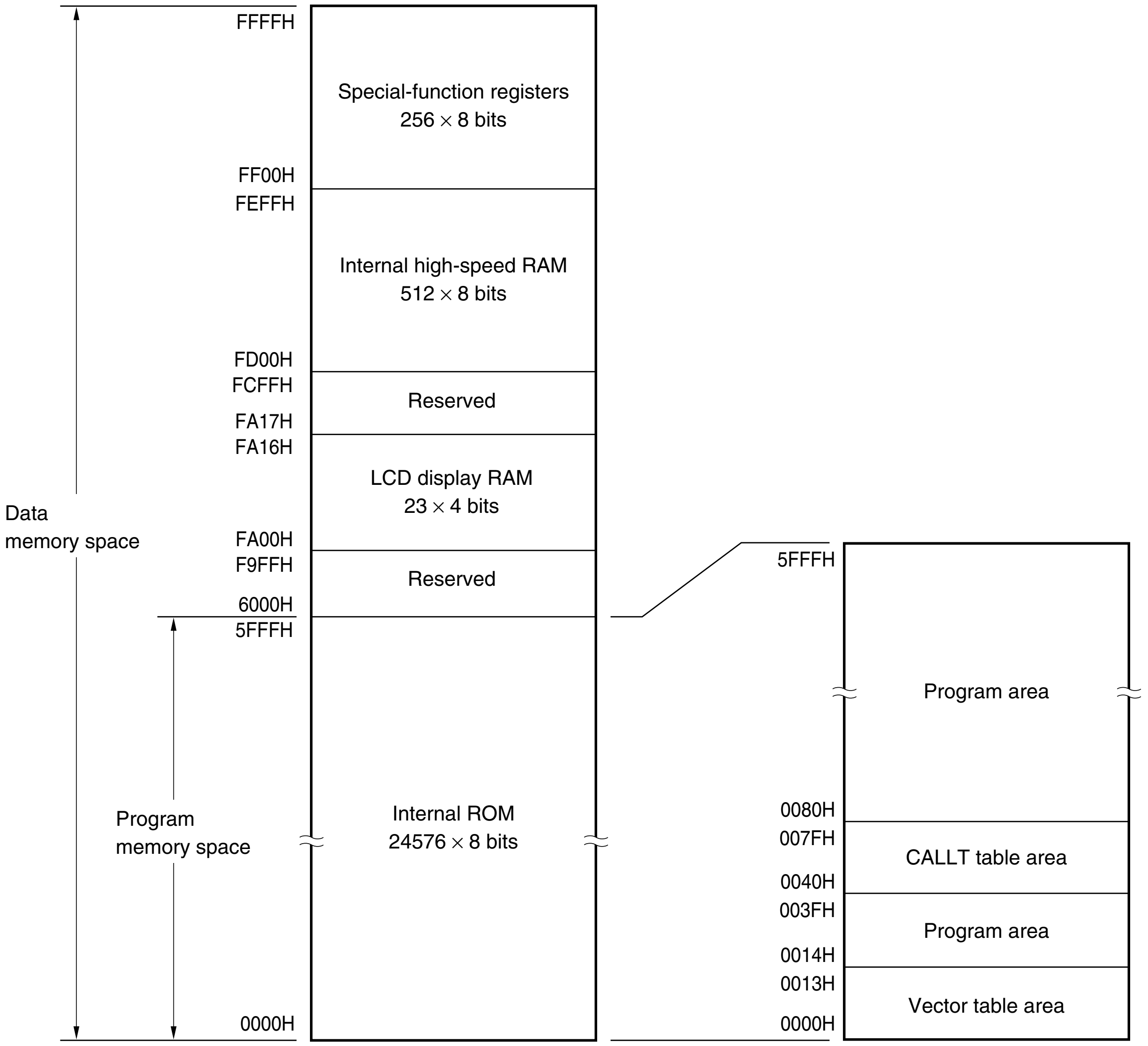
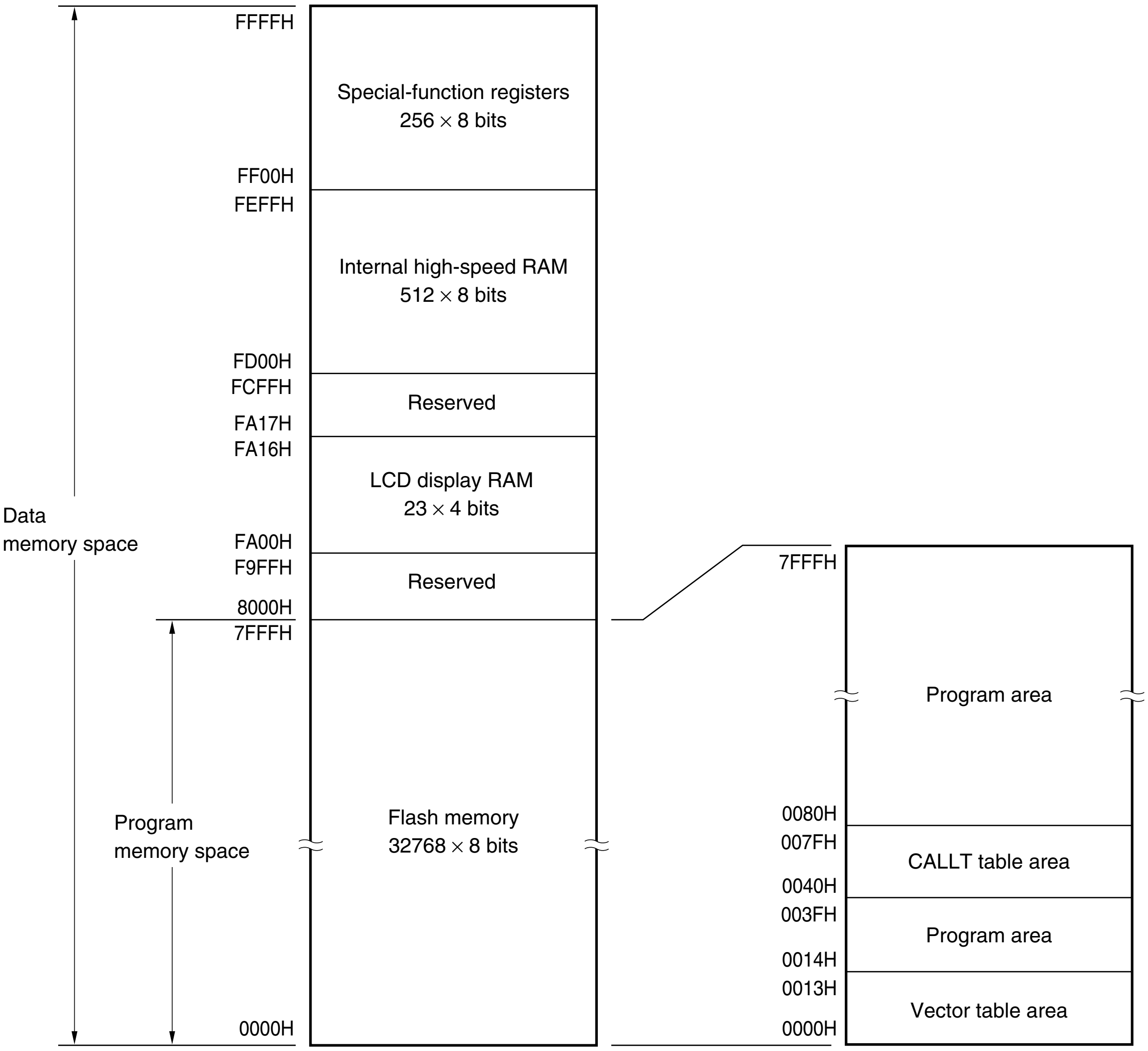


Figure 3-5. Memory Map (μPD78F9468)



3.1.1 Internal program memory space

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

The μPD789467 Subseries provides internal ROM (or flash memory) with the following capacity for each product.

Table 3-1. Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
μPD789462	Mask ROM	4096 × 8 bits
μPD789464		8192 × 8 bits
μPD789466		16384 × 8 bits
μPD789467		24576 × 8 bits
μPD78F9468	Flash memory	32768 × 8 bits

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

(1) Vector table area

The 20-byte area of addresses 0000H to 0013H is reserved as a vector table area. This area stores program start addresses to be used when branching by $\overline{\text{RESET}}$ input or interrupt request generation. Of a 16-bit program address, the lower 8 bits are stored in an even address, and the higher 8 bits are stored in an odd address.

Table 3-2. Vector Table

Vector Table Address	Interrupt Request	Vector Table Address	Interrupt Request
0000H	$\overline{\text{RESET}}$ input	000CH	INTTM30
0004H	INTWDT	000EH	INTTM40
0006H	INTP0	0010H	INTKR00
0008H	INTAD	0012H	INTWTI
000AH	INTWT		

(2) CALLT instruction table area

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

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

μPD789467 Subseries products incorporate the following RAM.

(1) Internal high-speed RAM

Internal high-speed RAM is incorporated in the area between FE00H and FEFFH in the μPD789462 and 789464, and in the area between FD00H and FEFFH in the μPD789466, 789467, and 78F9468.

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

(2) LCD display RAM

LCD display RAM is allocated in the area between FA00H and FA16H. The LCD display RAM can also be used as ordinary RAM.

3.1.3 Special-function register (SFR) area

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

3.1.4 Data memory addressing

The μPD789467 Subseries is provided with a variety of addressing modes to make memory manipulation as efficient as possible. At the addresses corresponding to data memory area especially, specific addressing modes that correspond to the particular function an area, such as the special-function registers, are available. Figures 3-6 to 3-10 show the data memory addressing modes.

Figure 3-6. Data Memory Addressing (μPD789462)

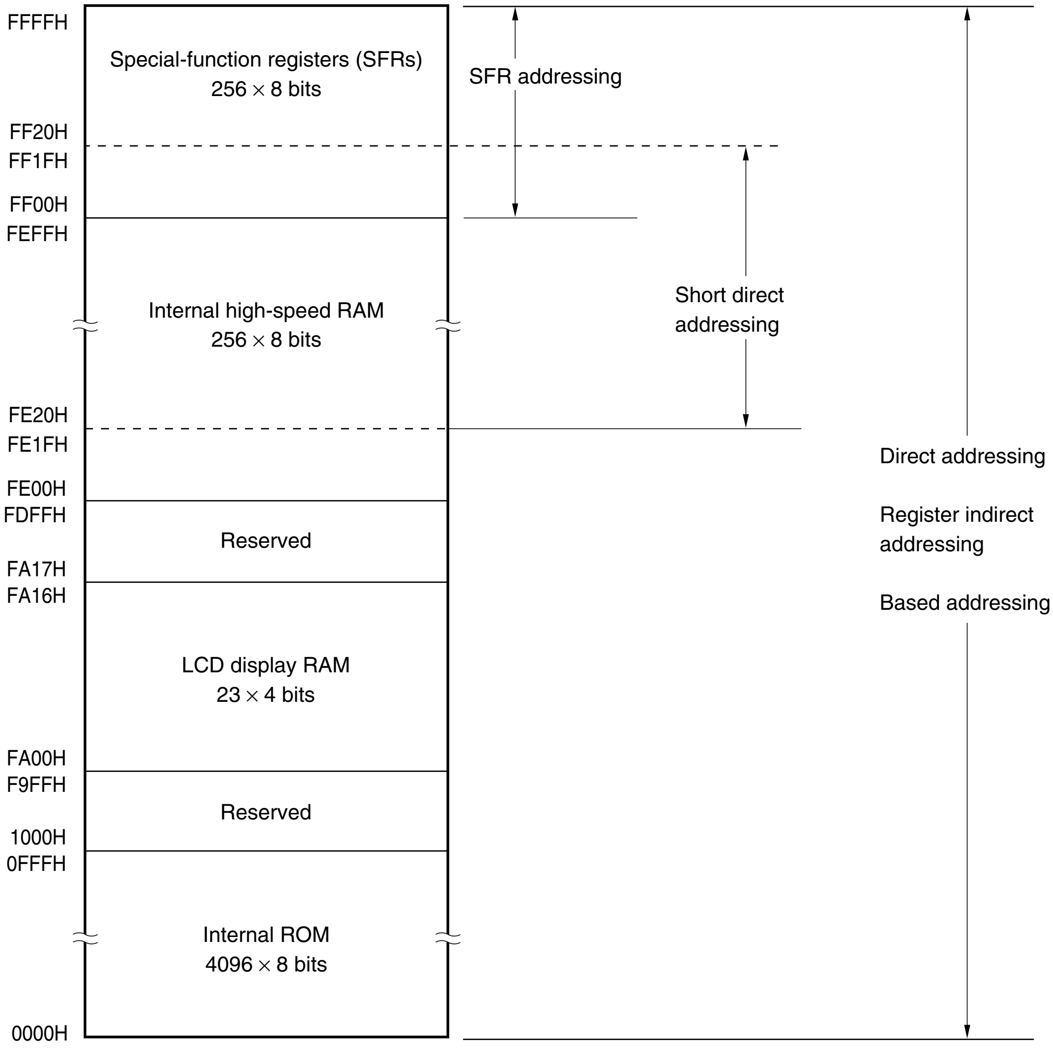


Figure 3-7. Data Memory Addressing (μPD789464)

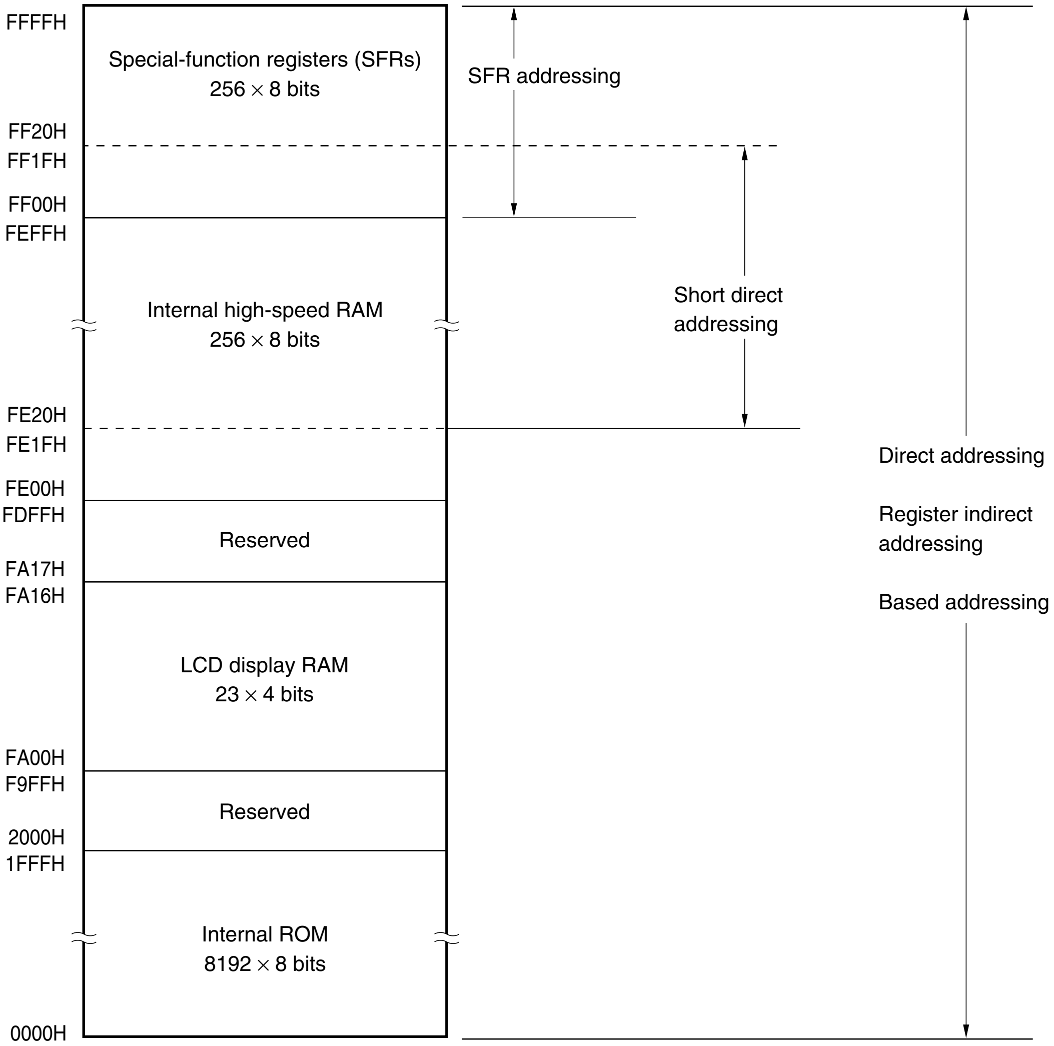


Figure 3-8. Data Memory Addressing (μPD789466)

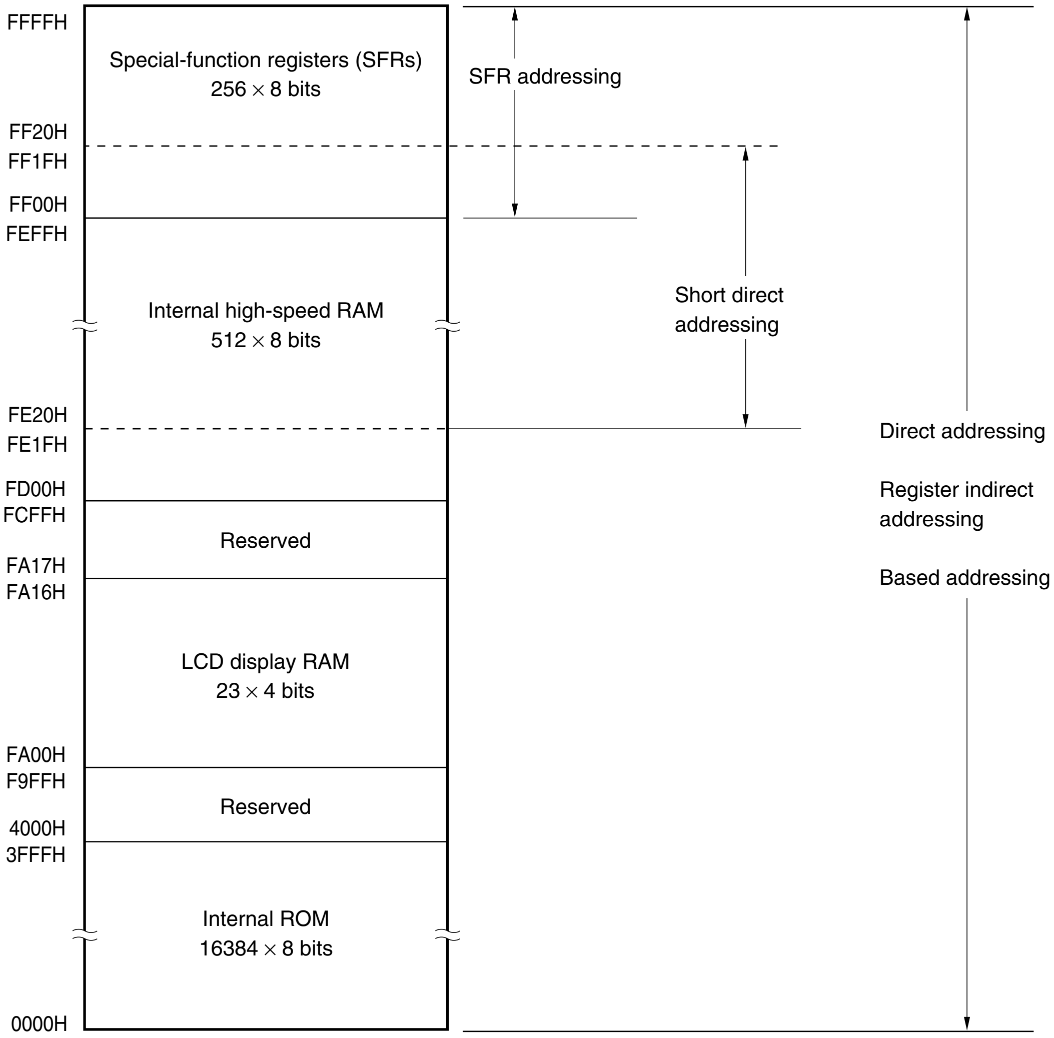


Figure 3-9. Data Memory Addressing (μPD789467)

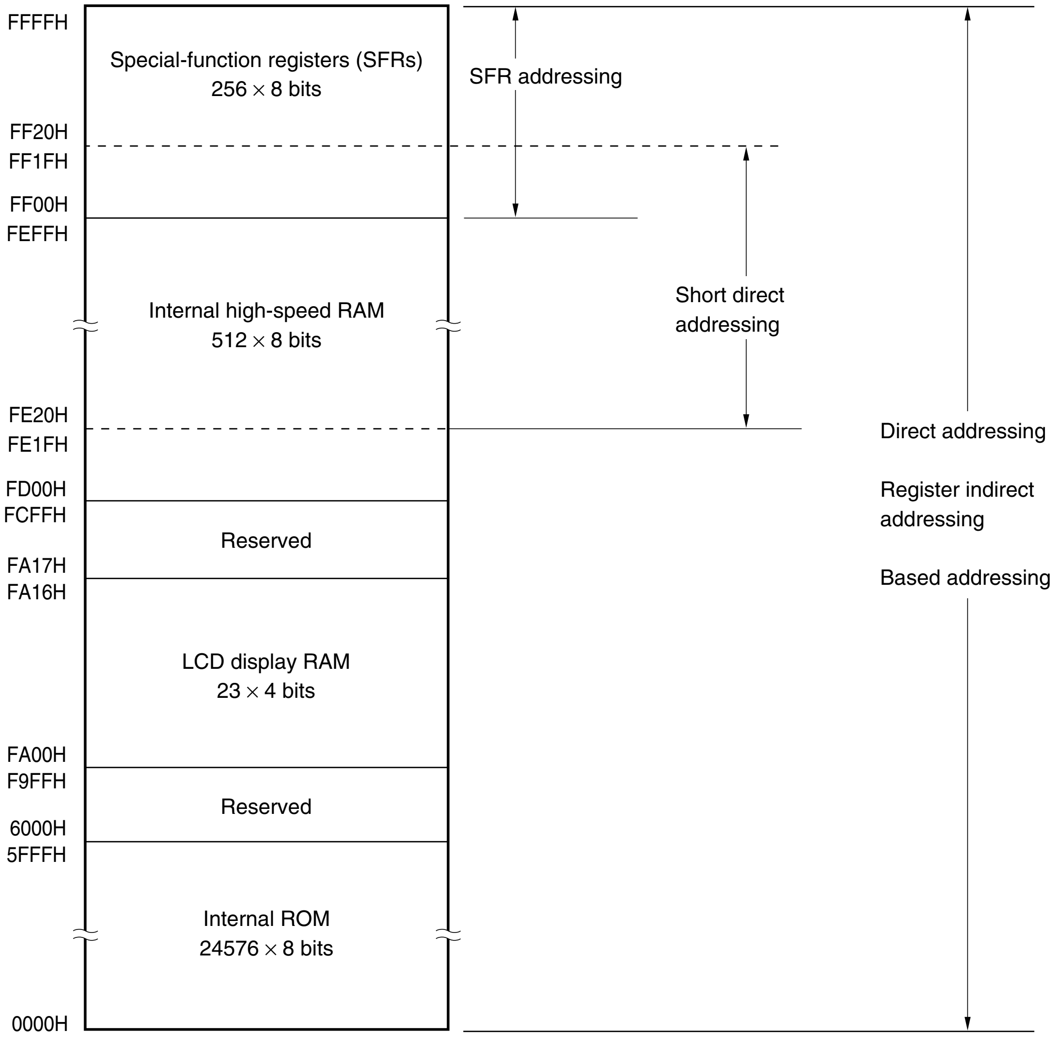
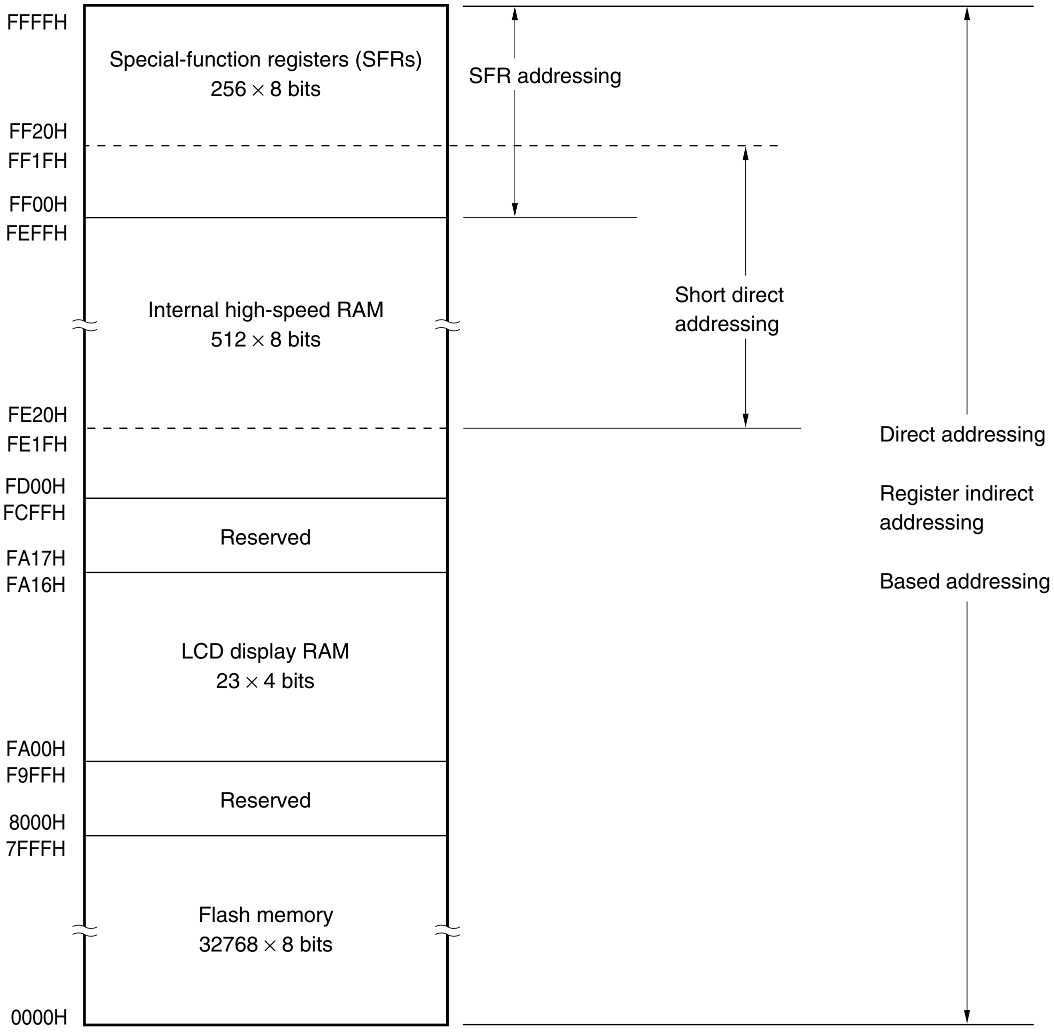


Figure 3-10. Data Memory Addressing (μPD78F9468)



3.2 Processor Registers

The μPD789467 Subseries provides the following on-chip processor registers.

3.2.1 Control registers

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

(1) Program counter (PC)

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

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

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

Figure 3-11. Program Counter Configuration

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

(2) Program status word (PSW)

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

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

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

Figure 3-12. Program Status Word Configuration

	7							0
PSW	IE	Z	0	AC	0	0	1	CY

(a) Interrupt enable flag (IE)

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

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

When 1, IE is set to the interrupt enabled status (EI). Interrupt request acknowledgment enable is controlled with an interrupt mask flag corresponding to each interrupt source.

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

(b) Zero flag (Z)

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

(c) Auxiliary carry flag (AC)

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

(d) Carry flag (CY)

This flag stores 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 used to hold the start address of the memory stack area. Only the internal high-speed RAM area can be set as the stack area.

Figure 3-13. Stack Pointer Configuration

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

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

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

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

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

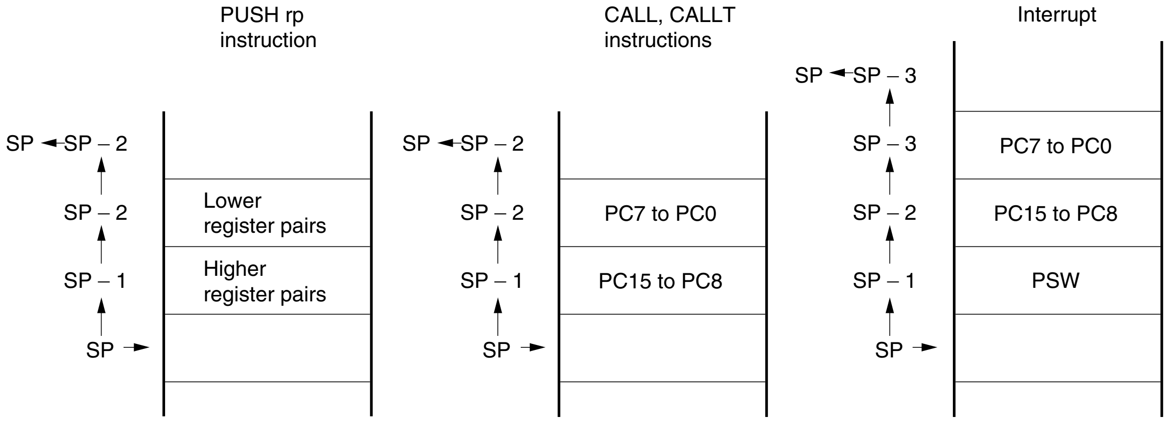
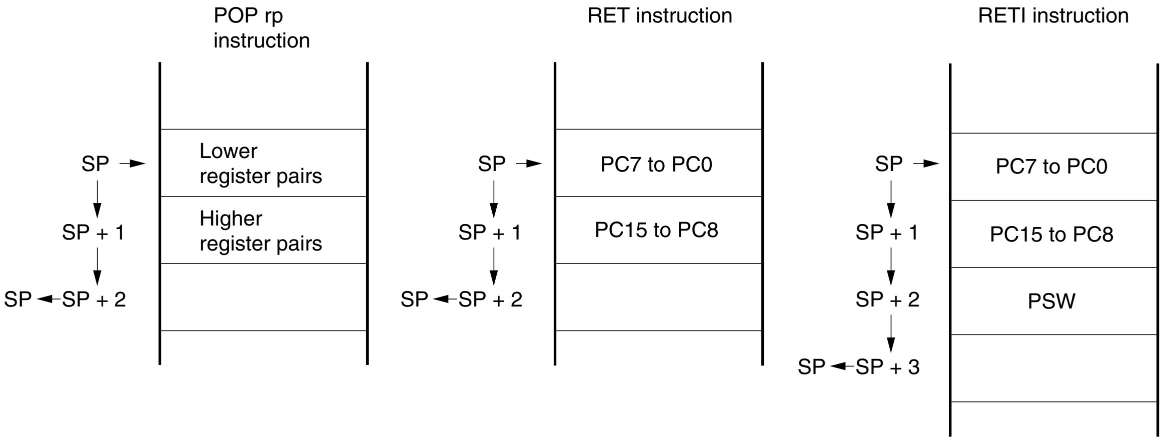


Figure 3-15. Data to Be Restored from Stack Memory



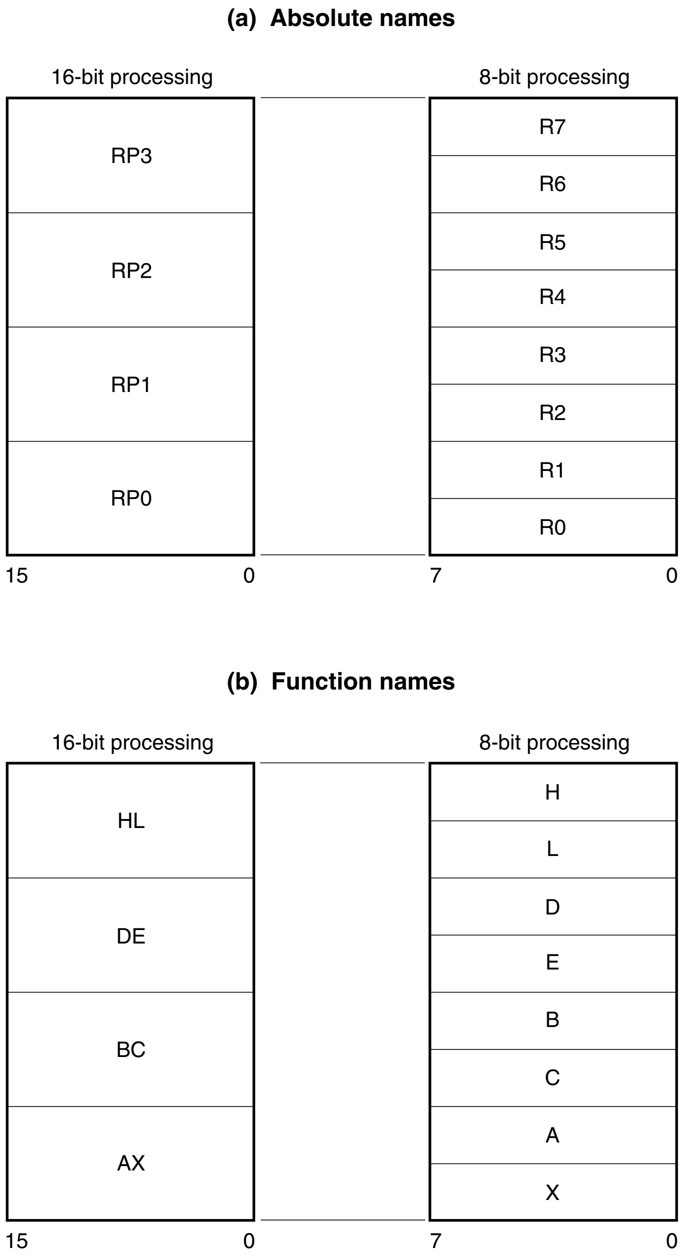
3.2.2 General-purpose registers

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

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

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

Figure 3-16. General-Purpose Register Configuration



(a) Absolute names

(b) Function names

3.2.3 Special-function registers (SFRs)

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

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

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

The manipulatable bits can be specified as follows.

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

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

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

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

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

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

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

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

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

Table 3-3. Special-Function Registers

Address	Special-Function Register (SFR) Name	Symbol	R/W	Bit Unit for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
FF00H	Port 0	P0	R/W	√	√	–	00H
FF01H	Port 1	P1		√	√	–	
FF03H	port 4	P4		√	√	–	
FF05H	Port 6	P6		√	√	–	
★ FF08H	Port 8	P8	R	√	√	–	Undefined
FF15H	A/D conversion result register	ADCR0		–	√	–	
FF20H	Port mode register 0	PM0	R/W	√	√	–	FFH
FF21H	Port mode register 1	PM1		√	√	–	
FF24H	Port mode register 4	PM4		√	√	–	
FF26H	Port mode register 6	PM6		√	√	–	
FF42H	Watchdog timer clock selection resister	TCL2		–	√	–	00H
FF4AH	Watch timer mode control register	WTM		√	√	–	
FF58H	Port function register 8	PF8		√	√	–	
FF63H	8-bit compare register 30	CR30	W	–	√	–	Undefined
FF64H	8-bit timer counter 30	TM30	R	–	√	–	00H
FF65H	8-bit timer mode control register 30	TMC30	R/W	√	√	–	
FF66H	8-bit compare register 40	CR40	W	–	√	–	Undefined
FF67H	8-bit H width compare register 40	CRH40		–	√	–	
FF68H	8-bit timer counter 40	TM40	R	–	√	–	00H
FF69H	8-bit timer mode control register 40	TMC40	R/W	√	√	–	
FF6AH	Carrier generator output control register 40	TCA40	W	–	√	–	
FF80H	A/D converter mode register 0	ADM0	R/W	√	√	–	
FF84H	A/D input selection register 0	ADS0		√	√	–	
FFB0H	LCD display mode register 0	LCDM0		√	√	–	00H$^{\text{Note}}$
FFB2H	LCD clock control register 0	LCDC0		√	√	–	00H
FFB3H	LCD voltage boost control register 0	LCDVA0		√	√	–	
FFE0H	Interrupt request flag register 0	IF0		√	√	–	
FFE4H	Interrupt mask flag register 0	MK0		√	√	–	FFH
FFECH	External interrupt mode register 0	INTM0		–	√	–	00H
FFF0H	Suboscillation mode register	SCKM		√	√	–	
FFF2H	Subclock control register	CSS		√	√	–	
FFF5H	Key return mode register 00	KRM00		√	√	–	
FFF7H	Pull-up resistor option register 0	PU0		√	√	–	
FFF9H	Watchdog timer mode register	WDTM		√	√	–	
★ FFFAH	Oscillation stabilization time selection register	OSTS		√	√	–	04H
FFFBH	Processor clock control register	PCC		√	√	–	02H

Note Bit 2 (LCDM02) must be set to 1 after reset.

3.3 Instruction Address Addressing

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

3.3.1 Relative addressing

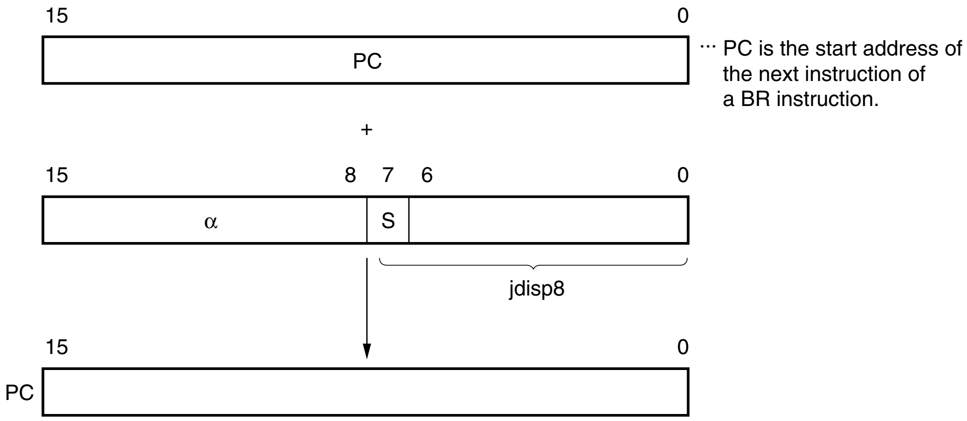
[Function]

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

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

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

[Illustration]


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

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

3.3.2 Immediate addressing

[Function]

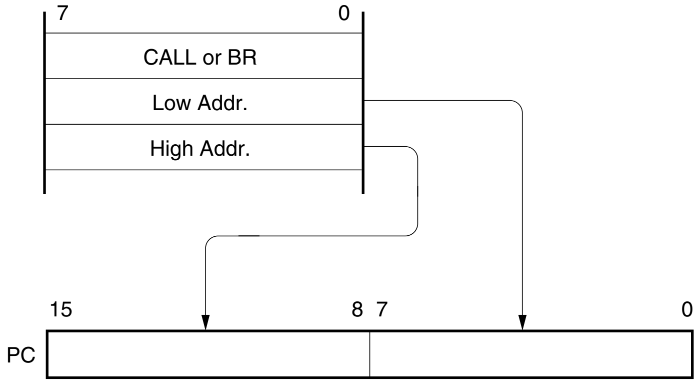
Immediate data in the instruction word is transferred to the program counter (PC) and branched.

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

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

[Illustration]

In case of CALL !addr16 and BR !addr16 instructions



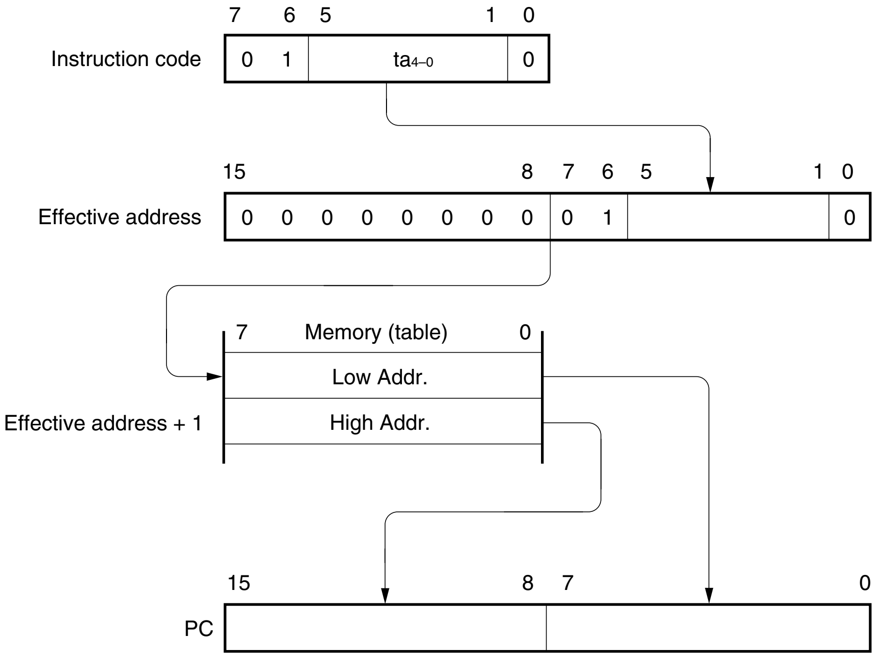
3.3.3 Table indirect addressing

[Function]

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

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

[Illustration]



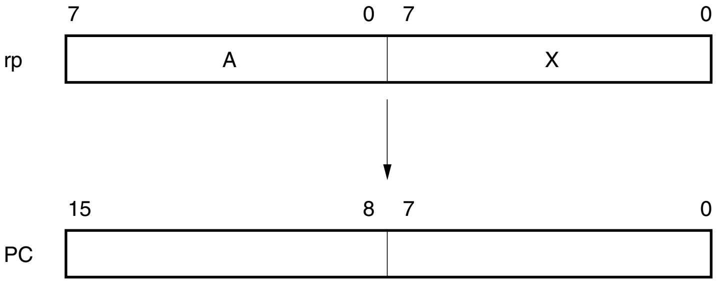
3.3.4 Register addressing

[Function]

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

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

[Illustration]



3.4 Operand Address Addressing

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

3.4.1 Direct addressing

[Function]

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

[Operand format]

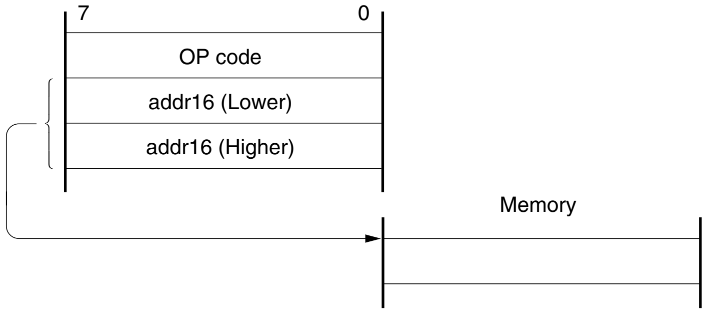
Identifier	Description
addr16	Label or 16-bit immediate data

[Description example]

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

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

[Illustration]



3.4.2 Short direct addressing

[Function]

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

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

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

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

[Operand format]

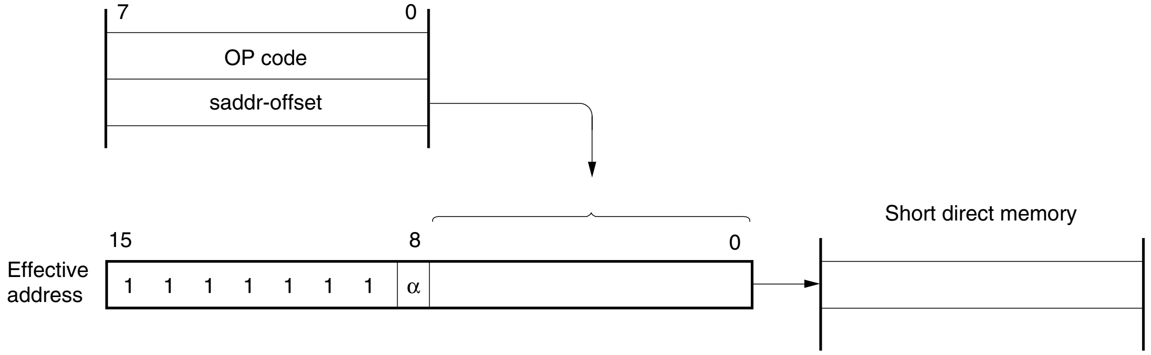
Identifier	Description
saddr	Label or FE20H to FF1FH immediate data
saddrp	Label or FE20H to FF1FH immediate data (even address only)

[Description example]

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

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

[Illustration]



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

3.4.3 Special-function register (SFR) addressing

[Function]

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

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

[Operand format]

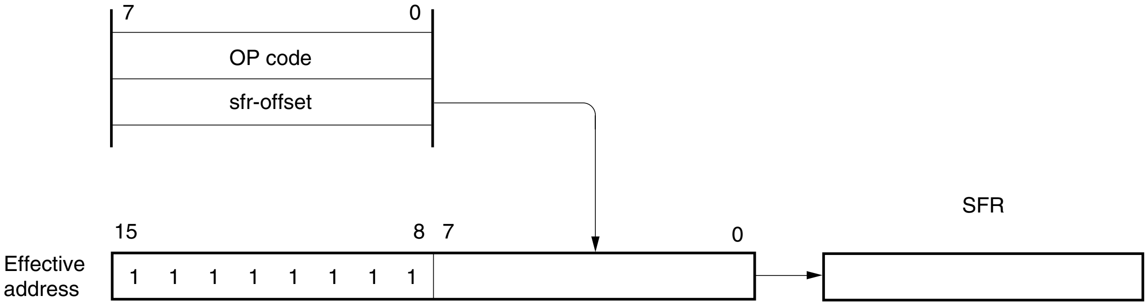
Identifier	Description
sfr	Special-function register name

[Description example]

MOV PM0, A; When selecting PM0 for sfr

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

[Illustration]



3.4.4 Register addressing

[Function]

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

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

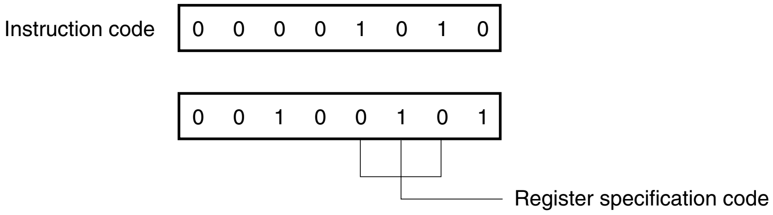
[Operand format]

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

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

[Description example]

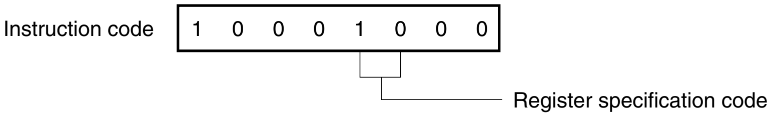
MOV A, C; When selecting the C register for r



Instruction code 0 0 0 0 1 0 1 0

0 0 1 0 0 1 0 1

INCW DE; When selecting the DE register pair for rp



Instruction code 1 0 0 0 1 0 0 0

3.4.5 Register indirect addressing

[Function]

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

[Operand format]

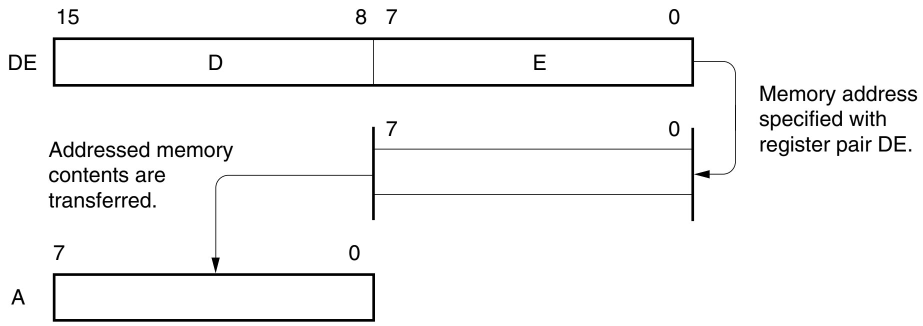
Identifier	Description
–	[DE], [HL]

[Description example]

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

Instruction code	0	0	1	0	1	0	1	1

[Illustration]


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

3.4.6 Based addressing

[Function]

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

[Operand format]

Identifier	Description
–	[HL+byte]

[Description example]

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

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

3.4.7 Stack addressing

[Function]

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

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

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

[Description example]

In the case of PUSH DE

Instruction code	1	0	1	0	1	0	1	0

CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

The μPD789467 Subseries provides the ports shown in Figure 4-1, enabling various methods of control.

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

Figure 4-1. Port Types

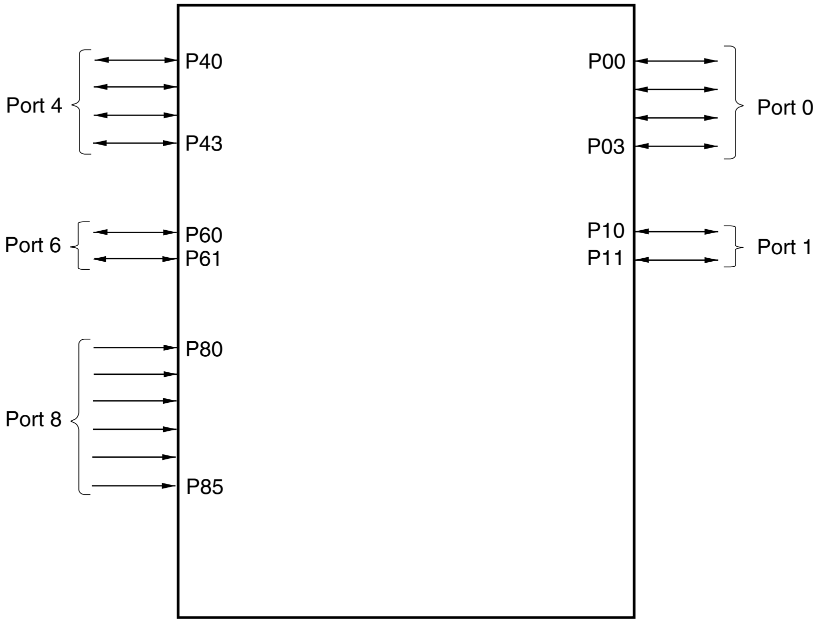


Table 4-1. Port Functions

Port Name	Pin Name	Function
Port 0	P00 to P03	This is an I/O port for which input and output can be specified in 1-bit units. When used as an input port, the use of on-chip pull-up resistors can be specified using pull-up resistor option register 0 (PU0).
Port 1	P10, P11	This is an I/O port for which input and output can be specified in 1-bit units. When used as an input port, the use of on-chip pull-up resistors can be specified using pull-up resistor option register 0 (PU0).
Port 4	P40 to P43	This is an I/O port for which input and output can be specified in 1-bit units. When used as an input port, the use of on-chip pull-up resistors can be specified using pull-up resistor option register 0 (PU0), or key return mode register 00 (KRM00).
Port 6	P60, P61	This is an I/O port for which input and output can be specified in 1-bit units.
Port 8	P80 to P85	This is an input port.

4.2 Port Configuration

The ports include the following hardware.

Table 4-2. Configuration of Port

Item	Configuration
Control registers	Port mode registers (PMm: m = 0, 1, 4, 6) Pull-up resistor option register 0 (PU0) Port function register 8 (PF8)
Ports	Total: 18 (CMOS I/O: 12, CMOS input: 6 (including pins shared with LCD)
Pull-up resistors	Total: 10 (software control: 10)

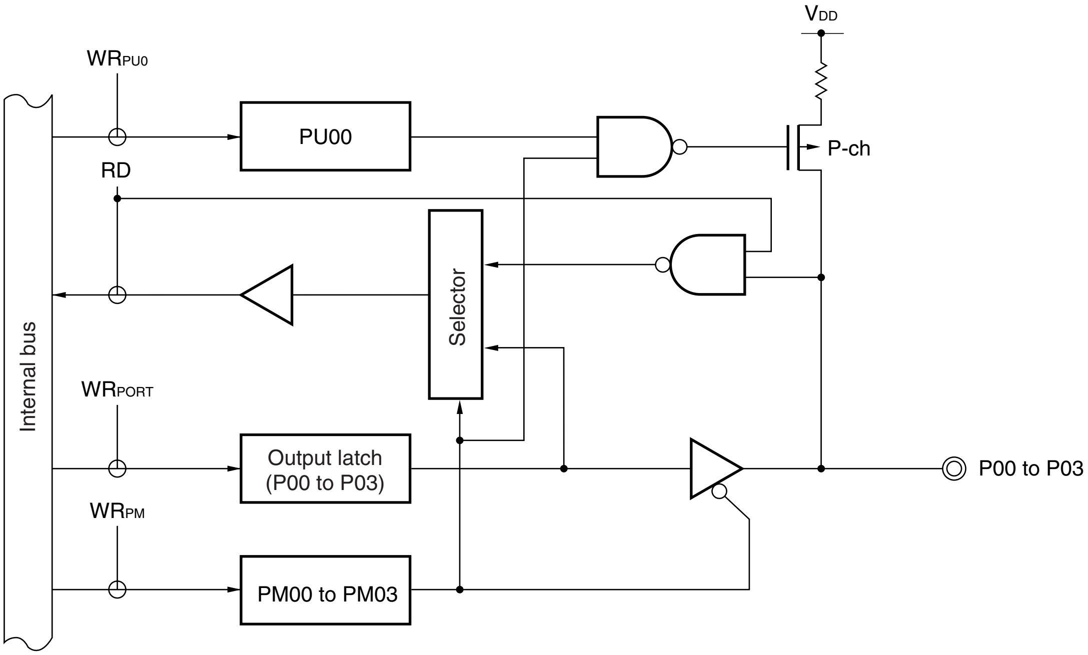
4.2.1 Port 0

This is a 4-bit I/O port with an output latch. Port 0 can be set to the input or output mode in 1-bit units using port mode register 0 (PM0). When the P00 to P03 pins are used as input port pins, on-chip pull-up resistors can be connected in 4-bit units using pull-up resistor option register 0 (PU0).

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

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

Figure 4-2. Block Diagram of P00 to P03



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

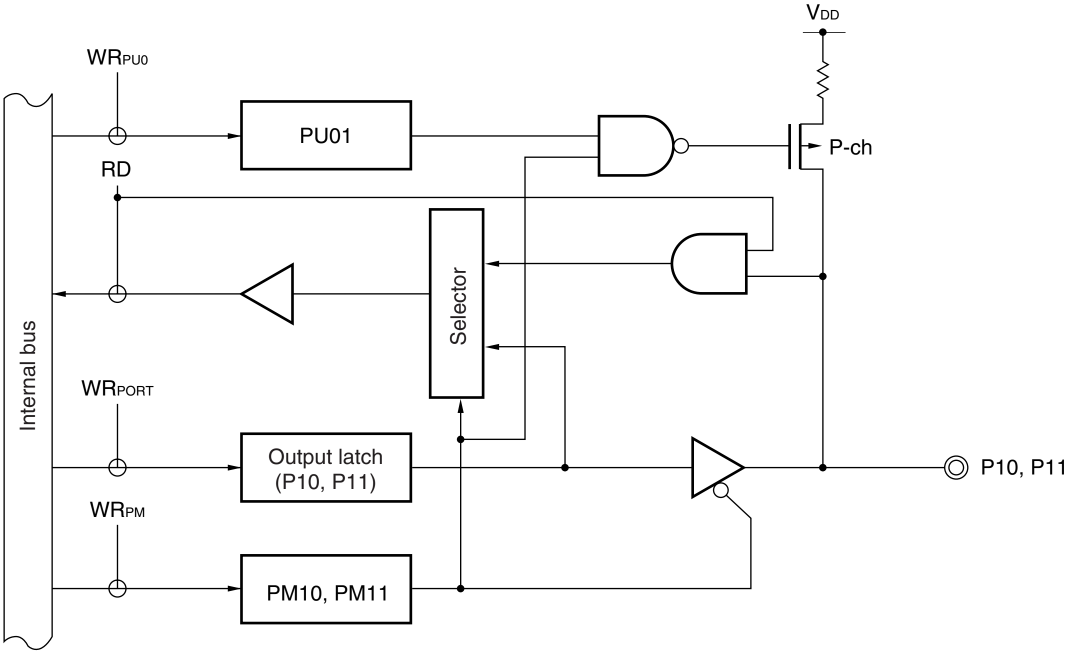
4.2.2 Port 1

This is a 2-bit I/O port with an output latch. Port 1 can be set to the input or output mode in 1-bit units using port mode register 1 (PM1). When using the P10 and P11 pins as input port pins, on-chip pull-up resistors can be connected in 2-bit units using pull-up resistor option register 0 (PU0).

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

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

Figure 4-3. Block Diagram of P10 and P11



PU0: Pull-up resistor option register 0

PM: Port mode register

RD: Port 1 read signal

WR: Port 1 write signal

4.2.3 Port 4

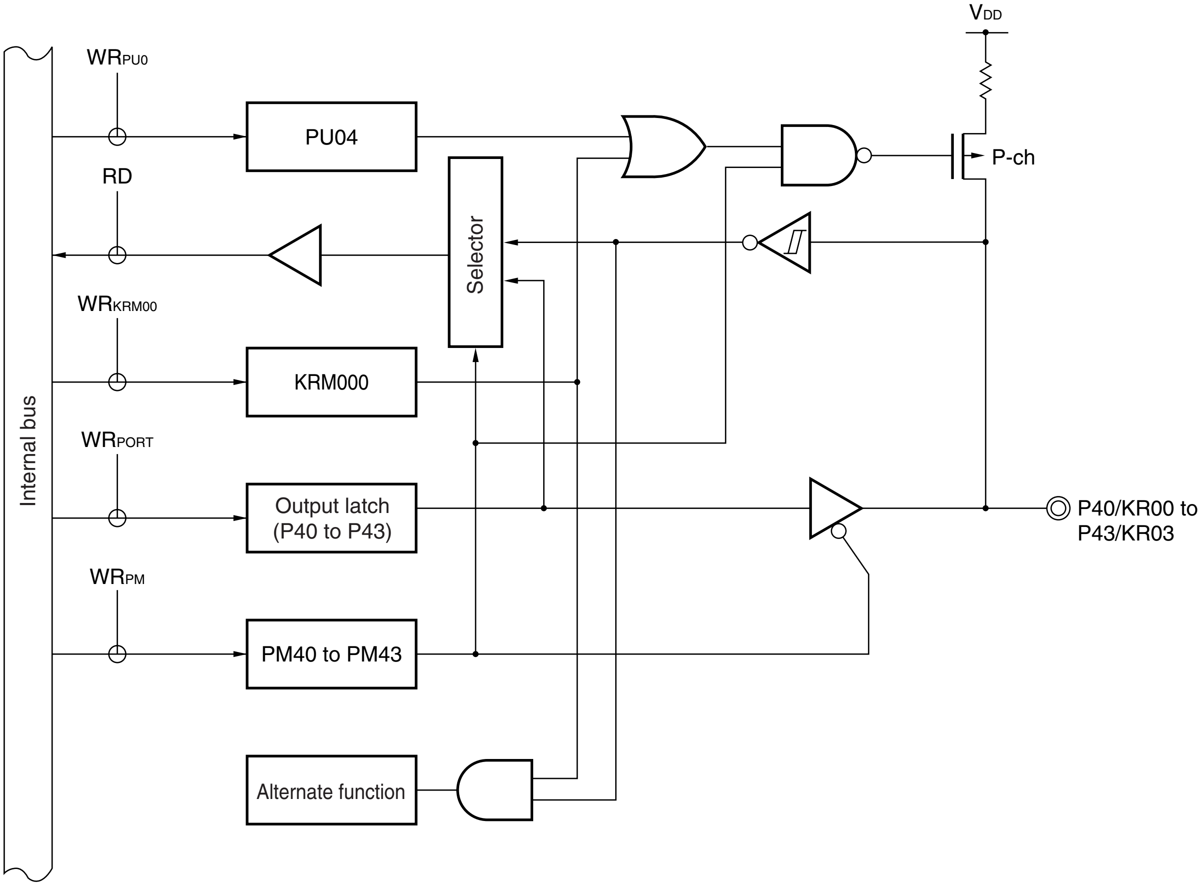
This is a 4-bit I/O port with an output latch. Port 4 can be set to the input or output mode in 1-bit units using port mode register 4 (PM4). When using the P40 to P43 pins as input port pins, on-chip pull-up resistors can be connected in 4-bit units using pull-up resistor option register 0 (PU0).

This port is also used as a key return input.

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

Figure 4-4 shows block diagram of port 4.

Figure 4-4. Block Diagram of P40 to P43



KRM00: Key return mode register 00
PU0: Pull-up resistor option register 0
PM: Port mode register
RD: Port 4 read signal
WR: Port 4 write signal

4.2.4 Port 6

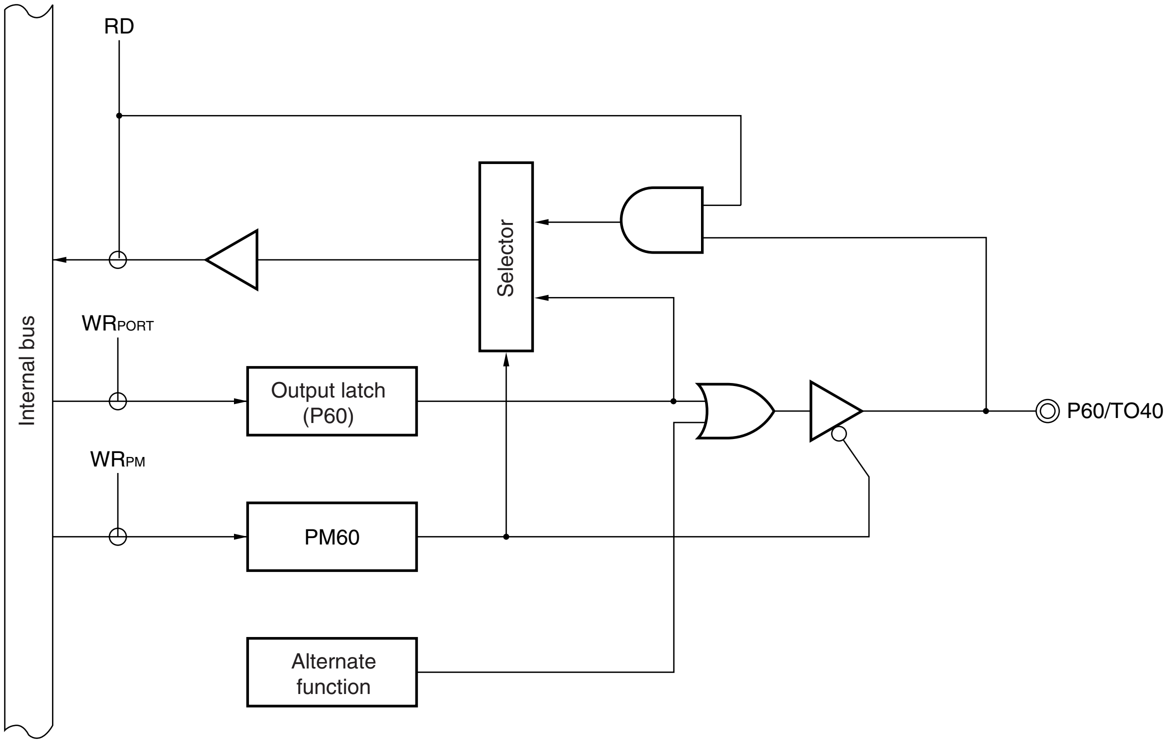
This is a 2-bit I/O port with an output latch. Port 6 can be set to the input or output mode in 1-bit units using port mode register 6 (PM6).

This port is also used as a timer output, external interrupt input, and analog input.

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

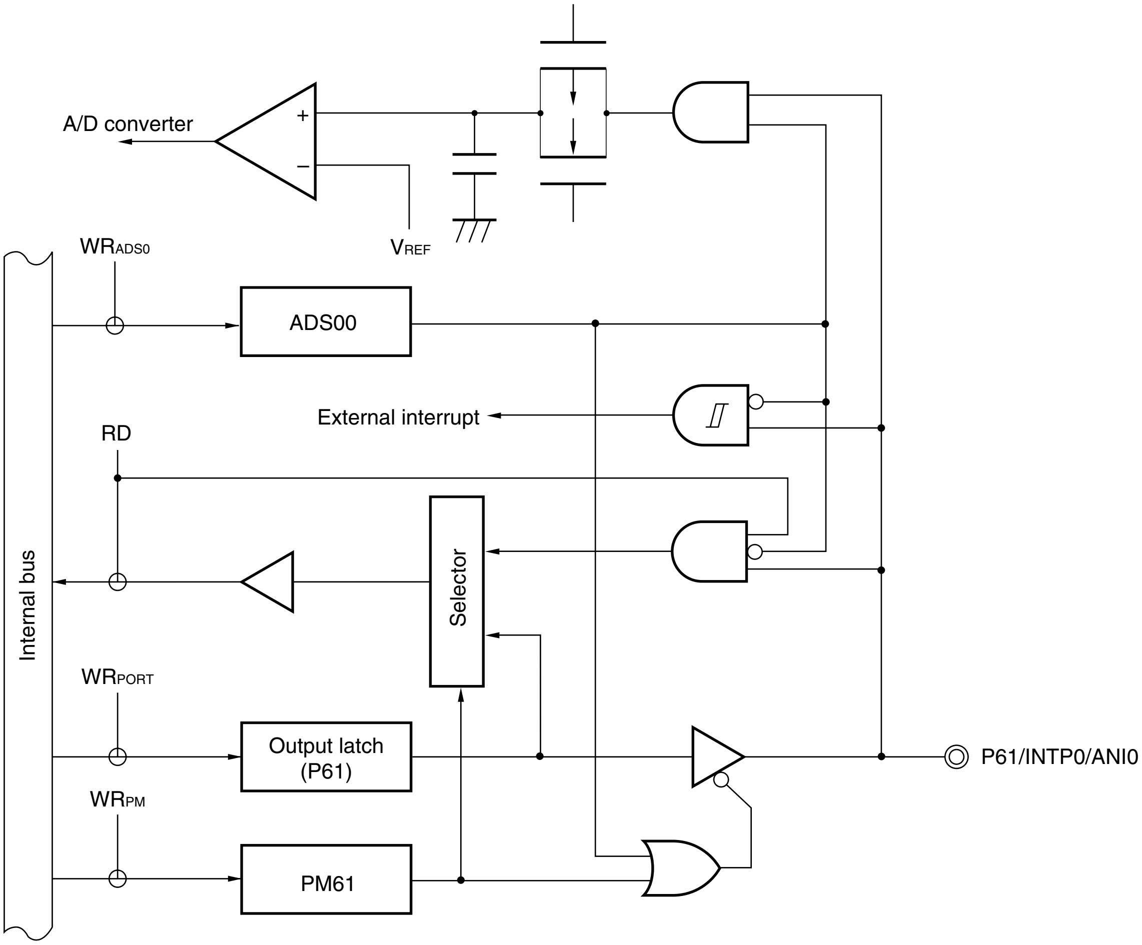
Figures 4-5 and 4-6 show block diagrams of port 6.

Figure 4-5. Block Diagram of P60



PM: Port mode register
RD: Port 6 read signal
WR: Port 6 write signal

Figure 4-6. Block Diagram of P61



ADS0: A/D input selection register 0

PM: Port mode register

RD: Port 6 read signal

WR: Port 6 write signal

4.2.5 Port 8

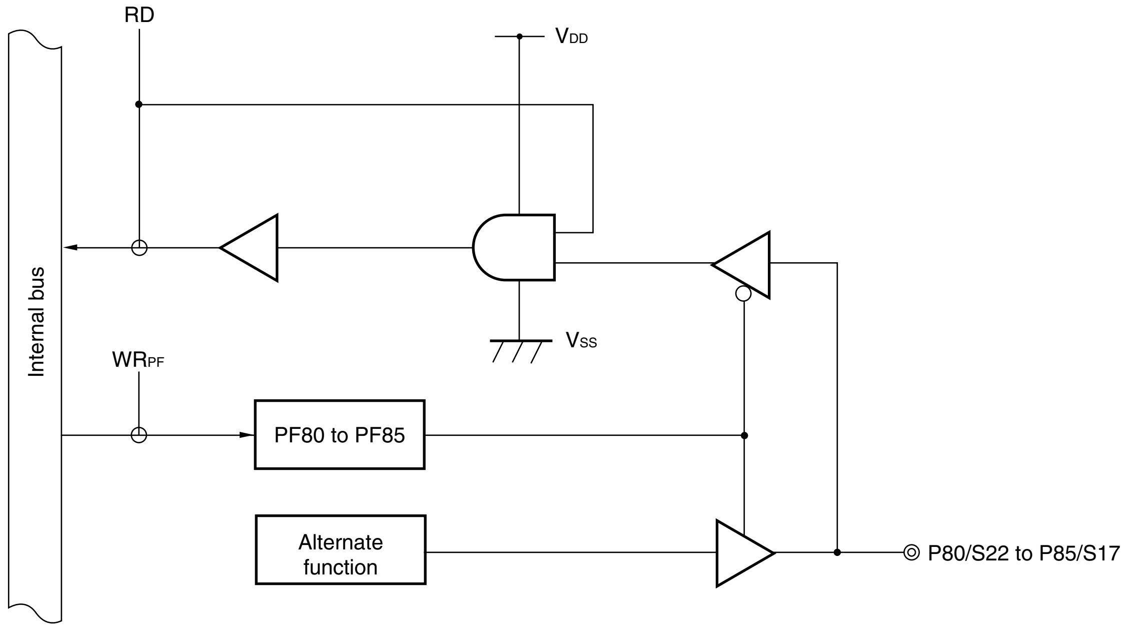
This is a 6-bit input port.

This port is also used as a segment output, and can be switched to the port function or segment output function in 1-bit units using port function register 8 (PF8).

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

Figure 4-7 shows a block diagram of port 8.

Figure 4-7. Block Diagram of P80 to P85



PF:	Port function register
RD:	Port 8 read signal
WR:	Port 8 write signal

4.3 Registers Controlling Port Function

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

- Port mode registers (PM0, PM1, PM4, PM6)
- Pull-up resistor option register 0 (PU0)
- Port function register 8 (PF8)

(1) Port mode registers (PM0, PM1, PM4, PM6)

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

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

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

When port pins are used as alternate-function pins, set the port mode register and output latch according to Table 4-3.

Caution As P61 has an alternate function as external interrupt input, when the port function output mode is specified and the output level is changed, the interrupt request flag is set. When the output mode is used, therefore, the interrupt mask flag (PMK0) should be preset to 1.

Figure 4-8. Format of Port Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	1	1	1	PM03	PM02	PM01	PM00	FF20H	FFH	R/W
PM1	1	1	1	1	1	1	PM11	PM10	FF21H	FFH	R/W
PM4	1	1	1	1	PM43	PM42	PM41	PM40	FF24H	FFH	R/W
PM6	1	1	1	1	1	1	PM61	PM60	FF26H	FFH	R/W

PMmn	Pmn pin input/output mode selection (m = 0, 1, 4, 6 n = 0 to 3)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

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

Pin Name	Alternate Function		PM××	PM××	ADS00
	Name	I/O			
P40 to P43	KR00 to KR03	Input	1	×	×
P60	TO40	Output	0	0	×
P61	INTP0	Input	1	×	0
	ANI0	Input	1	×	1

Remark ×: Don't care
PM××: Port mode register
P××: Port output latch
ADS00: Bit 0 of A/D input selection register 0 (ADS0)

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

Pull-up resistor option register 0 (PU0) sets whether the on-chip pull-up resistor at ports 0, 1, and 4 is used or not in port units.

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

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

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

Figure 4-9. Format of Pull-up Resistor Option Register 0

Symbol	7	6	5	<4>	3	2	<1>	<0>	Address	After reset	R/W
PU0	0	0	0	PU04	0	0	PU01	PU00	FFF7H	00H	R/W

PU0m	Pm on-chip pull-up resistor selection (m = 0, 1, 4)
0	On-chip pull-up resistor not used
1	On-chip pull-up resistor used

Caution Bits 2, 3, and 5 to 7 must be set to 0.

(3) Port function register 8 (PF8)

Port function register 8 (PF8) sets the port function of port 8 in 1-bit units.

The pins of port 8 are selected as either LCD segment signal outputs or general-purpose port pins according to the setting of PF8.

PF8 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PF8 to 00H.

Figure 4-10. Format of Port Function Register 8

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PF8	0	0	PF85	PF84	PF83	PF82	PF81	PF80	FF58H	00H	R/W

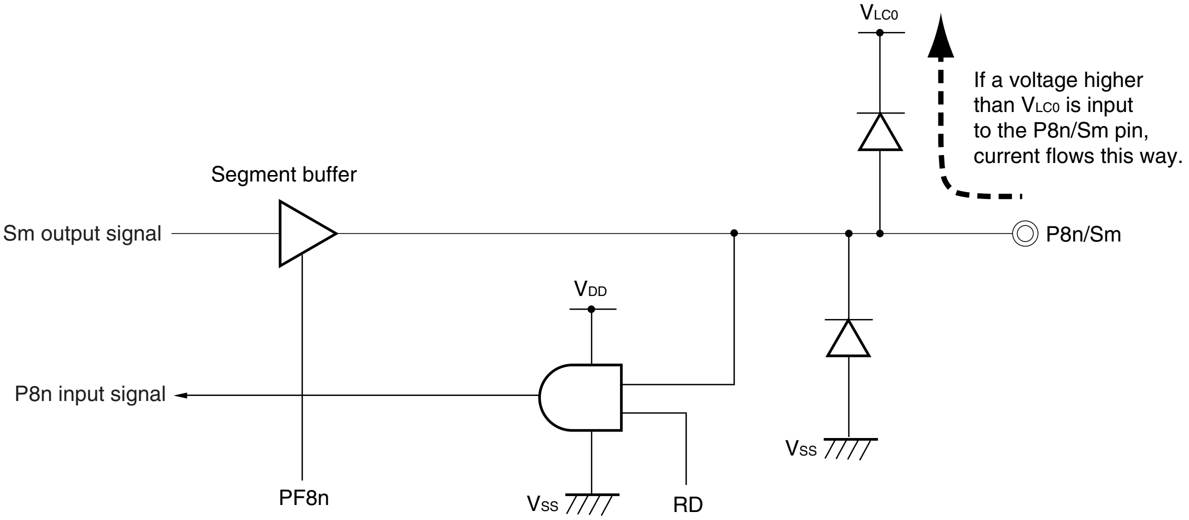
PF8n	P8n port function (n = 0 to 5)
0	Operates as a general-purpose port
1	Operates as an LCD segment signal output

Cautions 1. Bits 6 and 7 must be set to 0.

★ **2. When any one of pins P80 to P85 is used as a general-purpose pin, observe the following two restrictions (because an ESD protection circuit for the LCD pins is connected to the V_{LC0} side).**

- **Enable operation of the booster (VAON0 = 1).**
- **In V_{LC0} = 3.0 V mode (GAIN = 1)... Use the microcontroller in the range of V_{DD} = 1.8 to 3.0 V.**
  In V_{LC0} = 4.5 V mode (GAIN = 0)... Use the microcontroller in the range of V_{DD} = 1.8 to 4.5 V.

When all of pins P80 to P85 are used as LCD segment pins, the microcontroller can be used in the range of V_{DD} = 1.8 to 5.5 V.


VLC0
If a voltage higher than VLC0 is input to the P8n/Sm pin, current flows this way.
Segment buffer
Sm output signal
P8n/Sm
VDD
P8n input signal
VSS
PF8n
VSS
RD

Remark Sm: LCD segment output (m = 22 to 17)
P8n: Bit n of port 8 (n = 0 to 5)
PF8n: Bit n of port function register 8 (n = 0 to 5)
RD: Read signal of port 8n

4.4 Port Function Operation

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

4.4.1 Writing to I/O port

(1) In output mode

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

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

(2) In input mode

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

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

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

4.4.2 Reading from I/O port

(1) In output mode

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

(2) In input mode

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

4.4.3 Arithmetic operation of I/O port

(1) In output mode

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

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

(2) In input mode

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

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

CHAPTER 5 CLOCK GENERATOR

5.1 Clock Generator Functions

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

- **Main system clock (ceramic/crystal) oscillator**
  This circuit oscillates at 1.0 to 5.0 MHz. Oscillation can be stopped by executing the STOP instruction or setting the processor clock control register (PCC).
- **Subsystem clock oscillator**
  This circuit oscillates at 32.768 kHz. Oscillation can be stopped by the suboscillation mode register (SCKM).

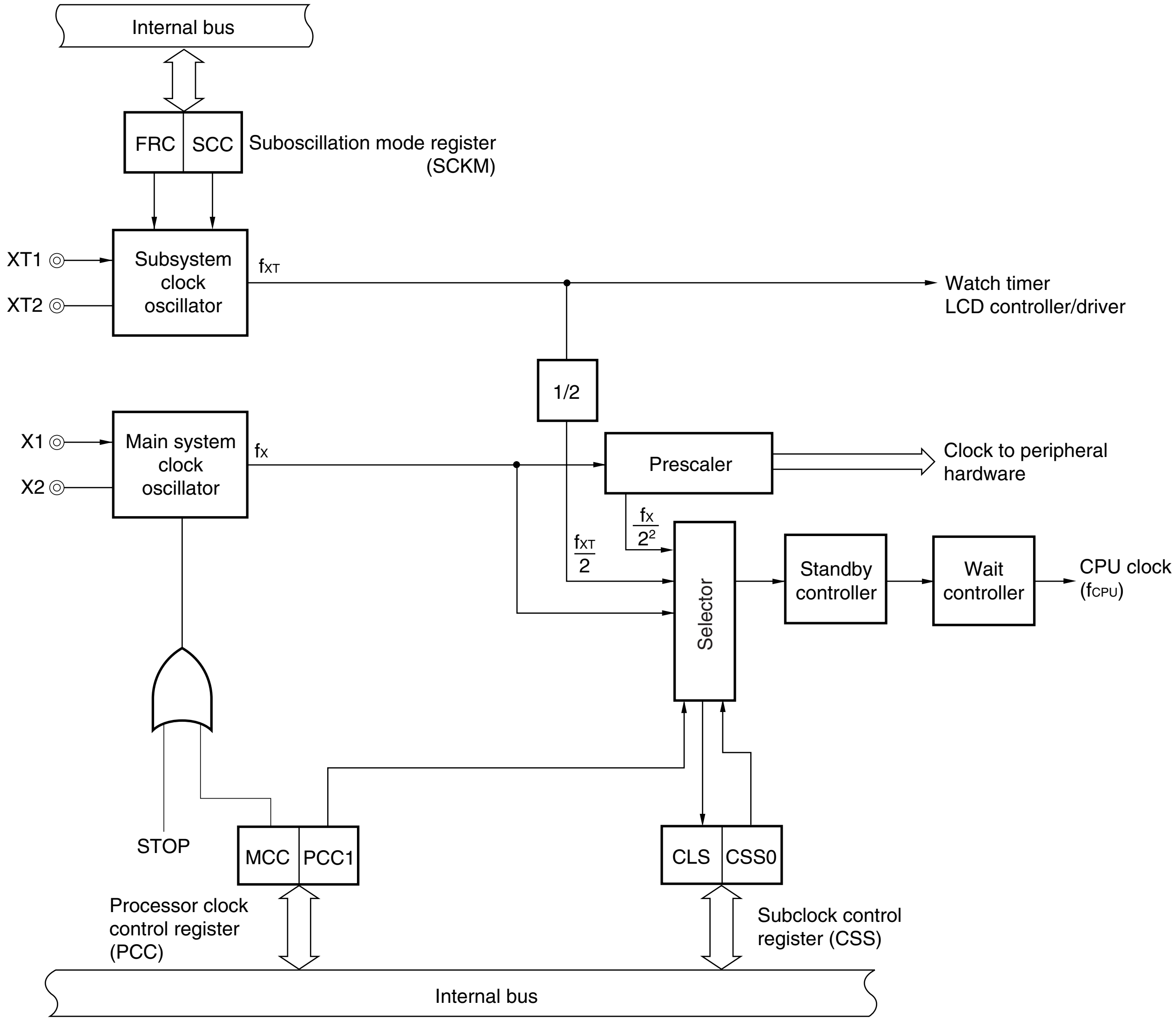
5.2 Clock Generator Configuration

The clock generator includes the following hardware.

Table 5-1. Configuration of Clock Generator

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

Figure 5-1. Block Diagram of Clock Generator



5.3 Registers Controlling Clock Generator

The clock generator is controlled by the following three registers.

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

(1) Processor clock control register (PCC)

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

Figure 5-2. Format of Processor Clock Control Register

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

MCC	Control of main system clock oscillator operation
0	Operation enabled
1	Operation disabled

CSS0	PCC1	CPU clock (f_{CPU}) selection[Note]	Maximum instruction execution time: $2/f_{CPU}$
			At f_X = 5.0 MHz or f_{XT} = 32.768 kHz Operation
0	0	f_X	0.4 µs
0	1	$f_X/2^2$	1.6 µs
1	×	$f_{XT}/2$	122 µs

Note The CPU clock is selected according to a combination of the PCC1 flag in the processor clock control register (PCC) and the CSS0 flag in the subclock control register (CSS) (refer to **5.3 (3) Subclock control register (CSS)**).

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

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

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

3. ×: Don't care

(2) Suboscillation mode register (SCKM)

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

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

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

Figure 5-3. Format of Suboscillation Mode Register

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

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

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

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

Caution Bits 2 to 7 must be set to 0.

(3) Subclock control register (CSS)

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

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

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

Figure 5-4. Format of Subclock Control Register

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

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

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

Note Bit 5 is read-only.

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

5.4 System Clock Oscillators

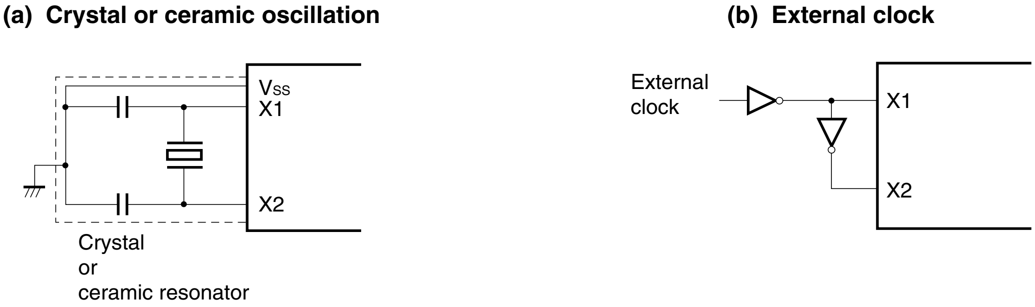
5.4.1 Main system clock oscillator

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

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

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

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



(a) Crystal or ceramic oscillation

(b) External clock

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

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

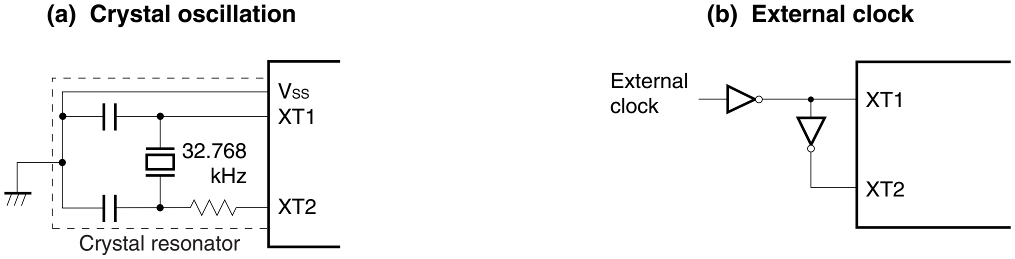
5.4.2 Subsystem clock oscillator

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

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

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

Figure 5-6. External Circuit of Subsystem Clock Oscillator


(a) Crystal oscillation

(b) External clock

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

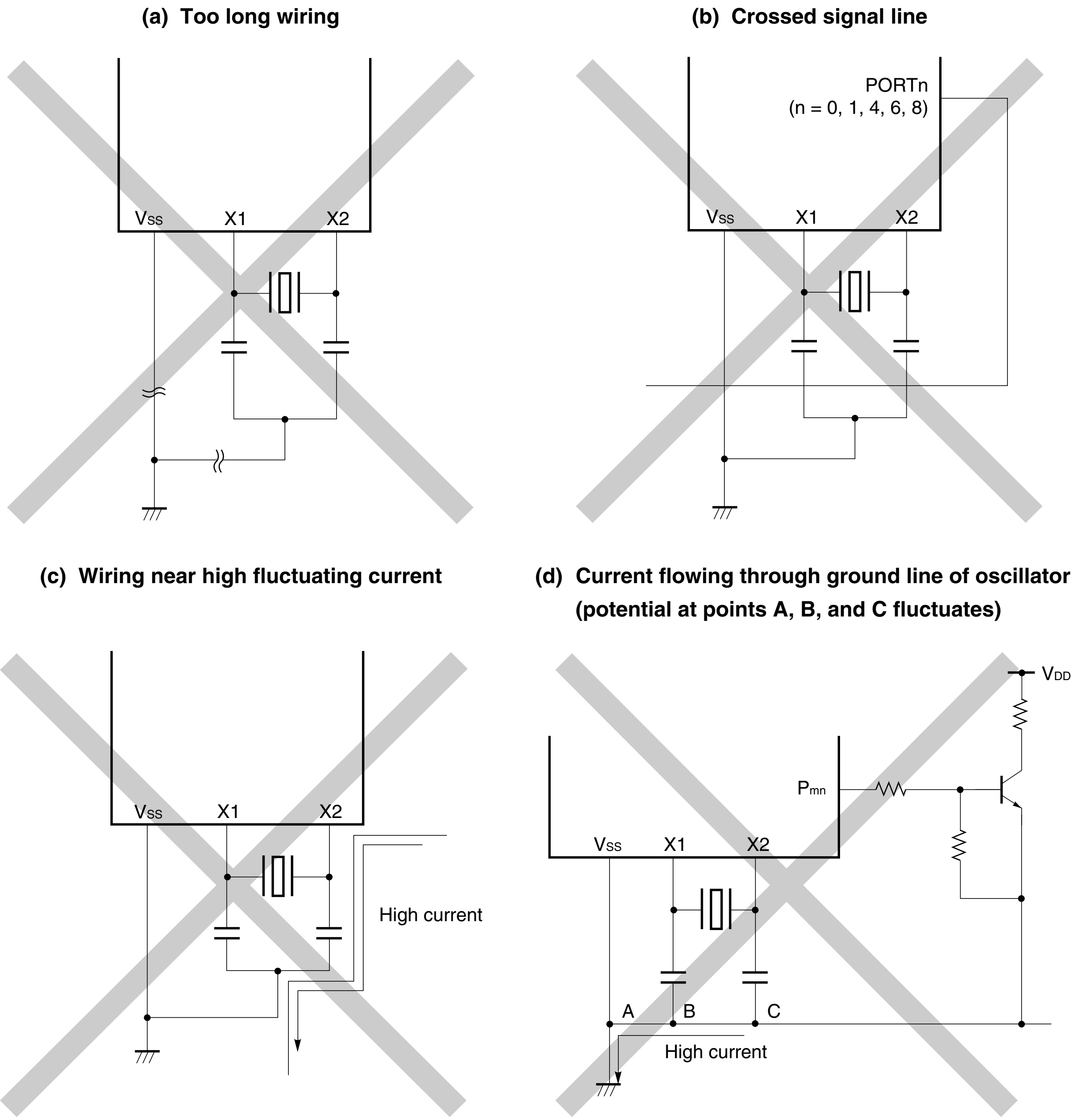
- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as V_{SS}. Do not ground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

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

5.4.3 Example of incorrect resonator connection

Figure 5-7 shows examples of incorrect resonator connection.

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



(a) Too long wiring

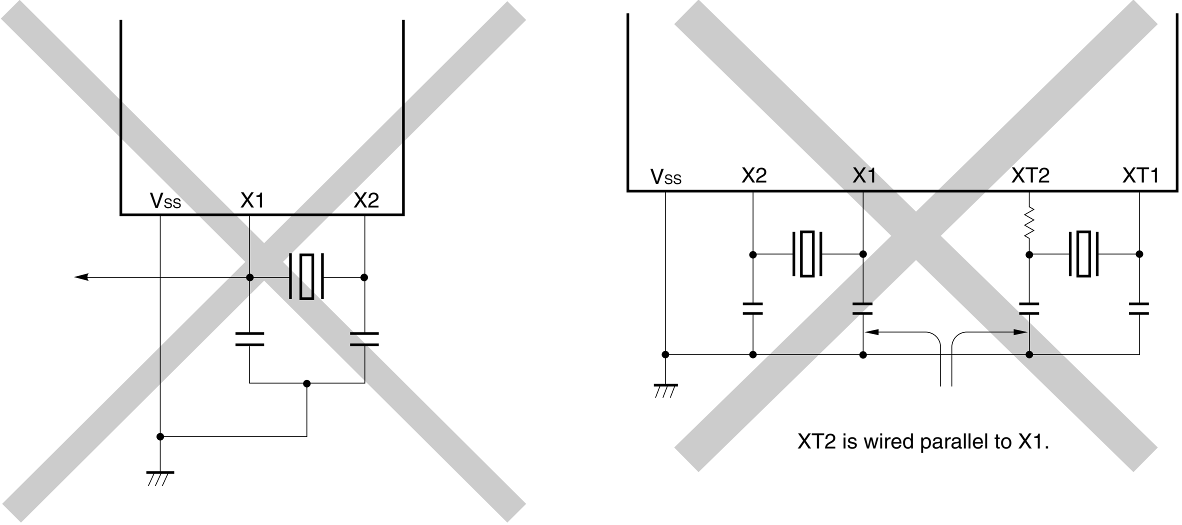
(b) Crossed signal line

(c) Wiring near high fluctuating current

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

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

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



(e) Signal is fetched

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

XT2 is wired parallel to X1.

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

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

5.4.4 Divider

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

5.4.5 When no subsystem clock is used

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

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

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

5.5 Clock Generator Operation

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

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

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

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

(b) Three types of minimum instruction execution time (0.4 μs and 1.6 μs: main system clock (at 5.0 MHz operation), 122 μs: subsystem clock (at 32.768 kHz operation)) can be selected by the PCC, SCKM, and CSS settings.

(c) Two standby modes, STOP and HALT, can be used with the main system clock selected. In a system where no subsystem clock is used, setting bit 1 (FRC) of SCKM so that the on-chip feedback resistor cannot be used reduces current consumption in STOP mode. In a system where a subsystem clock is used, setting SCKM bit 0 to 1 can cause the subsystem clock to stop oscillation.

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

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

(f) The clock pulse for the peripheral hardware is generated by dividing the frequency of the main system clock, but the subsystem clock pulse is only supplied to the watch timer and LCD controller/driver. The watch timer and LCD controller/driver can therefore keep running even during standby. The other hardware stops when the main system clock stops because it runs based on the main system clock (except for external input clock operations).

5.6 Changing Setting of System Clock and CPU Clock

5.6.1 Time required for switching between system clock and CPU clock

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

The maximum time indicated in Table 5-2 is required until the CPU clock actually switches (i.e. switching does not occur immediately after the PCC register is rewritten). Until this time has elapsed, therefore, it is impossible to ascertain whether the clock before or after the switch is operating

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

<table>
<tr><th colspan="2">Set Value Before Switching</th><th colspan="6">Set Value After Switching</th></tr>
<tr><td rowspan="2">CSS0</td><td rowspan="2">PCC1</td><td>CSS0</td><td>PCC1</td><td>CSS0</td><td>PCC1</td><td>CSS0</td><td>PCC1</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>×</td></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2"></td><td colspan="2">4 clocks</td><td colspan="2">$2f_X/f_{XT}$ clocks
(306 clocks)</td></tr>
<tr><td>1</td><td colspan="2">2 clocks</td><td colspan="2"></td><td colspan="2">$f_X/2f_{XT}$ clocks
(76 clocks)</td></tr>
<tr><td>1</td><td>×</td><td colspan="2">2 clocks</td><td colspan="2">2 clocks</td><td colspan="2"></td></tr>
</table>

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

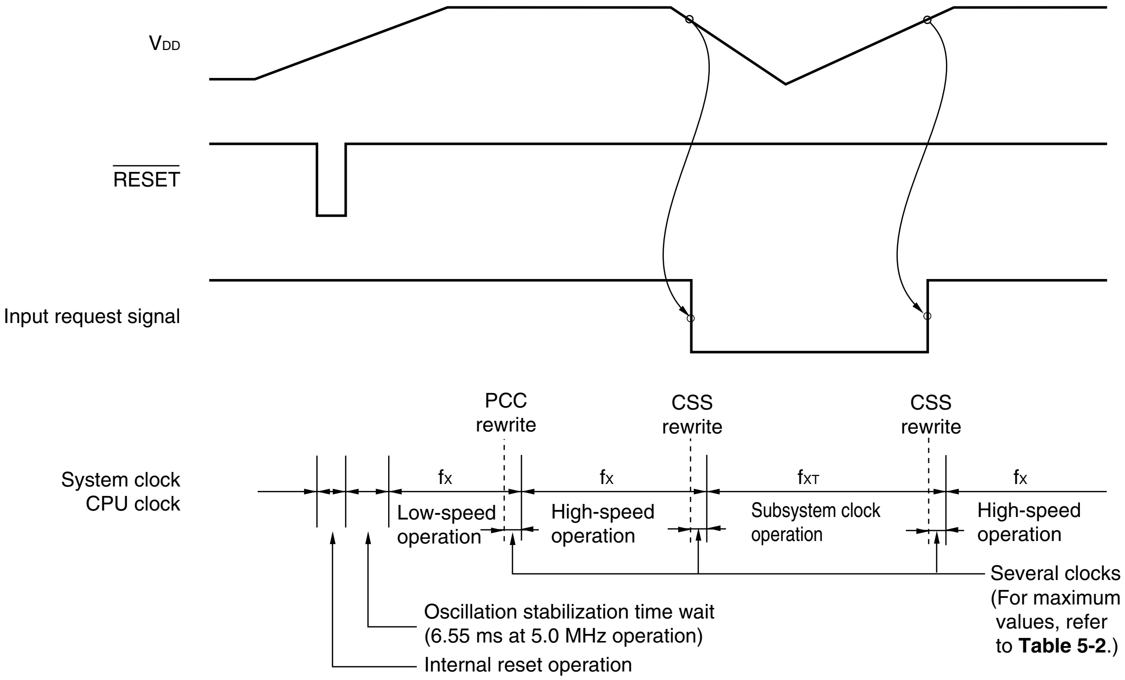
2. The parenthesized values apply to operation at f_X = 5.0 MHz or f_{XT} = 32.768 kHz.

3. ×: Don’t care

5.6.2 Switching between system clock and CPU clock

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

Figure 5-8. Example of Switching Between System Clock and CPU Clock



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

<2> After the time required for the V_{DD} voltage to rise to the level at which the CPU can operate at high speed has elapsed, bit 1 (PCC1) of the processor clock control register (PCC) is rewritten.

<3> After a few clocks have elapsed, the CPU clock is switched to high speed (0.4 μs at 5.0 MHz operation), and the CPU starts high-speed operation.

<4> A drop of the V_{DD} voltage is detected by an interrupt request signal. Bit 4 (CSS0) of the subclock control register (CSS) is rewritten so that the clock is switched to the subsystem clock (at this moment, the subsystem clock must be in the oscillation stabilized status).

<5> After a few clocks have elapsed, the CPU clock is switched to the subsystem clock operation (122 μs at 32.768 kHz operation). (At this time, bit 7 (MCC) of PCC can be set to 1 to stop the main system clock.)

<6> When recovery of the V_{DD} voltage is detected by an interrupt request signal, CSS0 is written so that the CPU clock is switched to the main system clock. (If the main system clock is stopped, set bit 7 (MCC) of PCC to 0 so that the main system clock starts oscillating. After the time required for the oscillation to stabilize has elapsed, rewrite CSS0.)

<7> After a few clocks, the CPU clock is switched to high speed (0.4 μs at 5.0 MHz operation), and the CPU returns to high-speed operation.

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

6.1 Functions of 8-Bit Timers 30 and 40

The 8-bit timer in the μPD789467 Subseries has 2 channels (timer 30 and timer 40). The operation modes listed in the following table can be set via mode register settings.

Table 6-1. Operation Modes

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

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

The following functions can be used in this mode.

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

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

Operation as a 16-bit timer is enabled during cascade connection mode.

The following functions can be used in this mode.

- Interval timer with 16-bit resolution
- Square-wave output with 16-bit resolution

(3) Carrier generator mode

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

(4) PWM output mode (timer 40 only)

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

6.2 Configuration of 8-Bit Timers 30 and 40

The 8-bit timers 30 and 40 include the following hardware.

Table 6-2. Configuration of 8-Bit Timers 30 and 40

Item	Configuration
Timer counters	8 bits × 2 (TM30, TM40)
Registers	Compare registers: 8 bits × 3 (CR30, CR40, CRH40)
Timer outputs	1 (TO40)
Control registers	8-bit timer mode control register 30 (TMC30) 8-bit timer mode control register 40 (TMC40) Carrier generator output control register 40 (TCA40) Port mode register 6 (PM6) Port 6 (P6)

Figure 6-1. Block Diagram of Timer 30

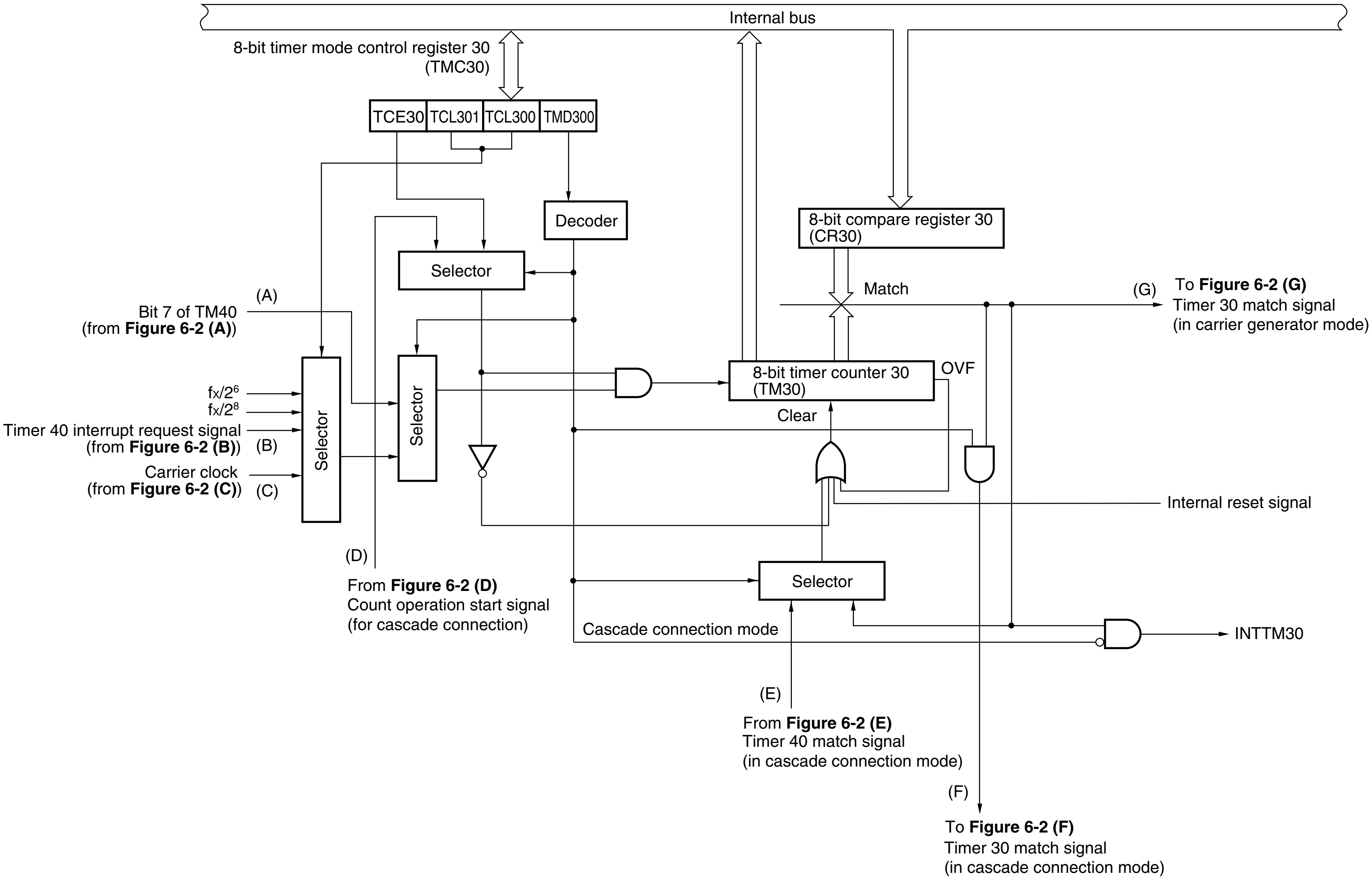
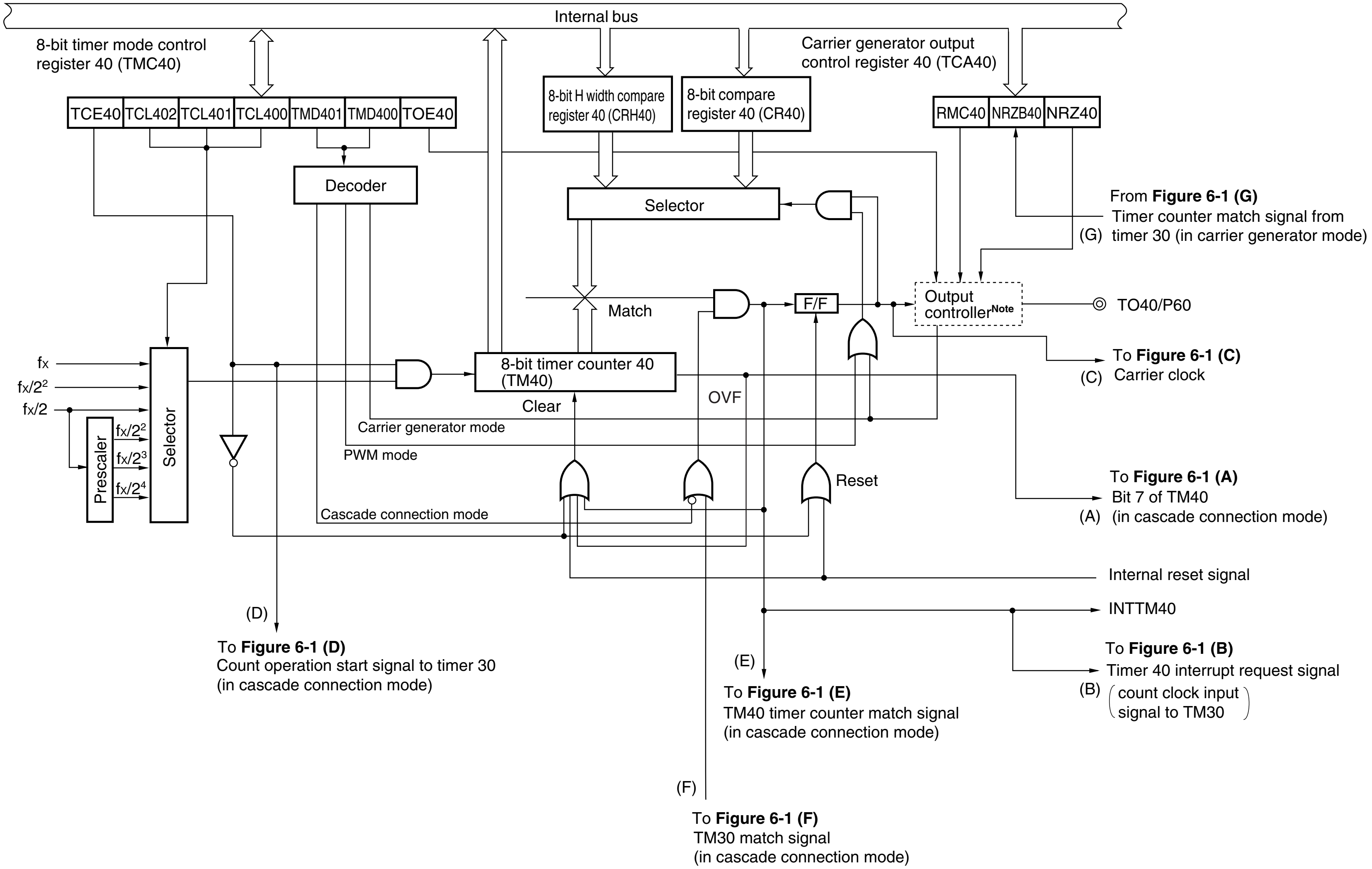
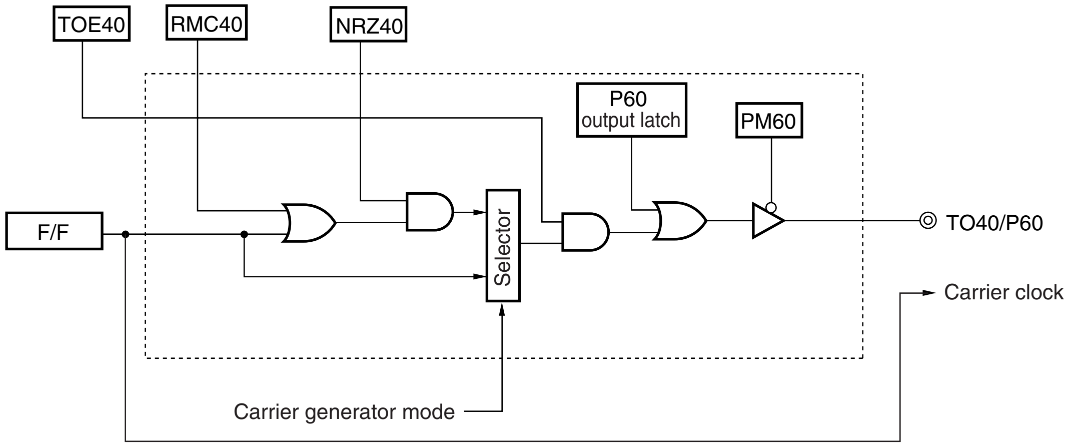


Figure 6-2. Block Diagram of Timer 40


Internal bus
8-bit timer mode control register 40 (TMC40)
TCE40
TCL402
TCL401
TCL400
TMD401
TMD400
TOE40
8-bit H width compare register 40 (CRH40)
8-bit compare register 40 (CR40)
Carrier generator output control register 40 (TCA40)
RMC40
NRZB40
NRZ40
Decoder
Selector
From Figure 6-1 (G)
(G) Timer counter match signal from timer 30 (in carrier generator mode)
Match
F/F
Output controller^Note
TO40/P60
To Figure 6-1 (C)
(C) Carrier clock
fx
fx/2^2
fx/2
Prescaler
fx/2^2
fx/2^3
fx/2^4
Selector
8-bit timer counter 40 (TM40)
Clear
OVF
Carrier generator mode
PWM mode
Reset
Cascade connection mode
To Figure 6-1 (A)
(A) Bit 7 of TM40 (in cascade connection mode)
Internal reset signal
INTTM40
To Figure 6-1 (B)
(B) Timer 40 interrupt request signal (count clock input signal to TM30)
(D)
To Figure 6-1 (D)
Count operation start signal to timer 30 (in cascade connection mode)
(E)
To Figure 6-1 (E)
TM40 timer counter match signal (in cascade connection mode)
(F)
To Figure 6-1 (F)
TM30 match signal (in cascade connection mode)

Note Refer to **Figure 6-3** for details.

★ **Figure 6-3. Block Diagram of Output Controller (Timer 40)**



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

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

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

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

Caution CR30 cannot be used in PWM output mode.

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

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

★ In carrier generator mode or PWM output mode, CR40 sets the timer output low-level width.

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

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

(3) 8-bit H width compare register 40 (CRH40)

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

★ The set value of CRH40 is always compared with the TM40 count value, and when they match, an interrupt request (INTTM40) is generated.

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

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

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

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

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

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

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

(a) Discrete mode

(i) TM30

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

(ii) TM40

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

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

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

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

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

6.3 Registers Controlling 8-Bit Timers 30 and 40

8-bit timer 30 and 40 are controlled by the following five registers.

- 8-bit timer mode control register 30 (TMC30)
- 8-bit timer mode control register 40 (TMC40)
- Carrier generator output control register 40 (TCA40)
- Port mode register 6 (PM6)
- Port 6 (P6)

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

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

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

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

Figure 6-4. Format of 8-Bit Timer Mode Control Register 30

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
TMC30	TCE30	0	0	TCL301	TCL300	0	TMD300	0	FF65H	00H	R/W

TCE30	Control of TM30 count operation[Note 1]
0	Clear TM30 count value and stop operation
1	Start count operation

TCL301	TCL300	Selection of timer 30 count clock
0	0	$f_X/2^6$ (78.1 kHz)
0	1	$f_X/2^8$ (19.5 kHz)
1	0	Timer 40 match signal
1	1	Carrier clock created for timer 40

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

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

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

Cautions 1. In cascade connection mode, the timer 40 output signal is forcibly selected for the count clock.

2. Be sure to clear bits 0, 2, 5, and 6 to 0.

Remarks 1. fx: Main system clock oscillation frequency

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

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

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

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

$\overline{\text{RESET}}$ input sets TMC40 to 00H.

Figure 6-5. Format of 8-Bit Timer Mode Control Register 40

Symbol	<7>	6	5	4	3	2	1	<0>	Address	After reset	R/W
TMC40	TCE40	0	TCL402	TCL401	TCL400	TMD401	TMD400	TOE40	FF69H	00H	R/W

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

TCL402	TCL401	TCL400	Selection of timer 40 count clock
0	0	0	fx (5 MHz)
0	0	1	fx/2^2 (1.25 MHz)
0	1	0	fx/2 (2.5 MHz)
0	1	1	fx/2^2 (1.25 MHz)
1	0	0	fx/2^3 (625 kHz)
1	0	1	fx/2^4 (313 kHz)
Other than above			Setting prohibited

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

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

Notes 1. Since the count operation is controlled by TCE40 in cascade connection mode, any setting for TCE30 (bit 7 of TMC30) is ignored.

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

Caution Be sure to clear bit 6 to 0.

Remarks 1. fx: Main system clock oscillation frequency

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

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

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

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

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

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

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TCA40	0	0	0	0	0	RMC40	NRZB40	NRZ40	FF6AH	00H	R/W

RMC40	Control of remote control output
0	When NRZ40 = 1, carrier pulse is output to TO40/P60 pin
1	When NRZ40 = 1, high-level signal is output to TO40/P60 pin

NRZB40	This is the bit that stores the next data to be output to NRZ40. Data is transferred to NRZ40 at the rising edge of the timer 30 match signal. Input the necessary value in NRZB40 in advance by program.

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

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

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

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

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

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

(4) Port mode register 6 (PM6)

This register is used to set the I/O mode of port 6 in 1-bit units.

When using the P60/TO40 pin as a timer output, set the PM60 and P60 output latch to 0.

PM6 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PM6 to FFH.

Figure 6-7. Format of Port Mode Register 6

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM6	1	1	1	1	1	1	PM61	PM60	FF26H	FFH	R/W

PM6n	I/O mode of P6n pin (n = 0, 1)
0	Output mode (output buffer is on)
1	Input mode (output buffer is off)

6.4 Operation of 8-Bit Timers 30 and 40

6.4.1 Operation as 8-bit timer counter

Timers 30 and 40 can be independently used as 8-bit timer counters.
The following modes can be used for the 8-bit timer counters.

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

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

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

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

When the count value of 8-bit timer counter n0 (TMn0) matches the value set in CRn0, TMn0 is cleared to 00H and continues counting. At the same time, an interrupt request signal (INTTMn0) is generated.
Tables 6-3 and 6-4 show the interval time, and Figures 6-8 to 6-12 show the timing of the interval timer operation.

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

Remark n = 3, 4

Table 6-3. Interval Time of Timer 30 (at fx = 5.0 MHz Operation)

TCL301	TCL300	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	2^6/fx (12.8 μs)	2^{14}/fx (3.28 ms)	2^4/fx (12.8 μs)
0	1	2^8/fx (51.2 μs)	2^{16}/fx (13.1 ms)	2^8/fx (51.2 μs)
1	0	Input cycle of timer 40 match signal	Input cycle of timer 40 match signal × 2^8	Input cycle of timer 40 match signal
1	1	Carrier clock cycle created with timer 40	Carrier clock cycle created with timer 40 × 2^8	Carrier clock cycle created with timer 40

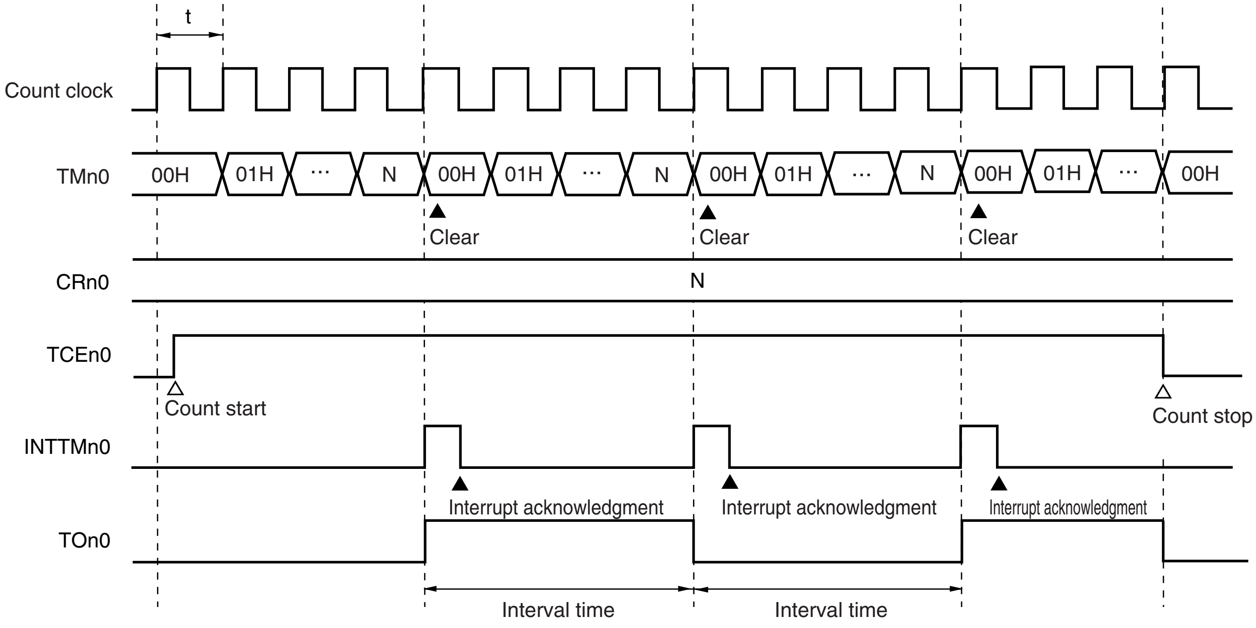
Remark fx: Main system clock oscillation frequency

Table 6-4. Interval Time of Timer 40 (at fx = 5.0 MHz Operation)

TCL402	TCL401	TCL400	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	0	1/fx (0.2 μs)	2^8/fx (51 μs)	1/fx (0.2 μs)
0	0	1	2^2/fx (0.8 μs)	2^{10}/fx (205 μs)	2^2/fx (0.8 μs)
0	1	0	2/fx (0.4 μs)	2^9/fx (102 μs)	2/fx (0.4 μs)
0	1	1	2^2/fx (0.8 μs)	2^{10}/fx (205 μs)	2^2/fx (0.8 μs)
1	0	0	2^3/fx (1.6 μs)	2^{11}/fx (410 μs)	2^3/fx (1.6 μs)
1	0	1	2^4/fx (3.2 μs)	2^{12}/fx (819 μs)	2^4/fx (3.2 μs)

Remark fx: Main system clock oscillation frequency

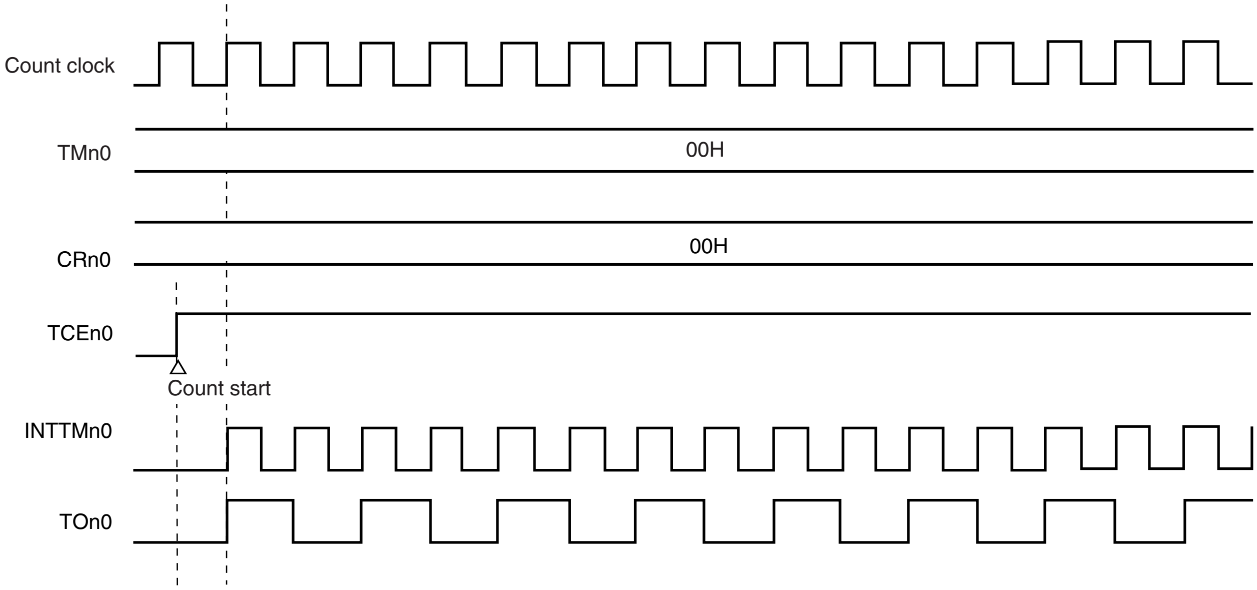
Figure 6-8. Timing of Interval Timer Operation with 8-Bit Resolution (Basic Operation)



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

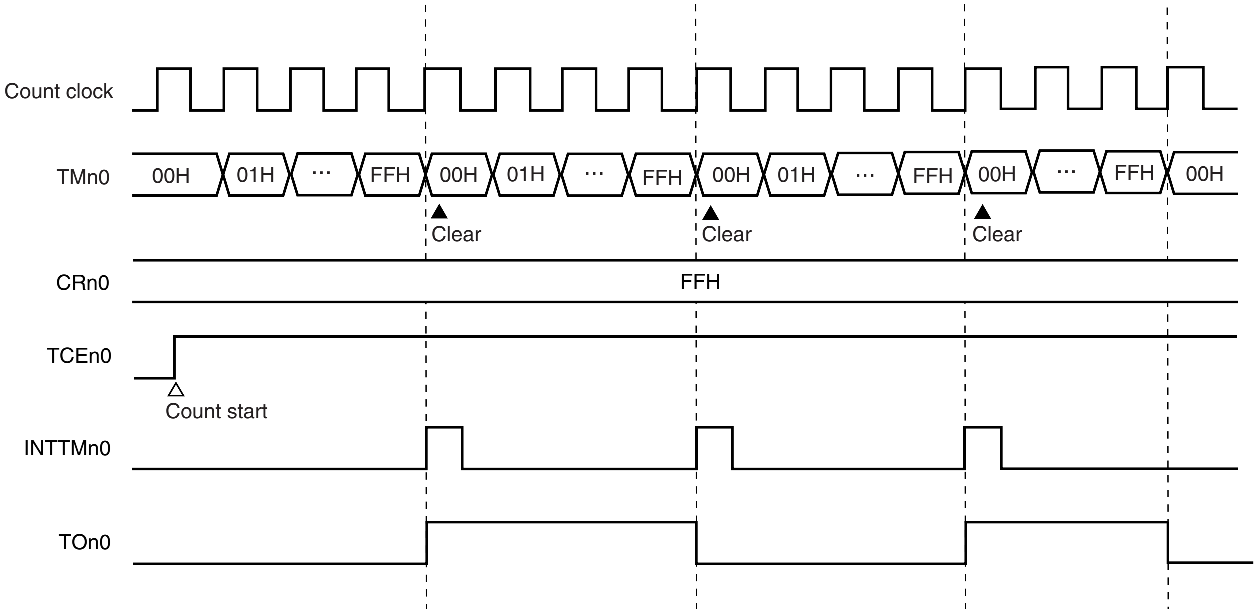
2. n = 3, 4

Figure 6-9. Timing of Interval Timer Operation with 8-Bit Resolution (When CRn0 Is Set to 00H)



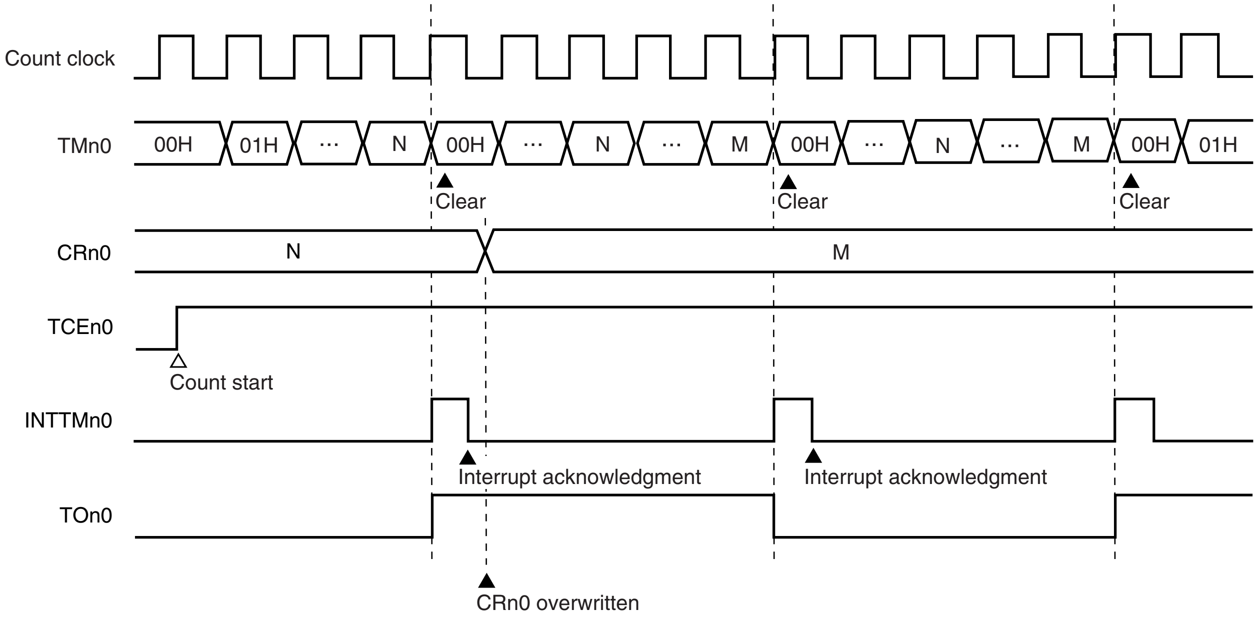
Remark n = 3, 4

Figure 6-10. Timing of Interval Timer Operation with 8-Bit Resolution (When CRn0 Is Set to FFH)



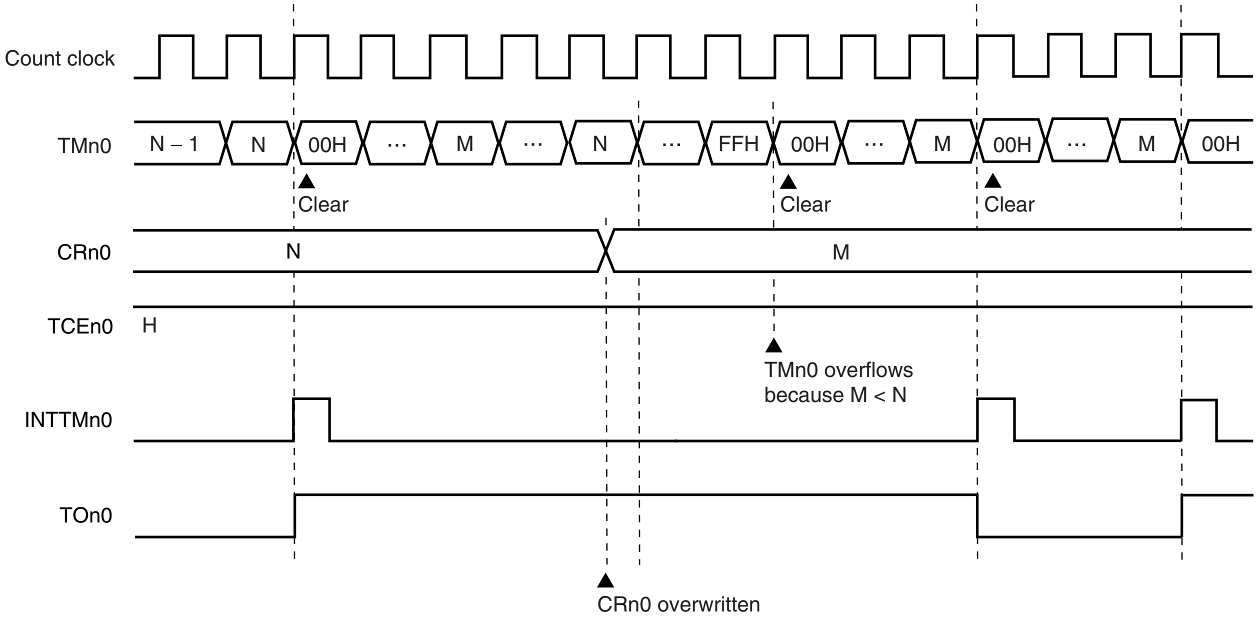
Remark n = 3, 4

Figure 6-11. Timing of Interval Timer Operation with 8-Bit Resolution (When CRn0 Changes from N to M (N < M))



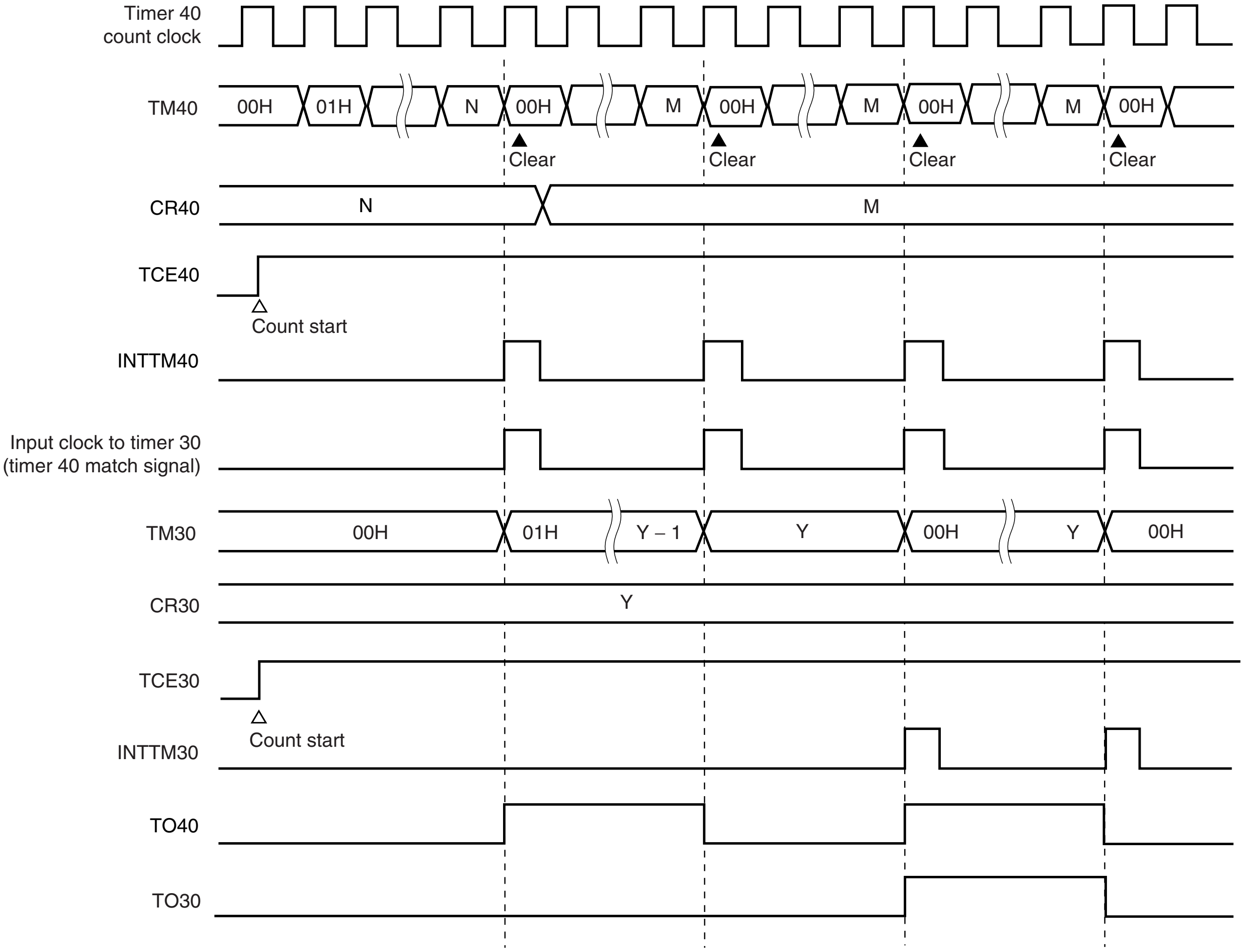
Remark n = 3, 4

Figure 6-12. Timing of Interval Timer Operation with 8-Bit Resolution (When CRn0 Changes from N to M (N > M))



Remark n = 3, 4

Figure 6-13. Timing of Interval Timer Operation with 8-Bit Resolution (When Timer 40 Match Signal Is Selected for Timer 30 Count Clock)



Remark n = 3, 4

(2) Operation as square-wave output with 8-bit resolution (timer 40 only)

Square waves of any frequency can be output at an interval specified by the value preset in 8-bit compare register 40 (CR40).

To operate timer 40 for square-wave output, settings must be made in the following sequence.

<1> Set P60 to output mode (PM60 = 0).
<2> Set the output latch of P60 to 0.
<3> Disable operation of timer counter 40 (TM40) (TCE40 = 0).
<4> Set a count clock for timer 40 and enable output of TO40 (TOE40 = 1).
<5> Set a count value in CR40.
<6> Enable the operation of TM40 (TCE40 = 1).

When the count value of TM40 matches the value set in CR40, the TO40 pin output will be inverted. Through application of this mechanism, square waves of any frequency can be output. As soon as a match occurs, TM40 is cleared to 00H and continues counting. At the same time, an interrupt request signal (INTTM40) is generated.

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

Table 6-5 shows the square-wave output range, and Figure 6-14 shows the timing of square-wave output.

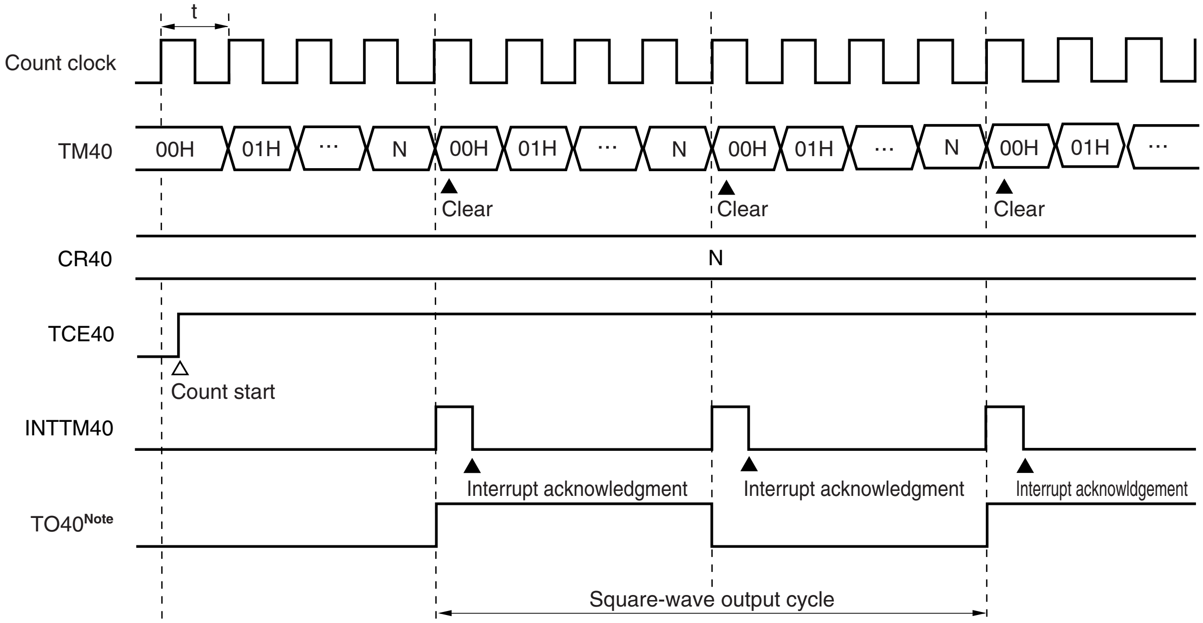
Caution Be sure to stop the timer operation before overwriting the count clock with different data.

Table 6-5. Square-Wave Output Range of Timer 40 (at fx = 5.0 MHz Operation)

TCL402	TCL401	TCL400	Minimum Pulse Width	Maximum Pulse Width	Resolution
0	0	0	1/fx (0.2 μs)	2^{8}/fx (51 μs)	1/fx (0.2 μs)
0	0	1	2^{2}/fx (0.8 μs)	2^{10}/fx (205 μs)	2^{2}/fx (0.8 μs)
0	1	0	2/fx (0.4 μs)	2^{9}/fx (102 μs)	2/fx (0.4 μs)
0	1	1	2^{2}/fx (0.8 μs)	2^{10}/fx (205 μs)	2^{2}/fx (0.8 μs)
1	0	0	2^{3}/fx (1.6 μs)	2^{11}/fx (410 μs)	2^{3}/fx (1.6 μs)
1	0	1	2^{4}/fx (3.2 μs)	2^{12}/fx (819 μs)	2^{4}/fx (3.2 μs)

Remark fx: Main system clock oscillation frequency

Figure 6-14. Timing of Square-Wave Output with 8-Bit Resolution



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

Remark Square-wave output cycle = 2 (N + 1) × t: N = 00H to FFH

6.4.2 Operation as 16-bit timer counter

Timers 30 and 40 can be used as 16-bit timer counters via a cascade connection. In this case, 8-bit timer counter 30 (TM30) is the higher 8 bits and 8-bit timer counter 40 (TM40) is the lower 8 bits. 8-bit timer 40 controls reset and clear.

The following modes can be used for the 16-bit timer counter.

- Interval timer with 16-bit resolution
- Square-wave output with 16-bit resolution

(1) Operation as interval timer with 16-bit resolution

The interval timer with 16-bit resolution repeatedly generates an interrupt at a time interval specified by the count value preset in 8-bit compare register 30 (CR30) and 8-bit compare register 40 (CR40).

To operate as an interval timer with 16-bit resolution, settings must be made in the following sequence.

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

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

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

Table 6-6 shows interval time, and Figure 6-15 shows the timing of the interval timer operation.

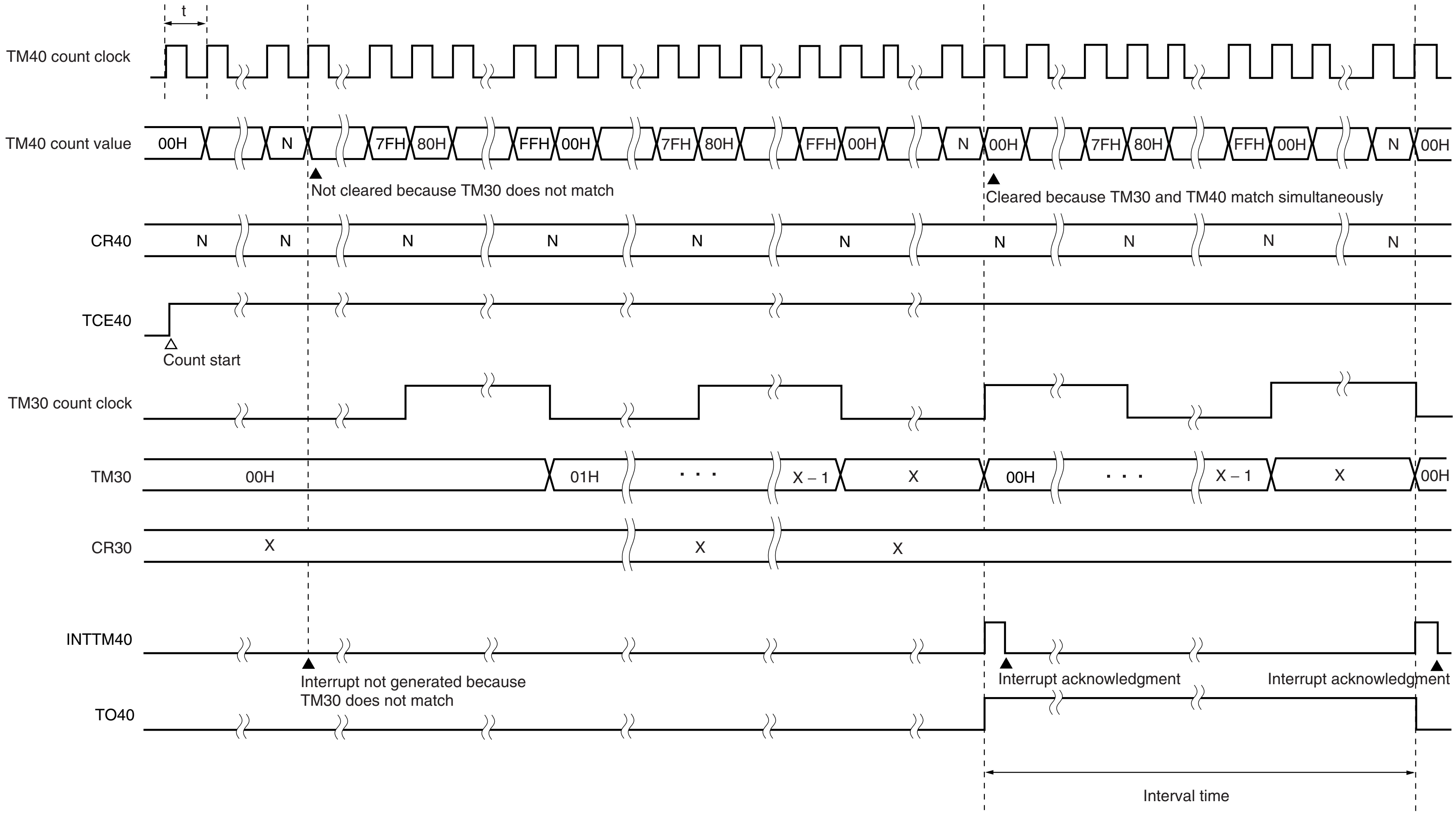
Caution Be sure to stop the timer operation before overwriting the count clock with different data.

Table 6-6. Interval Time with 16-Bit Resolution (at fx = 5.0 MHz Operation)

TCL402	TCL401	TCL400	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	0	1/fx (0.2 μs)	2^{16}/fx (13.1 ms)	1/fx (0.2 μs)
0	0	1	2^{2}/fx (0.8 μs)	2^{18}/fx (52.4 ms)	2^{2}/fx (0.8 μs)
0	1	0	2/fx (0.4 μs)	2^{17}/fx (26.2 ms)	2/fx (0.4 μs)
0	1	1	2^{2}/fx (0.8 μs)	2^{18}/fx (52.4 ms)	2^{2}/fx (0.8 μs)
1	0	0	2^{3}/fx (1.6 μs)	2^{19}/fx (105 ms)	2^{3}/fx (1.6 μs)
1	0	1	2^{4}/fx (3.2 μs)	2^{20}/fx (210 ms)	2^{4}/fx (3.2 μs)

Remark fx: Main system clock oscillation frequency

Figure 6-15. Timing of Interval Timer Operation with 16-Bit Resolution



Remark Interval time = (256X + N + 1) × t: X = 00H to FFH, N = 00H to FFH

(2) Operation as square-wave output with 16-bit resolution

Square waves of any frequency can be output at an interval specified by the count value preset in CR30 and CR40.

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

<1> Disable operation of TM30 and TM40 (TCE30 = 0, TCE40 = 0).
<2> Disable output of TO40 (TOE40 = 0).
<3> Set a count clock for timer 40.
<4> Set P60 to output mode (PM60 = 0) and P60 output latch to 0 and enable TO40 output (TOE40 = 1).
<5> Set count values in CR30 and CR40.
<6> Enable the operation of TM40 (TCE40 = 1[Note]).

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

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

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

Table 6-7 shows the square wave-output range, and Figure 6-16 shows timing of square-wave output.

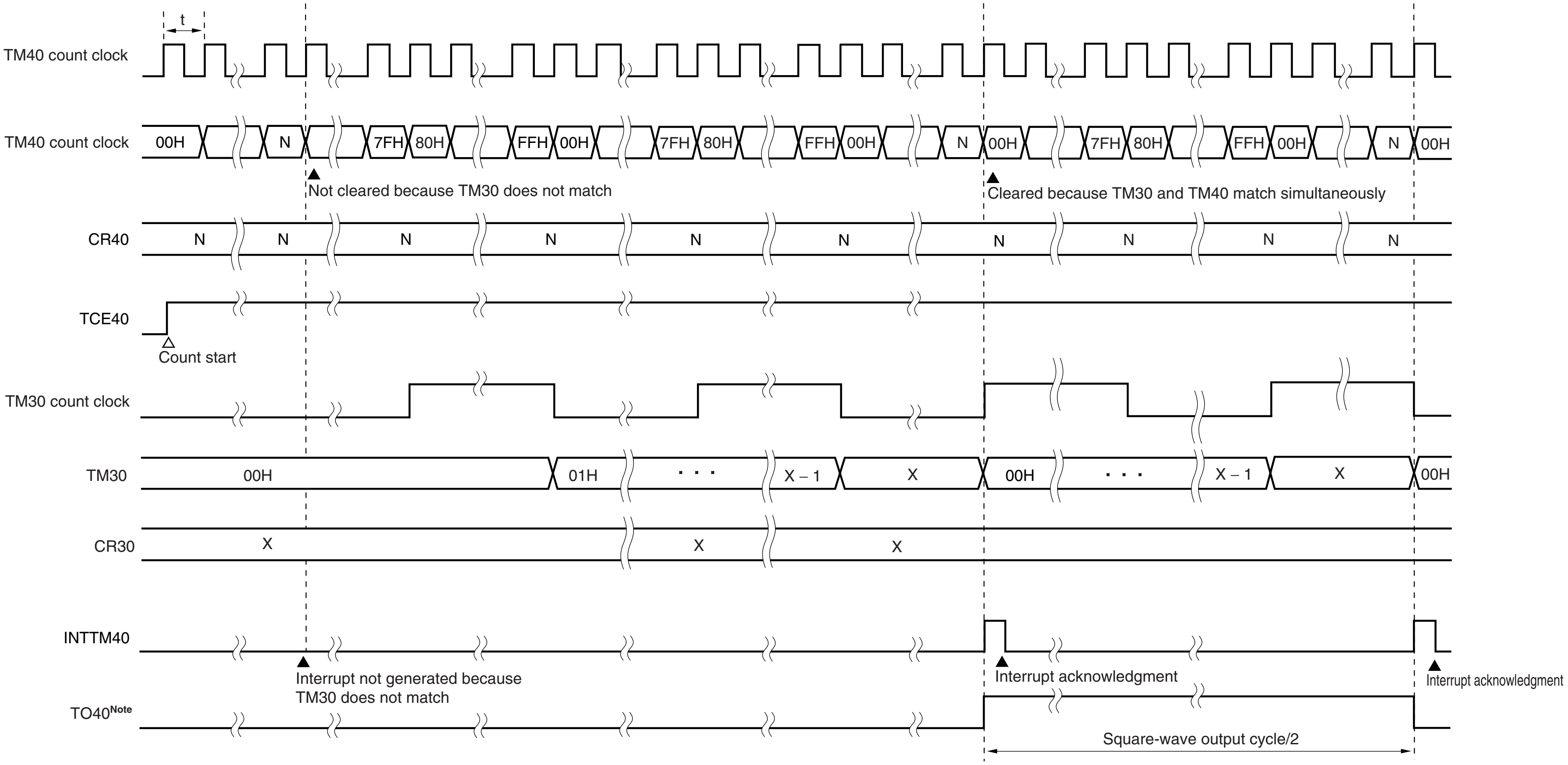
Caution Be sure to stop the timer operation before overwriting the count clock with different data.

Table 6-7. Square-Wave Output Range with 16-Bit Resolution (at fx = 5.0 MHz Operation)

TCL402	TCL401	TCL400	Minimum Pulse Width	Maximum Pulse Width	Resolution
0	0	0	1/fx (0.2 μs)	2^{16}/fx (13.1 ms)	1/fx (0.2 μs)
0	0	1	2^{2}/fx (0.8 μs)	2^{18}/fx (52.4 ms)	2^{2}/fx (0.8 μs)
0	1	0	2/fx (0.4 μs)	2^{17}/fx (26.2 ms)	2/fx (0.4 μs)
0	1	1	2^{2}/fx (0.8 μs)	2^{18}/fx (52.4 ms)	2^{2}/fx (0.8 μs)
1	0	0	2^{3}/fx (1.6 μs)	2^{19}/fx (105 ms)	2^{3}/fx (1.6 μs)
1	0	1	2^{4}/fx (3.2 μs)	2^{20}/fx (210 ms)	2^{4}/fx (3.2 μs)

Remark fx: Main system clock oscillation frequency

Figure 6-16. Timing of Square-Wave Output with 16-Bit Resolution



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

Remark Square-wave output cycle = 2 (256X + N + 1) × t: X = 00H to FFH, N = 00H to FFH

6.4.3 Operation as carrier generator

An arbitrary carrier clock generated by TM40 can be output in the cycle set in TM30.

To operate timers 30 and 40 as carrier generators, settings must be made in the following sequence.

<1> Disable operation of TM30 and TM40 (TCE30 = 0, TCE40 = 0).

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

<3> Set count values in CR30, CR40, and CRH40.

<4> Set the operation mode of timer 30 and timer 40 to carrier generator mode (see **Figures 6-4** and **6-5**).

<5> Set the count clock for timer 30 and timer 40.

<6> Set remote control output to carrier pulse (RMC40 (bit 2 of carrier generator output control register 40 (TCA40)) = 0).
Input the required value to NRZB40 (bit 1 of TCA40) by program.
Input a value to NRZ40 (bit 0 of TCA40) before it is reloaded from NRZB40.

<7> Set P60 to output mode (PM60 = 0) and the P60 output latch to 0 and enable TO40 output by setting TOE40 to 1.

<8> Enable the operation of TM30 and TM40 (TCE30 = 1, TCE40 = 1).

★ <9> Save the NRZB40 value to a general-purpose register.

★ <10> When INTTM30 rises, the NRZB40 value is transferred to NRZ40. After that, rewrite TCA40 using an 8-bit memory manipulation instruction. Input the value to be transferred next to NRZ40 to NRZB40, and input the value saved in step <9> to NRZ40.

★ <11> Generate the desired carrier signal by repeating steps <9> and <10>.

The operation of the carrier generator is as follows.

<1> When the count the value of TM40 matches the value set in CR40, an interrupt request signal (INTTM40) is generated and the output of timer 40 is inverted, which makes the compare register switch from CR40 to CRH40.

<2> After that, when the count the value of TM40 matches the value set in CRH40, an interrupt request signal (INTTM40) is generated and the output of timer 40 is inverted again, which makes the compare register switch from CRH40 to CR40.

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

<4> When the count value of TM30 matches the value set in CR30, an interrupt request signal (INTTM30) is generated. The rising edge of INTTM30 is the data reload signal of NRZB40 and is transferred to NRZ40.

<5> When NRZ40 is 1, a carrier clock is output from TO40 pin.

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

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

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

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

Figures 6-17 to 6-19 show the operation timing of the carrier generator.

Figure 6-17. Timing of Carrier Generator Operation (When CR40 = N, CRH40 = M (M > N))

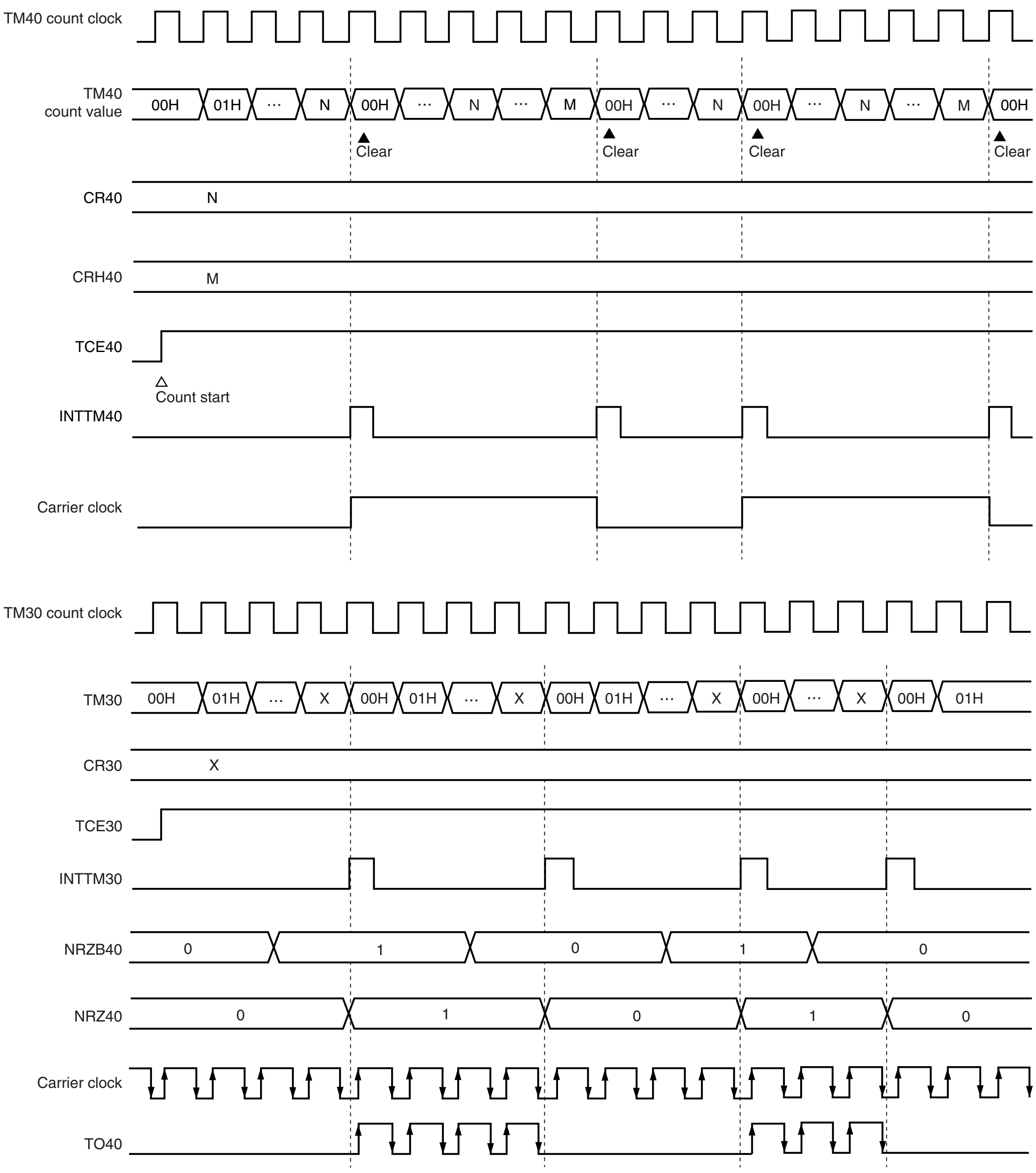
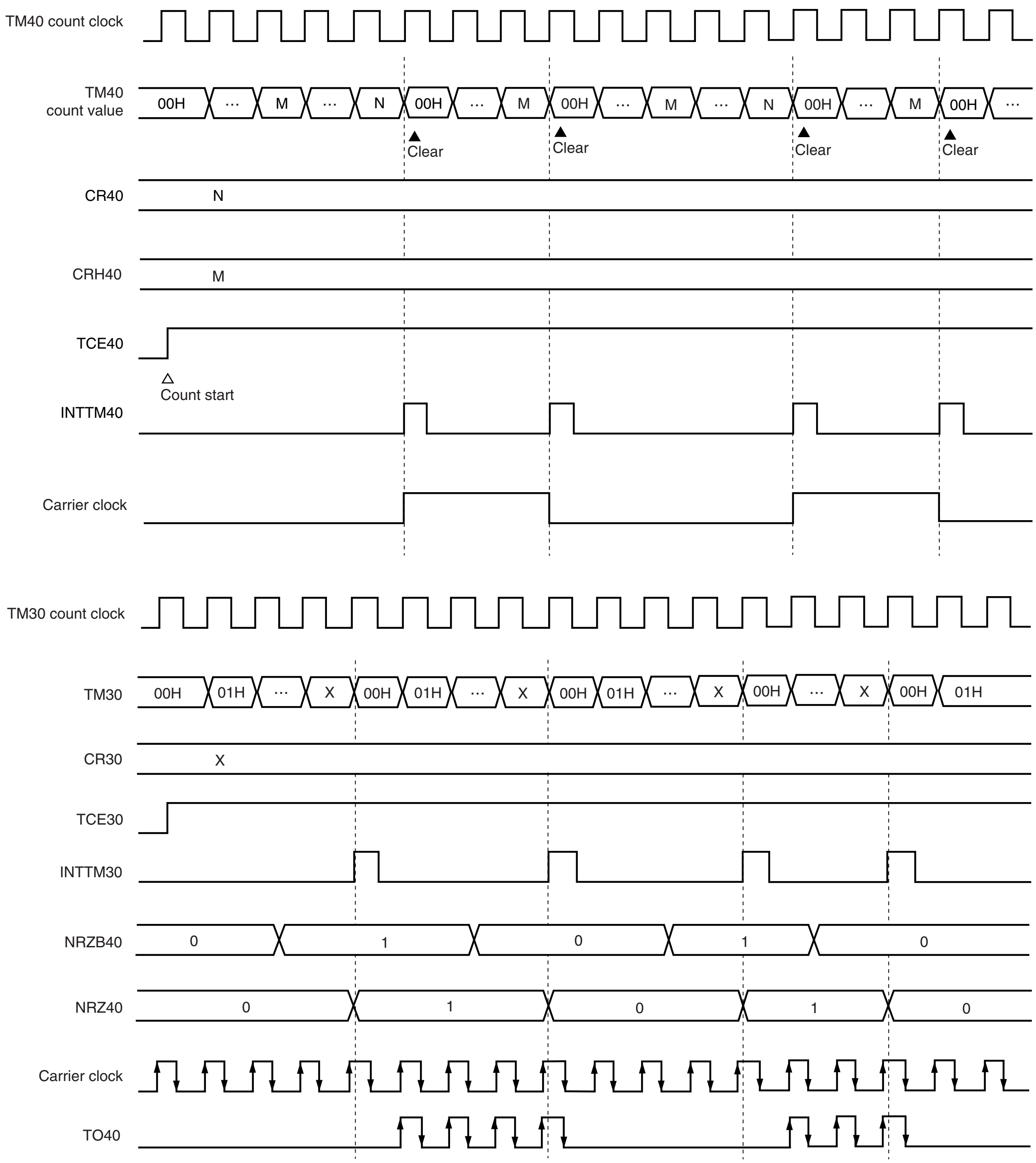
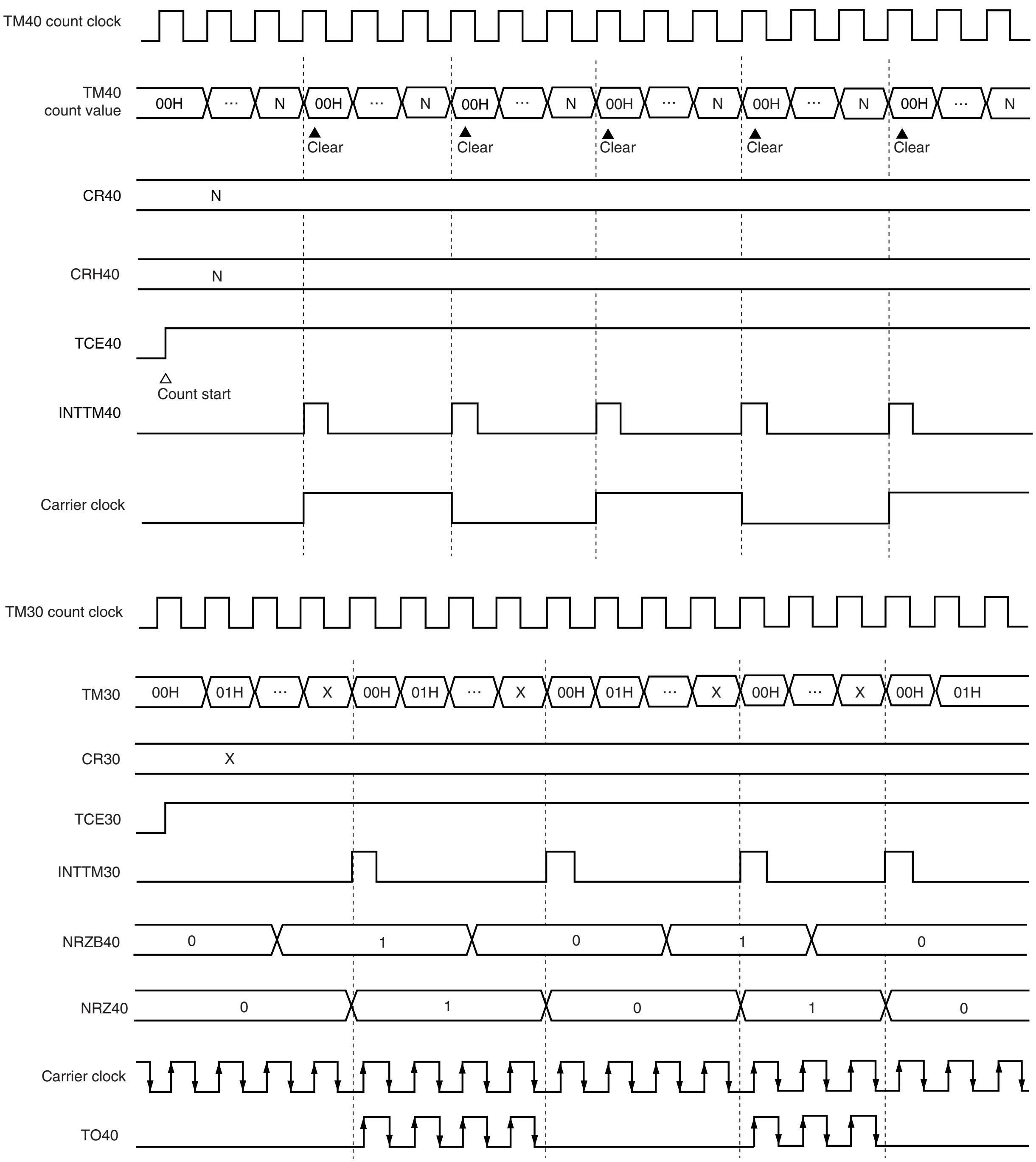


Figure 6-18. Timing of Carrier Generator Operation (When CR40 = N, CRH40 = M (M < N))



Remark This figure shows an example of when the NRZ40 value is changed while the carrier clock is high level.

Figure 6-19. Timing of Carrier Generator Operation (When CR40 = CRH40 = N)



6.4.4 Operation as PWM output (timer 40 only)

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

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

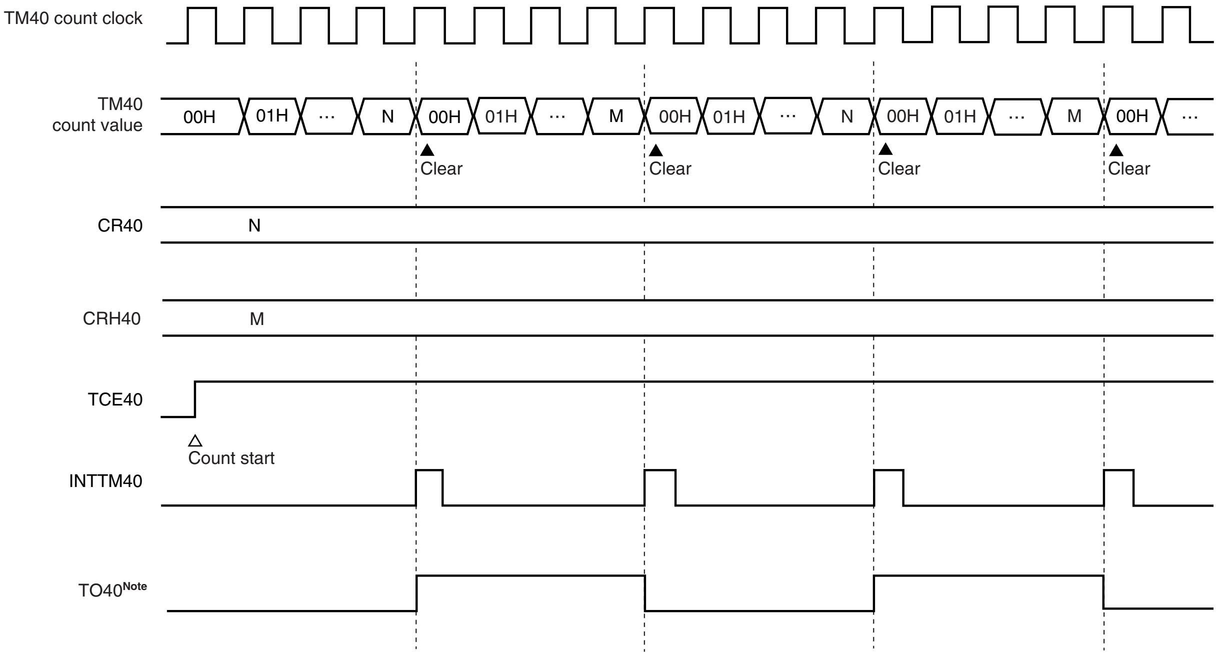
<1> Disable operation of TM40 (TCE40 = 0).
<2> Disable timer output of TO40 (TOE40 = 0).
<3> Set count values in CR40 and CRH40.
<4> Set the operation mode of timer 40 to carrier generator mode (see **Figure 6-5**).
<5> Set the count clock for timer 40.
<6> Set P60 to output mode (PM60 = 0) and the P60 output latch to 0 and enable timer output of TO40 (TOE40 = 1).
<7> Enable the operation of TM40 (TCE40 = 1).

The operation in the PWM output mode is as follows.

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

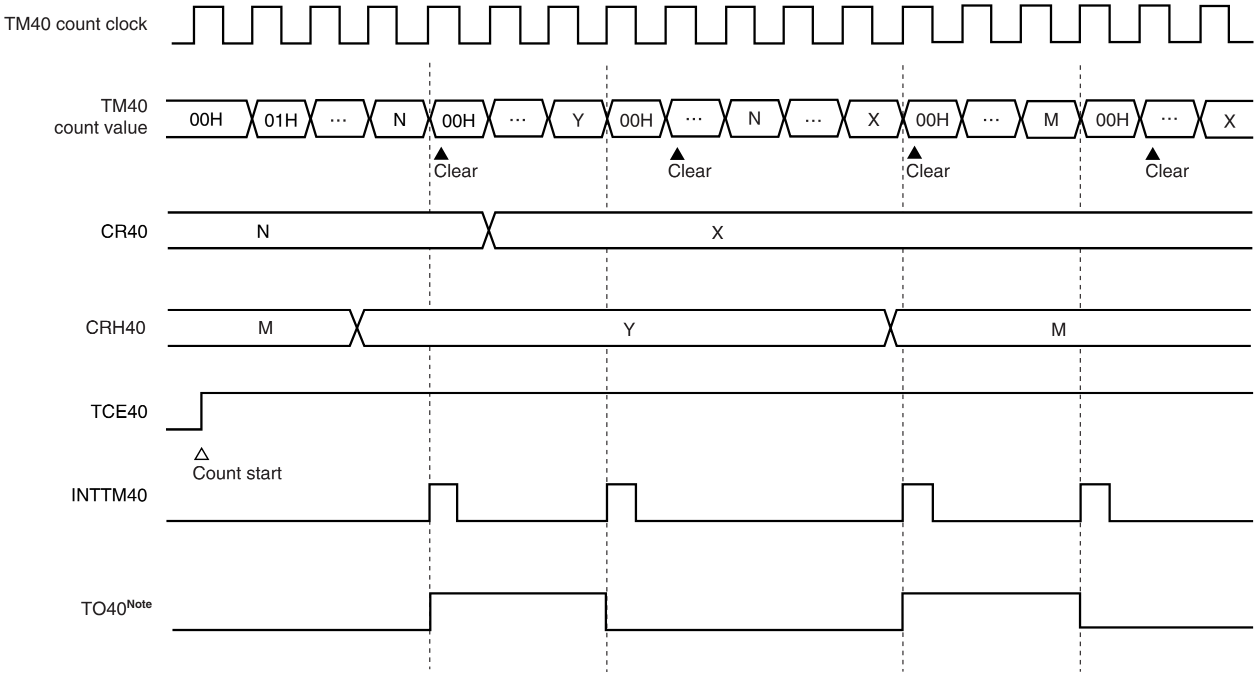
A pulse of any duty ratio is output by repeating <1> to <4> above. Figures 6-20 and 6-21 show the operation timing in the PWM output mode.

Figure 6-20. PWM Output Mode Timing (Basic Operation)



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

Figure 6-21. PWM Output Mode Timing (When CR40 and CRH40 Are Overwritten)



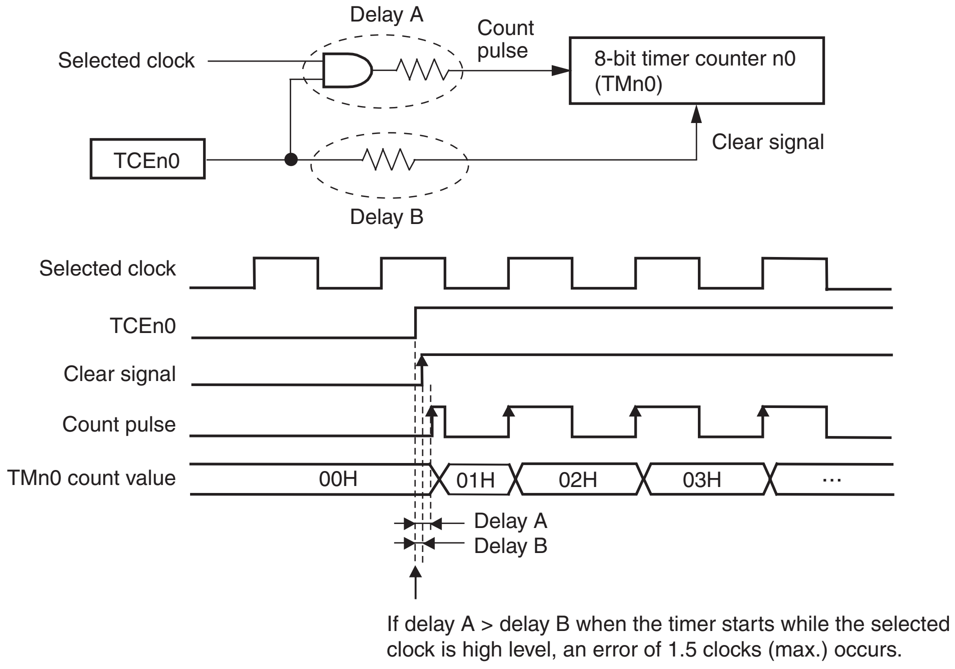
Note The initial value of TO40 is low level when output is enabled (TOE40 = 1).

6.5 Notes on Using 8-Bit Timers 30 and 40

★ **(1) Error on starting timer**

An error of up to 1.5 clocks is included in the time between when the timer is started and a match signal is generated. This is because the counter may be incremented by detecting a rising edge at the timing at which the timer starts while the count clock is high level (see **Figure 6-22**).

Figure 6-22. Case in Which Error of 1.5 Clocks (Max.) Occurs



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

Remark n = 3, 4

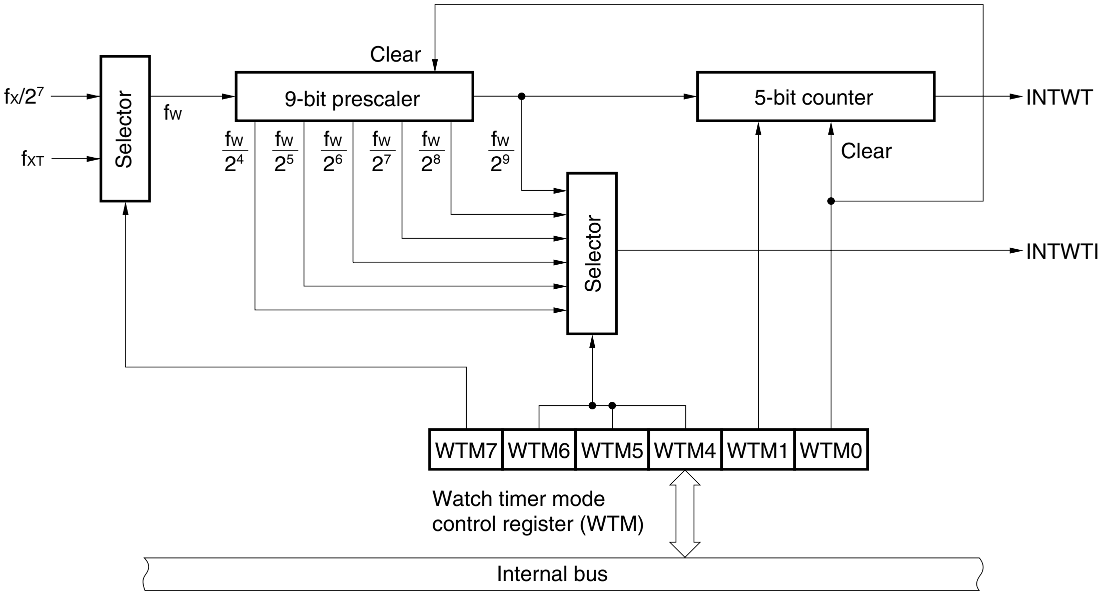
7.1 Watch Timer Functions

The watch timer has the following functions.

- Watch timer
- Interval timer

The watch and interval timers can be used at the same time.
Figure 7-1 shows a block diagram of the watch timer.

Figure 7-1. Block Diagram of Watch Timer



(1) Watch timer

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

Caution When the main system clock is operating at 5.0 MHz, it cannot be used to generate a 0.5-second interval. In this case, the subsystem clock, which operates at 32.768 kHz, should be used instead.

(2) Interval timer

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

Table 7-1. Interval Time of Interval Timer

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

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

2. f_X: Main system clock oscillation frequency

3. f_{XT}: Subsystem clock oscillation frequency

7.2 Watch Timer Configuration

The watch timer includes the following hardware.

Table 7-2. Configuration of Watch Timer

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

7.3 Register Controlling Watch Timer

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

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

Figure 7-2. Format of Watch Timer Mode Control Register

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

WTM7	Watch timer count clock (fW) selection
0	$f_X/2^7$ (39.1 kHz)
1	f_{XT} (32.768 kHz)

WTM6	WTM5	WTM4	Prescaler interval selection
0	0	0	$2^4/f_W$
0	0	1	$2^5/f_W$
0	1	0	$2^6/f_W$
0	1	1	$2^7/f_W$
1	0	0	$2^8/f_W$
1	0	1	$2^9/f_W$
Other than above			Setting prohibited

WTM1	Control of 5-bit counter operation
0	Cleared after stop
1	Started

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

Caution Be sure to clear bits 2 and 3 to 0.

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

2. f_X: Main system clock oscillation frequency

3. f_{XT}: Subsystem clock oscillation frequency

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

7.4 Watch Timer Operation

7.4.1 Operation as watch timer

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

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

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

7.4.2 Operation as interval timer

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

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

Table 7-3. Interval Time of Interval Timer

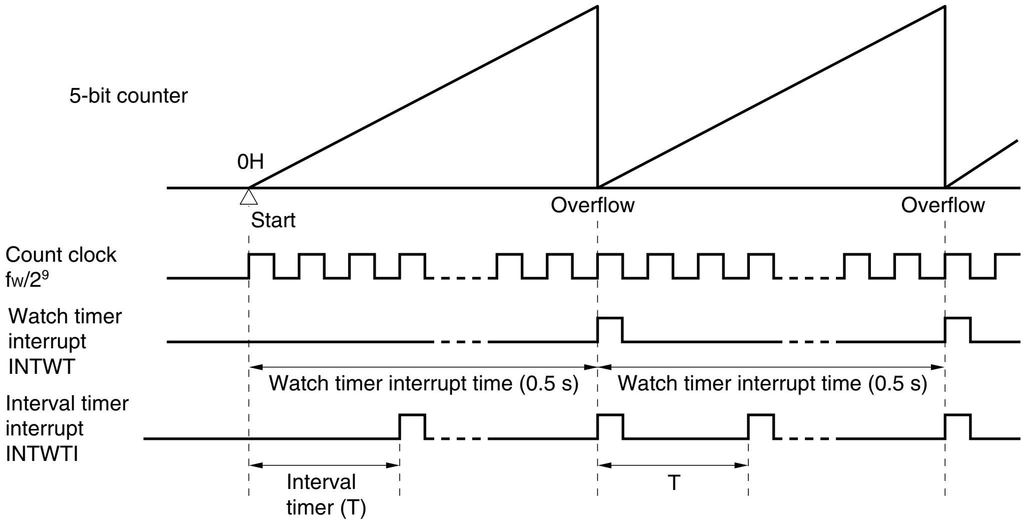
Interval	At f_X = 5.0 MHz Operation	At f_X = 4.19 MHz Operation	At f_{XT} = 32.768 kHz Operation
$2^4 \times 1/f_W$	409.6 μs	488 μs	488 μs
$2^5 \times 1/f_W$	819.2 μs	977 μs	977 μs
$2^6 \times 1/f_W$	1.64 ms	1.95 ms	1.95 ms
$2^7 \times 1/f_W$	3.28 ms	3.91 ms	3.91 ms
$2^8 \times 1/f_W$	6.55 ms	7.81 ms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

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

2. f_X: Main system clock oscillation frequency

3. f_{XT}: Subsystem clock oscillation frequency

Figure 7-3. Watch Timer/Interval Timer Operation Timing



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

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

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

CHAPTER 8 WATCHDOG TIMER

8.1 Watchdog Timer Functions

The watchdog timer has the following functions.

- Watchdog timer
- Interval timer

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

(1) Watchdog timer

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

Table 8-1. Program Loop Detection Time of Watchdog Timer

Program Loop Detection Time	At fx = 5.0 MHz Operation
2^{11} × 1/fx	410 μs
2^{13} × 1/fx	1.64 ms
2^{15} × 1/fx	6.55 ms
2^{17} × 1/fx	26.2 ms

Remark fx: Main system clock oscillation frequency

(2) Interval timer

The interval timer generates an interrupt at preset intervals.

Table 8-2. Interval Time of Watchdog Timer

Interval Time	At fx = 5.0 MHz Operation
2^{11} × 1/fx	410 μs
2^{13} × 1/fx	1.64 ms
2^{15} × 1/fx	6.55 ms
2^{17} × 1/fx	26.2 ms

Remark fx: Main system clock oscillation frequency

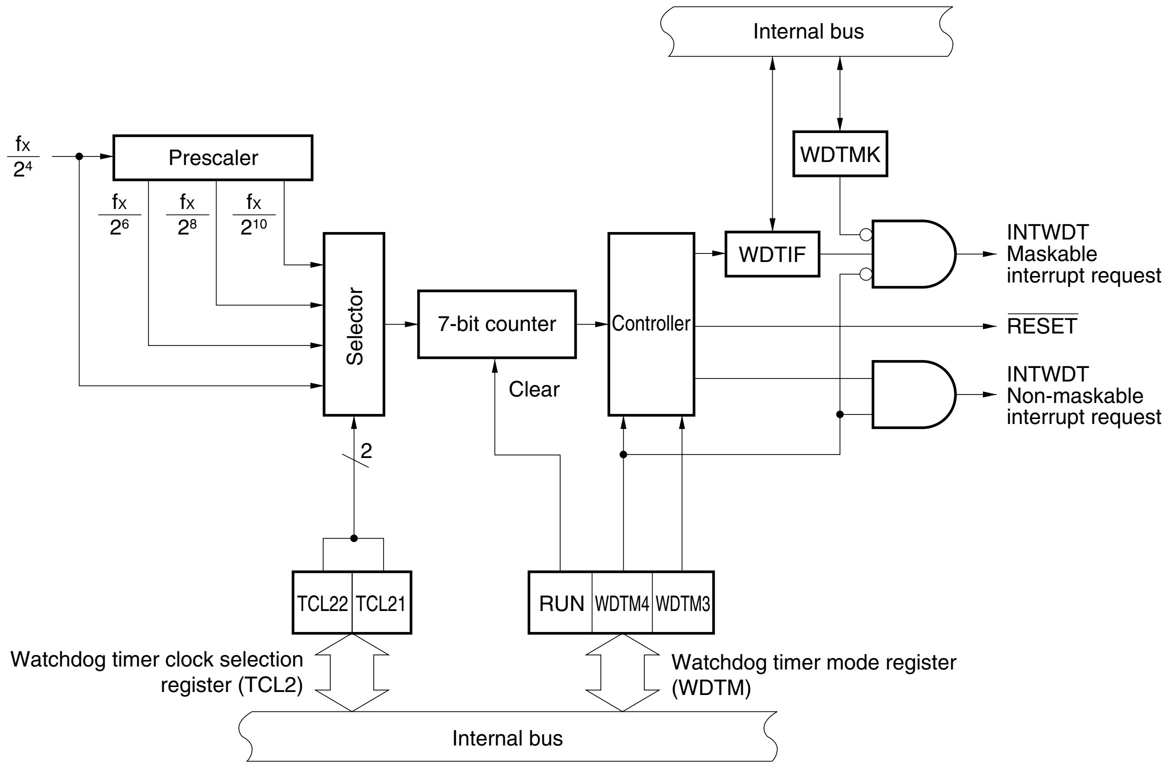
8.2 Watchdog Timer Configuration

The watchdog timer includes the following hardware.

Table 8-3. Configuration of Watchdog Timer

Item	Configuration
Control registers	Watchdog timer clock selection register (TCL2) Watchdog timer mode register (WDTM)

Figure 8-1. Block Diagram of Watchdog Timer



8.3 Registers Controlling Watchdog Timer

The watchdog timer is controlled by the following two registers.

- Watchdog timer clock selection register (TCL2)
- Watchdog timer mode register (WDTM)

(1) Watchdog timer clock selection register (TCL2)

This register sets the watchdog timer count clock.
TCL2 is set with an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets TCL2 to 00H.

Figure 8-2. Format of Watchdog Timer Clock Selection Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TCL2	0	0	0	0	0	TCL22	TCL21	0	FF42H	00H	R/W

TCL22	TCL21	Watchdog timer count clock selection	Program loop detection or interval time
0	0	fx/2^{4} (313 kHz)	2^{11}/fx (410 μs)
0	1	fx/2^{6} (78.1 kHz)	2^{13}/fx (1.64 ms)
1	0	fx/2^{8} (19.5 kHz)	2^{15}/fx (6.55 ms)
1	1	fx/2^{10} (4.88 kHz)	2^{17}/fx (26.2 ms)

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

Remarks 1. fx: Main system clock oscillation frequency

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

(2) Watchdog timer mode register (WDTM)

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

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

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

Figure 8-3. Format of Watchdog Timer Mode Register

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

RUN	Selection of operation of watchdog timer[Note 1]
0	Stop counting
1	Clear counter and start counting

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

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

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

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

Cautions 1. When the watchdog timer is cleared by setting RUN to 1, the actual overflow time is up to 0.8% shorter than the time set by the watchdog timer clock selection register (TCL2).

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

8.4 Watchdog Timer Operation

8.4.1 Operation as watchdog timer

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

The count clock (program loop detection time interval) of the watchdog timer can be selected by bits 1 and 2 (TCL21 and TCL22) of the watchdog timer clock selection register (TCL2). By setting bit 7 (RUN) of WDTM to 1, the watchdog timer is started. Set RUN to 1 within the set program loop detection time interval after the watchdog timer has been started. By setting RUN to 1, the watchdog timer can be cleared and start counting. If RUN is not set to 1, and the program loop detection time is exceeded, the system is reset or a non-maskable interrupt is generated according to the value of bit 3 (WDTM3) of WDTM.

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

Cautions 1. The actual program loop detection time may be up to 0.8% shorter than the set time.

2. When the subsystem clock is selected as the CPU clock, the watchdog timer stops counting. In this case, therefore, the watchdog timer stops operating even though the main system clock is oscillating.

Table 8-4. Program Loop Detection Time of Watchdog Timer

TCL22	TCL21	Program Loop Detection Time	At f_X = 5.0 MHz Operation
0	0	$2^{11} \times 1/f_X$	410 μs
0	1	$2^{13} \times 1/f_X$	1.64 ms
1	0	$2^{15} \times 1/f_X$	6.55 ms
1	1	$2^{17} \times 1/f_X$	26.2 ms

Remark f_X: Main system clock oscillation frequency

8.4.2 Operation as interval timer

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

Select a count clock (or interval time) by setting bits 1 and 2 (TCL21 and TCL22) of the watchdog timer clock selection register (TCL2). The watchdog timer starts operating as an interval timer when the RUN bit (bit 7 of WDTM) is set to 1.

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

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

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

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

Table 8-5. Interval Time of Watchdog Timer

TCL22	TCL21	Interval Time	At fX = 5.0 MHz Operation
0	0	$2^{11} \times 1/f_X$	410 μs
0	1	$2^{13} \times 1/f_X$	1.64 ms
1	0	$2^{15} \times 1/f_X$	6.55 ms
1	1	$2^{17} \times 1/f_X$	26.2 ms

Remark fX: Main system clock oscillation frequency

CHAPTER 9 8-BIT A/D CONVERTER

9.1 Functions of 8-Bit A/D Converter

The 8-bit A/D converter is an 8-bit resolution converter that converts analog inputs to digital signals. This converter can control one channel of analog inputs (ANI0).

A/D conversion can only be started by software.

A/D conversion is performed repeatedly, with an interrupt request (INTAD0) being issued each time an A/D conversion operation is completed.

Caution The A/D converter stops operating in STOP mode.

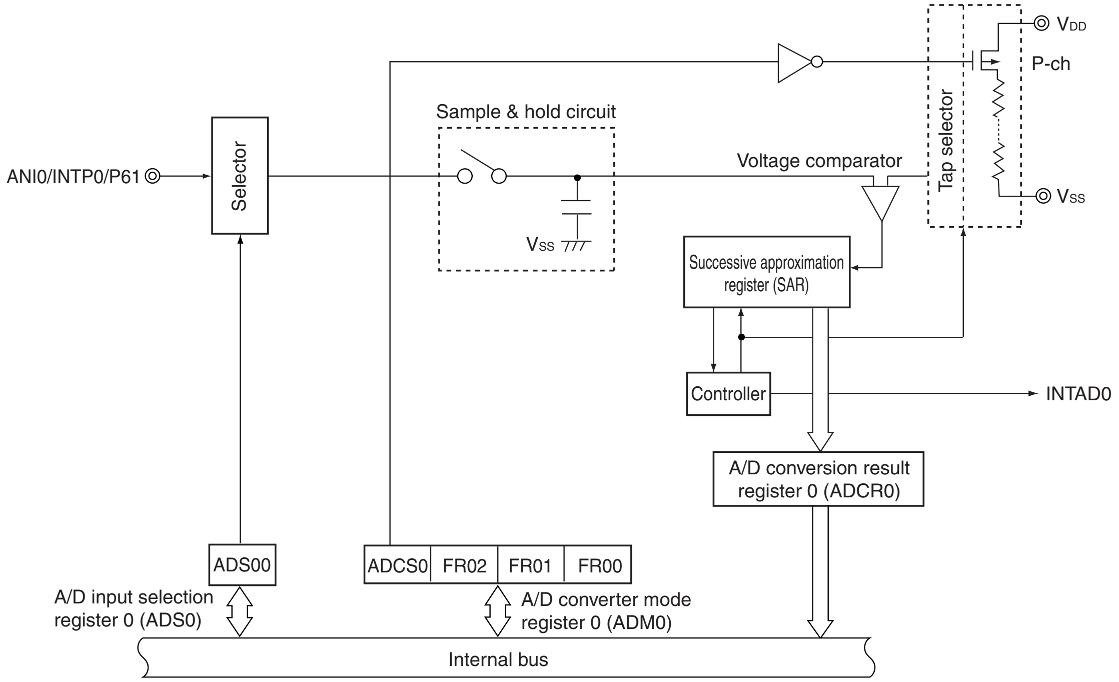
9.2 Configuration of 8-Bit A/D Converter

The 8-bit A/D converter includes the following hardware.

Table 9-1. Configuration of 8-Bit A/D Converter

Item	Configuration
Analog input	1 channel (ANI0)
Registers	Successive approximation register (SAR) A/D conversion result register 0 (ADCR0)
Control registers	A/D converter mode register 0 (ADM0) A/D input selection register 0 (ADS0)

Figure 9-1. Block Diagram of 8-Bit A/D Converter


VDD
P-ch
Sample & hold circuit
Tap selector
ANI0/INTP0/P61
Selector
Voltage comparator
VSS
VSS
Successive approximation register (SAR)
Controller
INTAD0
A/D conversion result register 0 (ADCR0)
ADS00
ADCS0
FR02
FR01
FR00
A/D input selection register 0 (ADS0)
A/D converter mode register 0 (ADM0)
Internal bus

(1) Successive approximation register (SAR)

The SAR receives the result of comparing an analog input voltage and a voltage at the voltage tap (comparison voltage) received from the series resistor string, starting from the most significant bit (MSB).

Upon receiving all the bits, down to the least significant bit (LSB), that is, upon the completion of A/D conversion, the SAR sends its contents to A/D conversion result register 0 (ADCR0).

(2) A/D conversion result register 0 (ADCR0)

ADCR0 holds the result of A/D conversion. Each time A/D conversion ends, the conversion result received from the successive approximation register is loaded into ADCR0, which is an 8-bit register that holds the result of A/D conversion.

ADCR0 can be read with an 8-bit memory manipulation instruction.

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

(3) Sample & hold circuit

The sample & hold circuit samples consecutive analog inputs from the input circuit, one by one, and sends them to the voltage comparator. The sampled analog input voltage is held during A/D conversion.

(4) Voltage comparator

The voltage comparator compares an analog input with the voltage output by the series resistor string.

(5) Series resistor string

The series resistor string is configured between V_{DD} and V_{SS}. It generates the reference voltages against which analog inputs are compared.

(6) ANI0 pin

The ANI0 pin is the 1-channel analog input pin for the A/D converter. It is used to receive the analog signals for A/D conversion.

Caution Do not supply the ANI0 pin with a voltage that falls outside the rated range. If a voltage greater than or equal to V_{DD} or less than or equal to V_{SS} (even if within the absolute maximum ratings) is supplied to this pin, the conversion value will be undefined.

9.3 Registers Controlling 8-Bit A/D Converter

The following two registers are used to control the 8-bit A/D converter.

- A/D converter mode register 0 (ADM0)
- A/D input selection register 0 (ADS0)

(1) A/D converter mode register 0 (ADM0)

ADM0 specifies the conversion time for analog inputs. It also specifies whether to enable conversion.
ADM0 is set with a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears ADM0 to 00H.

Figure 9-2. Format of A/D Converter Mode Register 0

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
ADM0	ADCS0	0	FR02	FR01	FR00	0	0	0	FF80H	00H	R/W

ADCS0	A/D conversion control
0	Conversion disabled
1	Conversion enabled

FR02	FR01	FR00	A/D conversion time selection[Note 1]
0	0	0	72/fx (14.4 μs)
0	0	1	60/fx (setting prohibited[Note 2])
0	1	0	48/fx (setting prohibited[Note 2])
1	0	0	144/fx (28.8 μs)
1	0	1	120/fx (24 μs)
1	1	0	96/fx (19.2 μs)
Other than above			Setting prohibited

Notes 1. The specifications of FR02, FR01, and FR00 must be such that the A/D conversion time is at least 14 μs.

2. These bit combinations must not be used when fx = 5.0 MHz, as the A/D conversion time will fall below 14 μs.

Cautions 1. The result of conversion performed immediately after bit 7 (ADCS0) is set is undefined. Use the second or later result.

2. The result of conversion performed after ADCS0 is cleared may become undefined. When reading the result of conversion, read it during A/D conversion. If reading the result of conversion after stopping A/D conversion, stop A/D conversion and then read the result between completion of A/D conversion and when the next A/D conversion starts.

3. Always set bits 0 to 2 and bit 6 to 0.

Remarks 1. fx: Main system clock oscillation frequency

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

(2) A/D input selection register 0 (ADS0)

The ADS0 register specifies the port used to input the analog voltage to be converted to a digital signal.

ADS0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ADS0 to 00H.

Figure 9-3. Format of A/D Input Selection Register 0

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
ADS0	0	0	0	0	0	0	0	ADS00	FF84H	00H	R/W

ADS00	Port function of P61
0	Operates as P61 (general-purpose port pin) or INTP0 (external interrupt pin)
1	Operates as ANI0 (analog input pin). External interrupts are prohibited.

Caution Always set bits 1 to 7 to 0.

9.4 8-Bit A/D Converter Operation

9.4.1 Basic operation of 8-bit A/D converter

<1> Set bit 0 of A/D input selection register 0 (ADS0) so that the P61/INTP0/ANI0 pin can be used as an analog input.

<2> The analog input voltage is sampled using the sample & hold circuit.

<3> After sampling continues for a certain period of time, the sample & hold circuit is put on hold to keep the input analog voltage until A/D conversion is completed.

<4> Bit 7 of the successive approximation register (SAR) is set. The series resistor string voltage tap at the tap selector is set to half V_{DD}.

<5> The series resistor string voltage tap is compared with the analog input voltage using the voltage comparator. If the analog input voltage is higher than half V_{DD}, the MSB of the SAR is left set. If it is lower than half V_{DD}, the MSB is reset.

<6> Bit 6 of the SAR is set automatically, and comparison shifts to the next stage. The next voltage tap of the series resistor string is selected according to bit 7, which reflects the previous comparison result, as follows.

- Bit 7 = 1: Three quarters of V_{DD}
- Bit 7 = 0: One quarter of V_{DD}

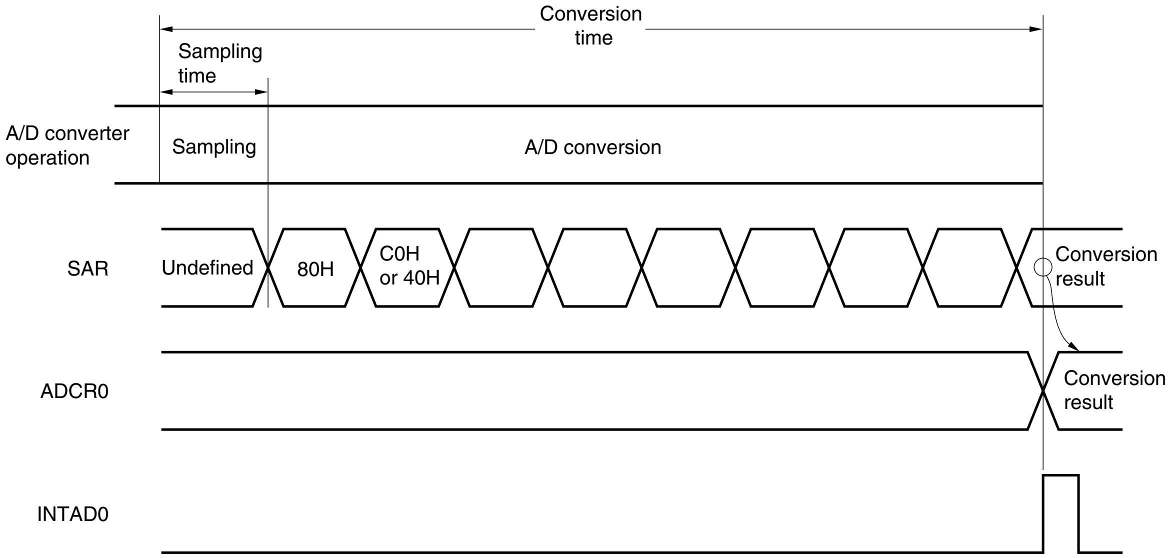
The voltage tap is compared with the analog input voltage. Bit 6 is set or reset according to the result of comparison.

- Analog input voltage ≥ voltage tap: Bit 6 = 1
- Analog input voltage < voltage tap: Bit 6 = 0

<7> Comparison is repeated until bit 0 of the SAR is reached.

<8> When comparison is completed for all of the 8 bits, a significant digital result is left in the SAR. This value is sent to and latched in A/D conversion result register 0 (ADCR0). At the same time, it is possible to generate an A/D conversion end interrupt request (INTAD0).

Figure 9-4. Basic Operation of 8-Bit A/D Converter



A/D conversion continues until bit 7 (ADCS0) of A/D converter mode register 0 (ADM0) is reset (0) by software.

If an attempt is made to write to ADM0 or A/D input selection register 0 (ADS0) during A/D conversion, the A/D conversion in progress is canceled. In this case, if ADCS0 is set (1), A/D conversion is restarted from the beginning.

$\overline{\text{RESET}}$ input makes the A/D conversion result register 0 (ADCR0) undefined.

Cautions 1. The first A/D conversion value immediately after starting the A/D conversion operation is undefined. Use the second or later conversion value.

2. When in standby mode, the A/D converter stops operation.

9.4.2 Input voltage and conversion result

The relationship between the analog input voltage at the analog input pin (ANI0) and the A/D conversion result (A/D conversion result register 0 (ADCR0)) is represented by:

$$\text{ADCR0} = \text{INT}\left(\frac{V_{IN}}{V_{DD}} \times 256 + 0.5\right)$$

or

$$(\text{ADCR0} - 0.5) \times \frac{V_{DD}}{256} \leq V_{IN} < (\text{ADCR0} + 0.5) \times \frac{V_{DD}}{256}$$

INT( ): Function that returns the integer part of the parenthesized value

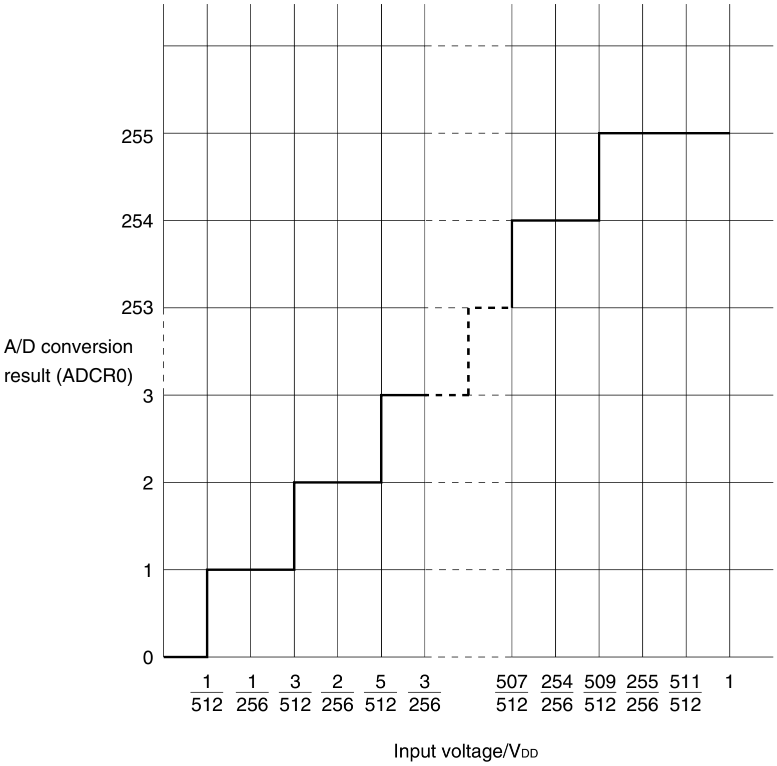
V_{IN}: Analog input voltage

V_{DD}: Supply voltage

ADCR0: Value in the A/D conversion result register 0 (ADCR0)

Figure 9-5 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 9-5. Relationship Between Analog Input Voltage and A/D Conversion Result



9.4.3 Operation mode of 8-bit A/D converter

A/D input selection register 0 (ADS0) is used to select the function of the P61/INTP0/ANI0 pin to be used as an analog input for A/D conversion.

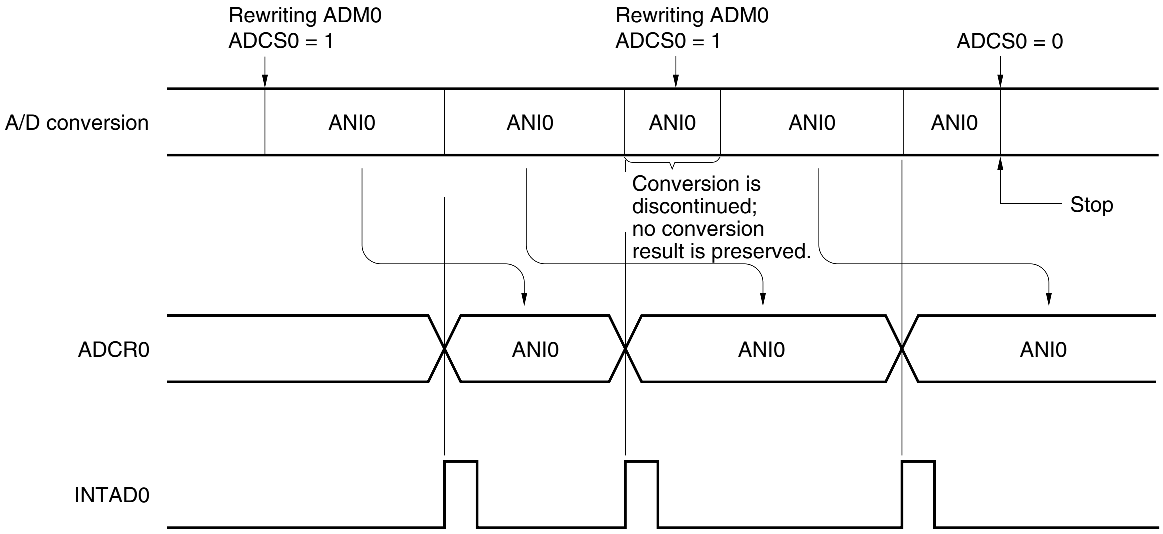
A/D conversion can only be started by software, that is, by setting A/D converter mode register 0 (ADM0).

The A/D conversion result is saved to A/D conversion result register 0 (ADCR0). At the same time, an interrupt request signal (INTAD0) is generated.

- **Software-started A/D conversion**

  Setting bit 7 (ADCS0) of A/D converter mode register 0 (ADM0) triggers A/D conversion for the voltage applied to the analog input pin specified by A/D input selection register 0 (ADS0). Upon completion of A/D conversion, the conversion result is saved to A/D conversion result register 0 (ADCR0). At the same, an interrupt request signal (INTAD0) is generated. Once A/D conversion is started, and completed, another A/D conversion operation is started. A/D conversion is repeated until new data is written to ADM0. If data in which ADCS0 is 1 is written to ADM0 again during A/D conversion, the A/D conversion in progress is discontinued, and an A/D conversion operation begins for the new data. If data in which ADCS0 is 0 is written to the ADM0 again during A/D conversion, A/D conversion is completely stopped.

Figure 9-6. Software-Started A/D Conversion



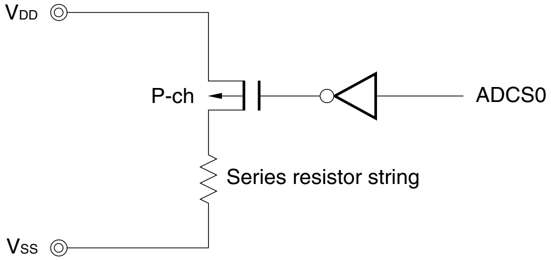
9.5 Cautions Related to 8-Bit A/D Converter

(1) Current consumption in standby mode

When the A/D converter enters a standby mode, it stops operating. Setting bit 7 (ADCS0) of A/D converter mode register 0 (ADM0) to 0 can reduce the current consumption.

Figure 9-7 shows how to reduce the current consumption in standby mode.

Figure 9-7. How to Reduce Current Consumption in Standby Mode



(2) Input range for the ANI0 pin

Be sure to keep the input voltage at ANI0 within the rating. If a voltage greater than or equal to V_{DD} or less than or equal to V_{SS} (even if within the absolute maximum ratings) is input to this channel, the conversion output of the channel becomes undefined.

(3) Conflict

<1> Conflict between writing to A/D conversion result register 0 (ADCR0) at the end of conversion and reading from ADCR0

Reading from ADCR0 takes precedence. After reading, the new conversion result is written to the ADCR0.

<2> Conflict between writing to ADCR0 at the end of conversion and writing to A/D converter mode register 0 (ADM0) or A/D input selection register 0 (ADS0)

Writing to ADM0 or ADS0 takes precedence. A request to write to ADCR0 is ignored. No conversion end interrupt request signal (INTAD0) is generated.

(4) Conversion results immediately following start of A/D conversion

The first A/D conversion value immediately following the start of A/D converter operation is undefined. Be sure to poll the A/D conversion end interrupt request (INTAD0), discard the first conversion result and use the second or later conversion result.

(5) Timing that makes the A/D conversion result undefined

If the timing of the end of A/D conversion and the timing of the stop of operation of the A/C converter conflict, the A/D conversion value may be undefined. Because of this, be sure to read out the A/D conversion result while the A/D converter is operating. Furthermore, when reading out an A/D conversion result after A/D converter operation has stopped, be sure to have done so by the time the next conversion result is complete. The conversion result readout timing is shown in Figures 9-8 and 9-9.

Figure 9-8. Conversion Result Readout Timing (When Conversion Result Is Undefined Value)

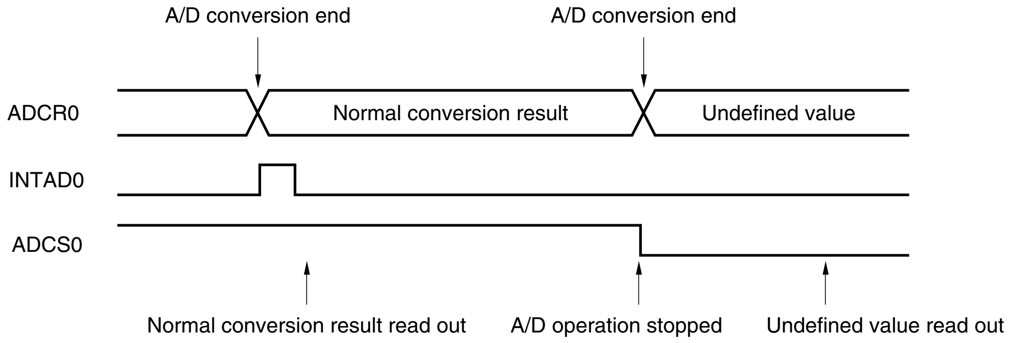
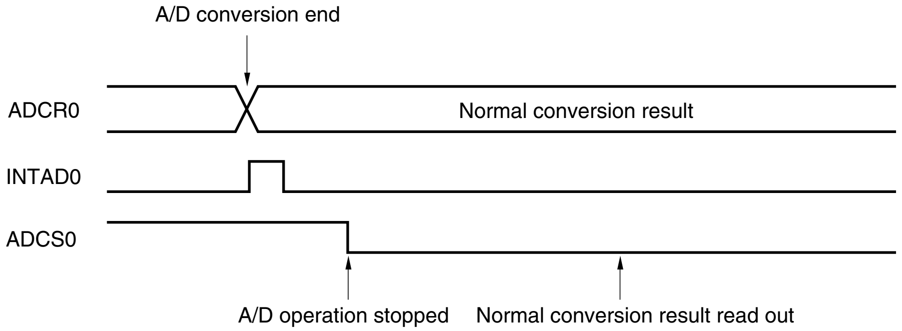


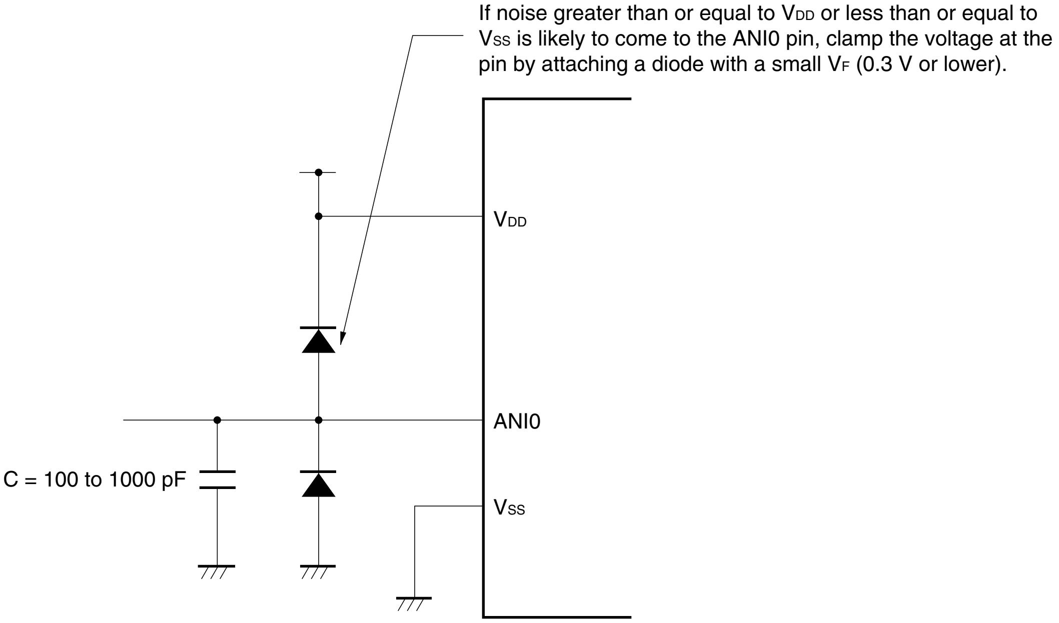
Figure 9-9. Conversion Result Readout Timing (When Conversion Result Is Normal Value)



(6) Noise prevention

To maintain a resolution of 8 bits, watch for noise at the V_{DD} and ANI0 pins. The higher the output impedance of the analog input source, the larger the effect by noise. To reduce noise, attach an external capacitor to the relevant pins as shown in Figure 9-8.

Figure 9-10. Analog Input Pin Handling


If noise greater than or equal to V_{DD} or less than or equal to V_{SS} is likely to come to the ANI0 pin, clamp the voltage at the pin by attaching a diode with a small V_F (0.3 V or lower).
V_{DD}
ANI0
V_{SS}
C = 100 to 1000 pF

(7) ANI0

The analog input pin (ANI0) is an alternate-function pin. It is also used as a port pin (P60).

If ANI0 has been selected for A/D conversion, do not execute input instructions for the ports; otherwise the conversion resolution may drop.

If a digital pulse is applied to a pin adjacent to the analog input pin being A/D converted, coupling noise may occur which prevents an A/D conversion result from being attained as expected. Avoid applying a digital pulse to pins adjacent to the analog input pin being A/D converted.

★ **(8) Input impedance of ANI0 pin**

This A/D converter executes sampling by charging the internal sampling capacitor for approximately 1/10 of the conversion time.

Therefore, only the leakage current flows during other than sampling, and the current for charging the capacitor flows during sampling. The input impedance therefore varies and has no meaning.

To achieve sufficient sampling, it is recommended that the output impedance of the analog input source be 10 kΩ or less, or attach a capacitor of around 100 pF to the ANI0 pin (see **Figure 9-10**).

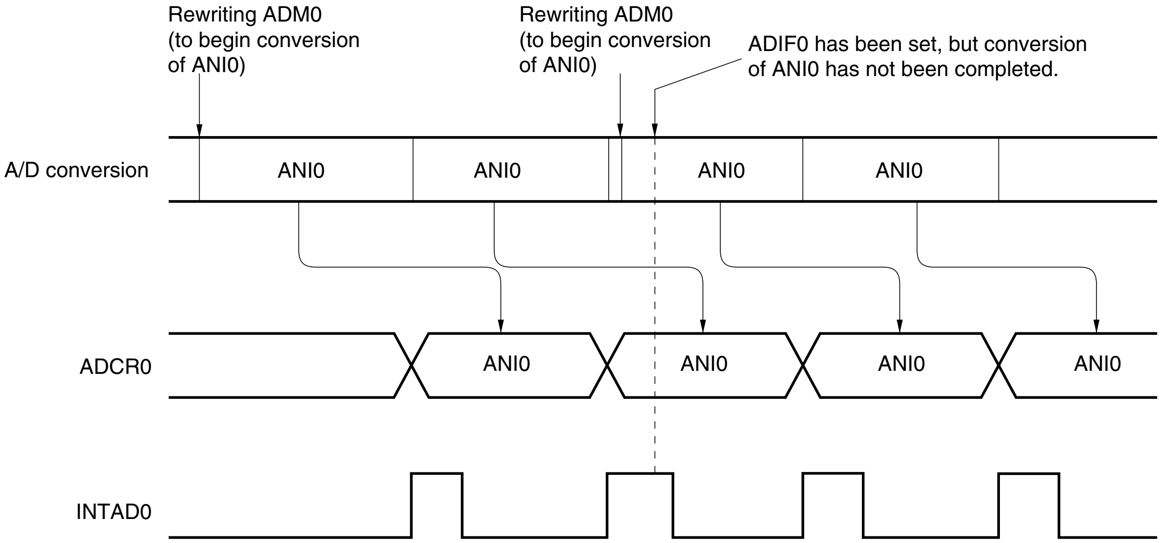
(9) Interrupt request flag (ADIF0)

Changing the contents of A/D converter mode register 0 (ADM0) does not clear the interrupt request flag (ADIF0).

If the analog input pin is changed during A/D conversion, therefore, the conversion result and the conversion end interrupt request flag may reflect the previous analog input immediately before writing to ADM0 occurs. In this case, ADIF0 may appear to be set if it is read-accessed immediately after ADM0 is write-accessed, even when A/D conversion has not been completed for the new analog input.

In addition, ADIF0 must be cleared before A/D conversion is restarted.

Figure 9-11. A/D Conversion End Interrupt Request Generation Timing



(10) Input impedance of V_{DD} pin

A series resistor string of several 10 kΩ is connected across the V_{DD} and V_{SS} pins.

Therefore, if the output impedance of the reference voltage source is high, this high impedance is eventually connected in series with the series resistor string across the V_{DD} and V_{SS} pins, leading to a higher reference voltage error.

CHAPTER 10 LCD CONTROLLER/DRIVER

10.1 Functions of LCD Controller/Driver

The features of the LCD controller/driver of the μPD789467 Subseries are as follows.

(1) Automatic output of segment and common signals based on automatic display data memory read
(2) Four different frame frequencies selectable
(3) Up to 23 segment signal outputs (S0 to S22) and four common signal outputs (COM0 to COM3)
(4) Operation with subsystem clock
(5) Voltage booster incorporated

Table 10-1 lists the maximum number of pixels that can be displayed in each display mode.

Table 10-1. Maximum Number of Pixels

Bias Mode	Number of Time Slices	Common Signals Used	Maximum Number of Pixels
1/3	4	COM0 to COM3	92 (23 segments × 4 commons)[Note]

Note 11-digit LCD panel, each digit having a 2-segment 8. configuration.

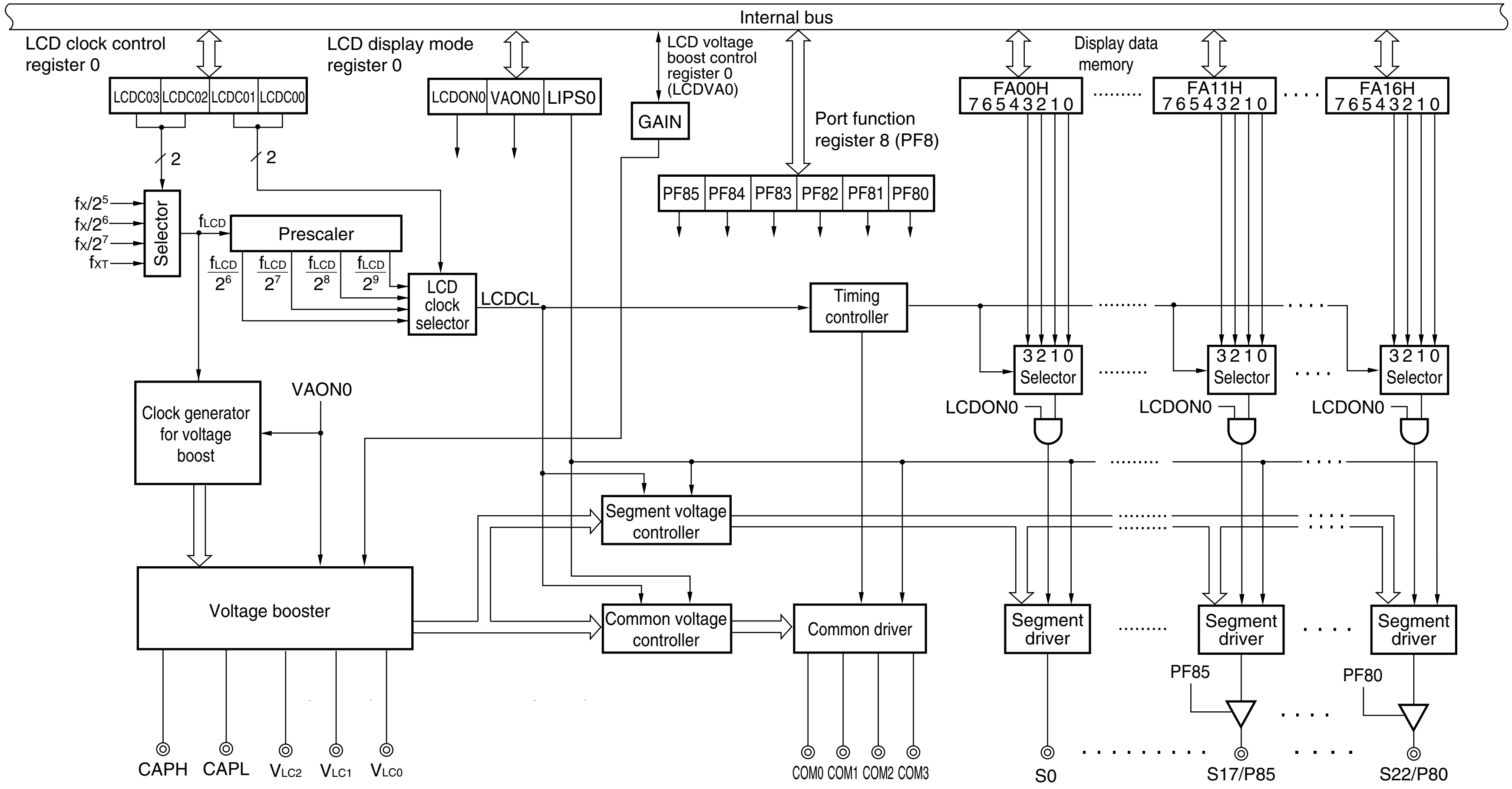
10.2 Configuration of LCD Controller/Driver

The LCD controller/driver includes the following hardware.

Table 10-2. Configuration of LCD Controller/Driver

Item	Configuration
Display outputs	Segment signals: 23 Common signals: 4
Control registers	LCD display mode register 0 (LCDM0) LCD clock control register 0 (LCDC0) LCD voltage boost control register (LCDVA0) Port function register 8 (PF8)

Figure 10-1. Block Diagram of LCD Controller/Driver



10.3 Registers Controlling LCD Controller/Driver

- LCD display mode register 0 (LCDM0)
- LCD clock control register 0 (LCDC0)
- LCD voltage boost control register 0 (LCDVA0)
- Port function register 8 (PF8)

(1) LCD display mode register 0 (LCDM0)

This register is used to enable/disable display operation, enable/disable voltage boost, and specify segment/common pin output and display mode.

LCDM0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets LCDM0 to 00H.

Figure 10-2. Format of LCD Display Mode Register 0

Symbol	<7>	<6>	5	4	3	2	1	0	Address	After reset	R/W
LCDM0	LCDON0	VAON0	0	LIPS0	0	LCDM02	0	0	FFB0H	00H	R/W

LCDON0	LCD display enable/disable
0	Display off (all segment outputs are unselected for signal output)
1	Display on

VAON0	Voltage booster enable/disable[Note 1]
0	Voltage boost disabled
1	Voltage boost enabled

LIPS0	Segment/common pin output control[Notes 1,2]
0	Ground level is output to segment/common pins
1	Unselected level and LCD waveform are output to segment and common pins, respectively.

LCDM02	Display mode selection[Note 3]
0	Four-time-slice, 1/3 bias mode
1	Static mode

Notes 1. When the LCD display panel is not used, set VAON0 and LIPS0 to 0 to reduce power ★ consumption. However, when any one of pins P80 to P85 is used as a general-purpose pin, set VAON0 to 1.

2. LIPS0 is valid only in mask ROM versions. However, writing to LIPS0 in the μPD78F9468 will not cause an error.

3. Ordinarily use in four-time-slice, 1/3 bias mode.

In either of the following cases, however, switch to the static mode (LCDM02 =1).

- When changing to the STOP mode when the main system clock is selected as the LCD source clock.
- Between reset and voltage boost start
- When disabling both display and voltage boost

Cautions 1. Bits 0, 1, 3, and 5 must be set to 0.

2. When the main system clock is selected as the LCD source clock, if the STOP mode is selected, an abnormal display may occur. Before selecting the STOP mode, disable display and select the static mode (LCDON0 = 0 and LCDM02 =1). If the subsystem clock is selected as the LCD source clock, a normal operation is performed in the STOP mode.

3. After reset, both display and voltage boost are disabled. Therefore, set LCDM02 to 1 to select the static mode. Otherwise, an abnormal display may occur until voltage boosting starts.

4. When operating VAON0, follow the procedure described below.

A. To stop voltage boost after switching display status from on to off:

1) Set to display off status by setting LCDON0 = 0.

2) Disable outputs of all the segment buffers and common buffers by setting LIPS0 = 0.

3) Select the static mode by setting LCDM02 = 1.

4) Stop voltage boost by setting VAON0= 0.

B. To stop voltage boost during display on status:

Setting prohibited. Be sure to stop voltage boost after setting display off.

C. To set display on from voltage boost stop status:

1) Start voltage boost by setting VAON0 = 1, then wait for about 500 ms.

2) Select the four-time-slice, 1/3 bias mode by setting LCDM02 = 0.

3) Set all the segment buffers and common buffers to the non-display output status by setting LIPS0 = 1.

4) Set display on by setting LCDON0 = 1.

★ **5. The combinations of settings of LCDON0, VAON0, and LIPS0 enabled/prohibited by the device file are as follows.**

LCDON0	VAON0	LIPS0	Combination enabled/prohibited
0	0	0	Enabled by device file
0	1	0	
0	1	1	
1	1	1	
0	0	1	Prohibited by device file (If this combination is set, an error occurs.)
1	0	0	
1	0	1	
1	1	0	

(2) LCD clock control register 0 (LCDC0)

LCDC0 specifies the LCD source clock and LCD clock. The frame frequency is determined by the LCD clock and the number of time slices.

LCDC0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets LCDC0 to 00H.

Figure 10-3. Format of LCD Clock Control Register 0

Symbol	7	6	5	4	3	2	1	<0>	Address	After reset	R/W
LCDC0	0	0	0	0	LCDC03	LCDC02	LCDC01	LCDC00	FFB2H	00H	R/W

LCDC03	LCDC02	LCD source clock (f_{LCD}) selection[Note]
0	0	f_{XT} (32.768 kHz)
0	1	$f_X/2^5$ (156.3 kHz)
1	0	$f_X/2^6$ (78.1 kHz)
1	1	$f_X/2^7$ (39.1 kHz)

LCDC01	LCDC00	LCD clock (LCDCL) selection
0	0	$f_{LCD}/2^6$
0	1	$f_{LCD}/2^7$
1	0	$f_{LCD}/2^8$
1	1	$f_{LCD}/2^9$

Note Specify an LCD source clock (f_{LCD}) frequency of at least 32 kHz.

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

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

Cautions 1. Bits 4 to 7 must be set to 0.

2. Be sure to turn the display off (LCDON0 = 0) and stop the voltage booster (VAON0 = 0) before changing the LCDC0 settings.

An example of frame frequencies when f_{XT} (32.768 kHz) is connected to the LCD source clock (f_{LCD}) is shown in Table 10-3.

Caution Set frame frequencies to 128 Hz or lower.

Table 10-3. Frame Frequencies (Hz)

LCD Clock (LCDCL) Time Slices	$f_{XT}/2^9$ (64 Hz)	$f_{XT}/2^8$ (128 Hz)	$f_{XT}/2^7$ (256 Hz)	$f_{XT}/2^6$ (512 Hz)
4	16	32	64	128

(3) LCD voltage boost control register 0 (LCDVA0)

LCDVA0 controls the voltage boost level during the voltage boost operation.

LCDVA0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets LCDVA0 to 00H.

Figure 10-4. Format of LCD Voltage Boost Control Register 0

Symbol	7	6	5	4	3	2	1	<0>	Address	After reset	R/W
LCDVA0	0	0	0	0	0	0	0	GAIN	FFB3H	00H	R/W

GAIN	Reference voltage (V_{LC2}) level selection[Note]
0	1.5 V (specification of the LCD panel used is 4.5 V.)
1	1.0 V (specification of the LCD panel used is 3 V.)

Note Select the settings according to the specifications of the LCD panel that is used.

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

2. Before changing the LCDVA0 setting, be sure to stop voltage boost (VAON0 = 0).

Remark The TYP. value is indicated as the reference voltage (V_{LC2}) value.

(4) Port function register 8 (PF8)

PF8 specifies whether S17/P85 to S22/P80 are used as LCD segment signal outputs or general-purpose ports in 1-bit units.

PF8 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PF8 to 00H.

Figure 10-5. Format of Port Function Register 8

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PF8	0	0	PF85	PF84	PF83	PF82	PF81	PF80	FF58H	00H	R/W

PF8n	Port function of P8n (n = 0 to 5)
0	Operates as a general-purpose port
1	Operates as an LCD segment signal output

(For cautions on this register, refer to **Figure 4-10**.)

10.4 Setting LCD Controller/Driver

Set the LCD controller/driver using the following procedure.
Before changing the LCD clock or voltage boost level, disable display and voltage boost.

10.4.1 Setting before starting display

<1> With the default setting after reset, select the static mode by setting bit 2 (LCDM02 = 1) of LCDM0.
<2> Input the initial data to the LCD display data memory.
<3> Set the LCD clock, using LCD clock control register 0 (LCDC0).
<4> Set the voltage boost level, using LCD voltage boost control register 0 (LCDVA0).
GAIN = 0: V_{LC0} = 4.5 V, V_{LC1} = 3.0 V, V_{LC2} = 1.5 V
GAIN = 0: V_{LC0} = 3.0 V, V_{LC1} = 2.0 V, V_{LC2} = 1.0 V
<5> Set bit 6 (VAON0 = 1) of LCD display mode register 0 (LCDM0) to enable the voltage booster.
<6> Wait for 500 ms or more after setting VAON0.
<7> Clear bit 2 (LCDM02 = 0) of LCDM0 to select four-time-slice, 1/3 bias mode.
<8> Set bit 4 (LIPS0 = 1) of LCDM0 to output the deselect electric potential.
<9> Set bit 7 (LCDON0 = 1) of LCDM0 to start output corresponding to each data memory.

10.4.2 Setting until disabling display and stopping voltage boost

<1> Clear bit 7 (LCDON0 = 0) of LCDM0 to select the display off status.
<2> Clear bit 4 (LIPS0 =0) of LCDM0 to prohibit all segment buffers and common buffers from outputting.
<3> Set bit 2 (LCDM02 = 1) of LCDM0 to select the static mode.
<4> Clear bit 6 (VAON0 = 0) of LCDM0 to disable voltage boost.

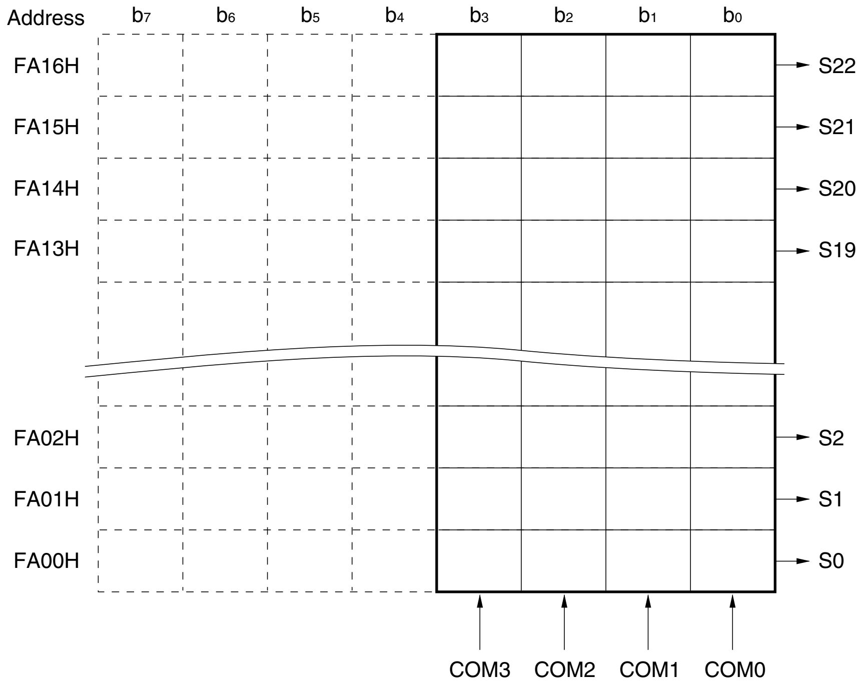
10.5 LCD Display Data Memory

The LCD display data memory is mapped at addresses FA00H to FA16H. Data in the LCD display data memory can be displayed on the LCD panel using the LCD controller/driver.

Figure 10-6 shows the relationship between the contents of the LCD display data memory and the segment/common outputs.

Parts of the display data memory not used for display can be used as ordinary RAM.

Figure 10-6. Relationship Between LCD Display Data Memory Contents and Segment/Common Outputs



Caution No memory has been assigned as the higher 4 bits of the LCD display data memory. Be sure to set these bits to 0.

10.6 Common and Segment Signals

Each pixel of the LCD panel turns on when the potential difference between the corresponding common and segment signals becomes higher than a specific voltage (LCD drive voltage, V_{LCD}). It turns off when the potential difference becomes lower than V_{LCD}.

Applying DC voltage to the common and segment signals for an LCD panel causes degradation. To avoid this problem, this LCD panel is driven with AC voltage.

(1) Common signals

Each common signal is selected sequentially according to the specified number of time slices at the timing listed in Table 10-4.

Table 10-4. COM Signals

COM Signal / Number of Time Slices	COM0	COM1	COM2	COM3
Four-time-slice mode				

(2) Segment signals

The segment signals correspond to 23 bytes of LCD display data memory (FA00H to FA16H). Bits 0, 1, 2, and 3 of each byte are read in synchronization with COM0, COM1, COM2, and COM3, respectively. If the contents of each bit are 1, it is converted to the select voltage, and if 0, it is converted to the deselect voltage. The conversion results are output to the segment pins (S0 to S22).

Check, using the information given above, what combination of the front-surface electrodes (corresponding to the segment signals) and the rear-surface electrodes (corresponding to the common signals) forms display patterns in the LCD display data memory, and write the bit data that corresponds to the desired display pattern on a one-to-one basis.

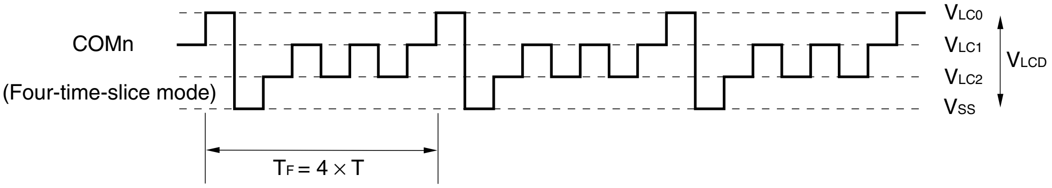
LCD display data memory bits 4 to 7 are fixed to 0.

(3) Output waveforms of common and segment signals

When both common and segment signals are at the select voltage, a display-on voltage of $\pm V_{LCD}$ is obtained. The other combinations of the signals correspond to the display-off voltage.

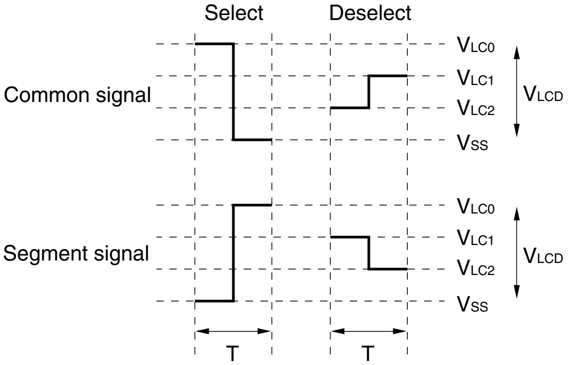
Figure 10-7 shows the common signal waveforms, and Figure 10-8 shows the voltages and phases of the common and segment signals.

Figure 10-7. Common Signal Waveforms



T: One LCD clock cycle TF: Frame frequency

Figure 10-8. Voltages and Phases of Common and Segment Signals



T: One LCD clock cycle

10.7 Display Modes

10.7.1 Four-time-slice display example

Figure 10-10 shows how the 11-digit LCD panel having the display pattern shown in Figure 10-9 is connected to the segment signals (S0 to S21) and the common signals (COM0 to COM3) of the μPD789467 Subseries chip. This example displays data “23456.789012” in the LCD panel. The contents of the display data memory (addresses FA00H to FA15H) correspond to this display.

The following description focuses on numeral “6.” (6.) displayed in the seventh digit from the right. To display “6.” in the LCD panel, it is necessary to apply the select or deselect voltage to the S12 and S13 pins according to Table 10-5 at the timing of the common signals COM0 to COM3.

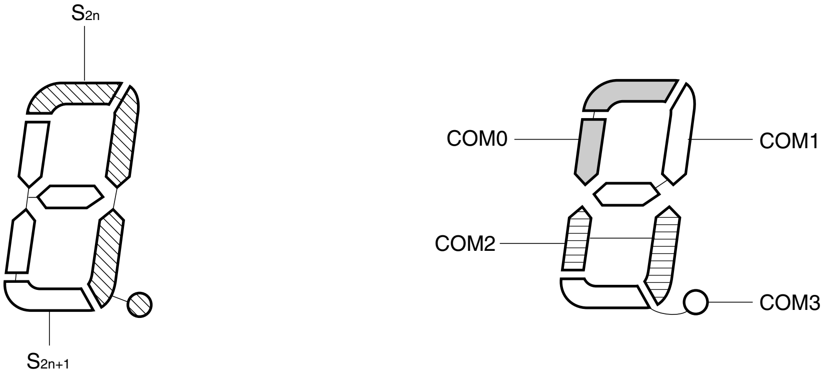
Table 10-5. Select and Deselect Voltages (COM0 to COM3)

Segment Common	S12	S13
COM0	Select	Select
COM1	Deselect	Select
COM2	Select	Select
COM3	Select	Select

According to Table 10-5, it is determined that the display data memory location (FA0CH) that corresponds to S12 must contain 1101.

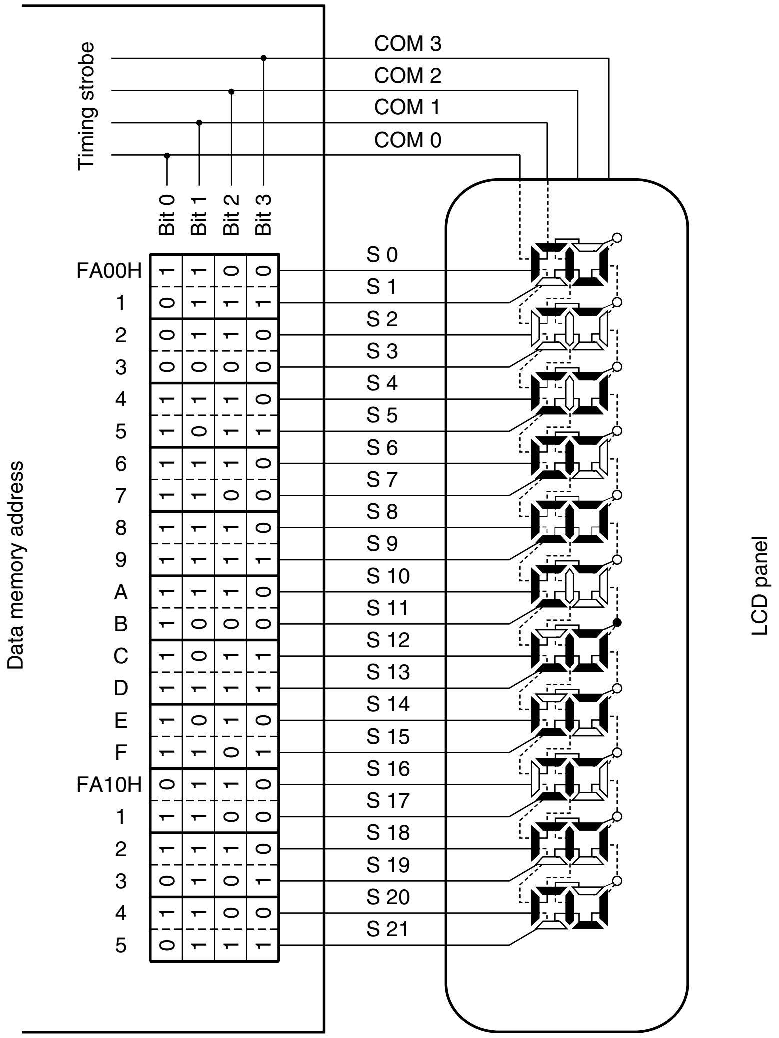
Figure 10-11 shows examples of LCD drive waveforms between the S12 signal and each common signal. When the select voltage is applied to S12 at the timing of COM0, an alternate rectangle waveform, +V_{LCD}/–V_{LCD}, is generated to turn on the corresponding LCD segment.

Figure 10-9. Four-Time-Slice LCD Display Pattern and Electrode Connections



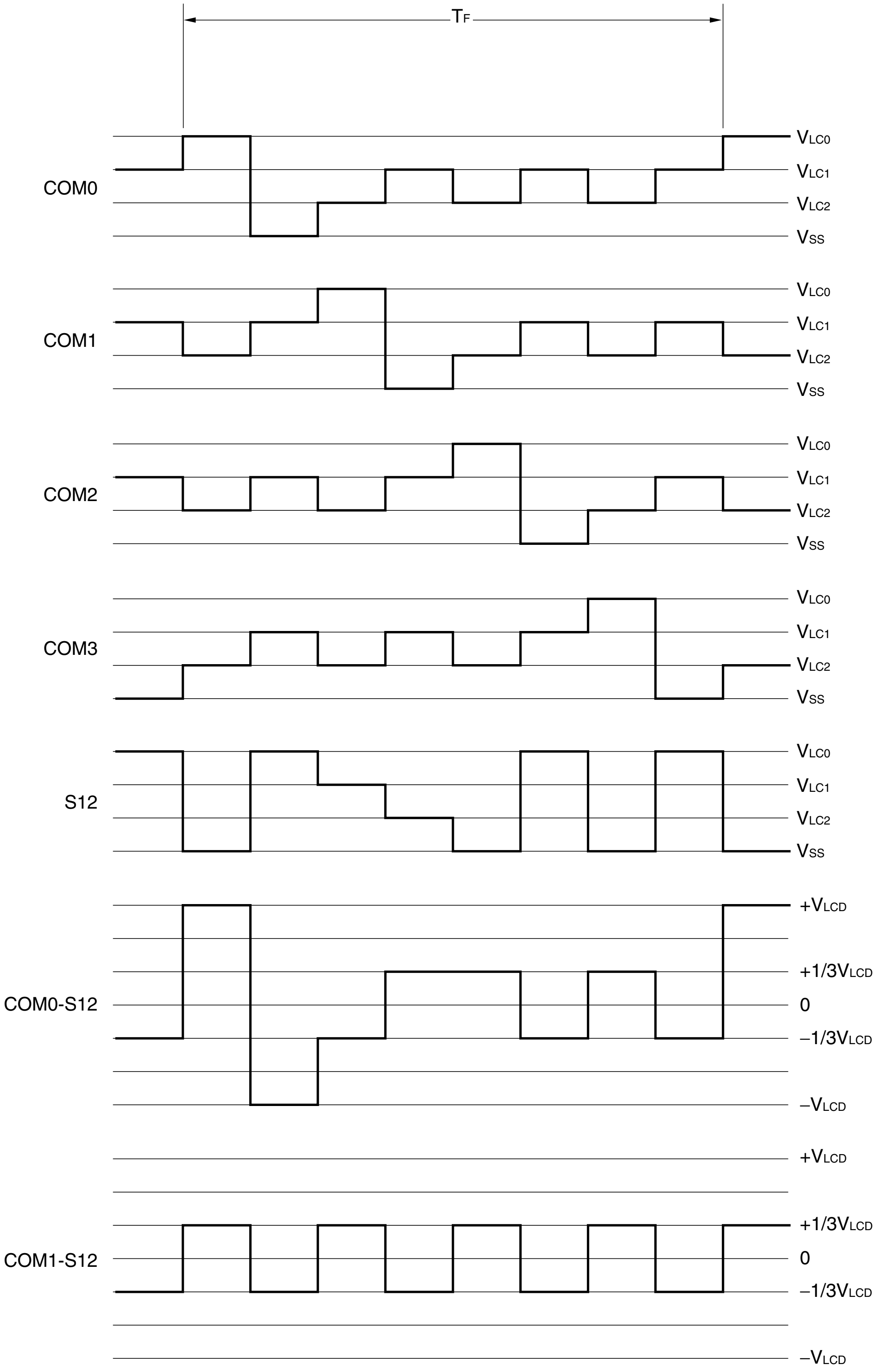
Remark n = 0 to 10

Figure 10-10. Example of Connecting Four-Time-Slice LCD Panel



	Bit 0	Bit 1	Bit 2	Bit 3
FA00H	1	1	0	0
1	0	1	1	1
2	0	1	1	0
3	0	0	0	0
4	1	1	1	0
5	1	0	1	1
6	1	1	1	0
7	1	1	0	0
8	1	1	1	0
9	1	1	1	1
A	1	1	1	0
B	1	0	0	0
C	1	0	1	1
D	1	1	1	1
E	1	0	1	0
F	1	1	0	1
FA10H	0	1	1	0
1	1	1	0	0
2	1	1	1	0
3	0	1	0	1
4	1	1	0	0
5	0	1	1	1

Figure 10-11. Examples of Four-Time-Slice LCD Drive Waveform



Remark The waveforms of COM2-S12 and COM3-S12 are not shown above.

10.8 Supplying LCD Drive Voltages V_{LC0}, V_{LC1}, and V_{LC2}

The μPD789467 Subseries contains a booster circuit (×3 only) to generate a supply voltage to drive the LCD. The internal LCD reference voltage is output from the V_{LC2} pin. A voltage two times higher than that on V_{LC2} is output from the V_{LC1} pin and a voltage three times higher than that on V_{LC2} is output from the V_{LC0} pin.

The LCD reference voltage (V_{LC2}) can be selected by setting LCD voltage boost control register 0 (LCDVA0).

The μPD789467 Subseries requires an external capacitor (recommended value: 0.47 μF) because it employs a capacitance division method to generate the supply voltage to drive the LCD.

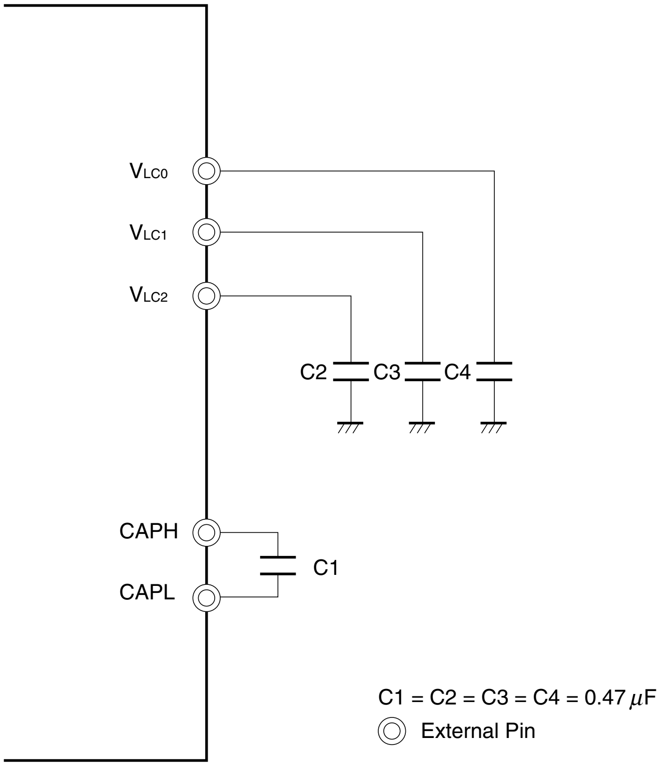
Table 10-6. Output Voltages of V_{LC0} to V_{LC2} Pins

LCDVA0 LCD driving power supply pin	GAIN = 0	GAIN = 1
V_{LC0}	4.5 V	3.0 V
V_{LC1}	3.0 V	2.0 V
V_{LC2} (LCD reference voltage)	1.5 V	1.0 V

Cautions 1. When using the LCD function, do not leave the V_{LC0}, V_{LC1}, and V_{LC2} pins open. Refer to Figure 10-12 for connection.

2. The power for the LCD drive is supplied separately from the μPD789467 Subseries; therefore, a constant voltage can be supplied regardless of the change of V_{DD}.

Figure 10-12. Example of Pin Connection for LCD Driver



Remark Use capacitors with as little leakage as possible. Use a non-polar capacitor for C1.

CHAPTER 11 POWER-ON-CLEAR CIRCUIT

The μPD789467 Subseries provides a power-on-clear (POC) circuit. In the flash memory version (μPD78F9468), the POC circuit is always operating. However, it can only be used when selected by a mask option in mask ROM versions (μPD789462, 789464, 789466, and 789467) (see **CHAPTER 16 MASK OPTION**).

11.1 Functions of Power-on-Clear Circuit

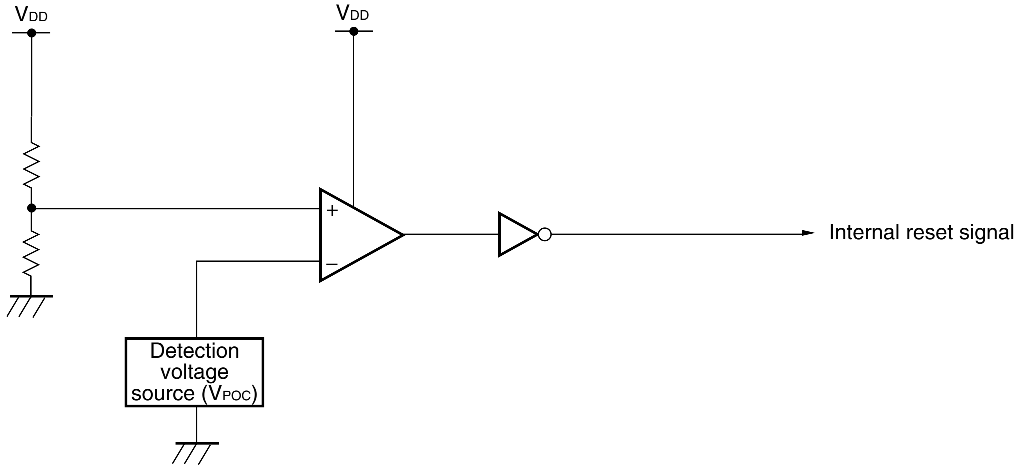
The power-on-clear circuit includes the following functions.

- Compares the detection voltage (V_{POC}) with the power supply voltage (V_{DD}) and generates an internal reset signal if $V_{DD} < V_{POC}$.
- Can operate even in STOP mode.

11.2 Configuration of Power-on-Clear Circuit

Figure 11-1 shows the block diagram of the power-on-clear circuit.

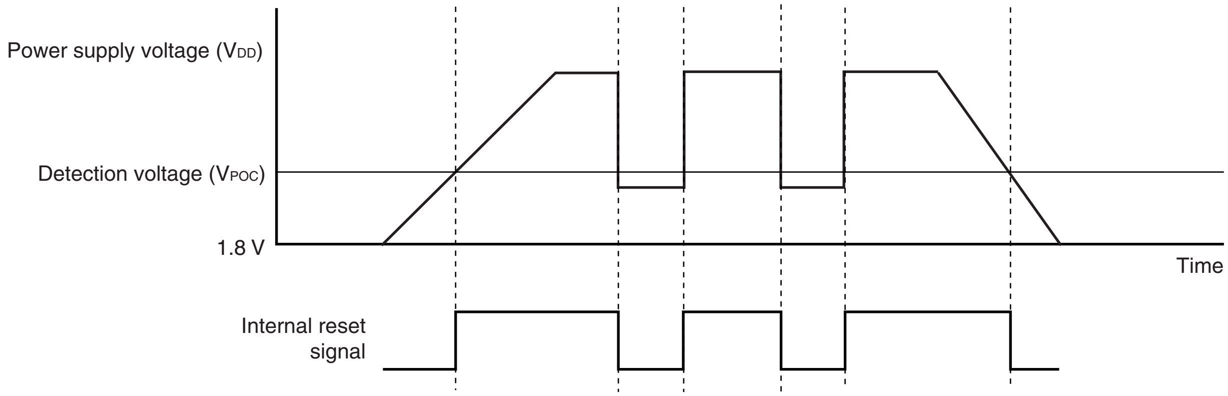
Figure 11-1. Block Diagram of Power-on-Clear Circuit



11.3 Power-on-Clear Circuit Operation

The POC circuit compares the detection voltage (V_{POC}) with the power supply voltage (V_{DD}) and generates an internal reset signal if $V_{DD} < V_{POC}$.

Figure 11-2. Timing of Internal Reset Signal Generation of POC Circuit



CHAPTER 12 INTERRUPT FUNCTION

12.1 Interrupt Types

The following two types of interrupts are used.

(1) Non-maskable interrupt

This interrupt is acknowledged unconditionally. It does not undergo interrupt priority control and is given top priority over all other interrupt requests.
A standby release signal is generated.
One interrupt source from the watchdog timer is incorporated as a non-maskable interrupt.

(2) Maskable interrupt

This interrupt undergoes mask control. If two or more interrupts with the same priority are simultaneously generated, each interrupt is serviced according to a predetermined priority as shown in Table 12-1.
A standby release signal is generated.
Two external and six internal interrupt sources are incorporated as maskable interrupts.

12.2 Interrupt Sources and Configuration

A total of nine non-maskable and maskable interrupts are incorporated as interrupt sources (see **Table 12-1**).

Table 12-1. Interrupt Source List

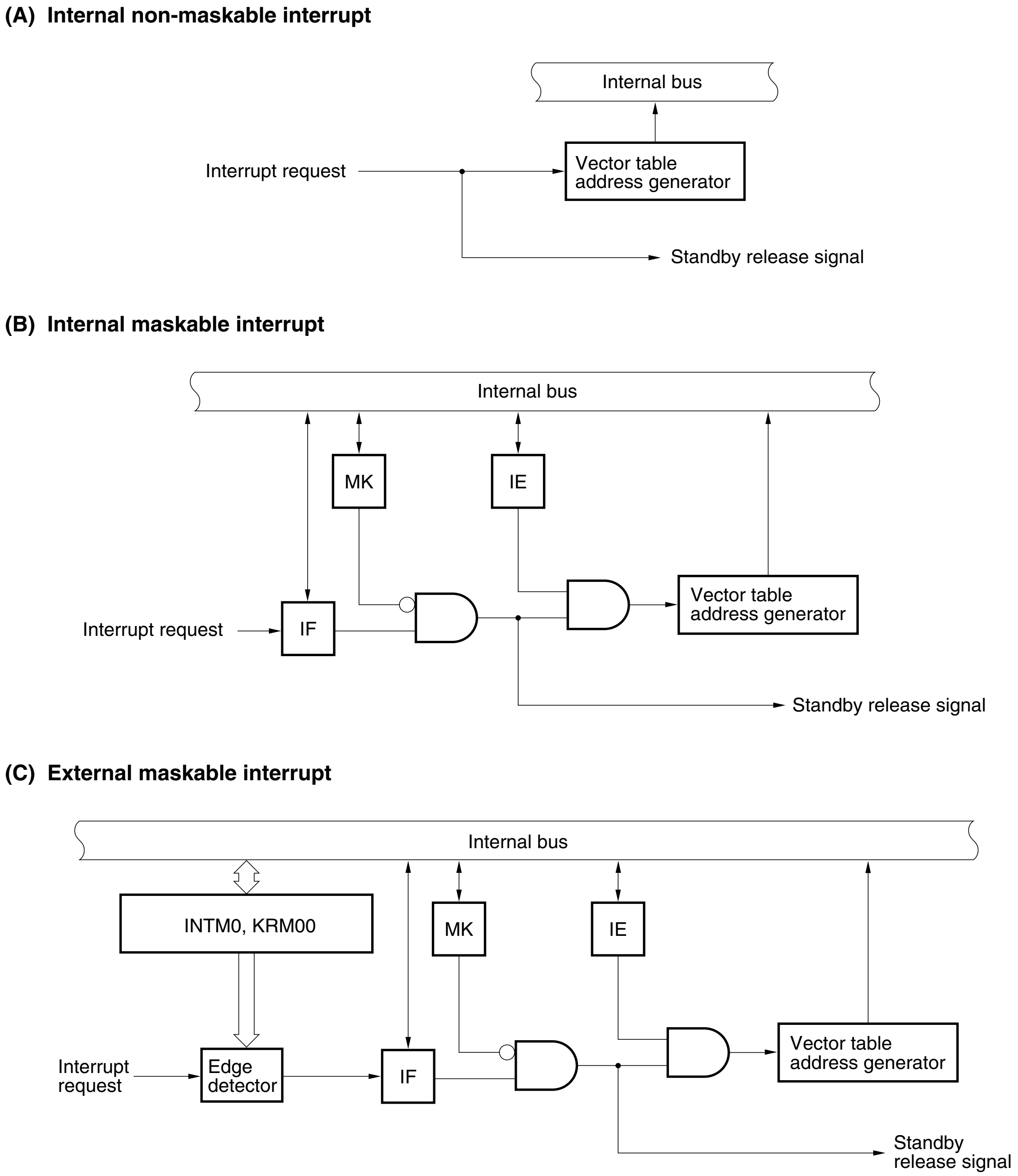
Interrupt Type	Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Non-maskable	–	INTWDT	Watchdog timer overflow (with watchdog timer mode 1 selected)	Internal	0004H	(A)
Maskable	0	INTWDT	Watchdog timer overflow (with interval timer mode selected)			(B)
	1	INTP0	Pin input edge detection	External	0006H	(C)
	2	INTAD0	A/D conversion completion signal	Internal	0008H	(B)
	3	INTWT	Watch timer interrupt		000AH	
	4	INTTM30	Generation of 8-bit timer 30 match signal		000CH	
	5	INTTM40	Generation of 8-bit timer 40 match signal		000EH	
	6	INTKR00	Key return signal detection	External	0010H	(C)
	7	INTWTI	Watch timer interval timer interrupt	Internal	0012H	(B)

Notes 1. "Priority" is the priority order when more than one maskable interrupt request is generated at the same time. 0 is the highest priority and 7 is the lowest.

2. Basic configuration types (A), (B), and (C) correspond to (A), (B), and (C) in Figure 12-1.

Remark There are two interrupt sources for the watchdog timer (INTWDT): non-maskable and maskable interrupts (internal). Either one (but not both) should be selected for actual use.

Figure 12-1. Basic Configuration of Interrupt Function



(A) Internal non-maskable interrupt

(B) Internal maskable interrupt

(C) External maskable interrupt

INTP0: External interrupt mode register 0
KRM00: Key return mode register 00
IF: Interrupt request flag
IE: Interrupt enable flag
MK: Interrupt mask flag

12.3 Registers Controlling Interrupt Function

The following five registers are used to control the interrupt function.

- Interrupt request flag register 0 (IF0)
- Interrupt mask flag register 0 (MK0)
- External interrupt mode register 0 (INTM0)
- Program status word (PSW)
- Key return mode register 00 (KRM00)

Table 12-2 gives a listing of interrupt request and interrupt mask flag names corresponding to interrupt requests.

Table 12-2. Flags Corresponding to Interrupt Request Signal Names

Interrupt Request Signal Name	Interrupt Request Flag	Interrupt Mask Flag
INTWDT	WDTIF	WDTMK
INTP0	PIF0	PMK0
INTAD0	ADIF0	ADMK0
INTWT	WTIF	WTMK
INTTM30	TMIF30	TMMK30
INTTM40	TMIF40	TMMK40
INTKR00	KRIF00	KRMK00
INTWTI	WTIIF	WTIMK

(1) Interrupt request flag register 0 (IF0)

An interrupt request flag is set (1) when the corresponding interrupt request is generated, or when an instruction is executed. It is cleared (0) when the interrupt request is acknowledged, when the $\overline{\text{RESET}}$ signal is input, or when an instruction is executed.

IF0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets IF0 to 00H.

Figure 12-2. Format of Interrupt Request Flag Register 0

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
IF0	WTIIF	KRIF00	TMIF40	TMIF30	WTIF	ADIF0	PIF0	WDTIF	FFE0H	00H	R/W

××IF×	Interrupt request flag
0	No interrupt request signal generated
1	An interrupt request signal is generated and an interrupt request made

Cautions 1. The WDTIF flag can be read/written only when the watchdog timer is being used as an interval timer. It must be cleared to 0 if the watchdog timer is used in watchdog timer mode 1 or 2.

2. Because P61 functions alternately as an external interrupt input, when the output level changes after the output mode of the port function is specified, the interrupt request flag will be inadvertently set. Therefore, be sure to preset the interrupt mask flag (PMK0) to 1 before using the port in output mode.

★ **3. When an interrupt is acknowledged, the interrupt request flag is automatically cleared and then the interrupt routine is started.**

(2) Interrupt mask flag register 0 (MK0)

Interrupt mask flags are used to enable and disable the corresponding maskable interrupts.

MK0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets MK0 to FFH.

Figure 12-3. Format of Interrupt Mask Flag Register 0

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
MK0	WTIMK	KRMK00	TMMK40	TMMK30	WTMK	ADMK0	PMK0	WDTMK	FFE4H	FFH	R/W

××MK	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Cautions 1. When the watchdog timer is being used in watchdog timer mode 1 or 2, any attempt to read the WDTMK flag results in an undefined value being detected.

2. Because P61 functions alternately as an external interrupt input, when the output level changes after the output mode of the port function is specified, the interrupt request flag will be inadvertently set. Therefore, be sure to preset the interrupt mask flag (PMK0) to 1 before using the port in output mode.

(3) External interrupt mode register 0 (INTM0)

INTM0 is used to specify the valid edge for INTP0.

INTM0 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets INTM0 to 00H.

Figure 12-4. Format of External Interrupt Mode Register 0

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
INTM0	0	0	0	0	ES01	ES00	0	0	FFECH	00H	R/W

ES01	ES00	INTP0 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

Cautions 1. Bits 0, 1, and 4 to 7 must be set to 0.

2. Before setting INTM0, set (1) the interrupt mask flag (PMK0) to disable interrupts. To enable interrupts, clear (0) the interrupt request flag (PIF0), then clear (0) the interrupt mask flag (PMK0).

(4) Program status word (PSW)

The program status word is a register used to hold the instruction execution result and the current status for interrupt requests. The IE flag, used to set maskable interrupt enable/disable, is mapped to the PSW.

Besides 8-bit unit read/write, this register can carry out operations via bit manipulation instructions and dedicated instructions (EI, DI). When a vectored interrupt is acknowledged, the PSW is automatically saved into a stack, and the IE flag is reset to 0.

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

Figure 12-5. Configuration of Program Status Word

Symbol	7	6	5	4	3	2	1	0	After reset
PSW	IE	Z	0	AC	0	0	1	CY	02H

Z, AC, CY → Used when normal instruction is executed

IE	Interrupt acknowledgment enable/disable
0	Disabled
1	Enabled

(5) Key return mode register 00 (KRM00)

KRM00 sets the pin that detects a key return signal (falling edge of port 4).

KRM00 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets KRM00 to 00H.

Figure 12-6. Format of Key Return Mode Register 00

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
KRM00	0	0	0	0	0	0	0	KRM000	FFF5H	00H	R/W

KRM000	Key return signal detection control
0	No detection
1	Detection (detecting falling edge of port 4)

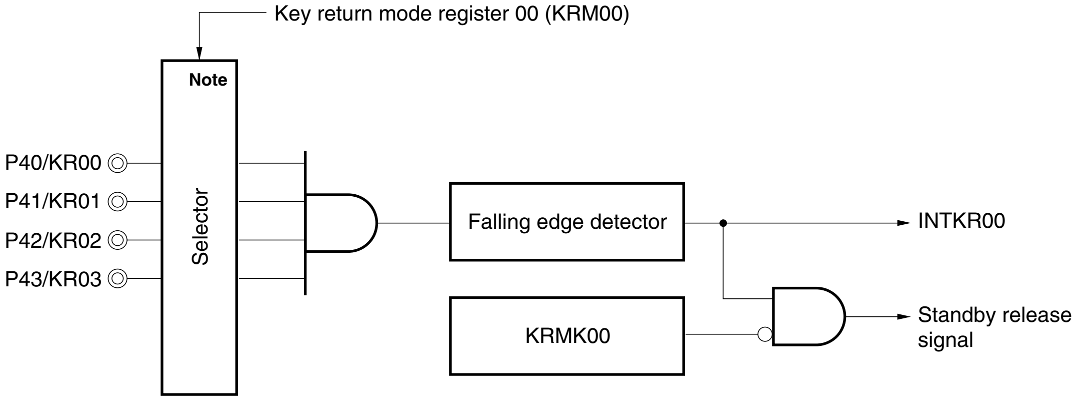
Cautions 1. Bits 1 to 7 must be set to 0.

2. Before setting KRM00, always set bit 6 of MK0 (KRMK00 = 1) to disable interrupts. After setting KRM00, clear KRMK00 after clearing bit 6 of IF0 (KRIF00 = 0) to enable interrupts.

3. On-chip pull-up resistors are automatically connected in input mode to the pins specified for key return signal detection (P40 to P43). Although these resistors are disconnected when the mode changes to output, rising edge detection continues unchanged. Therefore, when the output data of the pin falls, an interrupt (INTKR00) occurs.

★ **4. The key return signal can be detected while all of P40 to P43 are high level. The key return signal cannot be detected while even one of P40 to P43 is low, even if the falling edge of another key return pin is detected.**

Figure 12-7. Block Diagram of Falling Edge Detector



Note Selector that selects the pin used for falling edge input

12.4 Interrupt Servicing Operation

12.4.1 Non-maskable interrupt request acknowledgment operation

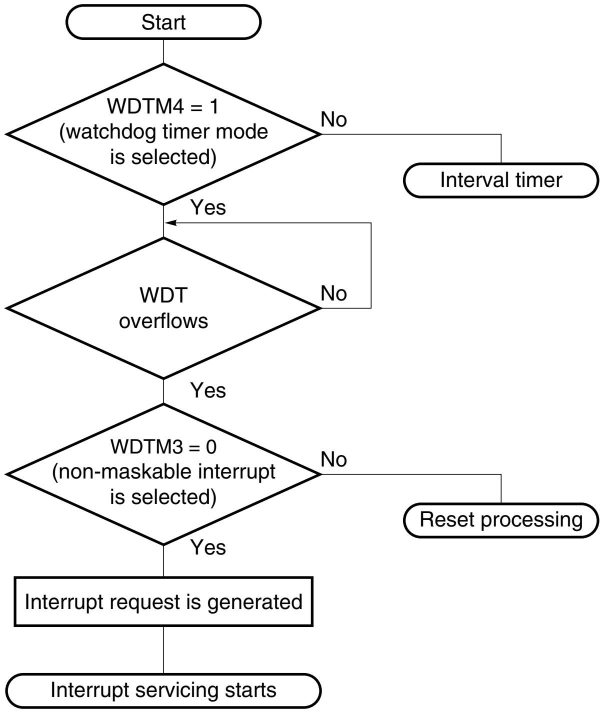
A non-maskable interrupt request is unconditionally acknowledged even when interrupts are disabled. It is not subject to interrupt priority control and takes precedence over all other interrupts.

When a non-maskable interrupt request is acknowledged, the PSW and PC are saved to the stack in that order, the IE flag is reset to 0, the contents of the vector table are loaded to the PC, and then program execution branches.

Figure 12-8 shows the flow from non-maskable interrupt request generation to acknowledgment, Figure 12-9 shows the timing of non-maskable interrupt acknowledgment, and Figure 12-10 shows the acknowledgment operation when a number of non-maskable interrupts are generated.

Caution During non-maskable interrupt service program execution, do not input another non-maskable interrupt request; if it is input, the service program will be interrupted and the new non-maskable interrupt request will be acknowledged.

Figure 12-8. Flow from Generation of Non-Maskable Interrupt Request to Acknowledgment


Start
WDTM4 = 1 (watchdog timer mode is selected)
No
Interval timer
Yes
WDT overflows
No
Yes
WDTM3 = 0 (non-maskable interrupt is selected)
No
Reset processing
Yes
Interrupt request is generated
Interrupt servicing starts

WDTM: Watchdog timer mode register
WDT: Watchdog timer

Figure 12-9. Timing of Non-Maskable Interrupt Request Acknowledgment

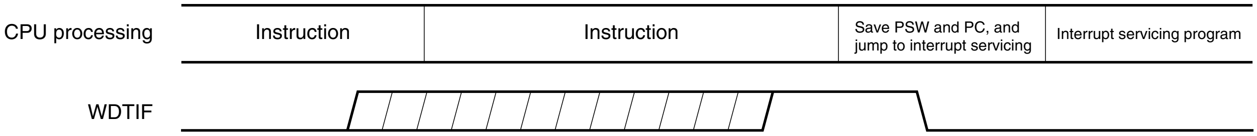
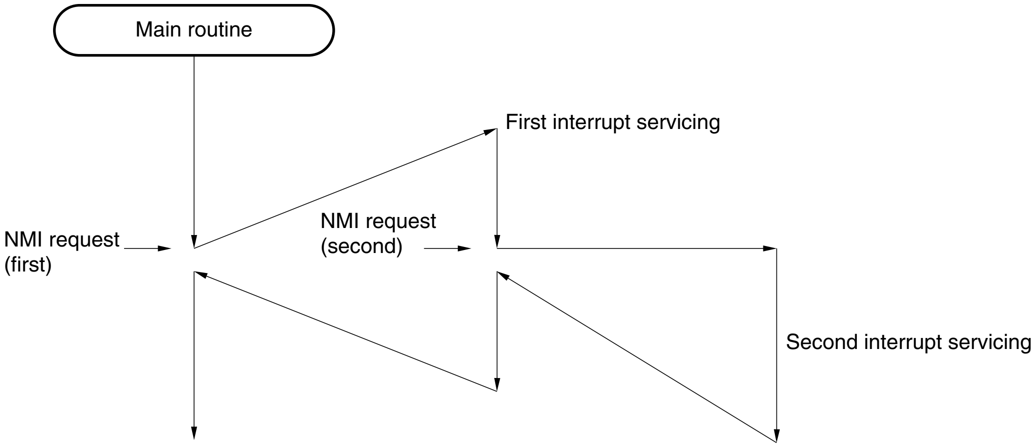

CPU processing
Instruction
Instruction
Save PSW and PC, and jump to interrupt servicing
Interrupt servicing program
WDTIF

Figure 12-10. Non-Maskable Interrupt Request Acknowledgment



12.4.2 Maskable interrupt request acknowledgment operation

A maskable interrupt request can be acknowledged when the interrupt request flag is set to 1 and the corresponding interrupt mask flag is cleared to 0. A vectored interrupt request is acknowledged in the interrupt enabled status (when the IE flag is set to 1).

The time required to start the interrupt servicing after a maskable interrupt request has been generated is shown in Table 12-3.

Refer to Figures 12-12 and 12-13 for the timing of interrupt request acknowledgment.

Table 12-3. Time from Generation of Maskable Interrupt Request to Servicing

Minimum Time	Maximum Time[Note]
9 clocks	19 clocks

Note The wait time is maximum when an interrupt request is generated immediately before the BT or BF instruction.

Remark 1 clock: $\frac{1}{f_{CPU}}$ (f_{CPU}: CPU clock)

When two or more maskable interrupt requests are generated at the same time, they are acknowledged starting from the one assigned the highest priority by the priority specification flag.

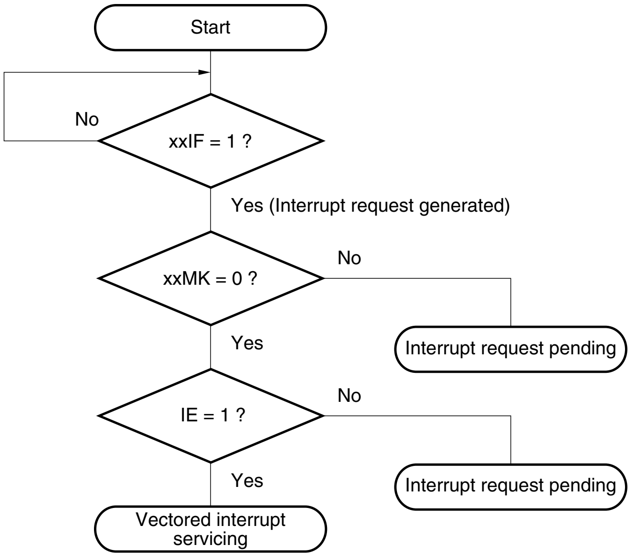
A pending interrupt is acknowledged when the status in which it can be acknowledged is set.

Figure 12-11 shows the algorithm of interrupt request acknowledgment.

When a maskable interrupt request is acknowledged, the PSW and PC are saved to the stack in that order, the IE flag is reset to 0, the data in the vector table determined for each interrupt request is loaded to the PC, and execution branches.

To return from interrupt servicing, use the RETI instruction.

Figure 12-11. Interrupt Request Acknowledgment Program Algorithm

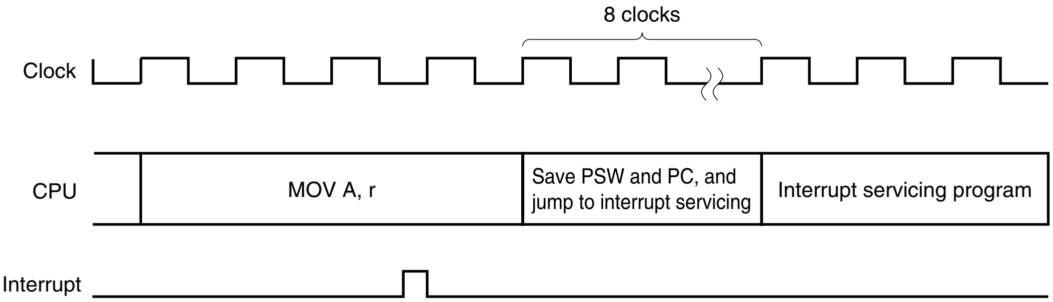

Start
No
xxIF = 1 ?
Yes (Interrupt request generated)
xxMK = 0 ?
No
Yes
Interrupt request pending
IE = 1 ?
No
Yes
Interrupt request pending
Vectored interrupt servicing

xxIF: Interrupt request flag

xxMK: Interrupt mask flag

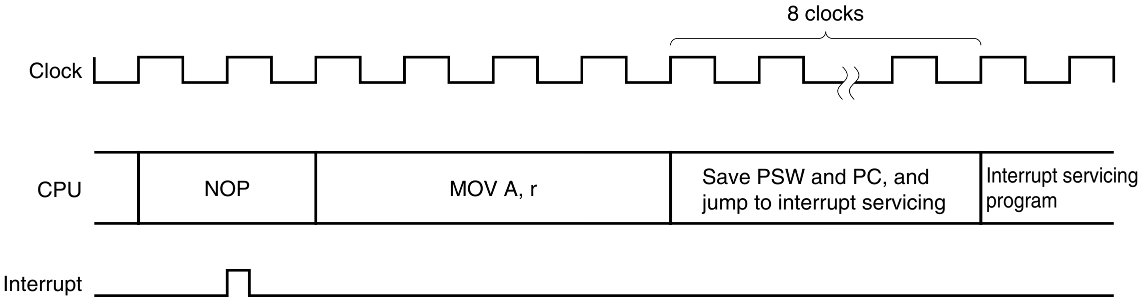
IE: Flag to control maskable interrupt request acknowledgment (1 = Enabled, 0 = Disabled)

Figure 12-12. Interrupt Request Acknowledgment Timing (Example: MOV A, r)


8 clocks
Clock
CPU
MOV A, r
Save PSW and PC, and jump to interrupt servicing
Interrupt servicing program
Interrupt

If the interrupt request has generated an interrupt request flag (XXIF) by the time the instruction clocks under execution, n clocks (n = 4 to 10), are n – 1, interrupt request acknowledgment processing will start following the completion of the instruction under execution. Figure 12-12 shows an example using the 8-bit data transfer instruction MOV A, r. Because this instruction is executed in 4 clocks, if an interrupt request is generated between the start of execution and the 3rd clock, interrupt request acknowledgment processing will take place following the completion of MOV A, r.

Figure 12-13. Interrupt Request Acknowledgment Timing (When Interrupt Request Flag Is Generated in Final Clock Under Execution)


8 clocks
Clock
CPU
NOP
MOV A, r
Save PSW and PC, and jump to interrupt servicing
Interrupt servicing program
Interrupt

If the interrupt request flag (XXIF) is generated in the final clock of the instruction, interrupt request acknowledgment processing will begin after execution of the next instruction is complete.

Figure 12-13 shows an example whereby an interrupt request was generated in the 2nd clock of NOP (a 2-clock instruction). In this case, the interrupt request will be serviced after execution of MOV A, r, which follows NOP, is complete.

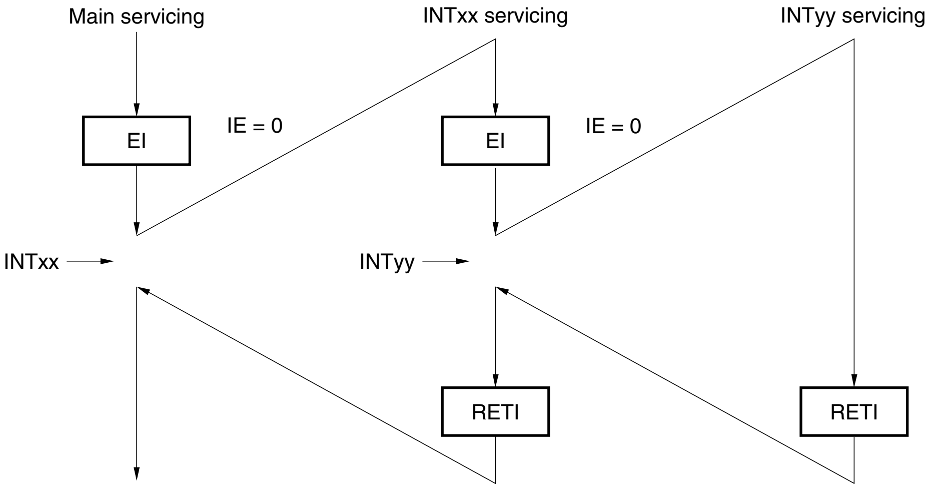
Caution When interrupt request flag register 0 (IF0) or interrupt mask flag register 0 (MK0) is being accessed, interrupt requests will be held pending.

12.4.3 Multiple interrupt servicing

Multiple interrupt servicing, in which another interrupt request is acknowledged while an interrupt request being serviced, can be executed using the priority order. If multiple interrupts are generated at the same time, they are serviced in the order according to the priority assigned to each interrupt request in advance (refer to **Table 12-1**).

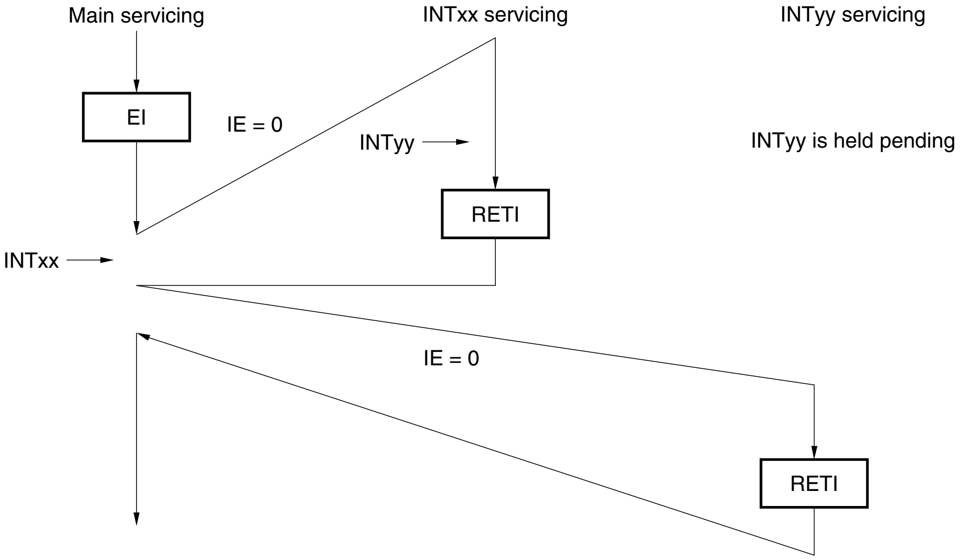
Figure 12-14. Example of Multiple Interrupt Servicing

Example 1. Acknowledging multiple interrupts



The interrupt request INTyy is acknowledged during the servicing of interrupt INTxx and multiple interrupt servicing is performed. Before each interrupt request is acknowledged, the EI instruction is issued and interrupt requests are enabled.

Example 2. Multiple interrupt servicing is not performed because interrupts are disabled



Because interrupt requests are disabled (the EI instruction has not been issued) in the interrupt INTxx servicing, the interrupt request INTyy is not acknowledged and multiple interrupt servicing is not performed. INTyy is held pending and is acknowledged after INTxx servicing is completed.

IE = 0: Interrupt requests disabled

12.4.4 Putting interrupt requests on hold

If an interrupt request (such as a maskable, non-maskable, or external interrupt) is generated when a certain type of instruction is being executed, the interrupt request will not be acknowledged until the instruction is completed. Such instructions (interrupt request pending instructions) are as follows.

- Instructions that manipulate interrupt request flag register 0 (IF0)
- Instructions that manipulate interrupt mask flag register 0 (MK0)

CHAPTER 13 STANDBY FUNCTION

13.1 Standby Function and Configuration

13.1.1 Standby function

The standby function is used to reduce the power consumption of the system and can be effected in the following two modes.

(1) HALT mode

This mode is set when the HALT instruction is executed. The HALT mode stops the operation clock of the CPU. The system clock oscillator continues oscillating. This mode does not reduce the power consumption as much as the STOP mode, but is useful for resuming processing immediately when an interrupt request is generated, or for intermittent operations.

(2) STOP mode

This mode is set when the STOP instruction is executed. The STOP mode stops the main system clock oscillator and stops the entire system. The power consumption of the CPU can be substantially reduced in this mode.

The data memory can be retained at a low voltage (V_{DD} = 1.8 V). Therefore, this mode is useful for retaining the contents of the data memory at an extremely low current.

The STOP mode can be released by an interrupt request, so this mode can be used for intermittent operation. However, some time is required until the system clock oscillator stabilizes after the STOP mode has been released. If processing must be resumed immediately by using an interrupt request, therefore, use the HALT mode.

In both modes, the previous contents of the registers, flags, and data memory before setting the standby mode are all retained. In addition, the statuses of the output latch of the I/O ports and output buffer are also retained.

Caution Be sure to stop the operations of the peripheral hardware before executing the STOP instruction.

13.1.2 Register controlling standby function

The wait time after the STOP mode is released upon interrupt request until oscillation stabilizes is controlled with the oscillation stabilization time selection register (OSTS).

★ OSTS is set with an 8-bit or 1-bit memory manipulation instruction.

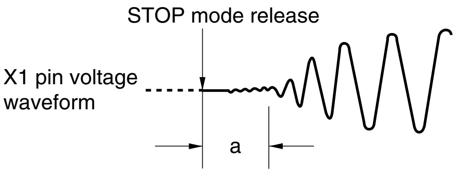
$\overline{\text{RESET}}$ input sets OSTS to 04H. Note that the time required for oscillation to stabilize after $\overline{\text{RESET}}$ input is $2^{15}/f_X$, independent of OSTS.

Figure 13-1. Format of Oscillation Stabilization Time Selection Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0	FFFAH	04H	R/W

OSTS2	OSTS1	OSTS0	
0	0	0	$2^{12}/f_X$ (819 μs)
0	1	0	$2^{15}/f_X$ (6.55 ms)
1	0	0	$2^{17}/f_X$ (26.2 ms)
Other than above			Setting prohibited

Caution The wait time after the STOP mode is released does not include the time from STOP mode release to clock oscillation start ("a" in the figure below), regardless of whether STOP mode is released by $\overline{\text{RESET}}$ input or by interrupt generation.



Remarks 1. f_X: Main system clock oscillation frequency

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

13.2 Standby Function Operation

13.2.1 HALT mode

(1) HALT mode

The HALT mode is set by executing the HALT instruction.

The operation statuses in the HALT mode are shown in the following table.

Table 13-1. Operation Statuses in HALT Mode

Item	HALT Mode Operation Status During Main System Clock Operation		HALT Mode Operation Status During Subsystem Clock Operation	
	Subsystem Clock Operating	Subsystem Clock Stopped	Main System Clock Operating	Main System Clock Stopped
Main system clock	Can be oscillated			Oscillation stopped
CPU	Operation stopped			
Ports (output latches)	Status before HALT mode setting retained			
8-bit timer 30, 40	Operable			Operation stopped
Watch timer	Operable	Operable[Note 1]	Operable	Operable[Note 2]
Watchdog timer	Operable		Operation stopped	
Power-on-clear circuit	Operable			
Key return circuit	Operable[Note 3]			
A/D converter	Operable			Operation stopped
LCD controller/driver	Operable[Note 4]	Operable[Notes 1, 4]	Operable[Note 4]	Operable[Notes 2, 4]
External interrupts	Operable[Note 3]			

Notes 1. Operation is enabled when the main system clock is selected

2. Operation is enabled when the subsystem clock is selected

3. Operation is enabled only for a maskable interrupt that is not masked

4. The HALT instruction can be set after display instruction execution

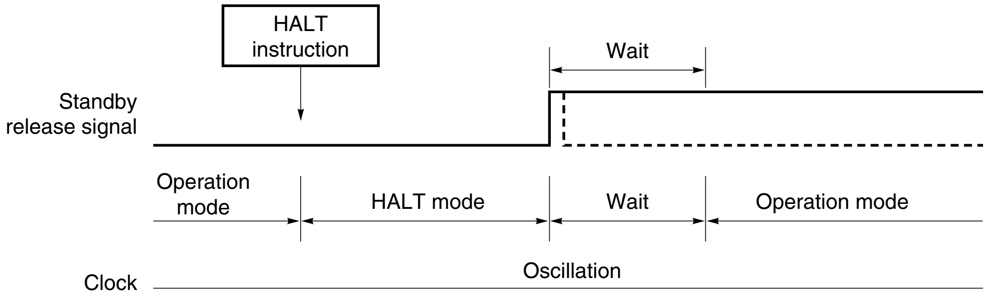
(2) Releasing HALT mode

The HALT mode can be released by the following three sources.

(a) Releasing by unmasked interrupt request

The HALT mode is released by an unmasked interrupt request. In this case, if interrupts are enabled to be acknowledged, vectored interrupt servicing is performed. If interrupts are disabled, the instruction at the next address is executed.

Figure 13-2. Releasing HALT Mode by Interrupt



Remarks 1. The broken lines indicate the case where the interrupt request that has released the standby mode is acknowledged.

2. The wait time is as follows:

- When vectored interrupt servicing is performed: 9 to 10 clocks
- When vectored interrupt servicing is not performed: 1 to 2 clocks

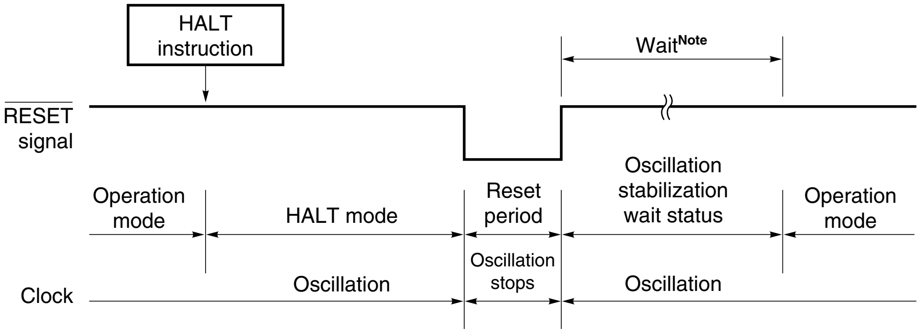
(b) Releasing by non-maskable interrupt request

The HALT mode is released regardless of whether interrupts are enabled or disabled, and vectored interrupt servicing is performed.

(c) Releasing by $\overline{\text{RESET}}$ input

When the HALT mode is released by the $\overline{\text{RESET}}$ signal, execution branches to the reset vector address in the same manner as the ordinary reset operation, and program execution is started.

Figure 13-3. Releasing HALT Mode by $\overline{\text{RESET}}$ Input



Note 2^{15}/fx: 6.55 ms (@ fx = 5.0 MHz operation)

Remark fx: Main system clock oscillation frequency

Table 13-2. Operation After Releasing HALT Mode

Releasing Source	MK××	IE	Operation
Maskable interrupt request	0	0	Next address instruction is executed
	0	1	Interrupt servicing is executed
	1	×	HALT mode is retained
Non-maskable interrupt request	–	×	Interrupt servicing is executed
$\overline{\text{RESET}}$ input	–	–	Reset processing is executed

×: Don’t care

13.2.2 STOP mode

(1) STOP mode

The STOP mode is set by executing the STOP instruction.

Caution Because the standby mode can be released by an interrupt request signal, the standby mode is released as soon as it is set if there is an interrupt source whose interrupt request flag is set and interrupt mask flag is reset. When the STOP mode is set, therefore, the HALT mode is set immediately after the STOP instruction has been executed, the wait time set by the oscillation stabilization time selection register (OSTS) elapses, and then the operation mode is set.

The operation statuses in the STOP mode are shown in the following table.

Table 13-3. Operation Statuses in STOP Mode

Item	STOP Mode Operation Status During Main System Clock Operation	
	Subsystem Clock Operating	Subsystem Clock Stopped
Main system clock	Oscillation stopped	
CPU	Operation stopped	
Ports (output latches)	Status before STOP mode setting retained	
8-bit timer 30, 40	Operation stopped	
Watch timer	Operable[Note 1]	Operation stopped
Watchdog timer	Operation stopped	
Power-on-clear circuit	Operable	
Key return circuit	Operable[Note 2]	
A/D converter	Operation stopped	
LCD controller/driver	Operable[Note 1]	Operation stopped[Note 3]
External interrupts	Operable[Note 2]	

Notes 1. Operation is enabled when the subsystem clock is selected.

2. Operation is enabled only for a maskable interrupt that is not masked

3. Before setting the STOP mode, disable display and select the static mode (refer to **10.3 (1) LCD display mode register 0 (LCDM0)**).

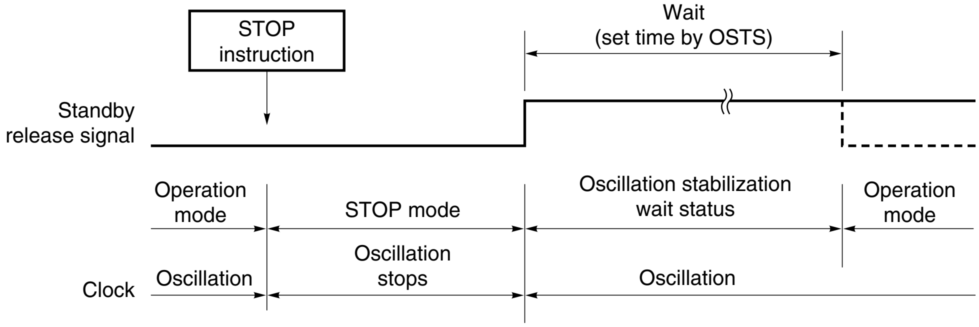
(2) Releasing STOP mode

The STOP mode can be released by the following two sources.

(a) Releasing by unmasked interrupt request

The STOP mode can be released by an unmasked interrupt request. In this case, if interrupts are enabled to be acknowledged, vectored interrupt servicing is performed, after the oscillation stabilization time has elapsed. If interrupts are disabled, the instruction at the next address is executed.

Figure 13-4. Releasing STOP Mode by Interrupt

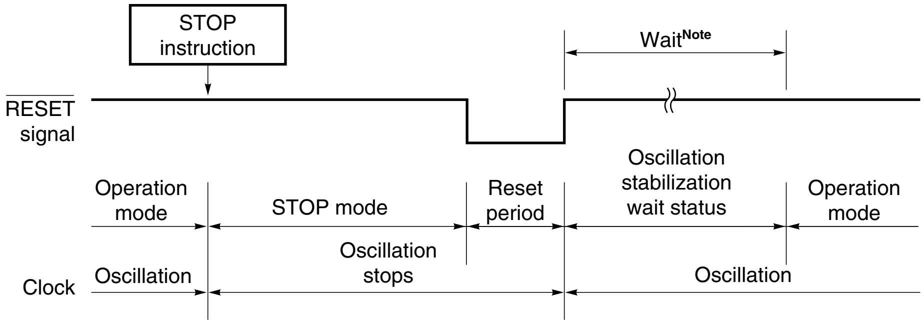


Remark The broken line indicates the case where the interrupt request that has released the standby mode is acknowledged.

(b) Releasing by $\overline{\text{RESET}}$ input

When the STOP mode is released by the $\overline{\text{RESET}}$ signal, the reset operation is performed after the oscillation stabilization time has elapsed.

Figure 13-5. Releasing STOP Mode by $\overline{\text{RESET}}$ Input



Note 2^{15}/fx: 6.55 ms (@ fx = 5.0 MHz operation)

Remark fx: Main system clock oscillation frequency

Table 13-4. Operation After Releasing STOP Mode

Releasing Source	MK××	IE	Operation
Maskable interrupt request	0	0	Next address instruction is executed
	0	1	Interrupt servicing is executed
	1	×	STOP mode is retained
$\overline{\text{RESET}}$ input	–	–	Reset processing is executed

×: Don’t care

CHAPTER 14 RESET FUNCTION

The following three operations are available to generate reset signals.

(1) External reset signal input via $\overline{\text{RESET}}$ pin
(2) Internal reset by detection of watchdog timer program loop time
(3) Internal reset using power-on-clear circuit (POC)

The external and internal reset signals are functionally equivalent. When $\overline{\text{RESET}}$ is input, program execution begins from the addresses written at 0000H and 0001H.

If a low-level signal is applied to the $\overline{\text{RESET}}$ pin, or if the watchdog timer overflows, a reset occurs, causing each item of the hardware to enter the states listed in Table 14-1. While a reset is being applied, or while the oscillation frequency is stabilizing immediately after the end of a reset sequence, each pin remains in the high-impedance state.

If a high-level signal is applied to the $\overline{\text{RESET}}$ pin, the reset sequence is terminated, and program execution is started after the oscillation stabilization time has elapsed. A reset sequence caused by a watchdog timer overflow is terminated automatically and program execution is started after the oscillation stabilization time has elapsed.

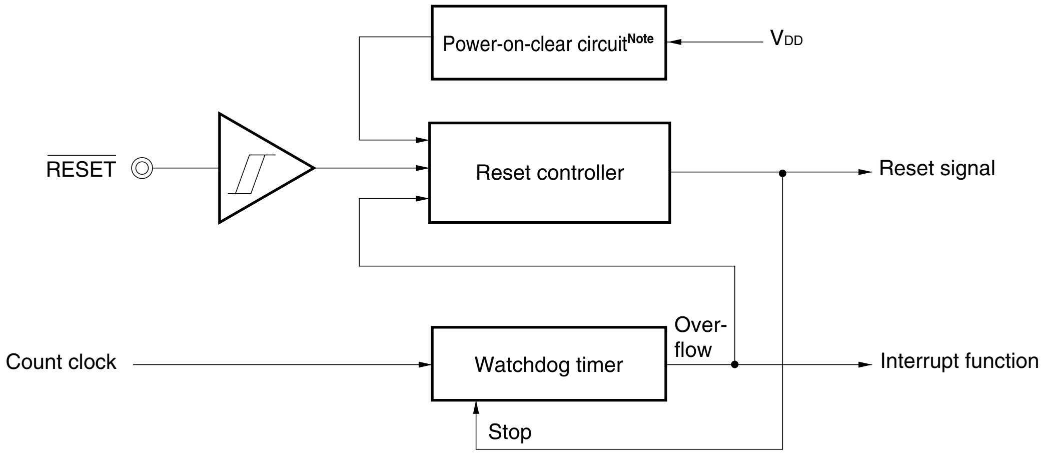
Reset by power-on clear (POC[Note]) is cleared if the supply voltage rises beyond a specific level, and program execution is started after the oscillation stabilization time has elapsed.

Note Enabled in mask ROM versions (μPD789462, 789464, 789466, and 789467) only when POC circuit usage is selected by a mask option.

Cautions 1. For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin.

2. When the STOP mode is cleared by reset, the STOP mode contents are held during reset input. However, the port pins become high impedance.

Figure 14-1. Block Diagram of Reset Function



Note Enabled in mask ROM versions (μPD789462, 789464, 789466, and 789467) only when POC circuit usage is selected by a mask option.

Figure 14-2. Reset Timing by $\overline{\text{RESET}}$ Input

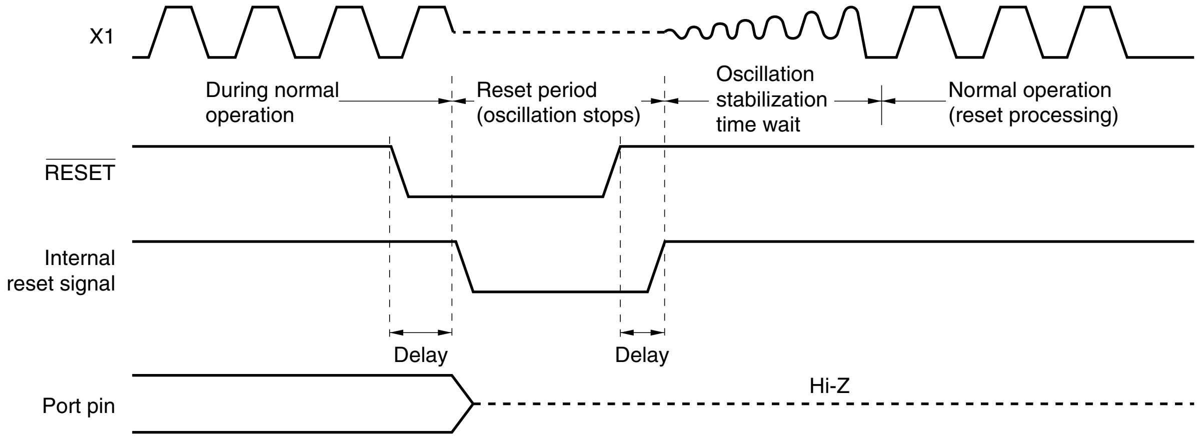

X1
During normal operation
Reset period (oscillation stops)
Oscillation stabilization time wait
Normal operation (reset processing)
RESET
Internal reset signal
Delay
Delay
Port pin
Hi-Z

Figure 14-3. Reset Timing by Overflow in Watchdog Timer

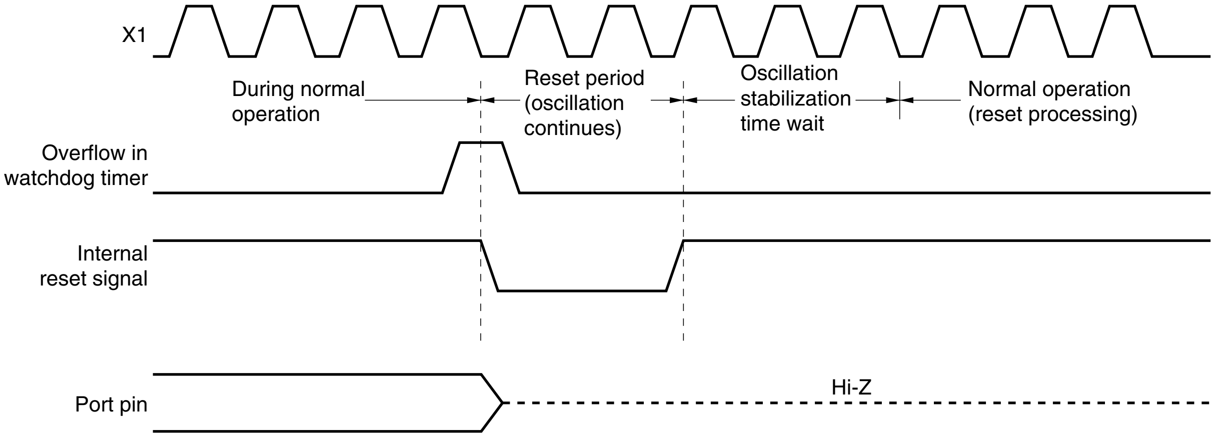

X1
During normal operation
Reset period (oscillation continues)
Oscillation stabilization time wait
Normal operation (reset processing)
Overflow in watchdog timer
Internal reset signal
Port pin
Hi-Z

Figure 14-4. Reset Timing by $\overline{\text{RESET}}$ Input in STOP Mode

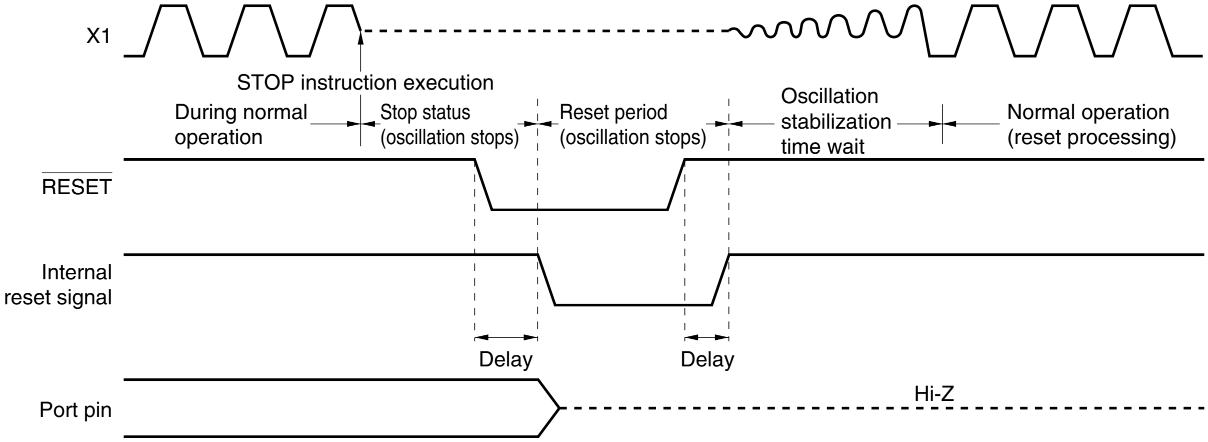
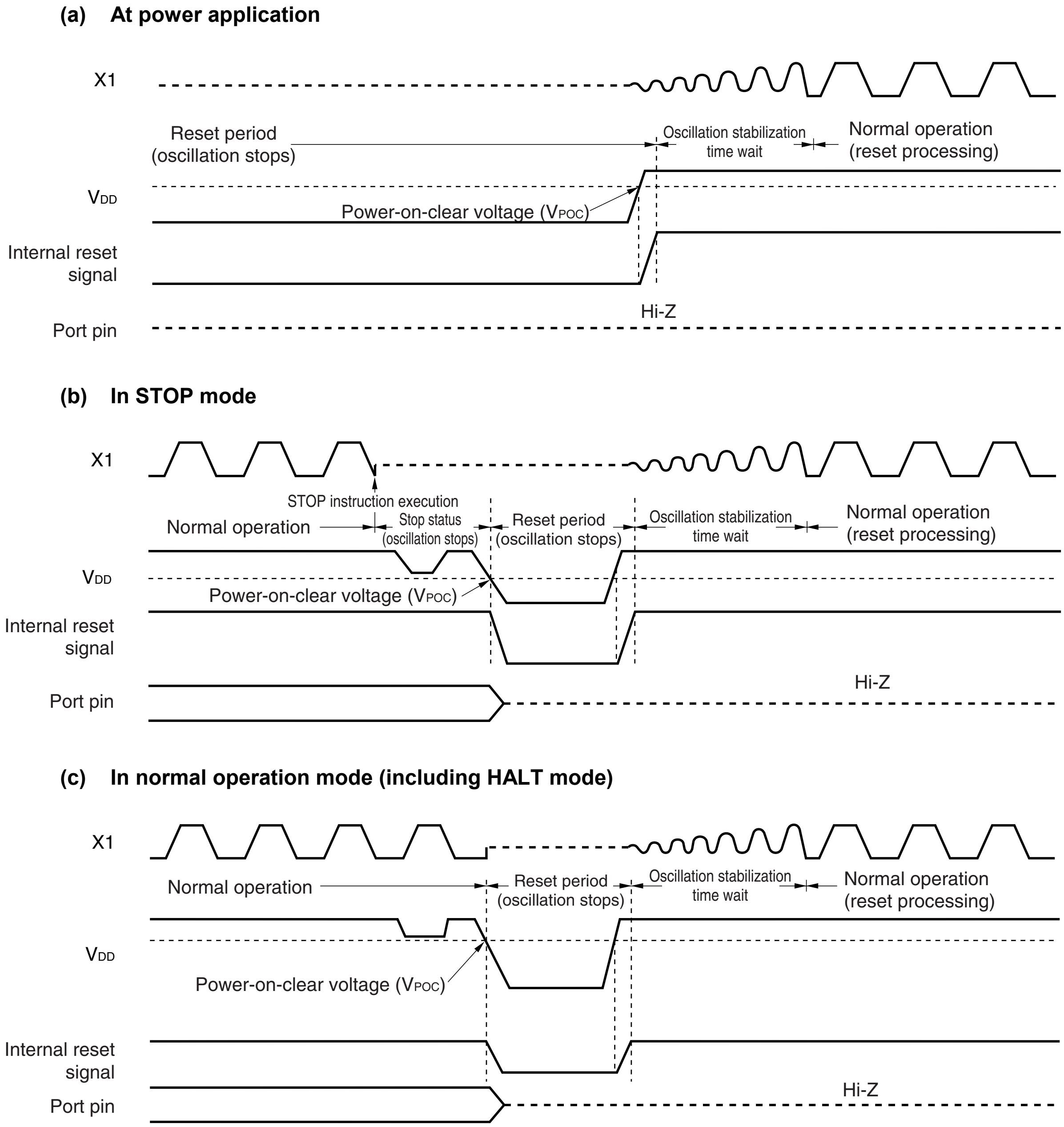

X1
STOP instruction execution
During normal operation
Stop status (oscillation stops)
Reset period (oscillation stops)
Oscillation stabilization time wait
Normal operation (reset processing)
RESET
Internal reset signal
Delay
Delay
Port pin
Hi-Z

Figure 14-5. Reset Timing by Power-on Clear



(a) At power application

(b) In STOP mode

(c) In normal operation mode (including HALT mode)

Table 14-1. Hardware Status After Reset

Hardware		Status After Reset
Program counter (PC)[Note 1]		Contents of reset vector table (0000H, 0001H) set
Stack pointer (SP)		Undefined
Program status word (PSW)		02H
RAM	Data memory	Undefined[Note 2]
	General-purpose registers	Undefined[Note 2]
Ports (P0, P1, P4, P6) (output latches)		00H
Port mode registers (PM0, PM1, PM4, PM6)		FFH
Port function register 8 (PF8)		00H
Pull-up resistor option register 0 (PU0)		00H
Processor clock control register (PCC)		02H
Suboscillation mode register (SCKM)		00H
Subclock control register (CSS)		00H
Oscillation stabilization time selection register (OSTS)		04H
8-bit timer 30, 40	Timer counters (TM30, TM40)	00H
	Compare registers (CR30, CR40, CRH40)	Undefined
	Mode control registers (TMC30, TMC40)	00H
	Carrier generator output control register (TCA40)	00H
Watch timer	Mode control register (WTM)	00H
Watchdog timer	Mode register (WDTM)	00H
	Clock selection register (TCL2)	00H
A/D converter	Mode register 0 (ADM0)	00H
	Input selection register 0 (ADS0)	00H
	Conversion result register 0 (ADCR0)	Undefined
LCD controller/driver	Display mode register 0 (LCDM0)	00H[Note 3]
	Clock control register 0 (LCDC0)	00H
	Voltage boost control register 0 (LCDVA0)	00H
Interrupts	Request flag register 0 (IF0)	00H
	Mask flag register 0 (MK0)	FFH
	External interrupt mode register 0 (INTM0)	00H
	Key return mode register 00 (KRM00)	00H

Notes 1. While a reset signal is being input, and during the oscillation stabilization time wait, only the contents of the PC will be undefined; the remainder of the hardware will be the same state as after reset.

2. In standby mode, RAM enters the hold state after reset.

3. Bit 2 (LCDM02) must be set to 1 after reset.

CHAPTER 15 μPD78F9468

The μPD78F9468 is available as the flash memory version of the μPD789467 Subseries.

The μPD78F9468 is a version with the internal ROM of the μPD789462, 789464, 789466, 789467 replaced with flash memory. The differences between the μPD78F9468 and the mask ROM versions are shown in Table 15-1.

Table 15-1. Differences Between μPD78F9468 and Mask ROM Versions

Part Number		Flash Memory Version	Mask ROM Version			
Item		μPD78F9468	μPD789462	μPD789464	μPD789466	μPD789467
Internal memory	ROM	32 KB (flash memory)	4 KB	8 KB	16 KB	24 KB
	High-speed RAM	512 bytes	256 bytes		512 bytes	
	LCD display RAM	23 × 4 bits				
LCD controller/driver		Refer to **Table 15-2 Differences in LCD Controller/Driver of μPD78F9468 and Mask ROM Version**.				
Power-on-clear (POC) circuit		Always operating	Enable/disable is selected by a mask option.			
IC0 pin		Not provided	Provided			
V_{PP} pin		Provided	Not provided			
Electrical specifications		Refer to **CHAPTER 18 ELECTRICAL SPECIFICATIONS**.				

Caution There are differences in noise immunity and noise radiation between the flash memory and mask ROM versions. When pre-producing an application set with the flash memory version and then mass-producing it with the mask ROM version, be sure to conduct sufficient evaluations for the commercial samples (not engineering samples) of the mask ROM version.

The μPD78F9468 and mask ROM version differ in the on-chip LCD controller/driver macro. The differences in the LCD controller/driver of μPD78F9468 and mask ROM version is shown in Table 15-2.

Table 15-2. Differences in LCD Controller/Driver of μPD78F9468 and Mask ROM Version

Item	Flash Memory Version (μPD78F9468)	Mask ROM Version
Output of pins COM0 to COM3 after reset	Pins COM0 to COM2: GND level output Pin COM3: V_{LC0} level output	Pins COM0 to COM3: GND level output
Bit 4 (LIPS0) of LCD display mode register 0 (LCDM0)	Setting is possible, but invalid	Valid Set (LIPS0 = 1) to enable display and clear (LIPS0 = 0) to disable display.
Output of segment/common pins when display and voltage booster disabled	Segment pin: GND level output Common pin: Two values, V_{LC0} and GND, output Therefore, an abnormal display may occur even though display is disabled. Select the static mode (LCDM02 = 1).	Segment pin: GND level output Common pin: GND level output Therefore display is disabled.
Operation procedure	If a program conforming to the instructions described in **10.4 Setting LCD Controller/Driver** is applied, the same program allows a successful operation both in flash memory and mask ROM versions.	

15.1 Flash Memory Characteristics

Flash memory programming is performed by connecting a dedicated flash programmer (Flashpro III (part no. FL-PR3, PG-FP3)/Flashpro IV (part no. FL-PR4, PG-FP4)) to the target system with the flash memory mounted on the target system (on-board write). A flash memory writing adapter (program adapter), which is a target board used exclusively for programming, is also provided.

Remark FL-PR3, FL-PR4, and the program adapter are products of Naito Densei Machida Mfg. Co., Ltd. (TEL +81-45-475-4191).

Programming using flash memory has the following advantages.

- Software can be modified after the microcontroller is solder-mounted on the target system.
- Distinguishing software facilities low-quantity, varied model production
- Easy data adjustment when starting mass production

15.1.1 Programming environment

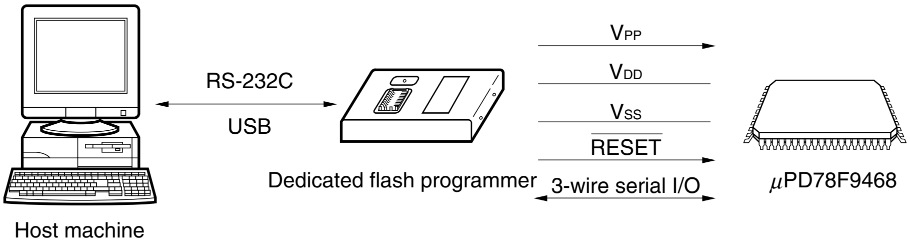
The following shows the environment required for μPD78F9468 flash memory programming.

When Flashpro III (part no. FL-PR3, PG-FP3) or Flashpro IV (Part no. FL-PR4, PG-FP4) is used as a dedicated flash programmer, a host machine is required to control the dedicated flash programmer. Communication between the host machine and flash programmer is performed via RS-232C/USB (Rev. 1.1).

For details, refer the manuals for Flashpro III/Flashpro IV.

Remark USB is supported by Flashpro IV only.

Figure 15-1. Environment for Writing Program to Flash Memory



15.1.2 Communication mode

Use the communication mode shown in Table 15-3 to perform communication between the dedicated flash programmer and μPD78F9468.

★ **Table 15-3. Communication Mode List**

Communication Mode	TYPE Setting[Note 1]					Pins Used	Number of V_{PP} Pulses
	COMM PORT	SIO Clock	CPU Clock		Multiple Rate		
			In Flashpro	On Target Board			
3-wire serial I/O	SIO ch-0 (3-wire, sync)	100 Hz to 1.25 MHz[Note2]	1, 2, 4, or 5 MHz[Notes 3, 4]	1 to 5 MHz[Note 3]	1.0	P00, P01, P02	1

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. When V_{DD} = 2.7 to 5.5 V. 100 Hz to 312.5 kHz when V_{DD} = 1.8 to 2.7 V.

3. The possible setting range differs depending on the voltage. For details, refer to **CHAPTER 18 ELECTRICAL SPECIFICATIONS**.

4. When using Flashpro III, only 2 MHz or 4 MHz can be selected.

Caution Be sure to select a communication mode according to the number of V_{PP} pulses shown in Table 15-3.

Figure 15-2. Communication Mode Selection Format

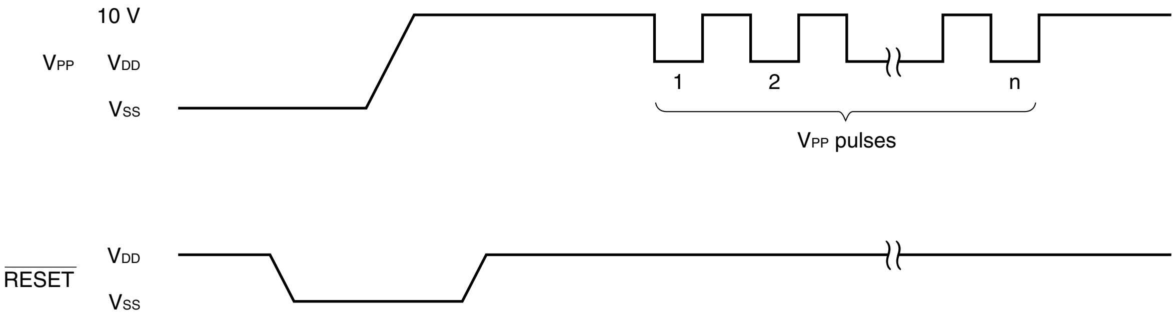
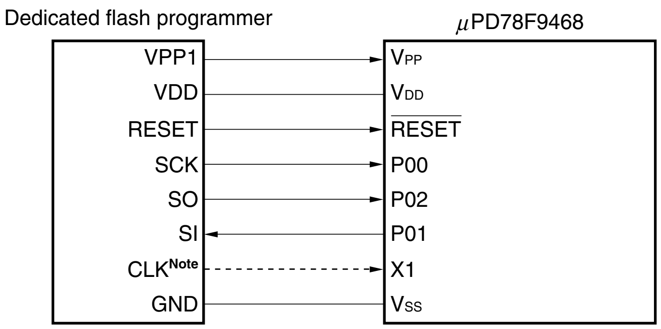


Figure 15-3. Example of Connection with Dedicated Flash Programmer



Note Connect this pin when the system clock is supplied by the dedicated flash programmer. If an oscillator is already connected to the X1 pin, do not connect X1 to the CLK pin.

Caution The VDD pin, if already connected to the power supply, must be connected to the VDD pin of the dedicated flash programmer. Before using the power supply connected to the VDD pin, supply voltage before starting programming.

If Flashpro III/Flashpro IV is used as the dedicated flash programmer, the following signals are generated for the μPD78F9468. For details, refer to the manual of Flashpro III/Flashpro IV.

Table 15-4. Pin Connection List

Signal Name	I/O	Pin Function	Pin Name	3-Wire Serial I/O
VPP1	Output	Write voltage	VPP	◎
VPP2	–	–	–	×
VDD	I/O	VDD voltage generation/voltage monitoring	VDD	◎[Note]
GND	–	Ground	VSS	◎
CLK	Output	Clock output	X1	○
RESET	Output	Reset signal	$\overline{\text{RESET}}$	◎
SI	Input	Reception signal	P01	◎
SO	Output	Transmit signal	P02	◎
SCK	Output	Transfer clock	P00	◎
HS	–	–	–	×

Note VDD voltage must be supplied before programming is started.

Remark ◎: Pin must be connected.
○: If the signal is supplied on the target board, pin does not need to be connected.
×: Pin does not need to be connected.

15.1.3 On-board pin processing

When performing programming on the target system, provide a connector on the target system to connect the dedicated flash programmer.

An on-board function that allows switching between normal operation mode and flash memory programming mode may be required in some cases.

<V_{PP} pin>

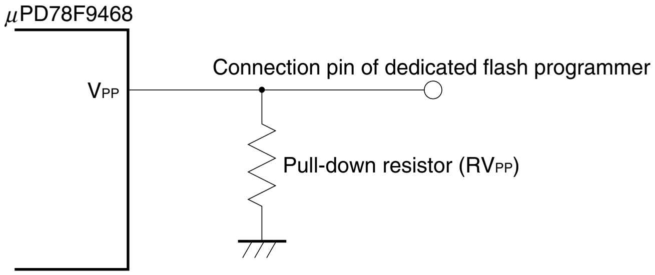
In normal operation mode, input 0 V to the V_{PP} pin. In flash memory programming mode, a write voltage of 10.0 V (TYP.) is supplied to the V_{PP} pin, so perform the following.

(1) Connect a pull-down resistor (RV_{PP} = 10 kΩ) to the V_{PP} pin.

(2) Use the jumper on the board to switch the V_{PP} pin input to either the programmer or directly to GND.

A V_{PP} pin connection example is shown below.

Figure 15-4. V_{PP} Pin Connection Example


μPD78F9468
VPP
Connection pin of dedicated flash programmer
Pull-down resistor (RVPP)

<Serial interface pin>

The following shows the pins used by the serial interface.

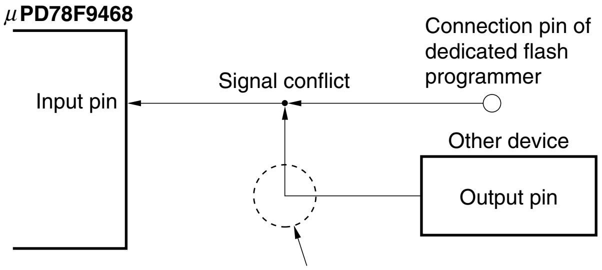
Serial Interface	Pins Used
3-wire serial I/O	P00, P01, P02

When connecting the dedicated flash programmer to a serial interface pin that is connected to another device on-board, signal conflict or abnormal operation of the other device may occur. Care must therefore be taken with such connections.

(1) Signal conflict

If the dedicated flash programmer (output) is connected to a serial interface pin (input) that is connected to another device (output), a signal conflict occurs. To prevent this, isolate the connection with the other device or set the other device to the output high impedance status.

Figure 15-5. Signal Conflict (Input Pin of Serial Interface)

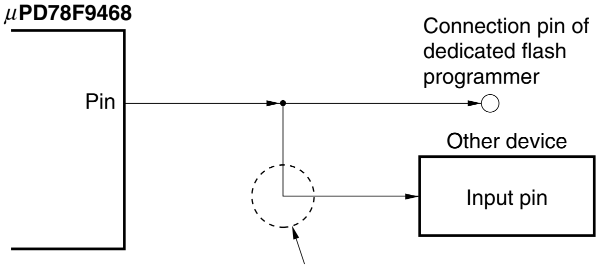


In the flash memory programming mode, the signal output by another device and the signal sent by the dedicated flash programmer conflict; therefore, isolate the signal of the other device.

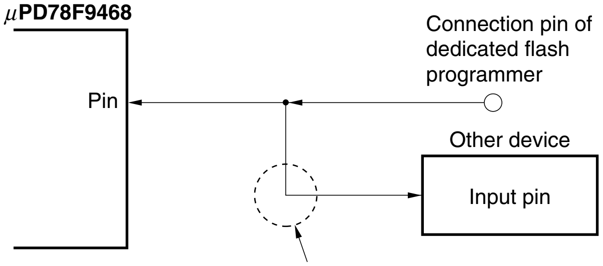
(2) Abnormal operation of other device

If the dedicated flash programmer (output or input) is connected to a serial interface pin (input or output) that is connected to another device (input), a signal is output to the device, and this may cause an abnormal operation. To prevent this abnormal operation, isolate the connection with the other device or set so that the signals input to the other device are ignored.

Figure 15-6. Abnormal Operation of Other Device



If the signal output by the μPD78F9468 affects another device in the flash memory programming mode, isolate the signals of the other device.



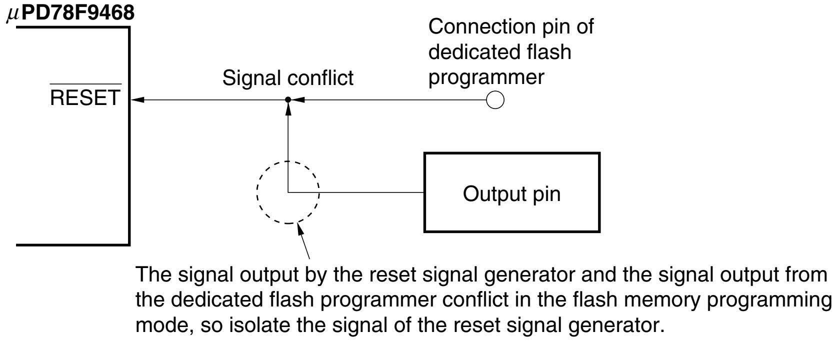
If the signal output by the dedicated flash programmer affects another device in the flash memory programming mode, isolate the signals of the other device.

<$\overline{\text{RESET}}$ 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 15-7. Signal Conflict (Reset Pin)


μPD78F9468
$\overline{\text{RESET}}$
Signal conflict
Connection pin of dedicated flash programmer
Output pin

The signal output by the reset signal generator and the signal output from the dedicated flash programmer conflict in the flash memory programming mode, so isolate the signal of the reset signal generator.

<Port pins>

When the μPD78F9468 enters the flash memory programming mode, all the pins other than those that communicate with the flash programmer are in the same status as immediately after reset.

If the external device does not recognize initial statuses such as the output high impedance status, therefore, connect the external device to V_{DD} or V_{SS} via a resistor.

<Oscillator>

When using the on-board clock, connect X1 and X2 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.

<Power supply>

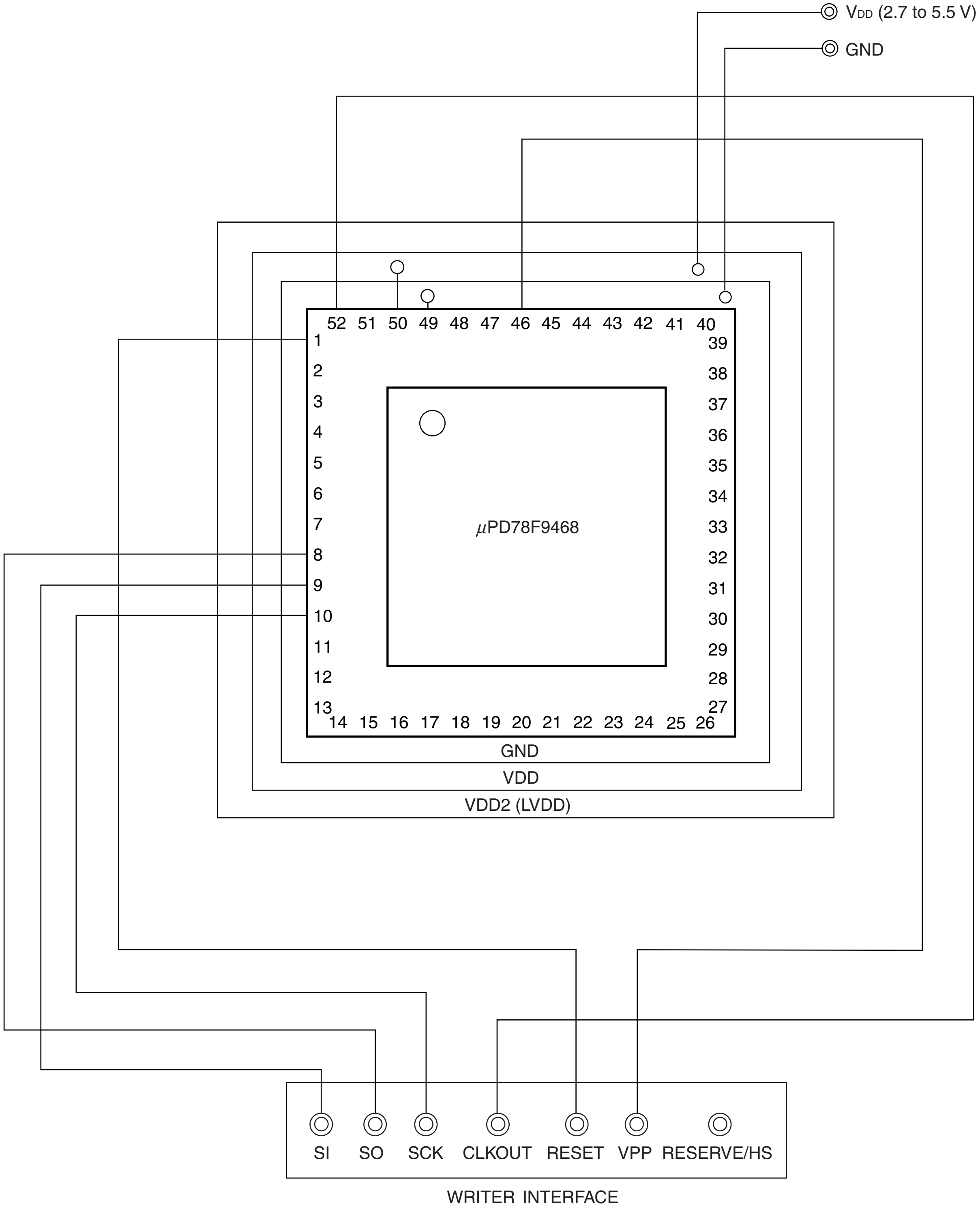
When using the power supply output of the flash programmer, connect the V_{DD} and V_{SS} pins to VDD and GND of the flash programmer, respectively.

When using the on-board power supply, connect it as required in the normal operation mode. Because the flash programmer monitors the voltage, however, VDD of the flash programmer must be connected.

15.1.4 Connection on flash memory writing adapter

The following shows an example of the recommended connection when using the flash memory writing adapter.

Figure 15-8. Wiring Example of Flash Memory Writing Adapter Using 3-Wire Serial I/O Mode



CHAPTER 16 MASK OPTION

The mask ROM versions (μPD789462, 789464, 789466, and 789467) have the following mask option.

- Power-on-clear (POC) circuit
  Use/non use of the POC circuit can be selected.
  <1> POC circuit used
  <2> POC circuit not used

Caution The POC circuit always operates in the flash memory version (μPD78F9468).

CHAPTER 17 INSTRUCTION SET

This chapter lists the instruction set of the μPD789467 Subseries. For details of the operation and machine language (instruction code) of each instruction, refer to **78K/0S Series Instructions User's Manual (U11047E)**.

17.1 Operation

17.1.1 Operand identifiers and description methods

Operands are described in the "Operands" column of each instruction in accordance with the description method of the instruction operand identifier (refer to the assembler specifications for details). When there are two or more description methods, select one of them. Uppercase letters and the symbols #, !, $, and [ ] are keywords and are described as they are. Each symbol has the following meaning.

- #: Immediate data specification
- !: Absolute address specification
- $: Relative address specification
- [ ]: Indirect address specification

In the case of immediate data, describe an appropriate numeric value or a label. When using a label, be sure to describe the #, !, $ and [ ] symbols.

For operand register identifiers r and rp, either functional names (X, A, C, etc.) or absolute names (names in parentheses in the table below: R0, R1, R2, etc.) can be used for description.

Table 17-1. Operand Identifiers and Description Methods

Identifier	Description Method
r rp sfr	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7) AX (RP0), BC (RP1), DE (RP2), HL (RP3) Special-function register symbol
saddr saddrp	FE20H to FF1FH Immediate data or labels FE20H to FF1FH Immediate data or labels (even addresses only)
addr16 addr5	0000H to FFFFH Immediate data or labels (only even addresses for 16-bit data transfer instructions) 0040H to 007FH Immediate data or labels (even addresses only)
word byte bit	16-bit immediate data or label 8-bit immediate data or label 3-bit immediate data or label

Remark See **Table 3-3 Special-Function Registers** for symbols of special-function registers.

17.1.2 Description of “Operation” column

A:	A register; 8-bit accumulator
X:	X register
B:	B register
C:	C register
D:	D register
E:	E register
H:	H register
L:	L register
AX:	AX register pair; 16-bit accumulator
BC:	BC register pair
DE:	DE register pair
HL:	HL register pair
PC:	Program counter
SP:	Stack pointer
PSW:	Program status word
CY:	Carry flag
AC:	Auxiliary carry flag
Z:	Zero flag
IE:	Interrupt request enable flag
NMIS:	Flag indicating non-maskable interrupt servicing in progress
( ):	Memory contents indicated by address or register contents in parentheses
X_H, X_L:	Higher 8 bits and lower 8 bits of 16-bit register
$\wedge$:	Logical product (AND)
$\vee$:	Logical sum (OR)
$\veebar$:	Exclusive logical sum (exclusive OR)
¯:	Inverted data
addr16:	16-bit immediate data or label
jdisp8:	Signed 8-bit data (displacement value)

17.1.3 Description of “Flag” column

(Blank):	Unchanged
0:	Cleared to 0
1:	Set to 1
x:	Set/cleared according to the result
R:	Previously saved value is restored

17.2 Operation List

Mnemonic	Operands	Bytes	Clocks	Operation	Flag		
					Z	AC	CY
MOV	r, #byte	3	6	r ← byte			
	saddr, #byte	3	6	(saddr) ← byte			
	sfr, #byte	3	6	sfr ← byte			
	A, r [Note 1]	2	4	A ← r			
	r, A [Note 1]	2	4	r ← A			
	A, saddr	2	4	A ← (saddr)			
	saddr, A	2	4	(saddr) ← A			
	A, sfr	2	4	A ← sfr			
	sfr, A	2	4	sfr ← A			
	A, !addr16	3	8	A ← (addr16)			
	!addr16, A	3	8	(addr16) ← A			
	PSW, #byte	3	6	PSW ← byte	x	x	x
	A, PSW	2	4	A ← PSW			
	PSW, A	2	4	PSW ← A	x	x	x
	A, [DE]	1	6	A ← (DE)			
	[DE], A	1	6	(DE) ← A			
	A, [HL]	1	6	A ← (HL)			
	[HL], A	1	6	(HL) ← A			
	A, [HL+byte]	2	6	A ← (HL + byte)			
	[HL+byte], A	2	6	(HL + byte) ← A			
XCH	A, X	1	4	A ↔ X			
	A, r [Note 2]	2	6	A ↔ r			
	A, saddr	2	6	A ↔ (saddr)			
	A, sfr	2	6	A ↔ sfr			
	A, [DE]	1	8	A ↔ (DE)			
	A, [HL]	1	8	A ↔ (HL)			
	A, [HL+byte]	2	8	A ↔ (HL + byte)			

Notes 1. Except r = A.

2. Except r = A, X.

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

Mnemonic	Operands	Bytes	Clocks	Operation	Flag		
					Z	AC	CY
MOVW	rp, #word	3	6	rp ← word			
	AX, saddrp	2	6	AX ← (saddrp)			
	saddrp, AX	2	8	(saddrp) ← AX			
	AX, rp [Note]	1	4	AX ← rp			
	rp, AX [Note]	1	4	rp ← AX			
XCHW	AX, rp [Note]	1	8	AX ↔ rp			
ADD	A, #byte	2	4	A, CY ← A + byte	x	x	x
	saddr, #byte	3	6	(saddr), CY ← (saddr) + byte	x	x	x
	A, r	2	4	A, CY ← A + r	x	x	x
	A, saddr	2	4	A, CY ← A + (saddr)	x	x	x
	A, !addr16	3	8	A, CY ← A + (addr16)	x	x	x
	A, [HL]	1	6	A, CY ← A + (HL)	x	x	x
	A, [HL+byte]	2	6	A, CY ← A + (HL + byte)	x	x	x
ADDC	A, #byte	2	4	A, CY ← A + byte + CY	x	x	x
	saddr, #byte	3	6	(saddr), CY ← (saddr) + byte + CY	x	x	x
	A, r	2	4	A, CY ← A + r + CY	x	x	x
	A, saddr	2	4	A, CY ← A + (saddr) + CY	x	x	x
	A, !addr16	3	8	A, CY ← A + (addr16) + CY	x	x	x
	A, [HL]	1	6	A, CY ← A + (HL) + CY	x	x	x
	A, [HL+byte]	2	6	A, CY ← A + (HL + byte) + CY	x	x	x
SUB	A, #byte	2	4	A, CY ← A – byte	x	x	x
	saddr, #byte	3	6	(saddr), CY ← (saddr) – byte	x	x	x
	A, r	2	4	A, CY ← A – r	x	x	x
	A, saddr	2	4	A, CY ← A – (saddr)	x	x	x
	A, !addr16	3	8	A, CY ← A – (addr16)	x	x	x
	A, [HL]	1	6	A, CY ← A – (HL)	x	x	x
	A, [HL+byte]	2	6	A, CY ← A – (HL + byte)	x	x	x

Note Only when rp = BC, DE, or HL.

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

Mnemonic	Operands	Bytes	Clocks	Operation	Flag		
					Z	AC	CY
SUBC	A, #byte	2	4	A, CY ← A – byte – CY	x	x	x
	saddr, #byte	3	6	(saddr), CY ← (saddr) – byte – CY	x	x	x
	A, r	2	4	A, CY ← A – r – CY	x	x	x
	A, saddr	2	4	A, CY ← A – (saddr) – CY	x	x	x
	A, !addr16	3	8	A, CY ← A – (addr16) – CY	x	x	x
	A, [HL]	1	6	A, CY ← A – (HL) – CY	x	x	x
	A, [HL+byte]	2	6	A, CY ← A – (HL + byte) – CY	x	x	x
AND	A, #byte	2	4	A ← A ∧ byte	x		
	saddr, #byte	3	6	(saddr) ← (saddr) ∧ byte	x		
	A, r	2	4	A ← A ∧ r	x		
	A, saddr	2	4	A ← A ∧ (saddr)	x		
	A, !addr16	3	8	A ← A ∧ (addr16)	x		
	A, [HL]	1	6	A ← A ∧ (HL)	x		
	A, [HL+byte]	2	6	A ← A ∧ (HL + byte)	x		
OR	A, #byte	2	4	A ← A ∨ byte	x		
	saddr, #byte	3	6	(saddr) ← (saddr) ∨ byte	x		
	A, r	2	4	A ← A ∨ r	x		
	A, saddr	2	4	A ← A ∨ (saddr)	x		
	A, !addr16	3	8	A ← A ∨ (addr16)	x		
	A, [HL]	1	6	A ← A ∨ (HL)	x		
	A, [HL+byte]	2	6	A ← A ∨ (HL + byte)	x		
XOR	A, #byte	2	4	A ← A $\overline{\vee}$ byte	x		
	saddr, #byte	3	6	(saddr) ← (saddr) $\overline{\vee}$ byte	x		
	A, r	2	4	A ← A $\overline{\vee}$ r	x		
	A, saddr	2	4	A ← A $\overline{\vee}$ (saddr)	x		
	A, !addr16	3	8	A ← A $\overline{\vee}$ (addr16)	x		
	A, [HL]	1	6	A ← A $\overline{\vee}$ (HL)	x		
	A, [HL+byte]	2	6	A ← A $\overline{\vee}$ (HL + byte)	x		

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

Mnemonic	Operands	Bytes	Clocks	Operation	Flag		
					Z	AC	CY
CMP	A, #byte	2	4	A – byte	x	x	x
	saddr, #byte	3	6	(saddr) – byte	x	x	x
	A, r	2	4	A – r	x	x	x
	A, saddr	2	4	A – (saddr)	x	x	x
	A, !addr16	3	8	A – (addr16)	x	x	x
	A, [HL]	1	6	A – (HL)	x	x	x
	A, [HL+byte]	2	6	A – (HL + byte)	x	x	x
ADDW	AX, #word	3	6	AX, CY ← AX + word	x	x	x
SUBW	AX, #word	3	6	AX, CY ← AX – word	x	x	x
CMPW	AX, #word	3	6	AX – word	x	x	x
INC	r	2	4	r ← r + 1	x	x	
	saddr	2	4	(saddr) ← (saddr) + 1	x	x	
DEC	r	2	4	r ← r – 1	x	x	
	saddr	2	4	(saddr) ← (saddr) – 1	x	x	
INCW	rp	1	4	rp ← rp + 1			
DECW	rp	1	4	rp ← rp – 1			
ROR	A, 1	1	2	$(CY, A_7 \leftarrow A_0, A_{m-1} \leftarrow A_m) \times 1$			x
ROL	A, 1	1	2	$(CY, A_0 \leftarrow A_7, A_{m+1} \leftarrow A_m) \times 1$			x
RORC	A, 1	1	2	$(CY \leftarrow A_0, A_7 \leftarrow CY, A_{m-1} \leftarrow A_m) \times 1$			x
ROLC	A, 1	1	2	$(CY \leftarrow A_7, A_0 \leftarrow CY, A_{m+1} \leftarrow A_m) \times 1$			x
SET1	saddr.bit	3	6	(saddr.bit) ← 1			
	sfr.bit	3	6	sfr.bit ← 1			
	A.bit	2	4	A.bit ← 1			
	PSW.bit	3	6	PSW.bit ← 1	x	x	x
	[HL].bit	2	10	(HL).bit ← 1			
CLR1	saddr.bit	3	6	(saddr.bit) ← 0			
	sfr.bit	3	6	sfr.bit ← 0			
	A.bit	2	4	A.bit ← 0			
	PSW.bit	3	6	PSW.bit ← 0	x	x	x
	[HL].bit	2	10	(HL).bit ← 0			
SET1	CY	1	2	CY ← 1			1
CLR1	CY	1	2	CY ← 0			0
NOT1	CY	1	2	CY ← CY			x

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

Mnemonic	Operands	Bytes	Clocks	Operation	Flag		
					Z	AC	CY
CALL	!addr16	3	6	(SP – 1) ← (PC + 3)$_H$, (SP – 2) ← (PC + 3)$_L$, PC ← addr16, SP ← SP – 2			
CALLT	[addr5]	1	8	(SP – 1) ← (PC + 1)$_H$, (SP – 2) ← (PC + 1)$_L$, PC$_H$ ← (00000000, addr5 + 1), PC$_L$ ← (00000000, addr5), SP ← SP – 2			
RET		1	6	PC$_H$ ← (SP + 1), PC$_L$ ← (SP), SP ← SP + 2			
RETI		1	8	PC$_H$ ← (SP + 1), PC$_L$ ← (SP), PSW ← (SP + 2), SP ← SP + 3, NMIS ← 0	R	R	R
PUSH	PSW	1	2	(SP – 1) ← PSW, SP ← SP – 1			
	rp	1	4	(SP – 1) ← rp$_H$, (SP – 2) ← rp$_L$, SP ← SP – 2			
POP	PSW	1	4	PSW ← (SP), SP ← SP + 1	R	R	R
	rp	1	6	rp$_H$ ← (SP + 1), rp$_L$ ← (SP), SP ← SP + 2			
MOVW	SP, AX	2	8	SP ← AX			
	AX, SP	2	6	AX ← SP			
BR	!addr16	3	6	PC ← addr16			
	$addr16	2	6	PC ← PC + 2 + jdisp8			
	AX	1	6	PC$_H$ ← A, PC$_L$ ← X			
BC	$saddr16	2	6	PC ← PC + 2 + jdisp8 if CY = 1			
BNC	$saddr16	2	6	PC ← PC + 2 + jdisp8 if CY = 0			
BZ	$saddr16	2	6	PC ← PC + 2 + jdisp8 if Z = 1			
BNZ	$saddr16	2	6	PC ← PC + 2 + jdisp8 if Z = 0			
BT	saddr.bit, $addr16	4	10	PC ← PC + 4 + jdisp8 if (saddr.bit) = 1			
	sfr.bit, $addr16	4	10	PC ← PC + 4 + jdisp8 if sfr.bit = 1			
	A.bit, $addr16	3	8	PC ← PC + 3 + jdisp8 if A.bit = 1			
	PSW.bit, $addr16	4	10	PC ← PC + 4 + jdisp8 if PSW.bit = 1			
BF	saddr.bit, $addr16	4	10	PC ← PC + 4 + jdisp8 if (saddr.bit) = 0			
	sfr.bit, $addr16	4	10	PC ← PC + 4 + jdisp8 if sfr.bit = 0			
	A.bit, $addr16	3	8	PC ← PC + 3 + jdisp8 if A.bit = 0			
	PSW.bit, $addr16	4	10	PC ← PC + 4 + jdisp8 if PSW.bit = 0			
DBNZ	B, $addr16	2	6	B ← B – 1, then PC ← PC + 2 + jdisp8 if B ≠ 0			
	C, $addr16	2	6	C ← C – 1, then PC ← PC + 2 + jdisp8 if C ≠ 0			
	saddr, $addr16	3	8	(saddr) ← (saddr) – 1, then PC ← PC + 3 + jdisp8 if (saddr) ≠ 0			
NOP		1	2	No Operation			
EI		3	6	IE ← 1 (Enable interrupt)			
DI		3	6	IE ← 0 (Disable interrupt)			
HALT		1	2	Set HALT mode			
STOP		1	2	Set STOP mode			

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

17.3 Instructions Listed by Addressing Type

(1) 8-bit instructions

MOV, XCH, ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP, INC, DEC, ROR, ROL, RORC, ROLC, PUSH, POP, DBNZ

2nd Operand 1st Operand	#byte	A	r	sfr	saddr	!addr16	PSW	[DE]	[HL]	[HL+byte]	$addr16	1	None
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV[Note] XCH[Note] ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP		ROR ROL RORC ROLC	
r	MOV	MOV											INC DEC
B, C											DBNZ		
sfr	MOV	MOV											
saddr	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV									DBNZ		INC DEC
!addr16		MOV											
PSW	MOV	MOV											PUSH POP
[DE]		MOV											
[HL]		MOV											
[HL+byte]		MOV											

Note Except r = A.

(2) 16-bit instructions

MOVW, XCHW, ADDW, SUBW, CMPW, PUSH, POP, INCW, DECW

2nd Operand 1st Operand	#word	AX	rp[Note]	saddrp	SP	None
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	
rp	MOVW	MOVW[Note]				INCW DECW PUSH POP
saddrp		MOVW				
SP		MOVW				

Note Only when rp = BC, DE, or HL.

(3) Bit manipulation instructions

SET1, CLR1, NOT1, BT, BF

2nd Operand 1st Operand	$addr16	None
A.bit	BT BF	SET1 CLR1
sfr.bit	BT BF	SET1 CLR1
saddr.bit	BT BF	SET1 CLR1
PSW.bit	BT BF	SET1 CLR1
[HL].bit		SET1 CLR1
CY		SET1 CLR1 NOT1

(4) Call instructions/branch instructions

CALL, CALLT, BR, BC, BNC, BZ, BNZ, DBNZ

2nd Operand 1st Operand	AX	!addr16	[addr5]	$addr16
Basic Instructions	BR	CALL BR	CALLT	BR BC BNC BZ BNZ
Compound Instructions				DBNZ

(5) Other instructions

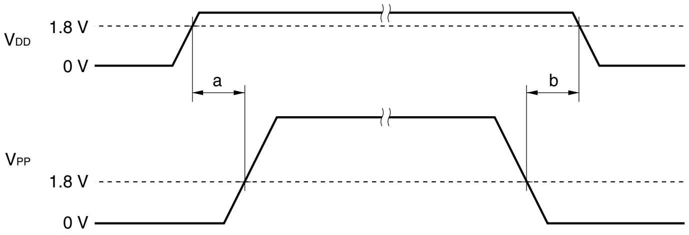
RET, RETI, NOP, EI, DI, HALT, STOP

Absolute Maximum Ratings (T_A = 25°C)

Parameter	Symbol	Conditions	Ratings	Unit
Supply voltage	V_{DD}		–0.3 to +6.5	V
	V_{PP}	μPD78F9468 only, **Note 1**	–0.3 to +10.5	V
Input voltage	V_I		–0.3 to V_{DD} + 0.3 Note 2	V
Output voltage	V_{O1}	P00 to P03, P10, P11, P40 to P43, P60, P61	–0.3 to V_{DD} + 0.3 Note 2	V
	V_{O2}	COM0 to COM3, S0 to S16, P80/S22 to P85/S17	–0.3 to V_{LC0} + 0.3 Note 2	V
Output current, high	I_{OH}	Pin P60/TO40	–30	mA
		Per pin (except P60/TO40)	–10	mA
		Total for all pins (except P60/TO40)	–30	mA
Output current, low	I_{OL}	Per pin	30	mA
		Total for all pins	80	mA
Operating ambient temperature	T_A	During normal operation	–40 to +85	°C
		During flash memory programming	10 to 40	°C
Storage temperature	T_{stg}	Mask ROM version	–65 to +150	°C
		Flash memory version	–40 to +125	°C

★ **Notes 1.** Make sure that the following conditions of the V_{PP} voltage application timing are satisfied when the flash memory is written.

- When supply voltage rises
  V_{PP} must exceed V_{DD} 10 μs or more after V_{DD} has reached the lower-limit value (1.8 V) of the operating voltage range (a in the figure below).
- When supply voltage falls
  V_{DD} must be lowered 10 μs or more after V_{PP} falls below the lower-limit value (1.8 V) of the range of V_{DD} (b in the figure below).

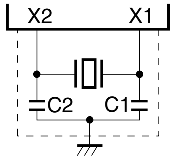
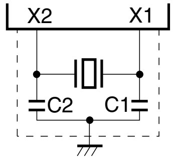
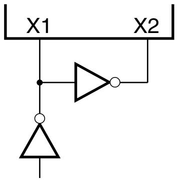


2. 6.5 V or less

Caution **Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.**

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		5.0	MHz
		Oscillation stabilization time[Note 2]	After V_{DD} has reached the oscillation voltage range MIN.			4	ms
Crystal resonator		Oscillation frequency [Note 1]		1.0		5.0	MHz
		Oscillation stabilization time[Note 2]	4.5 ≤ V_{DD} ≤ 5.5 V			10	ms
			1.8 ≤ V_{DD} ≤ 5.5 V			30	ms
External clock		X1 input frequency (f_X)[Note 1]		1.0		5.0	MHz
		X1 input high-/low-level width (t_{XH}, t_{XL})		85		500	ns

Notes 1. Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

2. Time required to stabilize oscillation after reset or STOP mode release. Use the resonator to stabilize oscillation within the oscillation wait time.

Cautions 1. When using the main system clock oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

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

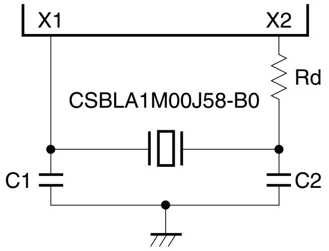
2. When the main system clock is stopped and the device is operating on the subsystem clock, wait until the oscillation stabilization time has been secured by the program before switching back to the main system clock.

★ Recommended Oscillator Constants

Ceramic resonator (T_A = –40 to +85°C) (mask ROM version)

Manufacturer	Part Number	Frequency (MHz)	Recommended Circuit Constant (pF)		Oscillation Voltage Range (V_{DD})		Remark
			C1	C2	MIN.	MAX.	
Murata Mfg.	CSBLA1M00J58-B0[Note]	1.0	100	100	2.0	5.5	Rd = 3.3 kΩ
	CSTCC2M00G53-R0	2.0	–	–	1.8	5.5	With internal capacitor
	CSTCR4M00G53-R0	4.0					
	CSTLS4M00G53-B0						
	CSTCR4M19G53-R0	4.194					
	CSTLS4M19G53-B0						
	CSTCR4M91G53-R0	4.915					
	CSTLS4M91G53-B0						
	CSTCR5M00G53-R0	5.0					
	CSTLS5M00G53-B0						
Kyocera	PBRC4.00HR	4.0	–	–	1.8	5.5	With internal capacitor
	PBRC4.19HR	4.19					
	PBRC4.91HR	4.91					
	PBRC5.00HR	5.0					
TDK	FCR4.0MC5	4.0	–	–	2.0	5.5	With internal capacitor
	FCR5.0MC5	5.0					

Note When using the CSBLA1M00J58-B0 (1.0 MHz) of Murata Mfg. as the ceramic resonator, a limiting resistor (Rd = 3.3 kΩ) is necessary (refer to the figure below). The limiting resistor is not necessary when other recommended resonators are used.

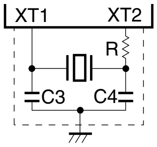
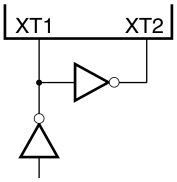


Caution The oscillator constant is a reference value based on evaluation under a specific environment by the resonator manufacturer. If optimization of oscillator characteristics is necessary in the actual application, apply to the resonator manufacturer for evaluation on the implementation circuit.

The oscillation voltage and oscillation frequency indicate only oscillator characteristics. Use the μPD78946x so that the internal operating conditions are within the specifications of the DC and AC characteristics.

Remark For the resonator selection and oscillator constant of the μPD78F9468, users are required to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

Subsystem Clock Oscillator Characteristics (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Crystal resonator		Oscillation frequency (f_{XT})[Note 1]		32	32.768	35	kHz
		Oscillation stabilization time[Note 2]	4.5 ≤ V_{DD} ≤ 5.5 V		1.2	2	s
			1.8 ≤ V_{DD} ≤ 5.5 V			10	s
External clock		XT1 input frequency (f_{XT})[Note 1]		32		35	kHz
		XT1 input high-/low-level width (t_{XTH}, t_{XTL})		14.3		15.6	μs

Notes 1. Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

2. The time required for oscillation to stabilize after V_{DD} reaches the MIN. oscillation voltage range. Use a resonator to stabilize oscillation during the oscillation wait time.

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 signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as V_{SS}.**
- **Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

2. The subsystem clock oscillator is designed as a low-amplitude circuit for reducing current consumption, and is more prone to malfunction due to noise than the main system clock oscillator. Particular care is therefore required with the wiring method when the subsystem clock is used.

Remark For the resonator selection and oscillator constant, users are required to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

DC Characteristics (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V) (1/2)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, low	I_{OL}	Per pin				10	mA
		Total for all pins				80	mA
Output current, high	I_{OH}	Per pin (except P60/TO40)				–1	mA
★		P60/TO40	V_{DD} = 3.0 V, V_{OH} = 2.0 V	–7	–15	–24	mA
		Total for all pins (except P60/TO40)				–15	mA
Input voltage, high	V_{IH1}	P00 to P03, P10, P11, P60, P80 to P85	2.7 ≤ V_{DD} ≤ 5.5 V	0.7V_{DD}		V_{DD}	V
			1.8 ≤ V_{DD} ≤ 5.5 V	0.9V_{DD}		V_{DD}	V
	V_{IH2}	$\overline{\text{RESET}}$, P40 to P43, P61	2.7 ≤ V_{DD} ≤ 5.5 V	0.8V_{DD}		V_{DD}	V
			1.8 ≤ V_{DD} ≤ 5.5 V	0.9V_{DD}		V_{DD}	V
	V_{IH3}	X1, X2		V_{DD} – 0.1		V_{DD}	V
	V_{IH4}	XT1, XT2		V_{DD} – 0.1		V_{DD}	V
Input voltage, low	V_{IL1}	P00 to P03, P10, P11, P60, P80 to P85	2.7 ≤ V_{DD} ≤ 5.5 V	0		0.3V_{DD}	V
			1.8 ≤ V_{DD} ≤ 5.5 V	0		0.1V_{DD}	V
	V_{IL2}	$\overline{\text{RESET}}$, P40 to P43, P61	2.7 ≤ V_{DD} ≤ 5.5 V	0		0.2V_{DD}	V
			1.8 ≤ V_{DD} ≤ 5.5 V	0		0.1V_{DD}	V
	V_{IL3}	X1, X2		0		0.1	V
	V_{IL4}	XT1, XT2		0		0.1	V
Output voltage, high	V_{OH11}	P00 to P03, P10, P11, P40 to P43, P61	1.8 ≤ V_{DD} ≤ 5.5 V, I_{OH} = –100 μA	V_{DD} – 0.5			V
	V_{OH12}		1.8 ≤ V_{DD} ≤ 5.5 V, I_{OH} = –500 μA	V_{DD} – 0.7			V
	V_{OH21}	P60/TO40	1.8 ≤ V_{DD} ≤ 5.5 V, I_{OH} = –400 μA	V_{DD} – 0.5			V
	V_{OH22}		1.8 ≤ V_{DD} ≤ 5.5 V, I_{OH} = –2 mA	V_{DD} – 0.7			V
Output voltage, low	V_{OL1}	P00 to P03, P10, P11, P40 to P43, P60, P61	1.8 ≤ V_{DD} ≤ 5.5 V, I_{OL} = 400 μA			0.5	V
	V_{OL2}		1.8 ≤ V_{DD} ≤ 5.5 V, I_{OL} = 2 mA			0.7	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) (2/2)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	$V_{IN} = V_{DD}$	P00 to P03, P10, P11, P40 to P43, P60, P61, P80 to P85, $\overline{\text{RESET}}$			3	μA
	I_{LIH2}		X1, X2, XT1, XT2			20	μA
Input leakage current, low	I_{LIL1}	V_{IN} = 0 V	P00 to P03, P10, P11, P40 to P43, P60, P61, P80 to P85, $\overline{\text{RESET}}$			–3	μA
	I_{LIL2}		X1, X2, XT1, XT2			–20	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
Software pull-up resistors	R_1	V_{IN} = 0 V	P00 to P03, P10, P11, P40 to P43	50	100	200	kΩ
★ Supply current[Note 1] (mask ROM version)	I_{DD1}	5.0 MHz crystal oscillation operating mode	V_{DD} = 5.5 V[Note 2]		1.5	3.0	mA
			V_{DD} = 3.3 V[Note 3]		0.6	1.2	mA
	I_{DD2}	5.0 MHz crystal oscillation HALT mode	V_{DD} = 5.5 V		0.8	2.0	mA
			V_{DD} = 3.3 V		0.4	0.8	mA
	I_{DD3}	32,768 kHz crystal oscillation operating mode[Note 4]	V_{DD} = 5.5 V		50	100	μA
			V_{DD} = 3.3 V		30	60	μA
	I_{DD4}	32.768 kHz crystal oscillation HALT mode[Note 4]	V_{DD} = 5.5 V		25	50	μA
			V_{DD} = 3.3 V		7	25	μA
	I_{DD5}	STOP mode (POC not operating)	V_{DD} = 5.5 V		0.1	10	μA
			V_{DD} = 3.3 V		0.05	5	μA
	I_{DD6}	STOP mode (POC operating)	V_{DD} = 5.5 V		2	20	μA
			V_{DD} = 3.3 V		1	10	μA
	I_{DD7}	5.0 MHz crystal oscillation A/D operating mode	V_{DD} = 5.5 V		2.3	5.0	mA
			V_{DD} = 3.3 V		1.1	2.4	mA
★ Supply current[Note 1] (μPD78F9468)	I_{DD1}	5.0 MHz crystal oscillation operating mode	V_{DD} = 5.5 V[Note 2]		5.0	15.0	mA
			V_{DD} = 3.3 V[Note 3]		2.0	5.0	mA
	I_{DD2}	5.0 MHz crystal oscillation HALT mode	V_{DD} = 5.5 V		1.2	3.6	mA
			V_{DD} = 3.3 V		0.5	1.5	mA
	I_{DD4}	32.768 kHz crystal oscillation HALT mode[Note 4]	V_{DD} = 5.5 V		25	70	μA
			V_{DD} = 3.3 V		10	35	μA
	I_{DD5}	STOP mode	V_{DD} = 5.5 V		2	20	μA
			V_{DD} = 3.3 V		1	10	μA

Notes 1. Excludes current flowing through ports (including current flowing through on-chip pull-up resistors).

2. High-speed mode operation (when the processor clock control register (PCC) is set to 00H).

3. Low-speed mode operation (when PCC is set to 02H)

4. When the main system clock operation is stopped.

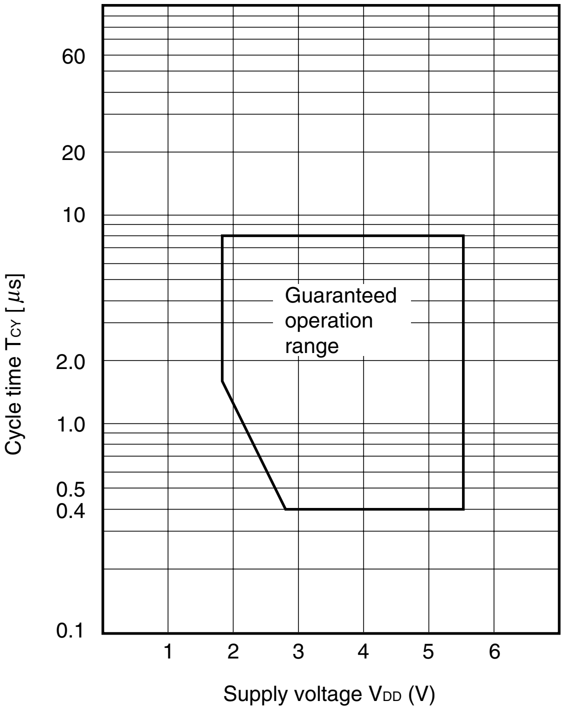
Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

AC Characteristics

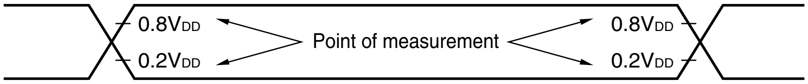
(1) Basic operation (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Cycle time (Min. instruction execution time)	T_{CY}	2.7 ≤ V_{DD} ≤ 5.5 V	0.4		8.0	μs
		1.8 ≤ V_{DD} ≤ 5.5 V	1.6		8.0	μs
Interrupt input high-/low-level width	t_{INTH}, t_{INTL}	INTP0	10			μs
Key return pin low-level width	t_{KRIL}	KR00 to KR03	10			μs
$\overline{\text{RESET}}$ low-level width	t_{RSL}		10			μs

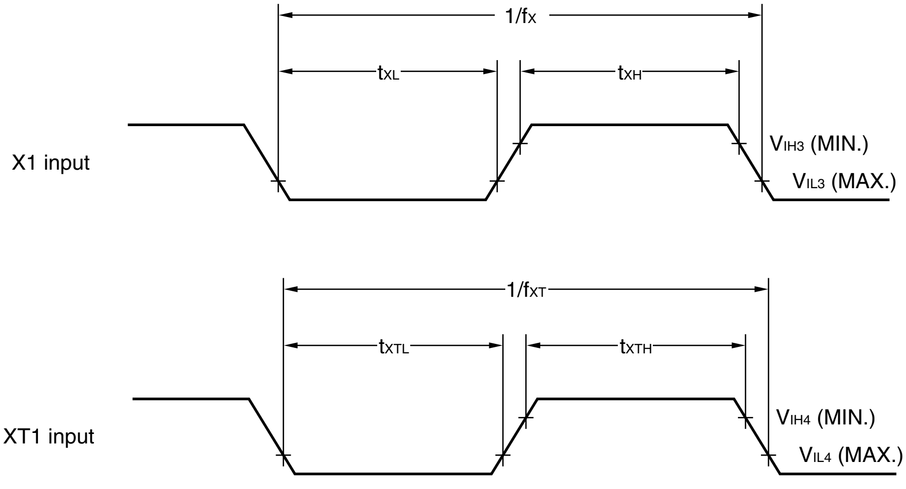
T_{CY} vs. V_{DD} (Main System Clock)



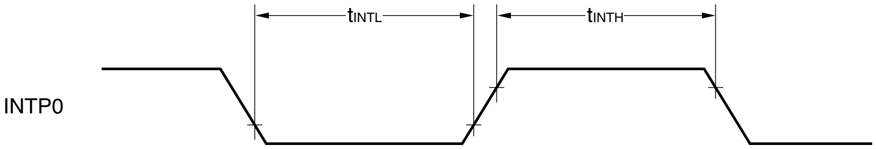
AC Timing Measurement Points (Excluding X1, XT1 Input)



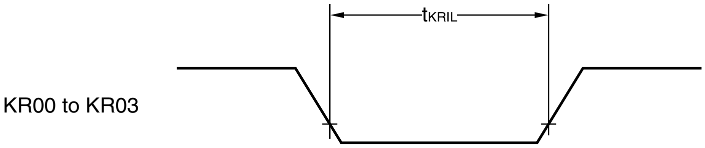
Clock Timing



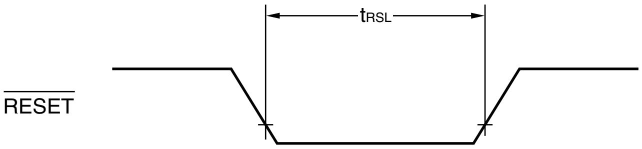
Interrupt Input Timing



Key Return Input Timing



$\overline{\text{RESET}}$ Input Timing



8-Bit A/D Converter Characteristics (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution			8	8	8	bit
Overall error[Notes 1, 2]		$2.7 \le V_{DD} \le 5.5$ V			±0.6	%FSR
		$1.8 \le V_{DD} \le 5.5$ V			±1.2	%FSR
Conversion time	t_{CONV}	$2.7 \le V_{DD} \le 5.5$ V	14		100	μs
		$1.8 \le V_{DD} \le 5.5$ V	28		100	μs

Notes 1. Excludes quantization error (±0.2% FSR).

2. The overall error is indicated as a ratio to the full-scale value (%FSR).

Remark FSR: Full scale range

LCD Characteristics (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
LCD boost output voltage	V_{LC20}	GAIN = 0	V_{LC2} pin	1.35	1.5	1.65	V
	V_{LC10}		V_{LC1} pin		3.0		V
	V_{LC00}		V_{LC0} pin		4.5		V
	V_{LC21}	GAIN = 1	V_{LC2} pin	0.9	1.0	1.1	V
	V_{LC11}		V_{LC1} pin		2.0		V
	V_{LC01}		V_{LC0} pin		3.0		V
LCD output voltage differential[Note] (common)	V_{ODC}	I_O = ±5 μA		0		±0.2	V
LCD output voltage differential[Note] (segment)	V_{ODS}	I_O = ±1 μA		0		±0.2	V

Note The voltage differential is the difference between the output voltage and the ideal value of the segment and common signal outputs.

Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage	V_{DDDR}		1.8		5.5	V
Low voltage detection (POC) voltage[Note 1]	V_{POC}	Response time: 2 ms[Note 2]	1.8	1.9	2.0	V
Release signal set time	t_{SREL}	STOP released by $\overline{\text{RESET}}$	10			μs
Oscillation stabilization wait time[Note 3]	t_{WAIT}	Cancelled by $\overline{\text{RESET}}$		$2^{15}/f_X$		s
		Cancelled by interrupt request		**Note 4**		s

Notes 1. In mask ROM versions, only when the POC circuit is selected by a mask option (refer to **CHAPTER 16 MASK OPTION**).

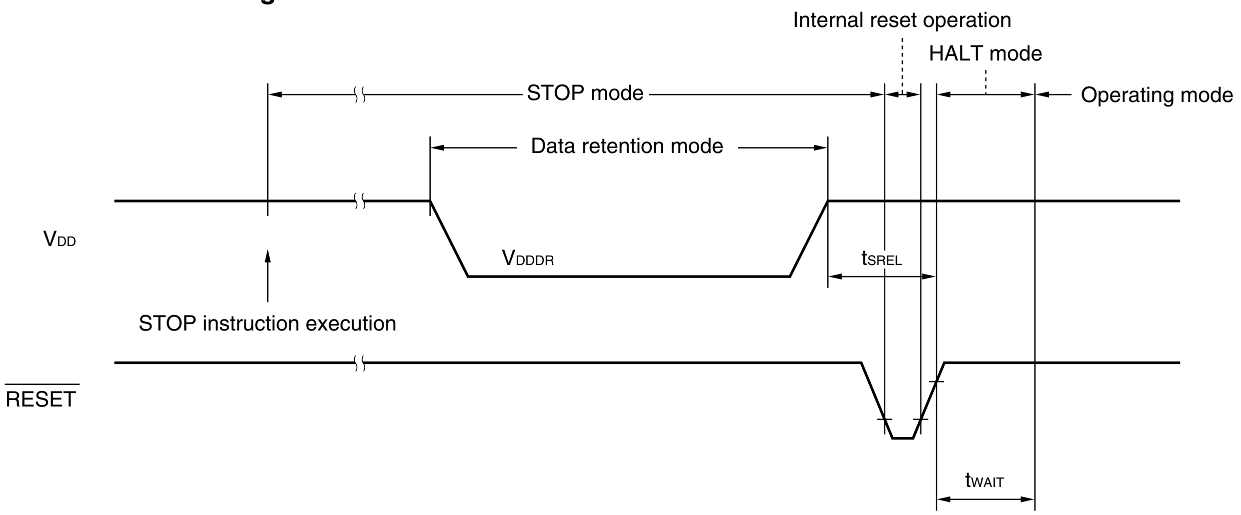
2. The response time is the time until the output is inverted following detection of voltage by POC, or the time until operation stabilizes after the shift from the operation stopped state to the operating state.

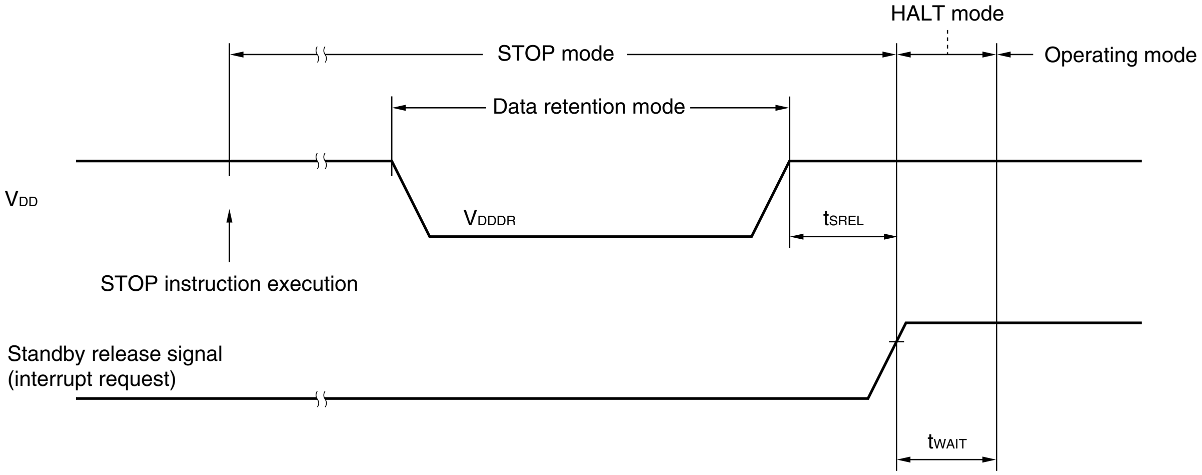
3. The oscillation stabilization time is the amount of time the CPU operation is stopped in order to avoid unstable operation at the start of oscillation. Program operation does not start until both the oscillation stabilization time and the time until oscillation starts have elapsed.

4. Selection of $2^{12}/f_X$, $2^{15}/f_X$, and $2^{17}/f_X$ is possible using bits 0 to 2 (OSTS0 to OSTS2) of the oscillation stabilization time selection register (OSTS) (refer to **13.1.2 Register controlling standby function**).

Remark f_X: Main system clock oscillation frequency

Data Retention Timing



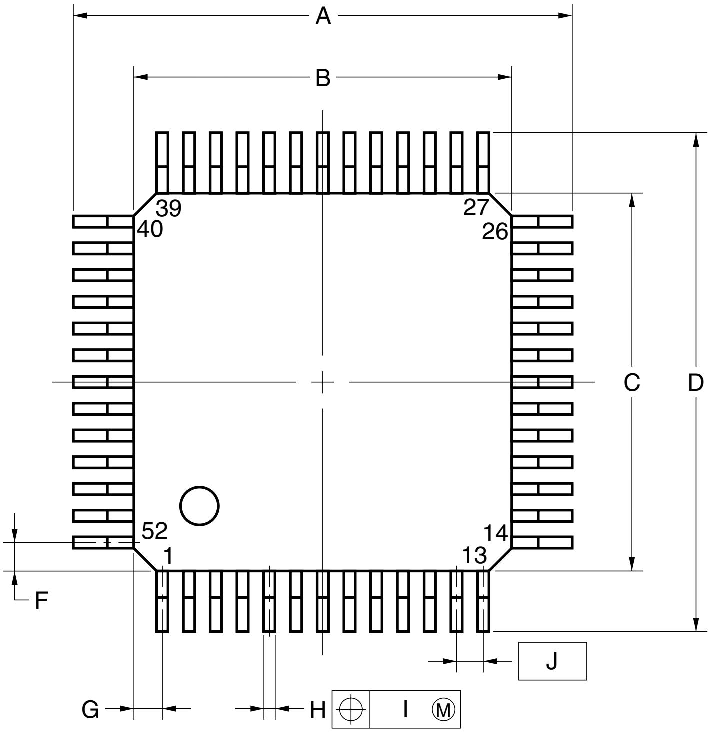


★ Write and Erase Characteristics (T_A = 10 to 40°C, V_{DD} = 1.8 to 5.5 V)

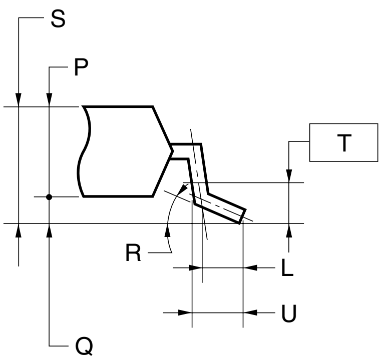
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Write operation frequency	f_X	V_{DD} = 2.7 to 5.5 V	1.0		5	MHz
		V_{DD} = 1.8 to 5.5 V	1.0		1.25	MHz
Write current (V_{DD} pin)$^{\text{Note}}$	I_{DDW}	When V_{PP} supply voltage = V_{PP1} (at 5.0 MHz operation)			7	mA
Write current (V_{PP} pin)$^{\text{Note}}$	I_{PPW}	When V_{PP} supply voltage = V_{PP1}			13	mA
Erase current (V_{DD} pin) $^{\text{Note}}$	I_{DDE}	When V_{PP} supply voltage = V_{PP1} (at 5.0 MHz operation)			7	mA
Erase current (V_{PP} pin)$^{\text{Note}}$	I_{PPE}	When V_{PP} supply voltage = V_{PP1}			100	mA
Unit erase time	t_{er}		0.5	1	1	s
Total erase time	t_{era}				20	s
Number of rewrites		Erase and write is considered as 1 cycle			20	Times
V_{PP} supply voltage	V_{PP0}	Normal operation	0		$0.2V_{DD}$	V
	V_{PP1}	Flash memory programming	9.7	10.0	10.3	V

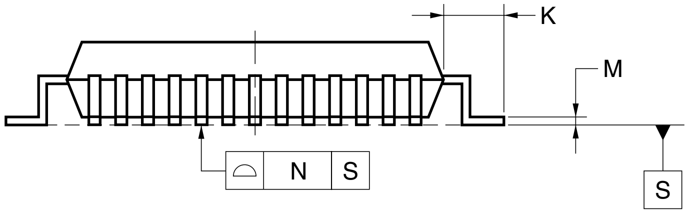
Note Excludes current flowing through ports (including on-chip pull-up resistors)

52-PIN PLASTIC LQFP (10x10)



detail of lead end





ITEM	MILLIMETERS
A	12.0±0.2
B	10.0±0.2
C	10.0±0.2
D	12.0±0.2
F	1.1
G	1.1
H	0.32±0.06
I	0.13
J	0.65 (T.P.)
K	1.0±0.2
L	0.5
M	$0.17^{+0.03}_{-0.05}$
N	0.10
P	1.4
Q	0.1±0.05
R	$3°^{+4°}_{-3°}$
S	1.5±0.1
T	0.25
U	0.6±0.15

S52GB-65-8ET-2

CHAPTER 20 RECOMMENDED SOLDERING CONDITIONS

The μPD789467 Subseries should be soldered and mounted under the following recommended conditions.

For soldering methods and conditions other than those recommended below, contact an NEC Electronics sales representative.

For technical information, see the following website.

Semiconductor Device Mount Manual (http://www.necel.com/pkg/en/mount/index.html)

Table 20-1. Surface Mounting Type Soldering Conditions

μPD789462GB-×××-8ET: 52-pin plastic LQFP (10 × 10)
μPD789464GB-×××-8ET: 52-pin plastic LQFP (10 × 10)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 235°C, Time: 30 seconds max. (at 210°C or higher), Count: Twice or less	IR35-00-2
VPS	Package peak temperature: 215°C, Time: 40 seconds max. (at 200°C or higher), Count: Twice or less	VP15-00-2
Wave soldering	Solder bath temperature: 260°C max., Time: 10 seconds max., Count: Once, Preheating temperature: 120°C max. (package surface temperature)	WS60-00-1
Partial heating	Pin temperature: 300°C max. Time: 3 seconds max. (per pin row)	—

Caution Do not use different soldering methods together (except for partial heating).

★ **μPD789466GB-×××-8ET: 52-pin plastic LQFP (10 × 10)**
★ **μPD789467GB-×××-8ET: 52-pin plastic LQFP (10 × 10)**
μPD78F9468GB-8ET: 52-pin plastic LQFP (10 × 10)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 235°C, Time: 30 seconds max. (at 210°C or higher), Count: Twice or less, Exposure limit: 7 days[Note] (after that, prebake at 125°C for 10 hours)	IR35-107-2
VPS	Package peak temperature: 215°C, Time: 40 seconds max. (at 200°C or higher), Count: Twice or less, Exposure limit: 7 days[Note] (after that, prebake at 125°C for 10 hours)	VP15-107-2
Wave soldering	Solder bath temperature: 260°C max., Time: 10 seconds max., Count: Once, Preheating temperature: 120°C max. (package surface temperature), Exposure limit: 7 days[Note] (after that, prebake at 125°C for 10 hours)	WS60-107-1
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 DEVELOPMENT TOOLS

The following development tools are available for development of systems using the μPD789467 Subseries. Figure A-1 shows development tools.

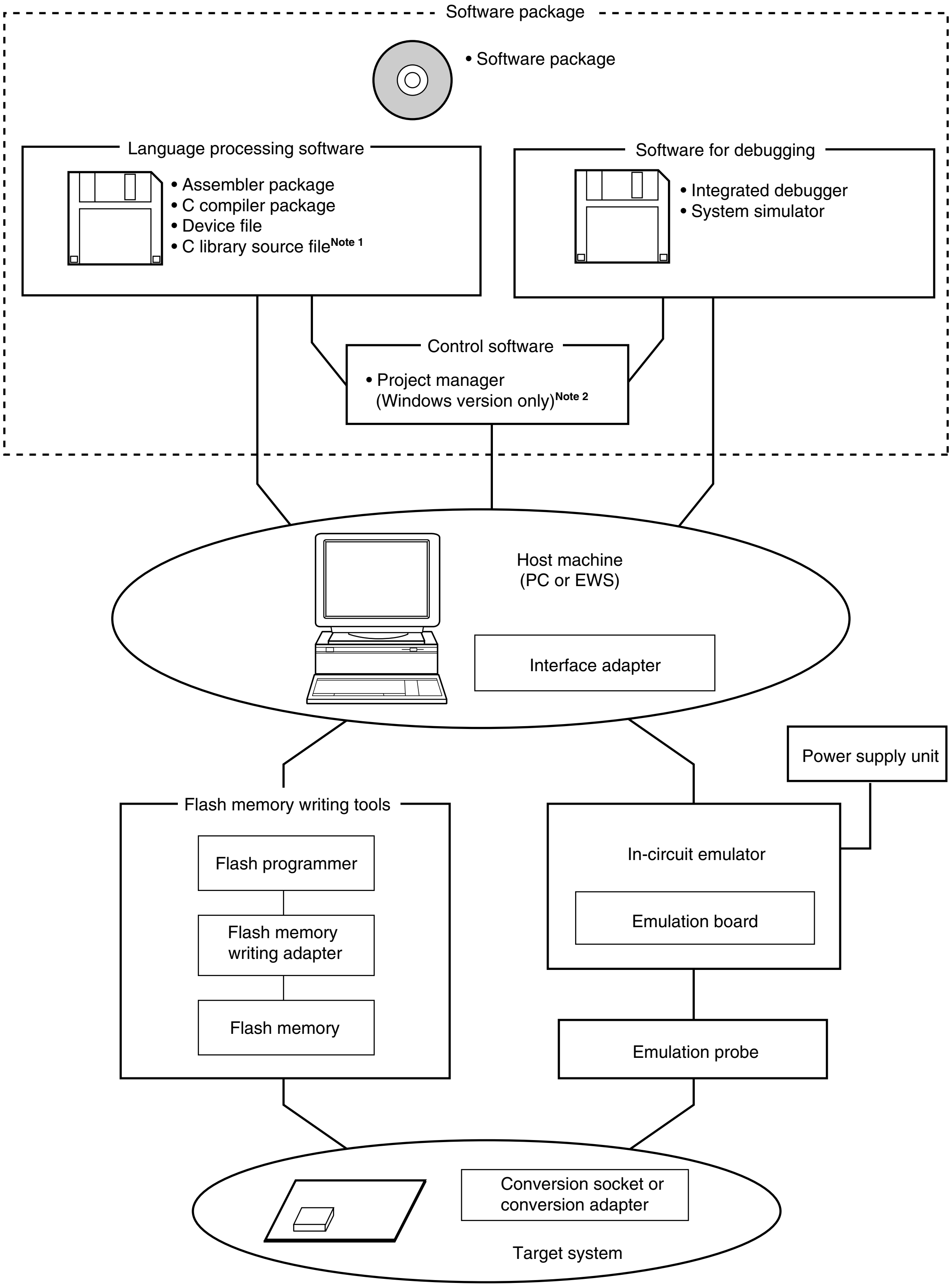
- Support of PC98-NX series

  Unless specified otherwise, the products supported by IBM PC/AT™ compatibles can be used in the PC98-NX series. When using the PC98-NX Series, refer to the explanation of IBM PC/AT compatibles.

- Windows

  Unless specified otherwise, "Windows" indicates the following operating systems.
  - Windows 3.1
  - Windows 95, 98, 2000
  - Windows NT™ Ver.4.0

Figure A-1. Development Tools



Notes 1. The C library source file is not included in the software package.

2. The project manager is included in the assembler package. The project manager is used only for Windows.

A.1 Software Package

SP78K0S Software package	Various software tools for 78K/0S Series development are integrated into one package. The following tools are included. RA78K0S, CC78K0S, ID78K0S-NS, SM78K0S, various device files
	Part number: μS××××SP78K0S

Remark ×××× in the part number differs depending on the operating system to be used.

μS××××SP78K0S

××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Japanese Windows	CD-ROM
BB17		English Windows	

A.2 Language Processing Software

RA78K0S Assembler package	Program that converts program written in mnemonic into object codes that can be executed by microcontroller. In addition, automatic functions to generate symbol tables and optimize branch instructions are also provided. Used in combination with a device file (DF789468) (sold separately). **<Caution when used in PC environment>** The assembler package is a DOS-based application but may be used in the Windows environment by using the Project Manager of Windows (included in the assembler package).
	Part number: μS××××RA78K0S
CC78K0S C compiler package	Program that converts program written in C language into object codes that can be executed by microcontroller. Used in combination with an assembler package (RA78K0S) and device file (DF789468) (both sold separately). **<Caution when used in PC environment>** The C compiler package is a DOS-based application but may be used in the Windows environment by using the Project Manager of Windows (included in the assembler package).
	Part number: μS××××CC78K0S
DF789468[Note 1] Device file	File containing the information specific to the device. Used in combination with the RA78K0S, CC78K0S, and SM78K0S (sold separately).
	Part number: μS××××DF789468
CC78K0S-L[Note 2] C library source file	Source file of functions for generating object library included in C compiler package. Necessary for changing object library included in C compiler package according to customer's specifications. Since this is a source file, its working environment does not depend on any particular operating system.
	Part number: μS××××CC78K0S-L

Notes 1. DF789468 is a common file that can be used with the RA78K0S, CC78K0S, ID78K0S-NS, and SM78K0S.

2. CC78K0S-L is not included in the software package (SP78K0S).

Remark ×××× in the part number differs depending on the host machine and operating system to be used.

μS××××RA78K0S
μS××××CC78K0S

××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT compatibles	Japanese Windows	3.5” 2HD FD
BB13		English Windows	
AB17		Japanese Windows	CD-ROM
BB17		English Windows	
3P17	HP9000 series 700™	HP-UX™ (Rel.10.10)	
3K17	SPARCstation™	SunOS™ (Rel.4.1.4), Solaris™ (Rel.2.5.1)	

μS××××DF789468
μS××××CC78K0S-L

××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT compatibles	Japanese Windows	3.5” 2HD FD
BB13		English Windows	
3P16	HP9000 series 700	HP-UX (Rel.10.10)	DAT
3K13	SPARCstation	SunOS (Rel.4.1.4), Solaris (Rel.2.5.1)	3.5” 2HD FD
3K15			1/4” CGMT

A.3 Control Software

Project manager	Control software designed so that the user program can be efficiently developed in the Windows environment. A series of jobs for user program development including starting the editor, building, and starting the debugger, can be executed on the project manager. <Caution> The project manager is included in the assembler package (RA78K0S). It cannot be used in an environment other than Windows.

A.4 Flash Memory Writing Tools

Flashpro III (FL-PR3, PG-FP3) Flashpro IV (FL-PR4, PG-FP4) Flash programmer	Dedicated flash programmer for microcontrollers incorporating flash memory
FA-52GB-8ET Flash memory writing adapter	Adapter for writing to flash memory and connected to Flashpro III or Flashpro IV. FA-52GB-8ET: for 52-pin plastic LQFP (GB-8ET type)

Remark The FL-PR3, FL-PR4, and FA-52GB-8ET are products of Naito Densei Machida Mfg. Co., Ltd. (TEL +81-45-475-4191).

A.5 Debugging Tools (Hardware)

IE-78K0S-NS In-circuit emulator	In-circuit emulator for debugging hardware and software of application system using 78K/0S Series. Can be used with integrated debugger (ID78K0S-NS). Used in combination with AC adapter, emulation probe, and interface adapter for connecting the host machine.
IE-78K0S-NS-A In-circuit emulator	A coverage function has been added to the IE-78K0S-NS function and the debug function has been further enhanced, enhancing the tracer and timer functions.
IE-70000-MC-PS-B AC adapter	Adapter for supplying power from AC 100 to 240 V outlet.
IE-70000-98-IF-C Interface adapter	Adapter necessary when using PC-9800 series PC (except notebook type) as host machine (C bus supported)
IE-70000-CD-IF-A PC card interface	PC card and interface cable necessary when using notebook PC as host machine (PCMCIA socket supported)
IE-70000-PC-IF-C Interface adapter	Interface adapter necessary when using IBM PC/AT compatible as host machine (ISA bus supported)
IE-70000-PCI-IF-A Interface adapter	Adapter necessary when using personal computer incorporating PCI bus as host machine
IE-789468-NS-EM1 Emulation board	Board for emulating peripheral hardware specific to device. Used in combination with in-circuit emulator.
NP-H52GB-TQ Emulation probe	Probe for connecting in-circuit emulator and target system. Used in combination with TGB-052SBP.
★ TGB-052SBP Conversion adapter	Conversion adapter to connect NP-H52GB-TQ and target system board on which 52-pin plastic LQFP (GB-8ET type) can be mounted

Remarks **1.** The NP-H52GB-TQ is a product of Naito Densei Machida Mfg. Co., Ltd. (TEL +81-45-475-4191).

2. The TGB-052SBP is a product of TOKYO ELETECH CORPORATION.

For further information, contact: Daimaru Kogyo, Ltd.
Tokyo Electronics Department (TEL +81-3-3820-7112)
Osaka Electronics Department (TEL +81-6-6244-6672)

A.6 Debugging Tools (Software)

ID78K0S-NS Integrated debugger	A debugger supporting in-circuit emulators for the 78K/0S Series: IE-78K0S-NS and IE-78K0S-NS-A. The ID78K0S-NS is Windows-based software. This program enhances the debugging functions for C language. Therefore, it can display the trace results corresponding to the source program by using the window integration function that links the source program, disassembled display, and memory display with the trace results. Use this program in combination with a device file (DF789468) (sold separately).
	Part number: μS××××ID78K0S-NS
SM78K0S System simulator	A system simulator for the 78K/0S Series. The SM78K0S is Windows-based software. C-source-level or assembler level debugging is possible while simulating the operation of the target system on the host machine. Using the SM78K0S enables logical and performance verification of an application independently of the hardware development. This enhances development efficiency and improves software quality. Use this program in combination with a device file (DF789468) (sold separately).
	Part number: μS××××SM78K0S
DF789468[Note] Device file	File containing information specific to the device. Use this file in combination with the RA78K0S, CC78K0S, ID78K0S-NS, and SM78K0S (sold separately).
	Part number: μS××××DF789468

Note DF789468 is a common file that can be used with the RA78K0S, CC78K0S, ID78K0S-NS, and SM78K0S.

Remark ×××× in the part number differs depending on the operating system to be used and the supply medium.

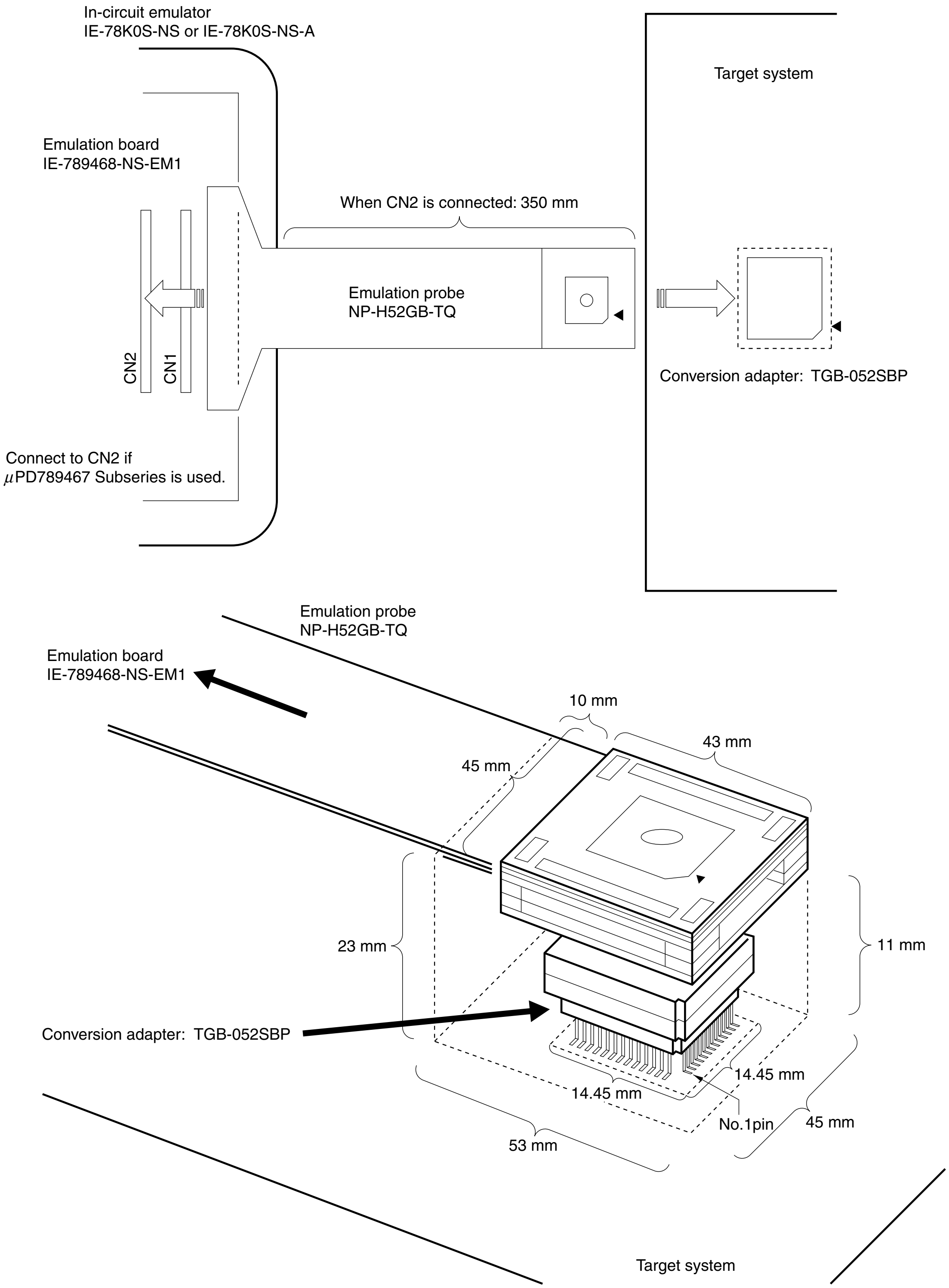
μS××××ID78K0S-NS
μS××××SM78K0S

××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT compatibles	Japanese Windows	3.5" 2HD FD
BB13		English Windows	
AB17		Japanese Windows	CD-ROM
BB17		English Windows	

A.7 Cautions When Designing Target System

The following shows the conditions when connecting the emulation probe to the conversion connector and conversion socket. Design the system considering shapes and other conditions of the components to be mounted on the target system and be sure to follow the configuration below.

Figure A-2. Condition Diagram of Connection to Target System



APPENDIX B REGISTER INDEX

B.1 Register Index (Alphabetic Order of Register Name)

B.2 Register Index (Alphabetic Order of Register Symbol)

★

APPENDIX C REVISION HISTORY

The revision history is described below. The “Applied to” column indicates the chapters in each edition.)

Edition	Major Revisions from Previous Edition	Applied to
2nd edition	Deletion of development status indication “under development” for μPD789466 and 789467	Throughout
	Addition of description on pin connections in **2.2.15 V_{PP} (μPD78F9468 only)**	**CHAPTER 2 PIN FUNCTIONS**
	Table 3-3 Special-Function Registers • Change of attribute of port 8 (P8) to read-only and change of value after reset to undefined • Modification of oscillation stabilization time selection register (OSTS) to allow use of 1-bit manipulation instruction.	**CHAPTER 3 CPU ARCHITECTURE**
	Modification of **Caution 2** in **Figure 4-10 Format of Port Function Register 8**	**CHAPTER 4 PORT FUNCTIONS**
	Addition of **Note** on feedback resistor in **Figure 5-3 Format of Suboscillation Mode Register**	**CHAPTER 5 CLOCK GENERATOR**
	6.2 Configuration of 8-Bit Timers 30 and 40 • **Modification of Figure 6-3 Block Diagram of Output Controller (Timer 40)** • **Addition of description to (2) 8-bit compare register 40 (CR40)** • **Addition of description to (3) 8-bit H width compare register 40 (CRH40)**	**CHAPTER 6 8-BIT TIMERS 30 AND 40**
	Addition to **Cautions** in **Figure 6-6 Format of Carrier Generator Output Control Register 40**	
	Addition to **Cautions** in **6.4.3 Operation as carrier generator**	
	Modification of description in **6.5 Notes on Using 8-Bit Timers 30 and 40**	
	Addition of **(8) Input impedance of ANI0 pin** to **9.5 Cautions Related to 8-Bit A/D Converter**	**CHAPTER 9 8-BIT A/D CONVERTER**
	Addition to **Notes** and **Cautions** in **Figure 10-2 Format of LCD Display Mode Register 0**	**CHAPTER 10 LCD CONTROLLER/DRIVER**
	Addition to **Cautions** in **Figure 12-2 Format of Interrupt Request Flag Register 0**	**CHAPTER 12 INTERRUPT FUNCTION**
	Addition to **Cautions** in **Figure 12-6 Format of Key Return Mode Register 00**	
	Modification of oscillation stabilization time selection register (OSTS) to allow use of 1-bit manipulation instruction in **13.1.2 Register controlling standby function**	**CHAPTER 13 STANDBY FUNCTION**
	Modification of **Table 15-3 Communication Mode List**	**CHAPTER 15 μPD78F9468**
	CHAPTER 18 ELECTRICAL SPECIFICATIONS • Change from target values to formal specifications • Addition of **Note 1** to **Absolute Maximum Ratings** • Addition of recommended oscillator constants for mask ROM versions • **Modification of Write and Erase Characteristics**	**CHAPTER 18 ELECTRICAL SPECIFICATIONS**
	Addition of recommended conditions for μPD789466 and 789467 in **CHAPTER 20 RECOMMENDED SOLDERING CONDITIONS**	**CHAPTER 20 RECOMMENDED SOLDERING CONDITIONS**
	Change of name “conversion connector” and “conversion socket” to “conversion adapter (TGB-052SBP)” in **A.5 Debugging Tools (Hardware)**	**APPENDIX A DEVELOPMENT TOOLS**
	Addition of **APPENDIX C REVISION HISTORY**	**APPENDIX C REVISION HISTORY**